Sun N1 Service Provisioning System User's Guide for OS Provisioning Plug-In 3.0

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

Part No: 819–4458–10
April 2006

Copyright 2006 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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.

Copyright 2006 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

Tables

Preface

The *Sun N1 Service Provisioning System User's Guide for OS Provisioning Plug-In 3.0* explains how to provision operating systems through the N1 Service Provisioning System (N1 SPS) product.

Who Should Use This Book

The main audience for the *Sun N1 Service Provisioning System User's Guide for OS Provisioning Plug-In 3.0* includes system administrators and operators of N1 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 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 SPS software, read the following books:

- *Sun N1 Service Provisioning System 5.2 System Administration Guide*
- *Sun N1 Service Provisioning System 5.2 Installation Guide*
- *Sun N1 Service Provisioning System 5.2 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 RedHat Linux systems.

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

Chapter 8 provides step-by-step instructions on how to provision Windows systems.

Chapter 9 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 describes the specific variables that apply to the RedHat Linux provisioning profiles.

Appendix D describes the specific variables that apply to the SuSe Linux provisioning profiles.

Appendix E describes the specific variables that apply to the Windows provisioning profiles.

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

Appendix G describes how to configure advanced properties for your provisioning plans.

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

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface	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	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*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

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

TABLE P–2 Shell Prompts

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

◆ ◆ ◆ 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, or systems that are currently running. 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" on page 19
- "Functions" on page 20
- "Provisioning Process Overview" on page 20
- "OS Provisioning System Software Architecture" on page 21
- "Supported Systems" on page 24

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 Service Provisioning System (N1 SPS) product. The N1 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 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 24 and Chapter 3.
2. Install the OS provisioning plug-in, as described in "Installing the OS Provisioning Plug-In" on page 45.
3. Create the OS provisioning server, as explained in "Creating the OS Provisioning Server" on page 50.
4. Define appropriate target hosts to be provisioned. See Chapter 9.
5. Provision the appropriate operating system onto the target hosts. See Chapter 5, Chapter 6, Chapter 7, and Chapter 8.

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, RSC, 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 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.

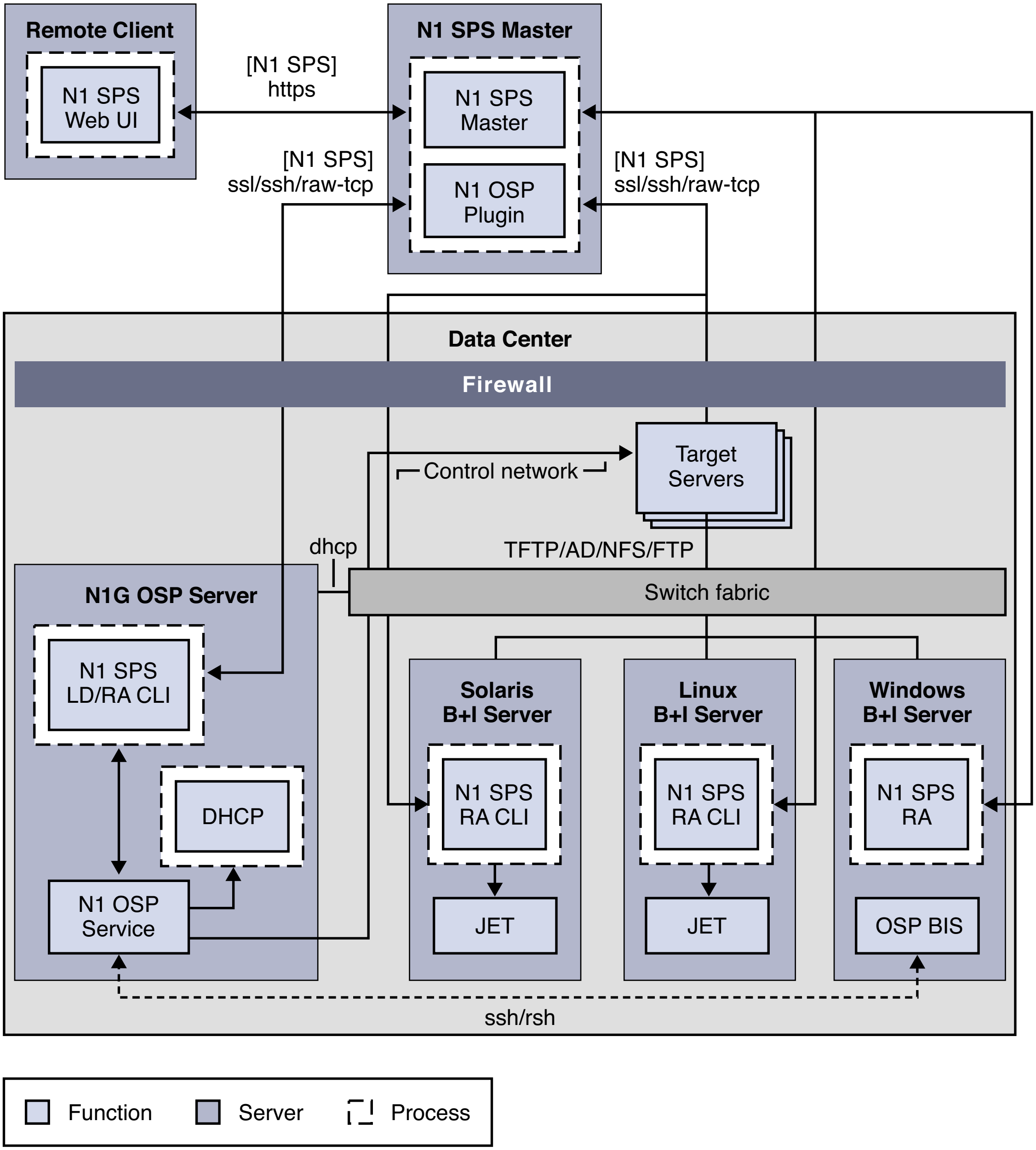

FIGURE 1–1 Software Architecture Diagram

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

- Remote Client – The N1 SPS remote client runs the browser interface and command-line interface. The remote client can be a separate system from the Master Server.
- N1 SPS Master – The N1 SPS Master Server is the main processing engine of the N1 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 (N1 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, RSC, 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 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, RSC, 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 SPS Component	OS Provisioning Component
Master Server	N1 SPS master, agent, and CLI	OS provisioning plug-in
OS provisioning server	N1 SPS RA and CLI	OS provisioning service
Solaris JET server	N1 SPS RA and CLI	OS provisioning boot and install service (BIS)
Linux Kickstart server	N1 SPS RA and CLI	OS provisioning boot and install service (BIS)
Windows RIS server	N1 SPS RA	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 8 (SPARC only) Solaris 9 (SPARC and x86), Solaris 10 (SPARC and x86), RedHat AS 2.1, RedHat AS 3.0, RedHat AS 4.0, Microsoft Windows Server 2000

TABLE 1–2 OS Provisioning Components by Operating System *(Continued)*

OS Provisioning Component	Host Operating System
OS provisioning service	Solaris8 (SPARC only), Solaris 9 (SPARC and x86), Solaris 10 (SPARC and x86), RedHat AS 3.0 (32- and 64-bit), RedHat AS 4.0 (32- and 64-bit)
Solaris boot and install server	Solaris 9 (SPARC and x86), Solaris 10 (SPARC and x86)
Linux boot and install server	RedHat AS 3.0 (32- and 64-bit), RedHat AS 4.0 (32- and 64-bit)
Windows boot and install server	Microsoft Windows 2003 Standard and Enterprise Editions 32 –bit (with Service Pack 1)

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	Custom JumpStart through JET
RedHat Linux	Kickstart
SuSE Linux	AutoYaST
Microsoft Windows 2003 Server Enterprise Edition with Service Pack 1 (32-bit)	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 139.

Target Host Operating Systems

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

- Solaris 8, 9, and 10 for SPARC based systems
- Solaris 9 and 10 for x86 based systems
- RedHat AS Linux 3.0 and 4.0 (32- and 64-bit)
- SuSE Linux 9.0 (32–bit)

- Microsoft Windows 2000 Server (with service packs)
- Microsoft Windows 2000 Advanced Server (with service packs)
- Microsoft Windows Server 2003 Standard and Enterprise Editions (32- and 64–bit, with Service Pack 1)
- Microsoft Windows Server 2003 Web Edition (32–bit, with Service Pack 1)

Sample Command Line Scripts for the OS Provisioning Plug-In

On the N1 SPS 5.2 DVD, several sample CLI script files are included in the `/plugins/com.sun.n1osp/SampleClis` directory. These scripts perform a variety of tasks, including the following.

- Create OS profiles
- Create target hosts
- Install boot and install servers
- Provision your target hosts

You can use these sample scripts as a reference for running CLI commands that are specific to your environment.

Note – These sample scripts are not supported by Sun Microsystems, Inc. You must edit the commands and arguments in these samples for your particular environment.

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 27
- "Runtime Issues" on page 27

Installation Issues

There are no known installation issues.

Runtime Issues

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

Windows Installations Switch to Interactive Mode on Multiboot Target Hosts (6360018)

Description: If you try to provision the Windows OS to a target host that is running the Solaris or Linux OS, and the free disk space on the system is not large enough for the Windows OS, the installation switches to the interactive mode.

Workaround: Choose one of the following workarounds.

- Before you provision the target host, use the `fdisk` utility to remove the Solaris or Linux partitions on the target host.
- If you encounter this issue during the installation, follow the instructions in the interactive screens to delete existing unknown partitions with the `DOS FDISK` utility.

Windows RIS Server Does Not Respond to DHCP PXE Requests (6333612)

Description: If the first interface on a Windows RIS server is not configured as the provisioning interface, the RIS server does not respond to DHCP PXE requests from target hosts. This issue occurs because the BINLSVC/RIS service on the Windows boot and install server listens on the first interface that is configured at the time of system boot. If the first interface is not configured as the provisioning interface, the RIS server cannot respond to DHCP PXE requests by target hosts. For additional information, see "Issues Related to PXE/DHCP/BINLSVC" on page 167.

The following error message is displayed.

```
CLIENT MAC ADDR: 00 09 3D 12 D4 16  GUID: DF5EF8FE E568 11D9 811E 0060B0B37D21
CLIENT IP: 172.0.0.1  MASK: 255.255.255.0  DHCP IP: 172.0.0.3
PXE-E55: ProxyDHCP service did not reply to request on port 4011.

PXE-M0F: Exiting Broadcom PXE ROM
```

Workaround: Choose one of the following workarounds.

- On the Windows RIS server, enable the provisioning interface, and disable all other interfaces.
- Change the order of the network connections to be accessed by network services on the Windows RIS server. Follow these steps.
  1. On the Windows RIS server, open the Control Panel window from the Start menu.
  2. In the Control Panel window, click Network Connections.
  3. In the Network Connections screen, select the Advanced tab.
  4. Click the Advanced Settings button.
  5. Click on the Adapters and Bindings tab.
  6. Change the order of connections to be accessed by the network services. Ensure that the provisioning interface is listed as the first interface.
  7. Stop and restart the BINLSVC. You can perform this operation by running the Services GUI, or by typing the following commands in an MS-DOS command window.

     ```
     > net stop binlsvc
     > net start binlsvc
     ```

SPARC: Invalid Errors Displayed During Detailed Preflight of Solaris OS Provisioning Server Create Plan (6376733)

Description: If you create an OS Provisioning server on a SPARC based system that is running the Solaris 9 release, 16 invalid warning messages are displayed if you select the Run Detailed Preflight option for the plan.

The warning messages use the following format.

```
Error verifying native procedure during preflight for the command
"/opt/SUNWn1sps_en/N1_Service_Provisioning_System/agent/data/systemcomps/com.sun.n1osp/pkgaddutil
/opt/SUNWn1sps_en/N1_Service_Provisioning_System/agent/data/systemcomps/com.sun.n1osp
adminfile package_name". (026084)
 User "root" does not have execute permissions for command
"/opt/SUNWn1sps_en/N1_Service_Provisioning_System/agent/data/systemcomps/com.sun.n1osp/pkgaddutil".
(026079)
```

In the previous example, *package_name* refers to a package that the warning indicates cannot be added to the OS Provisioning server.

Workaround: Ignore the invalid error messages. The packages are added to the OS Provisioning server, and the server is created successfully.

TFTP Server Not Started Automatically on Linux Boot and Install Servers (6382810)

Description: While provisioning the Linux OS by using a Solaris boot and install server, the TFTP service is not automatically started. You need to manually start the TFTP service on the boot and install server to provision your target hosts.

Workaround: Perform these steps on the boot and install server, with the OS distribution media accessible.

1. Plumb the IP addresses.
2. Export the NFS shares.
3. Start the TFTP services.

See the system administration documentation for your Solaris release for more information about how to start TFTP services.

Cancelling OS Image Import and Provisioning Plans Does Not Cancel Plan (6379004)

Description: If you cancel an OS image Import plan or a Provision OS plan, the plan operations continue in the background. This issue occurs under the following circumstances.

- **OS Image Import** - If you cancel an OS Image Import plan before the image is fully imported, the plan is cancelled, but the import operation continues in the background until the image is imported.
- **Provision OS** - If you configure the OS Provisioning server to perform provisioning operations in the foreground, and then cancel a Provision OS plan before the OS is provisioned, the provisioning continues in the background.

Workaround: Kill the import or provisioning process manually on the OS Provisioning server or boot and install server.

Cannot View or Manage OS Provisioning Components After Upgrading the OS Provisioning Plug-In (6391643)

Description: If you upgrade to the OS Provisioning Plug-In 3.0, you cannot view or manage installed OS Provisioning components in the N1 SPS browser interface.

Workaround: Use the Version History link on each Component Details page to view details about OS Provisioning components that were created in previous releases. For more information, see "How to Access Previous Versions of a Component " on page 49.

OS Profile Components Not Updated Properly After Upgrading the OS Provisioning Plug-In (6392983)

Description: If you recreate OS profile components after you upgrade to the OS Provisioning Plug-In 3.0, the component might include variable settings that are no longer included in the current plug-in version. The component variables are not properly updated after you recreate the component, and provisioning plans that reference this component might fail. An error that is similar to the following message is displayed.

```
The plan (or preflight) "/system/autogen/Solaris9_7_
sparc.standard-inst-provision_start-1141144282798" finished
with 1 failed host(s). (017034)

Unable to find accessible variable in component referenced by
substitution variable :[container:sysidcfg_network_interface_base_]. (04011
```

In the previous example, the plan fails because the OS Provisioning Plug-In 3.0 does not include the `sysidcfg_network_interface_base_config_` variable in the Solaris OS profile component.

Workaround: Choose one of the following workarounds.

- After you upgrade the OS Provisioning Plug-In, create a new OS profile component, rather than recreating the previous profile.
- Delete the previous OS profile component and create a new OS profile with the valid variable settings.
- Remove values for the deprecated variables from the old OS profile component.

Cannot Recreate Linux Image Server or Windows Image Server After Upgrading to OS Provisioning Plug-in 3.0 (6391832)

Description: If you try to recreate a Linux or Windows Image Server after you upgrade to the OS Provisioning Plug-in 3.0, and you change the value of the `installPath` variable, the recreation fails. An error similar to the following message is displayed.

```
Problems encountered during plan run or preflight

The plan (or preflight) "/com/sun/n1osp/untyped/LinuxServer-create"
 finished with 1 failed host(s). (017034)

This host has already been reserved or already exists. (011220)
```

Workaround: Choose one of the following workarounds.

- If the `installPath` variable value from the older version of the image server is still valid, do not change the value for the `installPath` variable.
- If you need to change the value of the installPath variable for the OS Image server, create a new OS Image server as described in the following procedures.
  - "How to Create the RedHat Linux Image Server (Browser Interface)" on page 84
  - "How to Create the SuSe Linux Image Server (Browser Interface)" on page 101
  - "How to Set Up the Windows RIS Server" on page 118

  When you create the new OS Image server, change the following plan variable settings.

  - Set the `installPath` variable to the valid value.
  - Edit the `linuxHost` or `windowsHost` variable to rename the virtual host for this plan. For example, `:[target:sys.hostName]-1-linux`.

Provisioning Non-English Windows OS Image Fails (6400419)

Description: If you try to provision a Windows OS image for a language other than English, the installation fails. This failure occurs because by default, the Microsoft RIS service looks for the English OSChooser files. An error that is similar to the following message is displayed.

```
The system cannot find the path specified.

  An error occurred on the server. Please notify your administrator.
  E:\RemoteInstall\OSChooser\English\LOGIN.OSC
```

Workaround: Choose one of the following workarounds.

- Modify the OSChooser files. Follow these steps.
  1. Edit the `multilng.osc` file to list the language for the image as the first entry in the file.
  2. In the `multilng.osc` file, add the following line before the `<TITLE>` tag.

     ```
     <META ACTION=AUTOENTER>
     ```
  3. Copy the `multilng.osc` to the `welcome.osc` file in the same directory.
  4. Edit the \OSChooser*language*\`welcome.osc` to include the following text line before the `<TITLE>` tag.

     ```
     <META ACTION=AUTOENTER>
     ```
- Copy the OSChooser files from the non-English language directory to the English directory.

  ```
  % cd RemoteInstall\OSChooser
  % copy -r Language English
  ```

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.

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.

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.

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.

CHAPTER 3

OS Provisioning Deployment Environment

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

- "Prerequisites" on page 35
- "Configuring New Environments" on page 40
- "Configuring Existing Environments" on page 41

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 SPS Master Server — A system upon which the N1 Service Provisioning System software is installed and that is configured to run as a Master Server.
- OS provisioning server — A Solaris or RedHat 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. For a list of supported OS releases for the OS Provisioning server, see "OS Provisioning Components" on page 24.

  For Solaris OS provisioning servers, you must install the Java 1.4.2 release on the system before you create the OS Provisioning server. The Java 1.4.2 release must be located, or linked to, the `/usr/bin/java` directory.

  For RedHat OS provisioning servers, you must install the Java 1.4.2 32–bit release and the `dhcp-3.01–10_EL3` RPM on the system before you create the OS Provisioning server. The Java 1.4.2 release must be located, or linked to, the `/usr/bin/java` directory. If a 64–bit Java version is installed on the system, link `/opt/SUNWn1osp/lib/libgridos.so` to `/opt/SUNWn1osp/lib/libgridos_rhel3_x86_64.so`.

- Boot and install server – The server that provides the OS images to provision to the target host. Choose the type of boot and install server that is appropriate for your environment.
  - Solaris boot and install server — To provision Solaris systems, you need a Solaris boot and install server.
  - Linux boot and install server — To provision Linux systems, you need a Linux or Solaris 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 with Service Pack 1 is installed.

  You must configure the network interfaces on the boot and install server before you can provision your target hosts with the provisioning software.
- 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 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 9.

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.

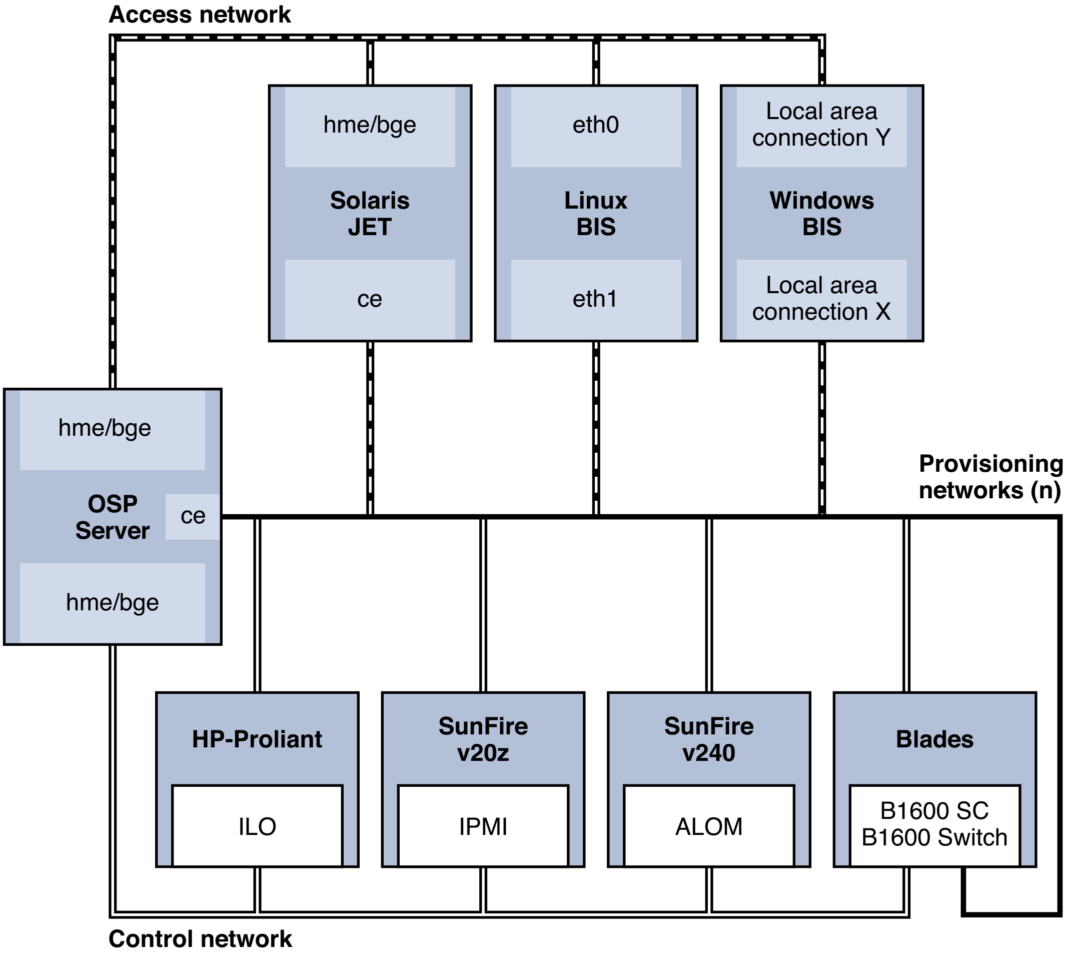

FIGURE 3–1 Network Environment Diagram for OS Provisioning

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 SPS security model. Most communication between the different servers occurs through the N1 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 159.

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 Set Up the Windows RIS Server" on page 118.

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 SPS Master Server, OS provisioning server, and boot and install servers.
2. Obtain the N1 SPS software.
3. Install the N1 SPS Master Server, as explained in "Installing the Sun N1 Service Provisioning System 5.2" in *Sun N1 Service Provisioning System 5.2 Installation Guide*.
4. Install the N1 SPS RA and N1 SPS command-line interface (CLI) on the OS provisioning server
5. Install the N1 SPS RA and N1 SPS CLI on the Solaris boot and install server
6. Install the N1 SPS RA and N1 SPS CLI on the Linux boot and install server
7. Install the N1 SPS RA on the Windows boot and install server
8. Prepare the RAs on the OS provisioning server, Solaris boot and install server, Linux boot and install server, and Windows boot and install server. For information, see "How to Prepare a Physical Host" in *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

Note – For safety, back up the N1 SPS database. For information, see Chapter 9, "Backing Up and Restoring," in *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

▼ How to Enable the Master Server to Use Session IDs

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

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

```
/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/server/config/config.properties
```

2 If this is an existing N1 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 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_Service_Provisioning_System_5.2/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 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 24 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. Configure the network interfaces on the boot and install server.

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.

If you want to use your network's DHCP service to install target hosts, you can disable the DHCP service provided by the OS provisioning server. You must ensure that your DHCP service is properly configured to respond to DHCP requests by the target hosts. For more information about how to disable the OS Provisioning DHCP service, see "Disabling the Plug-In Provided DHCP" on page 283.

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 9

N1 Service Provisioning System Software

Ensure that the N1 SPS software is version 5.1 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 59

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 59. 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 SPS software, , as explained in *Sun N1 Service Provisioning System 5.2 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

Acquiring the OS Provisioning Plug-In is a two-step process. First, you must add the package file that contains the OS Provisioning Plug-In JAR file to your system. Then you must import the OS Provisioning Plug-In JAR file. Perform the following steps to add the plug-in to your Master Server.

1. Add the file containing the JAR file:

   The OS Provisioning Plug-In is packaged as a *plug-in* to the N1 Service Provisioning System software. The plug-in files for the OS Provisioning Plug-In are available from the N1 SPS 5.2 DVD or from the Sun Download Center (`http://sunsolve.sun.com`).

   For instructions about how to add the plug-in package to your system, see the following procedures.

- "Adding the OS Provisioning Plug-In for Solaris" on page 46
- "Adding the OS Provisioning Plug-In for Linux" on page 46
- "Adding the OS Provisioning Plug-In for Windows" on page 47

2. Import the JAR file.

   Once the package file is added to your system, the OS Provisioning Plug-In is available for import from two different JAR files. Choose the correct file depending on your situation.

   - If you are importing the OS Provisioning Plug-In for the first time, acquire the *n1-basedir*/`com.sun.n1osp/com.sun.n1osp_3.0.jar` file.
   - If you have already imported the previous version of the OS Provisioning Plug-In, acquire the *n1-basedir*/`com.sun.n1osp/Upgrade/com.sun.n1osp_2.0_3.0.jar` file.

   For instructions about how to import the plug-in JAR file, see "Adding the OS Provisioning Plug-In to N1 SPS" on page 47.

Adding the OS Provisioning Plug-In for Solaris

The plug-in product name plug-in is contained in the `SUNWspsosp` package.

▼ To Add the OS Provisioning Plug-In Package for Solaris

1. **In a terminal window, become superuser.**

2. **Move to the directory containing the plug-in package.**

3. **Type the following command and press Return.**

   ```
   # pkgadd -d package_directory SUNWspsosp
   ```

   The standalone JAR file is in the `/opt/SUNWn1sps/plugins/com.sun.n1osp/` directory. The upgrade JAR file is in the `/opt/SUNWn1sps/plugins/com.sun.n1osp/Upgrade` directory.

Adding the OS Provisioning Plug-In for Linux

The plug-in product name plug-in is contained in the `sun-spsosp-3.0-1.noarch.rpm` file.

▼ To Add the OS Provisioning Plug-In Package for Linux

1. **In a terminal window, become superuser.**

2. **Move to the directory containing the `sun-spsosp-3.0-1.noarch.rpm` file.**

3. **Type the following command and press Return.**

   ```
   # rpm -i package_directory sun-spsosp-3.0-1.noarch.rpm
   ```

The standalone JAR file is in the `/opt/sun/N1_Service_Provisioning_System/plugins/com.sun.n1osp/` directory. The upgrade JAR file is in the `/opt/sun/N1_Service_Provisioning_System/plugins/com.sun.n1osp/Upgrade` directory.

Adding the OS Provisioning Plug-In for Windows

The plug-in product name plug-in is contained in the `sun-spsosp-3.0.msi` Microsoft Installer (MSI) package file.

▼ To Add the OS Provisioning Plug-In MSI File for Windows

1 **Move to the directory containing the `sun-spsosp-3.0.msi` file.**

2 **Double-click the `sun-spsosp-3.0.msi` file.**

The Installer GUI starts. The JAR file is copied to the `c:\Program Files\N1 Service Provisioning System\plugins\com.sun.n1osp` directory.

Adding the OS Provisioning Plug-In to N1 SPS

To make a given plug-in known to the N1 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 *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

1. In the Administrative section of the N1 SPS browser interface main window, click Plug-ins.
2. In the Action column of the Plug-ins page, click Import.
3. Browse to the location of the 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.

- If you have not previously used the OS provisioning plug-in, use the following command to import the plug-in:

```
% cr-cli -cmd plg.p.add -path com.sun.n1osp_3.0.jar -u username -p password
```

- If you already have an older version of the OS provisioning plug-in, use the following command to import the plug-in:

```
% cr-cli -cmd plg.p.add -path com.sun.n1osp_2.0_3.0.jar -u username -p password
```

▼ How to Upgrade the OS Provisioning Plug-In

To upgrade the OS provisioning plug-in from version 2.0 to version 3.0, follow these steps:

1. **Upgrade the N1 SPS software to version 5.2, as explained in Chapter 10, "Upgrading to the Sun N1 Service Provisioning System 5.2," in** *Sun N1 Service Provisioning System 5.2 Installation Guide***.**

2. **Acquire and import the** `com.sun.n1osp_2.0_3.0.jar` **file.**

   For instructions about how to install the OS Provisioning Plug-In, see "Installing the OS Provisioning Plug-In" on page 45.

3. **Upgrade the OS provisioning (control) server.**

   You must use the OS provisioning plug-in 3.0 software to recreate any existing OS provisioning servers. See "Creating the OS Provisioning Server" on page 50

4. **Upgrade the OS image servers in your environment.**

   You must use the OS provisioning plug-in 3.0 software to recreate any existing OS image servers.

   - **For instructions about how to create a JET Solaris image server, see "How to Create the JET Solaris Image Server" on page 59.**
   - **For instructions about how to create a Red Hat Linux image server, see "Creating a RedHat Linux Image Server" on page 84.**
   - **For instructions about how to create JET SuSe Linux image server, see "Creating a SuSe Linux Image Server" on page 101.**
   - **For instructions about how to create a Windows image server, see "Setting Up the Windows Server" on page 118.**

   Note – When you recreate the boot and install servers, you must use the value that you used previously for the `installPath` variable. If the previous `installPath` variable value is no longer valid, you must create a new boot and install server with the valid `installPath` variable value.

   ---

5. **Change the variable for the remote agent in any existing OS profiles.**

   The `product_version_spsra` variable in the profile must reflect the current N1 SPS RA software version. Edit any existing profiles and change the value of the `product_version_spsra` variable to `5.2`. For more information about this variable, see "Component Variables for Solaris Remote Agents" on page 219.

Troubleshooting

- If the OS Provisioning Plug-In package for the Solaris OS fails to install, remove the package, then try to install the package again.

- If you want to use new variables in the OS Provisioning Plug-In 3.0, you must recreate the target hosts in your N1 SPS environment. For more information about how to create target hosts, see Chapter 9.

  If you prefer to use your existing target host definitions, you can continue to provision these hosts with the OS Provisioning Plug-In 3.0.

Using the OS Provisioning Plug-In 3.0 With the N1 SPS 5.1 Release

Starting with the release N1 SPS 5.2 release, the OS Provisioning Plug-In 3.0 uses default values for N1 SPS 5.2 paths. If you are using the N1 SPS 5.1 release, the OS Provisioning Plug-In, you must edit the `sps_cli` variable value in your plans to enable the OS Provisioning Plug-In 3.0. Edit this value in the following procedures.

- "Creating the OS Provisioning Server" on page 50
- "Setting up the Solaris JET Server" on page 59
- "Creating a RedHat Linux Image Server" on page 84
- "Creating a SuSe Linux Image Server" on page 101

▼ How to Access Previous Versions of a Component

When you upgrade to a new version of a plug-in, the Common Tasks page for the plug-in is updated to provide links to components that are installed with the new plug-in version. This feature enables you to easily view components that contain the most current features and improvements. Components that rely on old functionality are not linked to from the Common Tasks page.

If you have trouble finding a component that you previously installed, the component was likely created from the previous version of the plug-in. Follow these steps to view or work with a previous version of the component:

1. **Click the appropriate component procedure on the Common Tasks page.**

   The Component Detail page appears.

2. **On the Component Details page, click Version History.**

   A list of components and versions appears.

3. **Click the appropriate link.**

   - **To perform a task with a component, click the version number or the Details link that applies to the component that you want to use.**

   - **To view where a component is installed, click Where Installed.**

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.

Before You Begin

- For OS Provisioning servers that are running the Solaris OS, you must install the Java 2 SE 1.4.2 release on the system before you create the OS Provisioning server. The Java 2 SE 1.4.2 release must be located in, or linked to, the `/usr/bin/java` directory.
- For OS Provisioning servers that are running RedHat Enterprise Linux, you must install the Java 2 SE 1.4.2 32–bit release and the DHCP RPM for your RedHat distribution on the system before you create the OS Provisioning server. The Java 2 SE 1.4.2 release must be located in, or linked to, the `/usr/bin/java` directory. If a 64–bit Java version is installed on the system, link `/opt/SUNWn1osp/lib/libgridos.so` to `/opt/SUNWn1osp/lib/libgridos_rhel3_x86_64.so`. Verify that the host name of the OS Provisioning server resolves to a primary interface on the system, and not a loopback address.

1. **In the Common Tasks section of the N1 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.
`run_provisioniing_foreground`	Specifies whether to run the provisioning operation in the foreground. If you set this variable value to `true`, all deployments are run in the foreground. You cannot reset this value for individual deployments. **Note –** To enable this feature, you must disable locking of physical hosts, which is an unsupported configuration. For more information about how to disable physical locking of hosts, see "Running Deployment Jobs in the Foreground" on page 284.
`java_bin`	Path to the Java binary.
`dhcpd_bin`	Path to the DHCP daemon binary.
`sps_cli`	Location of the N1 SPS command line interface (CLI) binaries. **Note –** If you are using the OS Provisioning Plug-In 3.0 with the N1 SPS 5.1 release, you must edit this value to point to the location of the `cr_cli` directory in your environment. Do not accept the default value.

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`.

See Also In some environments, you might want to use your network's DHCP server rather than the ISC DHCP server that is created and started after you complete the previous procedure. If you want to disable the ISC DHCP server that is set up by the provisioning software, see "Disabling the Plug-In Provided DHCP" on page 283.

Troubleshooting If your OS provisioning server is located on the same physical host as the boot and install server, you must configure the interfaces on the system before your provision any target hosts. The OS provisioning server does not automatically configure the interfaces of the boot and install server.

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

◗ **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,H:NM:bikickstart  -comp - -vs + -pto 30 -nto 10
```

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

See Also In some environments, you might want to use your network's DHCP server rather than the ISC DHCP server that is created and started after you complete the previous procedure. If you want to disable the ISC DHCP server that is set up by the provisioning software, see "Disabling the Plug-In Provided DHCP" on page 283.

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 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.

1. **In the Common Tasks section of the N1 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`

Variable	Description	Example
`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:

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 53.

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. Configure the network interfaces on your boot and install server.

1 **Create a new subnet component as described in "How to Identify the Subnet for the OS Provisioning Server (Browser Interface)" on page 53.**

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 System

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 Solaris OS Configuration Variables” on page 207.
- spsra – Installs and configures the N1 SPS Remote Agent (RA) on a Solaris system. For information about spsra variables, see “Component Variables for Solaris Remote Agents” on page 219.
- 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 77.

Zones Support

The JET functionality that is provided with the OS Provisioning Plug-In is Solaris zone-aware and can be installed on a global zone without affecting any non–global zones. Because non–global zones do not currently support NFS share exports, JET is not supported on non–global zones

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

To provision the Solaris Operating System, you must perform these high-level tasks:

1. Set up the JET server.
2. Create the Solaris image or attach an existing image to the JET server.
3. Create a provisioning profile.

4. Create a target host.
5. 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 SPS browser interface or command-line interface. 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/Jet-create \
-tar H:NM:biss1 -comp - -vs solaris8 + -pto 30 -nto 10
```

Substitute the appropriate release value for the `solaris8` argument.

To create the server from the browser interface, follow these 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 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 61 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.**

- **If you want to use another component's variable settings, click Import Set From Component.**

The Import Variable Settings window displays.

a. **If necessary, navigate to the Folder that contains the component with the variable settings you want to import.**

b. **Select the component version.**

Note – Variable settings can vary between component versions. Ensure that the current components and the component from which you want to import variable settings share common variables. If the component from which you want to import variable settings does not share common variables with the component you want to use in your plan, the variable settings are not imported.

---

c. **Click Import Variable Settings.**

The variables settings are imported, and are displayed in the table.

d. **On the Plan Details Run page, select the variable settings that you imported from the Variable Settings drop-down list, then click Select.**

- **If you want to use component variable settings that are stored in a file, follow these steps.**

a. **In the Import Sets from File text field, enter the path to the variable settings file that you want to use.**

To browse through the file system to find the appropriate file, click the Browse button.

b. **Click Import.**

The variables settings are imported, and are displayed in the table.

Note – If the file from which you want to import variable settings does not share common variables with the component you want to use in your plan, the variable set is not imported.

---

c. **On the Plan Details Run page, select the variable settings that you imported from the Variable Settings drop-down list, then click Select.**

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`.

Troubleshooting If your OS provisioning server is located on the same physical host as the boot and install server, you must configure the interfaces on the system before you provision any target hosts. The OS provisioning server does not automatically configure the interfaces of the boot and install server.

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 SPS command-line interface.	`/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/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_`

Importing Solaris Patches to the JET Solaris Image Server

If you have acquired Solaris patches, and the JET Solaris image server can access those patches through NFS, you can import those patches to the JET Solaris image server. You can then add these patches to your provisioning plans.

▼ How to Import Solaris Patches to the JET Solaris Image Server

1. **In the Common Tasks section of the N1 SPS browser interface, 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, select the checkbox for `import_product_patches` in the Component Procedures table.**

4. **In the `import_product_patches` row, click Run in the Actions column.**

5. **On the Plan Details Run page, select the boot and install server to which you want to import the patches from the Current Installations list.**

6. **Click Run Selected Installations.**

7. **On the Plan Details Run page, specify the variables to use with this plan.**

   a. **In the JET Module Name field, specify the spsra module.**

   b. **Specify the product version.**

      For example, `5.2`.

8. **Specify the architecture of the patches.**

9. **In the Media Path field, specify the path to the patches.**

10. **Click Run (includes preflight).**

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.

You can import images created by the `setup_install_server` command, as well as ISO images. Edit the plan variable settings to specify which images to import.

Before You Begin If you want to fully automate the deployment of the Solaris 9 OS, you must create additional configuration files to accompany your Solaris 9 installation image. For more information, see "Automating Solaris 9 Installations on x86 Based Systems" on page 217.

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 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_path`	Path to where imported image will be stored.	`/export/osp_image/:[installPath]`
`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`

Variable Name	Description	Default Value
ISO_files	Space-separated names of ISO files in the media_src directory. The files must be unzipped. Specify the bootable CD as the first ISO in the list. **Note** – You cannot import cross-platform CD ISO images of the Solaris OS. For example, you cannot import Solaris x86 CD ISO files to a boot and install server running the Solaris OS for SPARC. Solaris SPARC CD ISO files cannot be imported to a Solaris x86 boot and install server.	sol-10-GA-sparc-v1-iso sol-10-GA-sparc-v2-iso sol-10-GA-sparc-v3-iso sol-10-GA-sparc-v4-iso

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).**

Creating Solaris Provisioning Profiles With JET

The JumpStart Enterprise Toolkit (JET) is a set of utilities that simplifies the process of creating the profiles necessary to provision the Solaris OS. Through the N1 SPS browser interface, you can generate the profile keywords and values that are required to perform an automated JET installation of your target host.

If you prefer to not use JET, and create the profile values manually, you can create the Solaris Provisioning profile without using JET modules. See "Creating Non-JET Solaris Provisioning Profiles" on page 69.

How to Create a Solaris Provisioning Profile With JET (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 59. For more information about creating the Solaris image, see "How to Import a Solaris Image" on page 63.
- If you are provisioning the Solaris OS on x86 target hosts, you must create a custom JumpStart profile that deletes any existing partitions, and point to that profile in the `profile_base_config` variable. For instructions about how to create a JumpStart profile that deletes existing partitions with the `fdisk` keyword, see *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

1. **In the Common Tasks section of the N1 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/Example
`architecture`	Architecture of the physical system, for example, SPARC or x86	`sparc` (Default)

Variable Name	Description	Default Value/Example
`installPath`	Solaris image name to attach	`Solaris:[version]_ :[release]_:[architecture]` For example: `Solaris9_u7_sparc`
`image_path`	Path to location where image file resides	`/export/osp_image/ :[installPath]` (Default)
`image_subnet_addr`	Subnet address of the image server	`:[target(/):sys.ipAddress]` (Default)
`image_subnet_mask`	Subnet mask of the image server	`255.255.255.0` (Default)
`media_src`	Path to the image files.	`/cdrom/cdrom0/s0` (Example)

`ISO_files`	List of ISO files located in the directory specified in the `media_src` variable	`sol-10-GA-sparc-v1-iso`
		`sol-10-GA-sparc-v2-iso`
		`sol-10-GA-sparc-v3-iso`
		`sol-10-GA-sparc-v4-iso`
		(Example)

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`.

By default, the plan variable `JET Modules Name(s)` lists two JET modules to include in the Solaris profile component:

- `base_config` – Installs and configures the Solaris OS. For information about `base_config` variables, see "Basic Solaris OS Configuration Variables" on page 207.
- `spsra` – Installs and configures the N1 SPS Remote Agent (RA) on a Solaris system. For information about `spsra` variables, see "Component Variables for Solaris Remote Agents" on page 219.

base_config is the required JET module for basic Solaris installation. All other /opt/SUNWjet/Products modules are optional. Adding a module name to this list adds the module installation function and configuration variables to the newly created Solaris provisioning profile.

One of the /opt/SUNWjet/Products optional modules is the custom module. The custom module installs arbitrary lists of Solaris packages, patches, and files, and can run arbitrary collections of scripts. See "Using the custom Module" on page 77 for further details. You can also write other JET modules. For information about creating JET modules, see Appendix F.

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.**

For information about Solaris profile variables, see Appendix B.

Note – If you are provisioning the Solaris OS to an x86 target host, you must create a custom JumpStart profile that deletes all existing partitions on the target host. For the profile_base_config variable value, point to the location of this profile

---

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 With JET (Command-Line Interface)

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 "solaris10sparc" -u admin -p admin -vars "version=10;release=ga;architecture=sparc; \
image_path=/export/install/s10ga-sparc;image_subnet_addr=10.42.42.2; \
image_subnet_mask=255.255.255.0;media_src="
```

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/SolarisImage-create-profile
-tar H:NM:biss1-jet -comp - -vs solaris10sparc -pto 30 -nto 10
-f /tmp/solaris-profile
```

Note the default N1 SPS user value you specified with the -u option. This user must exist after the OS installation for the N1 SPS Remote Agent software to start on the target host.

Creating Non-JET Solaris Provisioning Profiles

If you prefer to manually create the Solaris provisioning profiles, you can create the Solaris profile without using JET modules. This section describes how to create non-JET Solaris profiles.

If you prefer to use JET to generate your profiles, see "Creating Solaris Provisioning Profiles With JET" on page 65.

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

In certain circumstances, you might want to manually create the Solaris Provisioning profile, rather than generating a profile with JET. For example, if you have an existing custom JumpStart environment that you want to import into your N1 SPS environment, you need to manually create the OS Provisioning profile to include the information about your JumpStart files.

Before You Begin Make sure the Solaris image server exists and that the Solaris operating system software is available to that server. For more information about creating the Solaris image server, see "Setting up the Solaris JET Server" on page 59.

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 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 variables and change them, if needed.**

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

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

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 Solaris image server on which to create the profile.**

Tip – The host name of the Solaris image server ends in `-solaris`.

---

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

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

8 Deselect the checkbox for Use the Jet Modules Specified Above.

9 Click Run Plan (includes preflight).

10 To verify that the Solaris profile is created to the Solaris image server, click Components in the left pane of the N1 SPS browser interface.

You should find a `provision` component in the folder `/com/sun/n1osp/autogen-`*server-name*`-solaris/provision/`, where *server-name* is the Solaris server name that you provided in the variable set. For example: `/com/sun/n1osp/autogen-Solaris9_u7_sparc-solaris/provision/`.

11 (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.

For information about Solaris profile variables, see Appendix B.

Note – If you are provisioning the Solaris OS to an x86 target host, you must create a custom JumpStart profile that deletes all existing partitions on the target host. For the `profile_base_config` variable value, point to the location of this profile

---

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

ii. Change values.

Note – If you want to use an existing JumpStart environment for your provisioning plan, set the values for the `osp_profile_jumpstart_ip_address` and `osp_profile_jumpstart_directory` variables to point to the directory and IP address of your JumpStart files.

---

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

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

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; \
image_path=/export/install/Solaris_9;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-profile` that contains the following entries:**

```
standard-nojet
Standard Solaris
base_config spsra
false
```

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.
- The fourth line in the file corresponds to the Use the Jet Modules Specified Above checkbox in the plan variables section of the browser interface

3 **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/SolarisImage-create-profile
-tar H:NM:biss1-jet -comp - -vs solaris9 -pto 30 -nto 10
-f /tmp/solaris-profile
```

Note the default N1 SPS user value you specified with the `-u` option. This user must exist after the OS installation for the N1 SPS Remote Agent software to start on the target host.

Installing the Solaris 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 Solaris 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.

1. **In the Common Tasks section of the N1 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.

Troubleshooting If you are provisioning the Solaris 10 1/06 OS for x86 based systems, and the installation fails, verify that the /tftpboot/I86PC.Solaris_10–1 folder exists on the boot and install server. If the folder does not exist, create the folder, then reprovision the target host.

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

◗ **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 the Solaris OS Provisioning Status

1 **In the Common Tasks section of the N1 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 **Click Run Plan (includes preflight).**

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

▼ How to Stop a Solaris OS Provisioning Operation

1 **In the Common Tasks section of the N1 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 **Click Run Plan (includes preflight).**

Deleting Solaris Images and Profiles

You might want to periodically delete unused OS images and profiles from your N1 SPS environment.

▼ How to Delete Solaris OS Images

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 Solaris Images section.**

3 **On the Components Details page, select the checkbox for Uninstall: Delete.**

4 **Click Run.**

The Soalris OS image is removed from the N1 SPS environment. The image does not appear in the list of images in the View link for Solaris Images section.

Note – The Solaris OS image is not removed from the boot and install server.

▼ How to Delete Solaris OS Profiles

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 Solaris Images section.**

3 **On the Components page, select the checkbox for the profile that you want to delete.**

4 **Click Delete.**

JET Solaris Server Administration Tasks

▼ How to Import N1 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.

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 **Choose the JET Solaris Image server host on which to import the installer, and click Run Selected Installations.**

5 **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 SPS installer	`5.2`
Architecture	Architecture that applies to RA installer	`sparc`
Media Path	Path to directory that contains the RA installer	`/net/myothersys/export/ra`

6 **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 "Component Variables for Solaris Remote Agents" on page 219.

▼ 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.

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 157. 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

You can include the `custom` module in a Solaris profile component at Solaris Profile creation time. When you create the Solaris profile component, the plan variables include `JET Module Name(s)`.

By default, the Solaris profile component includes two JET modules:

- `base_config` – Installs and configures the Solaris OS. For information about `base_config` variables, see "Basic Solaris OS Configuration Variables" on page 207.
- `spsra` – Installs and configures the N1 SPS Remote Agent (RA) on a Solaris system. For information about `spsra` variables, see "Component Variables for Solaris Remote Agents" on page 219.

The third available module is `custom`, which installs arbitrary lists of Solaris packages, patches, and files, and can run arbitrary collections of scripts. To include the `custom` module in a Solaris profile component, add `custom` to the `JET Module Name(s)` plan variable before you create the Solaris profile.

When items in a `custom` module are 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 about creating JET modules, see Appendix F.

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 58.

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. 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 F.

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 F.

CHAPTER 6

Provisioning the RedHat Linux Operating Environment

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

The chapter contains the following information:

RedHat Kickstart Technology

The OS provisioning plug-in uses RedHat Linux kickstart technology.

RedHat Linux Provisioning Process Overview

1. Set up the RedHatLinux image server.
2. Create the RedHat 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 RedHat Linux Server

The OS provisioning plug-in uses the features provided through the JumpStart Enterprise Toolkit (JET) technology in conjunction with the RedHat Linux Kickstart functionality to provision the RedHat Linux Operating System (OS). When you set up a RedHat Linux server, you create JET modules that run Kickstart operations to provision the OS. For more information about JET, see "JumpStart Enterprise Toolkit (JET) Technology" on page 57.

The RedHat Linux boot and install server includes the following software:

- Linux (tested with RedHat Advanced Server 3.0) or Solaris OS
- TFTP server (`tftp-server` package)
- NFS server (`nfs-utils` package)
- N1 SPS Remote Agent (RA) for Linux. For information, see "How to Prepare a Physical Host" in *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

Preparing Your Network to Provision the RedHat Linux OS

Before you create the RedHat Linux image server, you need to perform the following tasks.

- Set up TFTP access to the RedHat Linux images on your boot and install server.

  For instructions, see "How to Set up TFTP Access" on page 82.
- Create the RedHat Linux OS installation image.

  For instructions, see "How to Copy the RedHat Linux Files" on page 83.

▼ How to Set up TFTP Access

If you use JET to create your RedHat Linux image server, the TFTP service is automatically configured to support provisioning. You need to create or copy additional directories and files to the TFTP root directory on the boot and install server. The OS distribution should be accessible through NFS for each subnet served from this boot and install server.

Before You Begin Be sure that the appropriate TFTP server software is installed on the boot and install server.

1 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.

  For instructions about how to create the PXE configuration file, see "How to Create a PXE Configuration File for RedHat Linux OS Provisioning" on page 91.

Note – This task uses `/tftpboot` as an example of the TFTP root directory.

2 **For RedHat Linux provisioning profiles that do not use JET, copy the following files to the TFTP root directory**

- `vmlinuz` – The installation kernel found under the distribution, in `/images/pxeboot`.
- `initrd.img` – The installation RAM disk found under the distribution, in `/images/pxeboot`.

If you want to support provisioning of multiple releases of the RedHat Linux OS, assign a unique name to this file, such as `vmlinuz_3.0`.

▼ How to Copy the RedHat Linux Files

This procedure describes how to create RedHat Linux OS images by copying the files from the product media. You can locate these copies on a system that is running either the RedHat Linux OS or the Solaris OS. You can also place the physical product media in the CD-ROM drives of a system, and point to these CDs as the image source for your provisioning plans. You are not required to locate these copies on your RedHat Linux boot and install server.

If you locate these copies on a system that is not the boot and install server, the copies must meet the following criteria.

- For RedHat Linux provisioning that uses JET, the boot and install server and target host must be able to access the copies through NFS.
- For RedHat Linux provisioning that does not use JET, the target host must be able to access the copies through NFS.

Follow these steps to create copies of the RedHat Linux files.

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 -o loop /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.**

Creating a RedHat Linux Image Server

After you copy the RedHat Linux images, you need to create the RedHat Linux image server in your N1 SPS environment. You must make the boot and install server an N1 SPS Remote Agent, and add the boot and install server as a Remote Agent to the Master Server.

▼ How to Create the RedHat Linux Image Server (Browser Interface)

To create the server from the browser interface, follow these 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 `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` or the `boot_server_tftp_root_directory` variables.

d. **Save the variables set.**

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

- **If you want to use another component's variable settings, click Import Set From Component.**

  The Import Variable Settings window displays.

  a. **If necessary, navigate to the Folder that contains the component with the variable settings you want to import.**

  b. **Select the component version.**

  Note – Variable settings can vary between component versions. Ensure that the current components and the component from which you want to import variable settings share common variables. If the component from which you want to import variable settings does not share common variables with the component you want to use in your plan, the variable settings are not imported.

  ---

  c. **Click Import Variable Settings.**

  The variables settings are imported, and are displayed in the table.

  d. **On the Plan Details Run page, select the variable settings that you imported from the Variable Settings drop-down list, then click Select.**

- **If you want to use component variable settings that are stored in a file, follow these steps.**

  a. **In the Import Sets from File text field, enter the path to the variable settings file that you want to use.**

  To browse through the file system to find the appropriate file, click the Browse button.

  b. **Click Import.**

  The variables settings are imported, and are displayed in the table.

  Note – If the file from which you want to import variable settings does not share common variables with the component you want to use in your plan, the variable set is not imported.

  ---

  c. **On the Plan Details Run page, select the variable settings that you imported from the Variable Settings drop-down list, then click Select.**

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`.

These steps install the `SUNWjet`, `JetRedHat`, and `JetSuse` packages on the boot and install server. The `SUNWjet` package sets up the TFTP server and the NFS share for the JET base directory, `/opt/SUNWjet` by default.

Troubleshooting If your OS provisioning server is located on the same physical host as the boot and install server, you must configure the interfaces on the system before your provision any target hosts. The OS provisioning server does not automatically configure the interfaces of the boot and install server.

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

You can create a RedHat Linux image server from the N1 SPS browser interface or command-line interface.

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

◗ **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
```

These commands install the `SUNWjet`, `JetRedHat`, and `JetSuse` packages on the boot and install server. The `SUNWjet` package sets up the TFTP server and the NFS share for the JET base directory, `/opt/SUNWjet` by default.

Troubleshooting If your OS provisioning server is located on the same physical host as the boot and install server, you must configure the interfaces on the system before your provision any target hosts. The OS provisioning server does not automatically configure the interfaces of the boot and install server.

Creating RedHat Linux Images and Profiles

After you create your RedHat Linux image server, you need to create the provisioning images and profiles.

- If you want the JumpStart Enterprise Toolkit (JET) to generate your image and profile, see“Creating RedHat Linux Images and Profiles With JET” on page 87.
- If you want to manually create your image and profile, see“Creating Non-JET RedHat Linux Images and Profiles” on page 91.

Creating RedHat Linux Images and Profiles With JET

The JumpStart Enterprise Toolkit (JET) is a set of utilities that simplifies the process of creating the profiles necessary to provision the RedHat Linux OS. Through the N1 SPS browser interface, you can generate the profile keywords and values that are required to perform an automated Kickstart installation of your target host.

If you prefer to not use JET, and create the profile values manually, you can create the RedHat Linux image and profile without using JET modules. See “Creating Non-JET RedHat Linux Images and Profiles” on page 91.

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

Before You Begin Make sure the Linux image server exists and that the RedHat Linux operating system software is available to that server. For more information about creating the RedHat Linux image server, see “Setting up the RedHat Linux Server” on page 82.

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 RedHat 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.**

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

Variable Name	Description	Default Value/Example
`osp_server`	Virtual server representing the OS provisioning service	`masterserver-osp` (Default)
`version`	Version of Linux to be deployed by this component	`3U4` (Example)
`architecture`	Architecture of the physical system, for example, x86 **Note –** This value is used to name the profile component and OS image. The value of this variable is not validated or enforced by the provisioning software.	`x86` (Default)
`installPath`	RedHat Linux image name to attach	`Linux:[version]_ :[architecture]` For example: `Linux_3U4_x86`
`description`	Optional description for this profile	`test linux` (Example)
`min_disk_size`	Minimum disk space required to provision the RedHat Linux OS	`10GB` (Default)
`boot_server_name`	Name of RedHat Linux boot and install server	`masterserver` (Example)
`boot_server_subnet_address`	Subnet served by the boot and install server	`10.10.31.0` (Example)
`boot_server_subnet_ ip_address`	IP address of the boot and install server on the subnet	`10.10.31.1` (Example)
`image_subnet_mask`	Netmask of the provisioning subnet	`255.255.255.0` (Default)
`kickstart_file` (Non-JET profiles only)	Absolute path of the Kickstart file `ks.cfg` on the boot server Do not specify a value for this variable for JET-generated profiles.	N/A

Variable Name	Description	Default Value/Example
`pxe_boot_file` (Non-JET profiles only)	Relative path from the `/tftpboot` directory of the `pxelinux.0` boot file on the boot server Do not specify a value for this variable for JET-generated profiles.	N/A
`pxe_configuration_file` (Non-JET profiles only)	Relative path from the `pxelinux.cfg` directory of the PXE Linux configuration file on boot server Do not specify a value for this variable for JET-generated profiles.	N/A
`linux_component_folder`	Location of autogenerated Linux Provision component folder	`/com/sun/n1osp/autogen-:[target(..):sys.hostName]-linux` (Default)
`image_path` (JET profiles only)	NFS path to the OS image. Do not specify a value for this variable for non-JET profiles.	`nfs://10.10.31.1/export/install/RHEL3` (Example)
`kernel` (JET profiles only)	Relative path to the kernel in the OS image Do not specify a value for this variable for non-JET profiles.	`images/pxeboot/vmlinuz` (Default)
`initrd` (JET profiles only)	Relative path to `initrd` in the OS image Do not specify a value for this variable for non-JET profiles.	`images/pxeboot/initrd.img` (Default)

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 **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`.

By default, the plan variable `JET Modules Name(s)` lists two JET modules to include in the RedHat Linux profile component:

- `base_config` – Installs and configures the RedHat Linux OS. For information about `base_config` variables, see "Basic RedHat OS Configuration Variables" on page 225.
- `spsra` – Installs and configures the N1 SPS Remote Agent (RA) on a Solaris system. For information about `spsra` variables, see "Component Variables for RedHat Remote Agents" on page 239.

`base_config` is the required JET module for basic RedHat Linux installation. All other `/opt/SUNWjet/Products` modules are optional. Adding a module name to this list adds the module installation function and configuration variables to the newly created RedHat Linux provisioning profile.

9 **Verify that the Use the Jet Modules Specified Above checkbox is selected.**

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

11 **To verify that the Linux profile is created to the Linux image server, click Components in the left pane of the N1 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-Linux3U4_x86-linux/provision/`.

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

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=test linux; \
min_disk_size=10GB;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 \
-f /tmp/redhat-profile
```

Note the default N1 SPS user value you specified with the `-u` option. This user must exist after the OS installation for the N1 SPS Remote Agent software to start on the target host.

Creating Non-JET RedHat Linux Images and Profiles

If you prefer to manually create the RedHat Linux images and profiles, you can create the RedHat Linux image and profile without using JET modules. This section describes how to create non-JET RedHat Linux images and profiles.

If you prefer to use JET to generate your images and profiles, see "Creating RedHat Linux Images and Profiles With JET" on page 87.

▼ How to Create a PXE Configuration File for RedHat Linux OS Provisioning

To deploy non-JET RedHat Linux images to your target hosts, you need to manually create a PXE configuration file. This file contains parameters specific to each hardware platform and RedHat Linux distribution.

1 **In a text editor, create a file with the following text.**

```
default el30 ks
serial 0 9600
label el30
kernel vmlinuz
append linksleep=30 ksdevice=eth0 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 must be all on one 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 is omitted, `eth0` is assumed.

`console=ttyS1,9600` specify console parameters to show the installation output on the console. The parameters vary depending on the target hardware platform.

`load_ramdisk` indicates to use `initrd.img` for the RAM disk. The path for the RAM disk is relative to `pxelinux.0`.

`network` indicates to perform a network installation.

2. **Save this file to the `pxelinux.cfg` subdirectory of the TFTP root directory on the boot and install server.**

   If you want to support provisioning of multiple releases of the RedHat Linux OS, assign a unique name to this file, such as `rh30.cfg`

▼ How to Create a Non-JET RedHat Linux Provisioning Profile (Browser Interface)

Before You Begin Make sure the Linux image server exists and that the RedHat Linux operating system software is available to that server. For more information about creating the RedHat Linux image server, see "Setting up the RedHat Linux Server" on page 82.

Create a PXE Linux configuration file, and locate this file in the `pxelinux.cfg` directory on the boot and install server. For an example of the PXE Linux configuration file, see "How to Create a PXE Configuration File for RedHat Linux OS Provisioning" on page 91.

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 RedHat 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.**

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

Variable Name	Description	Default Value/Example
`osp_server`	Virtual server representing the OS provisioning service	`masterserver-osp` (Default)
`version`	Version of Linux to be deployed by this component	`3U4` (Example)
`architecture`	Architecture of the physical system, for example, x86 **Note** – This value is used to name the profile component and OS image. The value of this variable is not validated or enforced by the provisioning software.	`x86` (Default)
`installPath`	RedHat Linux image name to attach	`Linux:[version]_:[architecture]` For example: `Linux_3U4_x86`
`description`	Optional description for this profile	`test linux` (Example)
`min_disk_size`	Minimum disk space required to provision the RedHat Linux OS	`10GB` (Default)
`boot_server_name`	Name of RedHat Linux boot and install server	`masterserver` (Example)
`boot_server_subnet_address`	Subnet served by the boot and install server	`10.10.31.0` (Example)
`boot_server_subnet_ip_address`	IP address of the boot and install server on the subnet	`10.10.31.1` (Example)
`image_subnet_mask`	Netmask of the provisioning subnet	`255.255.255.0` (Default)
`kickstart_file` (Non-JET profiles only)	Absolute path of the Kickstart file `ks.cfg` on the boot server. This file must be accessible to the boot and install server. Do not specify a value for this variable for JET-generated profiles.	`/tftpboot/v6x/as21.cfg` (Example)

Variable Name	Description	Default Value/Example
`pxe_boot_file` (Non-JET profiles only)	Relative path from the `/tftpboot` directory of the `pxelinux.0` boot file on the boot server Do not specify a value for this variable for JET-generated profiles.	`v6x/pxelinux.0` (Example)
`pxe_configuration_file` (Non-JET profiles only)	Relative path from the `pxelinux.cfg` directory of the PXE Linux configuration file on boot server Do not specify a value for this variable for JET-generated profiles.	`as21-console` (Example)
`linux_component_folder`	Location of autogenerated Linux Provision component folder	`/com/sun/n1osp/autogen-:[target(..):sys.hostName]-linux` (Default)
`image_path` (JET profiles only)	NFS path to the OS image. Do not specify a value for this variable for non-JET profiles.	N/A
`kernel` (JET profiles only)	Relative path to the kernel in the OS image Do not specify a value for this variable for non-JET profiles.	N/A
`initrd` (JET profiles only)	Relative path to `initrd` in the OS image Do not specify a value for this variable for non-JET profiles.	N/A

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 **Deselect the checkbox for Use the Jet Modules Specified Above.**

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

8 **To verify that the Linux profile is created to the Linux image server, click Components in the left pane of the N1 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-Linux3U4_x86-linux/provision/`.

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

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=test linux;min_disk_size=10GB; \
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 provide JET module information, create a file `/tmp/redhat-profile` that contains the following entries:**

```
standard-nojet
Standard Linux
base_config
false
```

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.
- The fourth line in the file corresponds to the Use the Jet Modules Specified Above checkbox in the plan variables section of the browser interface.

3 **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
-f /tmp/redhat-profile
```

Note the default N1 SPS user value you specified with the `-u` option. This user must exist after the OS installation for the N1 SPS Remote Agent software to start on the target host.

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

Before You Begin Ensure that the network interfaces on the boot and install server are properly configured.

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

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.**

Deleting RedHat Linux Images and Profiles

You might want to periodically delete unused OS images and profiles from your N1 SPS environment.

▼ How to Delete RedHat Linux OS Images

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 RedHat Linux Images section.**

3 **On the Components Details page, select the checkbox for Uninstall: Delete.**

4 **Click Run.**

The RedHat Linux OS image is removed from the N1 SPS environment. The image does not appear in the list of images in the View link for RedHat Linux Images section.

Note – The RedHat Linux OS image is not removed from the boot and install server.

▼ How to Delete RedHat Linux OS Profiles

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 RedHat Linux Images section.**

3 **On the Components page, select the checkbox for the profile that you want to delete.**

4 **Click Delete.**

CHAPTER 7

Provisioning the SuSe Linux Operating Environment

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

SuSe AutoYaST Technology

The OS provisioning plug-in uses SuSe AutoYaST technology.

SuSe Linux Provisioning Process Overview

1. Set up the SuSeLinux image server.
2. Create the SuSe 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 SuSe Linux Server

The OS provisioning plug-in uses the features provided through the JumpStart Enterprise Toolkit (JET) technology in conjunction with the AutoYaST functionality to provision the SuSe Linux Operating System (OS). When you set up a SuSe Linux server, you create JET modules that run AutoYaST operations to provision the OS. For more information about JET, see "JumpStart Enterprise Toolkit (JET) Technology" on page 57.

The SuSe Linux boot and install server includes the following software:

- Linux (tested with RedHat Advanced Server 3.0) or Solaris OS
- TFTP server (`tftp-server` package)
- NFS server (`nfs-utils` package)
- N1 SPS Remote Agent (RA) for Linux. For information, see "How to Prepare a Physical Host" in *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

Preparing Your Network to Provision the SuSe Linux OS

Before you create the SuSe Linux image server, you need to perform the following tasks.

- Set up TFTP access to the SuSe Linux images on your boot and install server.

  For instructions, see "How to Set up TFTP Access" on page 100.
- Create the SuSe Linux OS installation image.

  For instructions, see "Copying the SuSe Linux Files" on page 101.

▼ How to Set up TFTP Access

If you use JET to create your SuSe Linux image server, the TFTP service is automatically configured to support provisioning. You need to create or copy additional directories and files to the TFTP root directory on the boot and install server. The OS distribution should be accessible through NFS for each subnet served from this boot and install server.

Before You Begin Be sure that the appropriate TFTP server software is installed on the boot and install server.

1 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.

  For instructions about how to create the PXE configuration file, see "How to Create a PXE Configuration File for SuSe Linux OS Provisioning" on page 108.

Note – This task uses `/tftpboot` as an example of the TFTP root directory.

---

2 For SuSe Linux provisioning profiles that do not use JET, copy the following files to the TFTP root directory

- `linux` – The installation kernel found on the first CD of the distribution, in `boot/loader/`.

- `initrd` – The installation RAM disk found on the first CD of the distribution, in `boot/loader/`.

If you want to support provisioning of multiple releases of the SuSe Linux OS, assign a unique name to this file, such as `linux_9`.

Copying the SuSe Linux Files

To create SuSe Linux OS images, you copy the files from the product media to a boot and install server or other system that is accessible over the network through NFS. You can locate these copies on a system that is running either the SuSe Linux OS or the Solaris OS. You are not required to locate these copies on your SuSe Linux boot and install server.

To copy the SuSe Linux files, you extract the contents of the product CDs into specific directory structure. For instructions about how to copy the SuSe Linux files, see `http://www.suse.com/~ug/AutoYaST_FAQ.html#a6`.

If you locate these copies on a system that is not the boot and install server, the copies must meet the following criteria.

- For SuSe Linux provisioning that uses JET, the boot and install server and target host must be able to access the OS file copies through NFS.
- For SuSe Linux provisioning that does not use JET, the target host must be able to access the OS file copies through NFS.

Creating a SuSe Linux Image Server

After you copy the SuSe Linux images, you need to create the SuSe Linux image server, or boot and install server, in your N1 SPS environment. You must make the boot and install server an N1 SPS Remote Agent, and add the boot and install server as a Remote Agent to the Master Server.

▼ How to Create the SuSe Linux Image Server (Browser Interface)

To create the server from the browser interface, follow these 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 `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` or the `boot_server_tftp_root_directory` variables.

     ---

  d. **Save the variables set.**

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

- **If you want to use another component's variable settings, click Import Set From Component.**

  The Import Variable Settings window displays.

  a. **If necessary, navigate to the Folder that contains the component with the variable settings you want to import.**

  b. **Select the component version.**

     Note – Variable settings can vary between component versions. Ensure that the current components and the component from which you want to import variable settings share common variables. If the component from which you want to import variable settings does not share common variables with the component you want to use in your plan, the variable settings are not imported.

     ---

  c. **Click Import Variable Settings.**

     The variables settings are imported, and are displayed in the table.

  d. **On the Plan Details Run page, select the variable settings that you imported from the Variable Settings drop-down list, then click Select.**

- **If you want to use component variable settings that are stored in a file, follow these steps.**

  a. **In the Import Sets from File text field, enter the path to the variable settings file that you want to use.**

     To browse through the file system to find the appropriate file, click the Browse button.

  b. **Click Import.**

     The variables settings are imported, and are displayed in the table.

     Note – If the file from which you want to import variable settings does not share common variables with the component you want to use in your plan, the variable set is not imported.

     ---

  c. **On the Plan Details Run page, select the variable settings that you imported from the Variable Settings drop-down list, then click Select.**

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`.

These steps install the `SUNWjet`, `JetRedHat`, and `JetSuse` packages on the boot and install server. The `SUNWjet` package sets up the TFTP server and the NFS share for the JET base directory, `/opt/SUNWjet` by default.

Troubleshooting If your OS provisioning server is located on the same physical host as the boot and install server, you must configure the interfaces on the system before your provision any target hosts. The OS provisioning server does not automatically configure the interfaces of the boot and install server.

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

You can create a SuSe Linux image server from the N1 SPS browser interface or command-line interface.

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

◗ **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
```

These commands install the `SUNWjet`, `JetRedHat`, and `JetSuse` packages on the boot and install server. The `SUNWjet` package sets up the TFTP server and the NFS share for the JET base directory, `/opt/SUNWjet` by default.

Troubleshooting If your OS provisioning server is located on the same physical host as the boot and install server, you must configure the interfaces on the system before your provision any target hosts. The OS provisioning server does not automatically configure the interfaces of the boot and install server.

Creating SuSe Linux Images and Profiles

After you create your SuSe Linux image server, you need to create the provisioning images and profiles.

- If you want the JumpStart Enterprise Toolkit (JET) to generate your image and profile, see"Creating SuSe Linux Images and Profiles With JET" on page 104.
- If you want to manually create your image and profile, see"Creating Non-JET SuSe Linux Images and Profiles" on page 108.

Creating SuSe Linux Images and Profiles With JET

The JumpStart Enterprise Toolkit (JET) is a set of utilities that simplifies the process of creating the profiles necessary to provision the SuSe Linux. Through the N1 SPS browser interface, you can generate the profile keywords and values that are required to perform an automated Kickstart installation of your target host.

If you prefer to not use JET, and create the profile values manually, you can create the SuSe Linux image and profile without using JET modules. See "Creating Non-JET SuSe Linux Images and Profiles" on page 108.

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

Before You Begin Make sure the Linux image server exists and that the SuSe Linux operating system software is available to that server. For more information about creating the SuSe Linux image server, see "Setting up the SuSe Linux Server" on page 99.

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 SuSe 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.

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

Variable Name	Description	Default Value/Example
`osp_server`	Virtual server representing the OS provisioning service	`masterserver-osp` (Default)
`version`	Version of Linux to be deployed by this component	`SLES9` (Example)
`architecture`	Architecture of the physical system, for example, x86 **Note** – This value is used to name the profile component and OS image. The value of this variable is not validated or enforced by the provisioning software.	`x86` (Default)
`installPath`	SuSe Linux image name to attach	`SuSeLinux:[version]_:[architecture]` For example: `SuSeLinux_SLES9_x86`
`description`	Optional description for this profile	`test linux` (Example)
`min_disk_size`	Minimum disk space required to provision the SuSe Linux OS	`10GB` (Default)
`boot_server_name`	Name of SuSe Linux boot and install server	`masterserver` (Example)
`boot_server_subnet_address`	Subnet served by the boot and install server	`10.10.31.0` (Example)

Variable Name	Description	Default Value/Example
`boot_server_subnet_ip_address`	IP address of the boot and install server on the subnet	`10.10.31.1` (Example)
`image_subnet_mask`	Netmask of the provisioning subnet	`255.255.255.0` (Default)
`distro_root_directory`	Absolute path of the base directory of the SuSe Linux distribution on the boot server	`nfs://10.10.31.4/Suse/SLES-9-x86-64/installroot` (Example)
`kernel` (JET profiles only)	Relative path to the kernel in the OS image Do not specify a value for this variable for non-JET profiles.	`boot/loader/linux` (Default)
`initrd` (JET profiles only)	Relative path to `initrd` in the OS image Do not specify a value for this variable for non-JET profiles.	`boot/loader/initrd` (Default)
`autoyast_file` (Non-JET profiles only)	Absolute path of the AutoYaST file on the boot server Do not specify a value for this variable for JET-generated profiles.	N/A
`pxe_boot_file` (Non-JET profiles only)	Relative path from the `/tftpboot` directory of the `pxelinux.0` boot file on the boot server Do not specify a value for this variable for JET-generated profiles.	N/A
`pxe_configuration_file` (Non-JET profiles only)	Relative path from the `pxelinux.cfg` directory of the PXE Linux configuration file on boot server Do not specify a value for this variable for JET-generated profiles.	N/A
`SuSelinux_component_folder`	Location of autogenerated SuSe Linux Provision component folder	`/com/sun/n1osp/autogen-:[target(..):sys.hostName]-linux` (Default)

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 **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`.

By default, the plan variable `JET Modules Name(s)` lists two JET modules to include in the SuSe Linux profile component:

- `base_config` – Installs and configures the SuSe Linux OS. For information about `base_config` variables, see "Basic SuSe Linux OS Configuration Variables" on page 243.
- `spsra` – Installs and configures the N1 SPS Remote Agent (RA) on a Solaris system. For information about `spsra` variables, see "Component Variables for SuSe Linux Remote Agents" on page 264.

`base_config` is the required JET module for basic SuSe Linux installation. All other `/opt/SUNWjet/Products` modules are optional. Adding a module name to this list adds the module installation function and configuration variables to the newly created SuSe Linux provisioning profile.

9 **Verify that the Use the Jet Modules Specified Above checkbox is selected.**

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

11 **To verify that the Linux profile is created to the Linux image server, click Components in the left pane of the N1 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-SuSe_SLES9_x86-linux/provision/`.

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

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/SuseLinuxImage \
-name "SLES9" -u admin -p admin -vars "osp_server=masterserver-osp;version=9; \
architecture=x86;installPath=SuSeLinux:[version]_:[architecture]; \
description=SUSE SLES9;min_disk_size=10GB; \
boot_server_name=biautoyast;boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.3; \
distro_root_directory=nfs://10.42.42.3/export/SUSE/SLES-9-x86-64/installroot;\
pxe_boot_file=;pxe_configuration_file=; \
autoyast_file=;kernel=boot/loader/linux;initrd=boot/loader/initrd"
```

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/SuSeLinuxImage-create-profile \
-tar H:NM:bilinux-linux -comp - -vs SLES9 -pto 30 -nto 10 \
-f /tmp/suse-profile
```

Note the default N1 SPS user value you specified with the `-u` option. This user must exist after the OS installation for the N1 SPS Remote Agent software to start on the target host.

Creating Non-JET SuSe Linux Images and Profiles

If you prefer to manually create the SuSe Linux images and profiles, you can create the SuSe Linux image and profile without using JET modules. This section describes how to create non-JET SuSe Linux images and profiles.

If you prefer to use JET to generate your images and profiles, see "Creating SuSe Linux Images and Profiles With JET" on page 104.

How to Create a PXE Configuration File for SuSe Linux OS Provisioning

To deploy non-JET SuSe Linux images to your target hosts, you need to manually create a PXE configuration file. This file contains parameters specific to each hardware platform and SuSe Linux distribution.

1. **In a text editor, create a file with the following text.**

```
default sles
serial 0 9600
label sles
kernel linux
append console=ttyS0,9600 load_ramdisk=1 initrd=initrd textmode=1
```

Where:

- The first line (`default`) indicates to use label `sles` and do a AutoYaST installation
- The third line (`label`) defines a label `sles`.
- 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 must be all on one line. The specific parameters shown here are as follows:

  `console=ttyS0,9600` specify console parameters to show the installation output on the console. The parameters vary depending on the target hardware platform.

  `load_ramdisk` indicates to use `initrd` for the RAM disk. The path for the RAM disk is relative to `pxelinux.0`

2. **Save this file to the `pxelinux.cfg` subdirectory of the TFTP root directory on the boot and install server.**

   If you want to support provisioning of multiple releases of the SuSe Linux OS, assign a unique name to this file, such as `sles9.cfg`

▼ How to Create a Non-JET SuSe Linux Provisioning Profile (Browser Interface)

Before You Begin Make sure the Linux image server exists and that the SuSe Linux operating system software is available to that server. For more information about creating the SuSe Linux image server, see "Setting up the RedHat Linux Server" on page 82.

Create a PXE Linux configuration file, and locate this file in the `pxelinux.cfg` directory on the boot and install server. For an example of the PXE Linux configuration file, see "How to Create a PXE Configuration File for SuSe Linux OS Provisioning" on page 108.

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 SuSe 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.**

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

Variable Name	Description	Default Value/Example
`osp_server`	Virtual server representing the OS provisioning service	`masterserver-osp` (Default)
`version`	Version of Linux to be deployed by this component	`SLES9` (Example)
`architecture`	Architecture of the physical system, for example, x86 **Note –** This value is used to name the profile component and OS image. The value of this variable is not validated or enforced by the provisioning software.	`x86` (Default)
`installPath`	SuSe Linux image name to attach	`SuSeLinux:[version]_ :[architecture]` For example: `SuSeLinux_SLES9_x86`
`description`	Optional description for this profile	`test linux` (Example)
`min_disk_size`	Minimum disk space required to provision the SuSe Linux OS	`10GB` (Default)
`boot_server_name`	Name of SuSe Linux boot and install server	`masterserver` (Example)
`boot_server_subnet_address`	Subnet served by the boot and install server	`10.10.31.0` (Example)
`boot_server_subnet_ ip_address`	IP address of the boot and install server on the subnet	`10.10.31.1` (Example)
`image_subnet_mask`	Netmask of the provisioning subnet	`255.255.255.0` (Default)
`distro_root_directory`	Absolute path of the base directory of the SuSe Linux distribution on the boot server	`nfs://10.10.31.4/Suse/ SLES-9-x86-64/installroot` (Example)
`kernel` (JET profiles only)	Relative path to the kernel in the OS image Do not specify a value for this variable for non-JET profiles.	N/A

Variable Name	Description	Default Value/Example
`initrd` (JET profiles only)	Relative path to `initrd` in the OS image Do not specify a value for this variable for non-JET profiles.	N/A
`autoyast_file` (Non-JET profiles only)	Absolute path of the AutoYaST file on the boot server Do not specify a value for this variable for JET-generated profiles.	`nfs://10.10.31.4/export/v20z/sles9.xml` (Example)
`pxe_boot_file` (Non-JET profiles only)	Relative path from the `/tftpboot` directory of the `pxelinux.0` boot file on the boot server Do not specify a value for this variable for JET-generated profiles.	`v20z/suse/pxelinux.0` (Example)
`pxe_configuration_file` (Non-JET profiles only)	Relative path from the `pxelinux.cfg` directory of the PXE Linux configuration file on boot server Do not specify a value for this variable for JET-generated profiles.	`as21-console` (Example)
`SuSelinux_component_folder`	Location of autogenerated SuSe Linux Provision component folder	`/com/sun/n1osp/autogen-:[target(..):sys.hostName]-linux` (Default)

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

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

8 **Deselect the checkbox for Use the Jet Modules Specified Above.**

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

10 **To verify that the Linux profile is created to the Linux image server, click Components in the left pane of the N1 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-SuSeLinuxSLES9_x86-linux/provision/`.

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

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/SuSeLinuxImage \
-name "SLES9x86" -u admin -p admin -vars "osp_server=masterserver-osp;version=SLES9; \
architecture=x86;installPath=SuSeLinux:[version]_:[architecture]; \
description=SUSE SLES9;min_disk_size=10GB; \
boot_server_name=bilinux;boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.3; \
autoyast_file=nfs://10.10.31.4/export/v20z/sles9.xml;\
pxe_boot_file=pxelinux.0;pxe_configuration_file=pxesles9"
```

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

```
standard-nojet
Standard Linux
base_config
false
```

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.
- The fourth line in the file corresponds to the Use the Jet Modules Specified Above checkbox in the plan variables section of the browser interface.

3 **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/SuseLinuxImage-create-profile \
-tar H:NM:bilinux-linux -comp - -vs SLES9x86 -pto 30 -nto 10
-f /tmp/suse-profile
```

Note the default N1 SPS user value you specified with the `-u` option. This user must exist after the OS installation for the N1 SPS Remote Agent software to start on the target host.

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

Before You Begin Verify that the target host BIOS is set to the correct day and time. If the day and time in the BIOS is set to a past date, the installation of the `suse-build-key` RPM fails, and the installation switches to interactive mode. This failure occurs because the BIOS does not accept the RPM encryption keys that are used by the distribution, as these keys appear to apply to a future day and time.

Ensure that the network interfaces on the boot and install server are properly configured.

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/suse/SLES9_x86`

4 **Select 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

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.**

Deleting SuSe Linux Images and Profiles

You might want to periodically delete unused OS images and profiles from your N1 SPS environment.

▼ How to Delete SuSe Linux OS Images

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 SuSe Linux Images section.**

3. **On the Components Details page, select the checkbox for Uninstall: Delete.**

4. **Click Run.**

   The SuSe Linux OS image is removed from the N1 SPS environment. The image does not appear in the list of images in the View link for SuSe Linux Images section.

   Note – The SuSe Linux OS image is not removed from the boot and install server.

▼ How to Delete SuSeLinux OS Profiles

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 SuSe Linux Images section.**

3 **On the Components page, select the checkbox for the profile that you want to delete.**

4 **Click Delete.**

CHAPTER 8

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 profiles.
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 the following tasks:

1. Set up the Windows RIS server.
2. Create a component for the Windows RIS server in your N1 SPS environment.

▼ How to Set Up the Windows RIS Server

1 **Install Windows 2003 Server Standard or Enterprise Edition operating system with Service Pack 1 for 32–bit systems 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.

Note – If you already have a machine that is running Windows 2003 Server 32-bit Standard or Enterprise edition without Service Pack 1, you can install theWindows 2003 Service Pack 1 separately on the server. See Microsoft Knowledge Base Article #891128 .

---

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.

Note – While installing RIS server, add a Windows 2003 x64 `Risetup.exe` image or the Windows 2003-Service Pack 1 x86 `Risetup.exe` image on the RIS server.

If the Windows Server 2003 SP1 RISETUP image is the first image to be uploaded to the RIS server, the new OS chooser screen (`x8664.osc`) is copied to the RIS server by default.

If there are existing Windows OS images on the RIS server at the time the first Windows Server 2003 SP1 image or the Windows 2003 x64 is added, you must choose whether to overwrite (or backup and overwrite) the OS chooser screens during the installation to get the new screen. See Microsoft Knowledge Base Article #891128.

---

4. **Install Windows 2003 Resource Kit software and Support Tools on the Windows boot and install server.**

   This package includes tools like `setupmgr` and `xcacls` that are used by the OS provisioning scripts. You can install these tools in one of the following ways:

   - **Install the Windows 2003 Support Tools.**

     The Support Tools are not automatically installed when you install Windows 2003. The installation program is located on the OS CD-ROM in the `\support\tools` folder. You must manually open the setup file (`suptools.msi`) to launch the installation wizard.

   - **Install the Windows 2003 Resource Kit tools.**

     You can obtain the Resource Kit tools from a Windows 2003 Resource Kit CD or from `http://www.microsoft.com/downloads`.

5. **Install and configure the Windows RSH Server or the Windows SSH Server on the RIS server.**

Note – The N1 SPS OS Provisioning Plug-In 3.0 release supports only SSH for deploying Windows custom-based profiles and supports both RSH and SSH for Windows non-custom based profiles. SSH provides more secure and reliable communication between the RIS Server and the N1 OS Provisioning Server than the RSH service.

- **To install the Windows SSH Server on the Windows RIS Server, install and configure the Cygwin© Open Secure Shell Service (OpenSSH service) for Windows 2003 on the RIS Server as any user.**

  For more information, see the following web sites.

  - Cygwin home page (`http://www.cygwin.com/`)
  - Instructions to uninstall existing `cygwin` packages at `http://cygwin.com/faq/faq_2.html#SEC20`

- **To install the Windows RSH Server on the Windows RIS Server, follow these steps.**

  a. **Install the Windows 2003 Resource Kit and Windows Services for Unix 3.5 (SFU3.5) software on the Windows RIS server.**

  For instructions about how to install the Windows 2003 Resource Kit and Windows Services for Unix 3.5 (SFU3.5) software, see Microsoft Knowledge Base Article #321712 .

  The Windows 2003 Resource Kit and Windows Services for Unix 3.5 (SFU3.5) software are available on the Windows Resource Kit CD and from the Windows web site. This software includes the Windows RSH service, and other tools that are used by the OS provisioning plug-in scripts, such as `setupmgr`, `deploy.cab`, `xcacls`.

  b. **Install and configure the Windows Remote Shell Service (WinRsh service) on the RIS Server as user `root`.**

  For instructions, seeRemote Shell Services in SFU (`http://www.microsoft.com/technet/interopmigration/unix/sfu/sfu35rsh.mspx`).

  Note – If the permissions for the `.rhosts` file are not set correctly, the WinRsh service fails and the client receives an "Access denied" message. To prevent this failure, assign full permissions to the user group Administrators/SYSTEM in the `.rhosts` ACL list. For example add the following entry to the `.rhosts` file in `c://windows/system32/drivers/etc/`

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

  After editing the `.rhosts` file, ensure that you save the file as `.rhosts`, not `.rhosts.txt`.

  ---

6 **Install the N1 SPS Remote Agent on the Windows RIS server.**

7 **On the Master Server, prepare a new Remote Agent host for the Windows RIS server.**

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

After you set up the Windows RIS server, you must create a component for the server in your SPS environment. Follow these steps to create the component in the SPS browser user interface.

1 **Log in to the SPS browser interface.**

2 **In the Common Tasks section of the browser interface main window, selelct OS Provisioning.**

The OS Provisioning page is displayed.

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

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

The WindowsServer-create plan details page is displayed.

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

- **To use an existing variables set, select a name from the drop-down menu in the Variable Settings row of the Plan Parameters table.**
- **To create a new variables set, click Select from List in the Variable Settings column 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 124.

  d. **To change a variable setting, click the check box for the variable.**

  A check mark is displayed in the box, and a text field is displayed. Enter the new variable value in the text field.

  e. **Save the variables set.**

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

- **If you want to use another component's variable settings, click Import Set From Component.**

  The Import Variable Settings window displays.

  a. **If necessary, navigate to the Folder that contains the component with the variable settings you want to import.**

  b. **Select the component version.**

Note – Variable settings can vary between component versions. Ensure that the current components and the component from which you want to import variable settings share common variables. If the component from which you want to import variable settings does not share common variables with the component you want to use in your plan, the variable settings are not imported.

---

c. **Click Import Variable Settings.**

   The variables settings are imported, and are displayed in the table.

d. **On the Plan Details Run page, select the variable settings that you imported from the Variable Settings drop-down list, then click Select.**

- **If you want to use component variable settings that are stored in a file, follow these steps.**

a. **In the Import Sets from File text field, enter the path to the variable settings file that you want to use.**

   To browse through the file system to find the appropriate file, click the Browse button.

b. **Click Import.**

   The variables settings are imported, and are displayed in the table.

Note – If the file from which you want to import variable settings does not share common variables with the component you want to use in your plan, the variable set is not imported.

---

c. **On the Plan Details Run page, select the variable settings that you imported from the Variable Settings drop-down list, then click Select.**

6. **In the Target Host field, specify the host created in the N1 SPS Master Server for the Windows RIS server you created in "How to Set Up the Windows RIS Server" on page 118.**

7. **Type the password for the Active Directory user in the text field in the Plan Variables section.**

   Type the password in clear text only. Ensure that your password selection meets all Window password requirements. See the Microsoft documentation for more information.

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

   The plan performs the following tasks.

   - The Windows RIS server component is created on the Master Server.
   - The `N1ospRisUtil` directory is created on the Windows RIS server in the directory that is specified in the `installPath` variable. This directory contains the following subdirectories.
     - `N1ospScripts`

- `N1ospTemplates`
- `N1ospTools`
- `SampleOSCFiles`
- `SampleSIFFiles`

9 **To verify that the Windows RIS 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)

After you set up the Windows RIS server, you must create an component for the server in your SPS environment. Follow these steps to create the component in the SPS browser user interface.

1 **(Optional) Encrypt the RIS server access password.**

For more information about how to encrypt passwords, see "Password Encryption" on page 159.

2 **To create a variable set for the WindowsServer component, type 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 "installPath=C:\\\\N1 ISP\\\\IspScripts; \
boot_server_domain_name=n1lab.west.example.com;active_directory_userid=n1ospadmin; \
boot_server_access_protocol=ssh;boot_server_access_userid=Administrator; \
boot_server_access_password=yf7813jWUweB4SXBe0JIlw==; \
ris_share_directory=D:\\\\RemoteInstall"
```

For information about the WindowsServer variable set, see "Windows Image Server Component Variables" on page 124.

3 **Create a file named `/tmp/windows-bi-server` that includes the following entry:**

```
password4user!
```

In the previous example, **`password4user!`** specifies the Active Directory user password.

4 **To run the plan, type a command that is similar to the following example.**

```
# cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/untyped/WindowsServer-create -tar H:NM:risserver \
-comp - -vs winRisServer -pto 30 -nto 10 -f /tmp/windows-bi-server
```

The plan performs the following tasks.

- The Windows RIS server component is created on the Master Server.

- The N1ospRisUtil directory is created on the Windows RIS server in the directory that is specified in the installPath variable. This directory contains the following subdirectories.
  - N1ospScripts
  - N1ospTemplates
  - N1ospTools
  - SampleOSCFiles
  - SampleSIFFiles

Windows Image Server Component Variables

TABLE 8–1 Variables for Windows Image Server Component

Variable Name	Description	Default Value or Example
installPath	Location of the base directory where the N1 OS Provisioning scripts are installed. For example, C:\Program Files. Do not specify a trailing backslash in this variable.	C:\ (Default)
boot_srerver_domain_name	Windows active directory domain name for the RIS server.	winprov.n1lab.sun.com (Example)
active_directory_userid	Active directory user ID	n1ospadmin (Default)
boot_server_access_protocol	Protocol to use to access the RIS server, such as RSH or SSH **Note –** If you plan to use custom-based Windows RIS server profiles, RSH is not a supported value for the boot_server_access_protocol variable.	ssh (Default)
boot_server_access_userid	User ID to use to access the RIS server. For the RSH protocol, specify root for this variable.	No default
boot_server_access_password	Encrypted password to use to access the RIS server. This variable is required for RIS servers that use the SSH protocol. If you use the RSH protocol, this variable is not required. For instructions about how to create an encrypted password, see "Password Encryption" on page 159.	No default
ris_share_directory	Location of the RemInst share on the RIS server. For example, D:\RemoteInstall.	No default

TABLE 8–1 Variables for Windows Image Server Component *(Continued)*

Variable Name	Description	Default Value or Example
`ris_language`	Language of the OSChooser screens installed on the RIS server.	English
`ciw_directory`	Location of the CIW and OS Chooser files. For example, D:\RemoteInstall\OSChooser\English	`:[ris_share_directory]\OSChooser\:[ris_language]`
`temporary_sif_directory`	Location of the Sun N1 OS Provisioning temporary `sif` files in the RemInst share. For example, D:\RemoteInstall\N1ospSif.	`:[ris_share_directory]\N1ospSif`
`windowsHost`	Virtual host that represents the Windows RIS 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 Set Up a Windows Image on the Windows RIS Server

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

Use the `RiSetup.exe` utility. Select the Keep old existing OSC files option when you create the RIS image.

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.

If you are using a Windows distribution with integrated Service Packs, follow the instructions in the following sitesto create your RIS image.

- Service Pack 3 Installation and Deployment Guide for Windows 2000 Server Edition (`http://www.microsoft.com/Windows2000/downloads/servicepacks/sp3/spdeploy.htm#the_integrated_installation_fmay`)

  Similar documentation is available for other Windows OS versions at this site.

- Microsoft Knowledge Base Article #260910 (`http://support.microsoft.com/kb/260910/EN-US/`)

  This article describes how to obtain the latest Service Packs.

- Download Windows 2000 Service Pack 3 (`http://www.microsoft.com/Windows2000/downloads/servicepacks/sp3/download.aspx`)

  Download the appropriate service pack from this site. Select the Windows 2000 SP3 Network Installation Windows 2000 SP3 Network Installation download. Similar download pages are available for other Windows OS versions at this site.

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, and Microsoft Knowledge Base Article 254078.

Note – Create the distribution folder accordingly.

See the sample setup files and drivers in the `SampleSIFFiles` and `SampleDrivers` folders for more details.

3 **(Optional) Copy N1 SPS Remote Agent binary to the folder** *product-directory*`/$OEM$/$1/N1OSP/` **under the image distribution folder.**

product-directory is the Windows image distribution folder that has "i386" or “amd64” as a subfolder that contains the kernel files.

Note – If this folder layout does not exist, create the layout.

The script expects the `N1OSP` folder to be part of the image. Copy the tools required for provisioning to this directory path *product-directory*`/$OEM$/$1/N1OSP/`. Do not change the `N1OSP` folder name.

After you copy the N1 SPS Remote Agent binary to the folder, the directory contents resemble the following example.

```
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\N1OSP

     02/04/2005     02:39 PM    <DIR>        .
     02/04/2005     02:39 PM    <DIR>        ..
     01/31/2005     03:33 PM     27,595,776 cr_ra_win32_5.2.msi
```

Creating Custom Windows Provisioning Profiles

You can use the provisioning software to generate a custom Windows provisioning profile for your deployments. The OS Provisioning Plug-In can generate all the necessary variables and plans to provision the Windows OS to your target hosts.

If you want to manually create a Windows provisioning profile, see "How to Configure the Setup Information for the Windows Image" on page 131.

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

This procedure describes how to use the OS Provisioning browser interface to generate a custom Windows provisioning profile.

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 118.

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 135.

        Note – For custom-based profiles, you must not specify a value for the `sif_file` variable.

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

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

8 **Verify that the checkbox for the Create the Custom Profile option is selected.**

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

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

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

11 **Edit the variables for the custom-based Windows provisioning profile.**

You need to edit the component variables for the custom Windows profile to enter values for the following items.

- OS Provisioning control service
- Product key
- User name
- Organization name
- Time zone
- Terminal server properties
- Language and locale for the target host

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

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

c. **On the Components page, click the profile you previously created in the Components column.**

d. **On the Components Details page, click Edit.**

e. **On the Component Details Edit page, click the check box next to the variable that you want to edit, then type the value for the variable in the Value for This Component column.**

Edit the values for the following variables.

`osp_profile_control_service` Specifies whether to call the OSP Control Service, which automates power off, reboot, and power on functions. For Windows provisioning, set the value of this variable to `true`.

`user_full_name` Specifies the full name of the user for the target host.

`organization_name` Specifies the name of the organization of which the target host is a member.

`product_key` Specifies the product key for the OS distribution.

`timezone` Specifies the time zone for the target host.

`profile_language` Specifies the language or locale to install on the target host.

`profile_language_group` Specifies the language group to install on the target host.

`install_terminal_server` Specifies that the profile should install a terminal server on the target host.

`terminal_server_allow_connections` Specifies whether the target host should allow remote connections.

`terminal_server_licensing_mode` Specifies how the terminal services manages Client Access Licenses.

`terminal_server_permissions_setting` Enables administrators to choose the security mode for terminal server users.

f. **Click Check In.**

Next Steps To install the Windows OS on your target host, you must modify the BIOS boot settings, then run the provisioning plan. For more information, see "Installing the Windows OS on the Target Host" on page 136.

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

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

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/untyped/WindowsImage \
-name "windows2003custom" -u admin -p admin -vars \
"osp_server=masterserver-osp;os_name="WIN2003_trial"version=2003; \
installPath="2003customprofile"; \
description=test windows image 2003;min_disk_size=10GB; \
boot_server_name=biwl;boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.4;\
boot_srv_active_directory_domain=n1.west.example.com;\
windows_product_directory=D:\\\\RemoteInstall\\\\Setup\\\\English
\\\\Images\\\\Win2003_trial"
```

Note – The `windows_product_directory` should be included all on one line. For display purposes, this entry appears on two separate lines.

For more information about the variables, see "Windows Provisioning Profile Variables" on page 135.

2 **Create a file named `/tmp/windows-profile` with the following entries.**

```
custom
Windows 2003 EE x86 Profile
true
```

The entries in this file correspond to the following plan variables.

- `custom` specifies the Profile Name.
- `Windows 2003 EE x86 Profile` specifies the Profile Description.
- `true` specifies that this plan will generate a custom profile.

3 **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/WindowsImage-create-profile \
-tar H:NM:risserver-windows -comp - -vs windows2003custom -pto 30 -nto 10 \
-f /tmp/windows-profile
```

Note the default N1 SPS user value you specified with the `-u` option. This user must exist after the OS installation for the N1 SPS Remote Agent software to start on the target host.

Creating Non-Custom Windows Provisioning Profiles

Through the N1 SPS browser or command line interface, you can manually create a Windows Provisioning profile. When you manually create a provisioning profile, you must supply the appropriate variable values to use in your provisioning plans.

If you want the OS Provisioning plug-in to generate a Windows provisioning profile for you, see "How to Create a Custom Windows Provisioning Profile (Browser Interface)" on page 127.

To create a non-custom Windows Provisioning Profile, complete the following tasks.

- "How to Configure the Setup Information for the Windows Image" on page 131
- "How to Create a Non-Custom Windows Provisioning Profile (Browser Interface)" on page 133

▼ How to Configure the Setup Information for the Windows Image

This procedure describes how to manually configure the setup information for a Windows image. You must complete this procedure if you are creating a non-custom Windows provisioning profile.

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

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 = "N1osp 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 **Edit the software information file (`sif`) to automate download of critical security updates after the installation.**

Add the following lines to the `sif` file.

```
[WindowsFirewall]
Profiles=WindowsFirewall.TurnOffFirewall

[WindowsFirewall.TurnOffFirewall]
Mode=0
```

7 **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.

---

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

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

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

This procedure describes how to use the OS Provisioning browser interface to manually create your own Windows provisioning profile.

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 118 and "How to Set Up the Windows RIS Server" on page 118.

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 **Click Select from List in the WindowsImage component row of the Plan Parameters table.**

5 **Click Create Set.**

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

7 **Specify the variable values to use with this plan.**

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 135.

Note – You must specify a value for the `sif_file` variable.

---

8 **Save the variables set.**

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

10 **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`.

---

11 **Type values for the Profile Name and Profile Description plan variables.**

12 **Deselect the Create Custom Profile check box.**

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

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

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

Next Steps To install the Windows OS on your target host, you must modify the BIOS boot settings, then run the provisioning plan. For more information, see "Installing the Windows OS on the Target Host" on page 136.

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

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/WindowsImage \
-name "windows2003" -u admin -p admin -vars "osp_server=masterserver-osp;version=2003; \
os_name="WIN2003_trial";description=test windows image 2003;min_disk_size=10GB; \
boot_server_name=biw1;boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.4;boot_srv_active_directory_domain=n1isp.sfbay.sun.com; \
windows_product_directory=D:\\\\RemoteInstall\\\\Setup\\\\English\\\\Images\\\\Win2003_trial; \
sif_file=D:\\\\RemoteInstall\\\\Setup\\\\English\\\\Images\\\\Win2003_trial\\\\i386\\\\templates\\\\
Ispristndrd.sif"
```

Note – The `windows_product_directory` should be included all on one line. For display purposes, this entry appears on three separate lines.

For more information about the variables, see "Windows Provisioning Profile Variables" on page 135.

2 **Create a file named `/tmp/windows-profile` with the following entries.**

```
noncustom
Windows 2003 EE x86 Profile
false
```

The entries in this file correspond to the following plan variables.

- `noncustom` specifies the Profile Name.

- `Windows 2003 EE x86 Profile` specifies the Profile Description.
- `false` specifies that this plan will not use a custom profile.

3 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/WindowsImage-create-profile \
-tar H:NM:risserver-windows -comp + -vs windows2003 -pto 30 -nto 1 -f /tmp/windows-profile0
```

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 8–2 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` (Default)
`os_name`	Name of the Windows OS distribution on the RIS server	`Win2003_SP1_x86` (Example)
`version`	Windows version	`2003` (Default)
`architecture`		`x86` (Default)
`os_language`	Language of the Windows OS distribution	`English` (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)
`boot_server_name`	Name of Windows boot server	`Winbiserver1` (Example)
`boot_server_subnet_address`	Subnet served by boot server	`10.42.42.0` (Example)
`boot_server_subnet_ip_address`	IP address of boot server on this subnet	`10.42.42.1` (Example)
`boot_srv_active_directory_domain`	Windows active directory domain for this subnet	`:[targetableComponent: boot_server_domain_name]` (Default)

TABLE 8–2 Component Variables for Windows Provisioning Profile *(Continued)*

Variable Name	Description	Default or Sample Value
`windows_product_directory`	Windows product directory name on boot server	`:[targetableComponent:ris_share_directory]\Setup\:[os_language]\Images\:[os_name]` (Default)
`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 Windows 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

Before You Begin Before you provision the Windows OS, create the target hosts that you want to install. For more information, see Chapter 9.

Ensure that the GUID is specified in the variable set for the component for the target host that you want to install. For more information about target host variables, see Appendix A.

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).**

Managing Windows Installations

The OS Provisioning Plug-In enables you to perform the following administration task on your Windows installations.

- Check the status of an installation
- Stop a provisioning operation
- Register installed target hosts as N1 SPS Remote Agents
- Delete prestaged hosts from your N1 SPS environment

▼ How to Check the Windows OS Provisioning Status

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

2 **On the OS Provisioning Common Tasks page, click Status in the Host 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 **Click Run Plan (includes preflight).**

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

▼ How to Stop a Windows OS Provisioning Operation

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

2 **On the OS Provisioning Common Tasks page, click Abort in the Host Admin 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 **Click Run Plan (includes preflight).**

▼ How to Register Installed Windows Hosts in N1 SPS

After you install the Windows OS on a target host, register the host in your N1 SPS environment.

1. **In the Common Tasks section of the N1 SPS browser interface, select OS Provisioning.**
2. **On the OS Provisioning Common Tasks page, click Register Host in the Host Admin section.**
3. **On the Plans Details page, click Run.**
4. **On the Plan Details Run page, select the target host that you want to register.**
5. **Click Run Plan (includes preflight).**

▼ How to Delete Prestaged Hosts on the Windows RIS Server

You can use the OS Provisioning Plug-In to delete prestaged target hosts from the Active Directory on the Windows RIS server.

1. **In the Common Tasks section of the N1 SPS browser interface, select OS Provisioning.**
2. **On the OS Provisioning Common Tasks page, click Manage in Windows Image Servers section.**
3. **On the Components Details page, click the checkbox next to delete_prestage_hosts in the Component Procedures table.**
4. **Click Run in the Actions column of the Component Procedures table.**
5. **On the Plan Details Run page, select the name of the prestaged host that you want to delete in the Current Installations list.**

   The name of the prestaged host does not have the `-target` suffix.
6. **Click Run Selected Installations.**
7. **Click the Details link to view the status of the deletion.**

◆ ◆ ◆ CHAPTER 9

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" on page 139
- "Example Tasks for Defining Target Hosts" on page 144
- "Password Encryption" on page 159
- "Terminal Server Support" on page 160

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
- Sun Advanced Remote System Control (RSC) Targets
- 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.

As an alternate means to IP connectivity, the OS Provisioning Plug-In supports the use of the Cisco Terminal Server 2621 series and Cyclades Alterpath ACS Terminal Server series terminal server or concentrator types to connect to the Network Management controllers. In the variable set for your target host, you can specify a terminal server for the target host to use. For more information about terminal server support and the variables for these terminal servers, see "Terminal Server Support" on page 160.

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 9–1 Target Hosts by Platform

Target Platform	Firmware or Service Processor	Configuration Requirements	Information Needed for OS Provisioning Target Variables	Target Host Type
SunFire x4100, x4200 family	SP version 6464	■ Provide IP (static) to Network Management port (routable from OS provisioning server)	■ Service Processor IP ■ ssh user/password ■ ipmi user/password ■ MAC address of the target provisioning interface ■ GUID of the provisioning interface	Sun IPMI target
SunFire x2100 family	SP v4.11	■ Provide IP (static) to Network Management port (routable from OS provisioning server)	■ Service Processor IP ■ ipmi user/password ■ MAC address of the target provisioning interface ■ GUID of the provisioning interface	Sun IPMI target

TABLE 9–1 Target Hosts by Platform *(Continued)*

Target Platform	Firmware or Service Processor	Configuration Requirements	Information Needed for OS Provisioning Target Variables	Target Host Type
SunFire v480, v880 family	RSC 2.2.3, RSCboot	■ Provide IP (static) to RSC port (routable from OS provisioning server) or connect RSC serial port to terminal server (Terminal server IP should be routable from OS provisioning server) ■ Enable Telnet (for IP-based managment)	■ RSC IP, or terminal server information (ip, type, port connected to RSC serial port) ■ RSC user/password ■ MAC address of target provisioning interface	Sun RSC target
SunFire v240, v210 family	Sun Advanced Lights Out Manager 1.3 (ALOM)	■ 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	Sun ALOM target

TABLE 9–1 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”	■ Service Processor IP ■ ssh user/password ■ ipmi password ■ MAC address of the target provisioning interface ■ GUID of the provisioning interface	Sun IPMI target

TABLE 9–1 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 IPMI port (routable from OS provisioning server) ■ Enable ipmi, ipmi channel "lan" ■ Configure BIOS to boot from PXE-enabled NICs first	■ 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 IP address, user ID, and password ■ Domain name, user ID and password ■ MAC address of the target provisioning interface	Sun Mid Range Target
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

TABLE 9–1 Target Hosts by Platform *(Continued)*

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

1. **In the Common Tasks section of the N1 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 179.

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 159.

   - 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. **If the system is connected to a terminal server, 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 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)

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 179.

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

```
true
root
true
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:**

```
# 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 several system families, including the Sun Fire™ x2100, x4100, x4200, and V20z,systems.

▼ 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 x2100 target host.

1. **In the Common Tasks section of the N1 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 175.

     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 159.

        - 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 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-x2100`, the virtual host is `sunfire-x2100-target`.

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

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 "x2100" \
-u admin -p admin -vars "installPath=x2100host;description=test x2100 host; \
device_type=sunfirex2100;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= \
console_service=false"
```

For more information about the variables, see "Sun IPMI Target Host Variables" on page 175.

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 Sun IPMI target, type a command similar to the following example:**

```
# 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 x2100 -pto 30 -nto 10 -f /tmp/test-x2100
```

The previous example applies to the Sun Fire x2100.

Defining a Sun RSC Target Host

The Sun RSC target host type is used for systems such as a Sun Fire v490 or v890 system.

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

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

2 **On the OS Provisioning Common Tasks page, click Create in the Sun RSC Targets 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 SunRSC component row of the Plan Parameters table.**

- **To create a new variables set, click Select from List in the SunRSC 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 RSC Target Host Variables" on page 182.

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:

     ```
     rsc_access_userid
     rsc_access_password
     terminal_server_userid
     terminal_server_password
     ```

     For information about encrypting the passwords, see "Password Encryption" on page 159.

   - 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 SunRSC 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 RSC password below."**

b. **Type the password in the Password to Access RSC 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 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 `test490`, the virtual host is `test490-target`.

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

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

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/targets/SunRSC  \
-name "testv490" -u admin -p admin -vars "installPath=testv490; \
description=To test v490;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;rsc_ip_address=10.5.133.150;rsc_access_userid=admin; \
rsc_access_password="
```

For more information about the variables, see "Sun RSC Target Host Variables" on page 182.

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 RSC password below" box in the plan variables section of the browser interface.
- The second line in the file is the password to access RSC.
- 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/testv490-target`.

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

```
# cat /tmp/RSC-target | cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/targets/SunRSC-create -tar H:NM:masterserver-osp \
-comp + -vs testv490 -pto 30 -nto 10 -f /tmp/testv490-target
```

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)

1. **In the Common Tasks section of the N1 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 191.

     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 159.

        - 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 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)

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 191.

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 hardware abstraction layer (HAL) JAR file that handles the remote network management capabilities. The HAL JAR files are located at `/opt/SUNWn1osp/drivers` on the OS provisioning server. You need to be aware of the services, such as power, boot and console, that is supported by each HAL JAR file. For example, the `b1600sc.jar` file includes support for power, boot, and console services for the Sun Fire B100s and B100x Blade series. Before you populate the variable sets for the remotely-managed host type, ensure that you are familiar with these services and the HAL JARs that support these services.

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.

▼ How to Create an Other Remotely-Managed Target Host (Browser Interface)

1 **In the Common Tasks section of the N1 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 197.

  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 159.

     - 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 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-b100s-1`, the virtual host is `sunfire-b100s-1-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.

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 197.

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)

1 In the Common Tasks section of the N1 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 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 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 203.

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 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)

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 203.

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 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 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.

Terminal Server Support

As an alternate means to IP connectivity, the OS Provisioning Plug-In supports the use of the following terminal server or concentrator types to connect to the Network Management controllers.

- Cisco Terminal Server 2621 series

  For product information, see `http://www.cisco.com`.
- Cyclades Alterpath ACS Terminal Server series

  For product information, see `http://www.cyclades.com`.

The OS Provisioning Plug-In supports the `telnet` protocol (for Cisco Terminal Server 2621) and SSH protocol (for Cyclades Alterpath ACS Terminal Server) as the access protocols for these terminal servers/concentrators. See the vendor specific documentation for the technical specifications and setup instructions for these terminal servers.

In the OS Provisioning Plug-In, the Cisco 2621 is the default terminal server binding for the server network management controllers. To use the Cyclades Alterpath ACS terminal server, edit the appropriate terminal server variables for your target hosts. Table 9–2 lists the terminal server variables and the values for both the Cisco 2621 and Cyclades Alterpath ACS terminal servers.

TABLE 9–2 Terminal Server Variables and Values

Variable (Description)	Cisco 2621 Variable Value	Cyclades Alterpath Variable Value
`terminal_server_type` (Terminal server type)	`Cisco2621`	`cyclades`
`terminal_server_hal_jar` (Terminal server hal jar)	`file://:[targetableComponent:installPath]/SUNWn1osp/drivers/cisco/c2621.jar`	`file://:[targetableComponent:installPath]/SUNWn1osp/drivers/cyclades/cycladesTS.jar`
`terminal_server_port` (Port of terminal server)	Port values of 2032 and above	Port values of 7000 and above
`terminal_server_protocol` (Terminal server access protocol)	`telnet` (`ssh` is not supported)	`ssh` (`telnet` is not supported)

In addition, you must provide values for the following variables to enable your target hosts to use terminal servers.

`terminal_server_ip_address` Specifies the IP address of terminal server connected to the target host.

`terminal_server_userid` Specifies the user ID for the terminal server.

`terminal_server_password` Specifies the encrypted password for the terminal server.

For instructions about how to encrypt passwords, see "Password Encryption" on page 159.

CHAPTER 10

Troubleshooting

This chapter provides information that might enable you to more easily resolve OS provisioning issues.

The chapter includes the following information:

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 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 on Solaris based OS Provisioning servers, and the`/var/log/n1osp*` files on Linux based OS Provisioning servers. 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-core.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 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 *Sun N1 Service Provisioning System 5.2 Installation Guide*.
2. Verify that the N1 SPS command-line interface (CLI) is installed on the OS provisioning server and the Solaris and Linux 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_Service_Provisioning_System_5.2/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: While provisioning the Solaris 10 1/06 OS for x86 based systems, the installation fails.

Solution: Verify that the `/tftpboot/I86PC.Solaris_10–1` folder exists on the boot and install server. If the folder does not exist, create the folder, then reprovision the target host.

Problem: While provisioning the Solaris OS to a headless target host, the installation enters interactive mode, launching the `kdmconfig` utility.

Solution: Remove the `SUNWxwssu` and `SUNWxwscf` packages from the provisioning plant. In the Solaris OS Provisioning Profile, set the value of the `profile_del_packages_base_config` variable to `SUNWxwssu SUNWxwscf`.

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 and <reminst_share>\OSChooser\amd64 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: Choose one of the following workarounds.

- On the Windows RIS server, enable the provisioning interface, and disable all other interfaces.
- Change the order of the network connections to be accessed by network services on the Windows RIS server. Follow these steps.
  1. On the Windows RIS server, open the Control Panel window from the Start menu.
  2. In the Control Panel window, click Network Connections.
  3. In the Network Connections screen, select the Advanced tab.
  4. Click the Advanced Settings button.
  5. Click on the Adapters and Bindings tab.
  6. Change the order of connections to be accessed by the network services. Ensure that the provisioning interface is listed as the first interface.
  7. Stop and restart the BINLSVC. You can perform this operation by running the Services GUI, or by typing the following commands in an MS-DOS command window.

     ```
     > net stop binlsvc
     > net start binlsvc
     ```

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/multilng.osc) under (<reminst share>/OSChooser/<language>) 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 for data in the following sections: Unattended, UserData, GuiRunOnce, and GuiUnattended.
- Make sure you provide the default Administator'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.

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 list shows all component variables for the Sun IPMI target host. .

This list has been augmented to support the Sun Fire x2100, x4100, and x4200 systems. A new variable, `device_type`, is provided to enable you to specify the IPMI system type. This variable accepts the following values.

- `sunfirev20z`
- `sunfirev40z`
- `sunfirex2100`
- `sunfirex4100`
- `sunfirex4200`

The variable prompts in the browser interface indicate whether a variable must not be set for a specific device type.

The new variable `console_service` specifies whether to disable console service on IPMI target hosts. The default value for this variable is `true`.

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.

`installPath`

Description: Physical host name for the target system.

Example: `sunfire-v20z-host`

`target_host`

Description: Virtual host name for the target system.

Default value: `:[installPath]-target`

`description`

Description: Optional description of the target.

Example: `test v20z`

`ethernet_mac_address`

Description: Mac address.

Example: `00:0E:0C:07:E9:31`

`ethernet_ip_address`

Description: IP address.

Example: `10.42.42.77`

`ethernet_netmask`

Description: Network mask.

Default value: `255.255.255.0`

`architecture`

Description: Target architecture (for example, sun4u, x86).

Default value: `x86`

`kernel_arch`

Description: Kernel architecture (sun4u, i86pc).

Default value: `i86pc`

`disk_size`

Description: Size of disk.

Example: `20GB`

`default_router`

Description: Default router to use for network connectivity

Example:10.42.42.1

`sysidcfg_default_route`

Description: Default router IP address for Solaris deployments.

Example: 10.42.42.1

`sysidcfg_network_interface`

Description: Network interface to configure for Solaris deployments.

Default value: PRIMARY

`networkifs_base_config`

Description: Additional network interfaces to configure for Solaris deployments.

Example: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

Description: Configures IP Multipathing for Solaris deployments.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`osp_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False).

Default value: `TRUE`

`sps_ra_parent`

Description: Name of parent for Remote Agent (RA).

Default value: `masterserver`

`sps_ra_parent_type`

Description: Whether RA parent is an MS (masterserver) or an LD (local distributor).

Default value: `MS`

`sps_ra_host`

Description: Name or IP address for RA host.

Default value: `:[installPath]`

`sps_ra_conn`

Description: Type of RA connection (raw, ssl, ssh).

Default value: `raw`

`sps_ra_ssl_auth`

Description: Whether RA has SSL authorization (y,n).

Default value: `N`

`sps_ra_port`

Description: Port for RA.

Default value: `1131`

`sps_ra_adv_parms`

Description: Advanced parameters for RA.

`linux_network_interface`

Description: Network interfaces to use to provision Linux to target hosts. Use this variable for Linux OS profiles created with JET.

Default value: `eth0`

`networkifs_redhat_base_config`

Description: Additional network interfaces to configure for RedHat Linux deployments. Use this variable for RedHat Linux OS profiles created with JET.

Example: `eth1!--ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

`networkifs_suse_base_config`

Description: Additional network interfaces to configure for SuSe Linux deployments. Use this variable for SuSe Linux OS profiles created with JET.

Example: `th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255`

`guid`

Description: GUID of the target specified within {}

Note – The GUID must be in the form {*dddddddd-dddd-dddd-dddd-dddddddddddd*}, where *d* is a hexadecimal text digit.

`device_type`

Description: Specifies the system type. For example, `sunfirev20z`, `sunfirev40z`, `sunfirex2100`, `sunfirex4100`, `sunfirex4200`

Default value: `sunfirev20z`

`ipmi_ip_address`

Description: IPMI IP address

Example: `10.5.133.135`

`ipmi_access_userid`

Description: User ID to access IPMI

`ipmi_access_password`

Description: Encrypted Password to access IPMI

`boot_service`

Description: Specifies whether the provisioning software should use the boot net option to install the target host (true/false).

Default value: `true`

`console_service`

Description: Specifies if the provisioning software should use the console provided by the network management interface (true/false).

Default value: `true`

`ssh_access_userid`

Description: SSH user ID to access the remote management interface.

`ssh_access_password`

Description: Encrypted SSH password to access the remote management interface.

Sun ALOM Target Host Variables

The following list shows 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.

`installPath`

Description: Physical host name for the target system

Example: `sunfire-v240-host`

`target_host`

Description: Virtual host name for the target system

Default value: `:[installPath]-target`

`description`

Description: Optional description of the target

Example: `test v240`

`ethernet_mac_address`

Description: Mac address

Example: `00:0E:0C:07:E9:31`

`ethernet_ip_address`

Description: IP address

Example: `10.42.42.77`

`ethernet_netmask`

Description: Network mask

Default value: `255.255.255.0`

`architecture`

Description: Target architecture (for example, sun4u, x86)

Default value: `sun4u`

`kernel_arch`

Description: Kernel architecture (sun4u, i86pc)

Default value: `sun4u`

`disk_size`

Description: Size of disk

Default value: `20GB`

`default_router`

Description: Default router to use for network connectivity

Example: 10.42.42.1

`sysidcfg_default_route`

Description: Default router IP address for Solaris deployments.

Example: 10.42.42.1

`sysidcfg_network_interface`

Description: Network interface to configure for Solaris deployments.

Default value: PRIMARY

`networkifs_base_config`

Description: Additional network interfaces to configure for Solaris deployments.

Example: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

Description: Configures IP Multipathing for Solaris deployments.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`osp_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)

Default value: `TRUE`

`sps_ra_parent`

Description: Name of parent for RA

Default value: `masterserver`

`sps_ra_parent_type`

Description: Whether RA parent is an MS or an LD

Default value: `MS`

`sps_ra_host`

Description: Name or IP address of RA host

Default value: `:[installPath]`

`sps_ra_conn`

Description: Type of RA connection (raw, ssl, ssh)

Default value: `raw`

`sps_ra_ssl_auth`

Description: Whether RA has SSL authorization (y,n)

Default value: `N`

`sps_ra_port`

Description: Port for RA

Default value: `1131`

`sps_ra_adv_parms`

Description: Advanced parameters for RA

`alom_ip_address`

Description: ALOM IP address

Example: `10.5.133.135`

`alom_access_userid`

Description: User id to access ALOM

`alom_access_password`

Description: Encrypted password to access ALOM

`terminal_server_ip_address`

Description: IP address of terminal server connected to this system

Example: `10.5.133.135`

`terminal_server_protocol`

Description: Communication protocol to use to connect to the terminal server.

If you are using a Cyclades terminal server, set this variable to `ssh`.

Default value: `telnet`

`terminal_server_port`

Description: Port of terminal server

`terminal_server_userid`

Description: Terminal server user id

`terminal_server_password`

Description: Encrypted password for terminal server

`terminal_server_type`

Description: Terminal server type

Default value: `Cisco2621`

`terminal_server_version`

Description: Terminal server version

`terminal_server_hal_jar`

Description: Path to terminal server hardware abstraction layer (HAL) JAR file.

If you are using a Cyclades terminal server, set this value to `file:///opt/n1osp/drivers/cyclades/cycladesTS.jar`

Default value: `file:///opt/n1osp/drivers/cisco/c2621.jar`

Sun RSC Target Host Variables

The following list shows all component variables for the Sun RSC 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.

`installPath`
: **Description:** Physical host name for the target system

  Example: `sunfire-v880-host`

`target_host`
: **Description:** Virtual host name for the target system

  Default value: `:[installPath]-target`

`description`
: **Description:** Optional description of the target

  Example: `test v880`

`ethernet_mac_address`
: **Description:** Mac address

  Example: `00:0E:0C:07:E9:31`

`ethernet_ip_address`
: **Description:** IP address

  Example: `10.42.42.77`

`ethernet_netmask`
: **Description:** Network mask

  Default value: `255.255.255.0`

`architecture`
: **Description:** Target architecture (for example, sun4u)

  Default value: `sun4u`

`kernel_arch`
: **Description:** Kernel architecture (sun4u)

  Default value: `sun4u`

`disk_size`
: **Description:** Size of disk

  Default value: `20GB`

`default_router`
: **Description:** Default router to use for network connectivity

Example: 10.42.42.1

`sysidcfg_default_route`

Description: Default router IP address for Solaris deployments.

Example: 10.42.42.1

`sysidcfg_network_interface`

Description: Network interface to configure for Solaris deployments.

Default value: PRIMARY

`networkifs_base_config`

Description: Additional network interfaces to configure for Solaris deployments.

Example: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

Description: Configures IP Multipathing for Solaris deployments.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`osp_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)

Default value: `TRUE`

`sps_ra_parent`

Description: Name of parent for RA

Default value: `masterserver`

`sps_ra_parent_type`

Description: Whether RA parent is an MS or an LD

Default value: `MS`

`sps_ra_host`

Description: Name or IP address of RA host

Default value: `:[ethernet_ip_address]`

`sps_ra_conn`

Description: Type of RA connection (raw, ssl, ssh)

Default value: `raw`

`sps_ra_ssl_auth`

Description: Whether RA has SSL authorization (y,n)

Default value: `N`

`sps_ra_port`

Description: Port for RA

Default value: `1131`

`sps_ra_adv_parms`

Description: Advanced parameters for RA

`rsc_ip_address`

Description: RSC IP address

Example: `10.5.133.135`

`rsc_access_userid`

Description: User id to access RSC

`rsc_access_password`

Description: Encrypted password to access RSC

`terminal_server_ip_address`

Description: IP address of terminal server connected to this system

Example: `10.5.133.135`

`terminal_server_protocol`

Description: Communication protocol to use to connect to the terminal server.

If you are using a Cyclades terminal server, set this variable to `ssh`.

Default value: `telnet`

`terminal_server_port`

Description: Port of terminal server

`terminal_server_userid`

Description: Terminal server user id

`terminal_server_password`

Description: Encrypted password for terminal server

`terminal_server_type`

Description: Terminal server type

Default value: `Cisco2621`

`terminal_server_version`

Description: Terminal server version

`terminal_server_hal_jar`

Description: Path to terminal server hardware abstraction layer (HAL) JAR file

If you are using a Cyclades terminal server, set this value to `file:///opt/n1osp/drivers/cyclades/cycladesTS.jar`

Default value: `file:///opt/n1osp/drivers/cisco/c2621.jar`

Sun LOM Target Host Variables

The following list shows 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.

`installPath`
Description: Physical host name for the target system

Example: `sunfire-v120-host`

`target_host`
Description: Virtual host name for the target system

Default value: `:[installPath]-target`

`description`
Description: Optional description of the target

Example: `test v120`

`ethernet_mac_address`
Description: Mac address

Example: `00:0E:0C:07:E9:31`

`ethernet_ip_address`
Description: IP address

Example: `10.42.42.77`

`ethernet_netmask`
Description: Network mask

Default value: `255.255.255.0`

`architecture`
Description: Target architecture (for example, sun4u, x86)

Default value: `sun4u`

`kernel_arch`
Description: Kernel architecture (sun4u, i86pc)

Default value: `sun4u`

`disk_size`
Description: Size of disk

Default value: `20GB`

`default_router`
Description: Default router to use for network connectivity

Example: 10.42.42.1

`sysidcfg_default_route`

Description: Default router IP address for Solaris deployments.

Example: 10.42.42.1

`sysidcfg_network_interface`

Description: Network interface to configure for Solaris deployments.

Default value: PRIMARY

`networkifs_base_config`

Description: Additional network interfaces to configure for Solaris deployments.

Example: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

Description: Configures IP Multipathing for Solaris deployments.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`osp_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)

Default value: `TRUE`

`sps_ra_parent`

Description: Name of parent for RA

Default value: `masterserver`

`sps_ra_parent_type`

Description: Whether RA parent is an MS or an LD

Default value: `MS`

`sps_ra_host`

Description: Name or IP address of RA host

`sps_ra_conn`

Description: Type of RA connection (raw, ssl, ssh)

Default value: `raw`

`sps_ra_ssl_auth`

Description: Whether RA has SSL authorization (y,n)

Default value: `N`

`sps_ra_port`

Description: Port for RA

Default value: `1131`

`sps_ra_adv_parms`
Description: Advanced parameters for RA

`disk_type`
Description: SCSI or IDE

Default value: `scsi`

`lom_version`
Description: LOM version

`lom_hal_jar`
Description: Path to LOM HAL JAR file

Default value: `file://:[targetableComponent:installPath]/SUNWn1osp/drivers/sun/lom.jar`

`lom_access_userid`
Description: User ID to access LOM.

`lom_access_password`
Description: Encrypted password to access LOM.

`terminal_server_ip_address`
Description: IP address of terminal server connected to this system

Example: `10.5.133.135`

`terminal_server_protocol`
Description: Communication protocol to use to connect to the terminal server.

If you are using a Cyclades terminal server, set this variable to `ssh`.

Default value: `telnet`

`terminal_server_port`
Description: Port of terminal server

`terminal_server_userid`
Description: Terminal server user id

`terminal_server_password`
Description: Encrypted value for terminal server password

`terminal_server_type`
Description: Terminal server type

Default value: `Cisco2621`

`terminal_server_version`
Description: Terminal server version

`terminal_server_hal_jar`

Description: Path to terminal server HAL JAR file

If you are using a Cyclades terminal server, set this value to `file:///opt/n1osp/drivers/cyclades/cycladesTS.jar`

Default value: `file://:[targetableComponent:installPath]/SUNWn1osp/drivers/cisco/c2621.jar`

`power_service_true_false`

Description: Does the network management interface provide power on/off service

Default value: `true`

`boot_service_true_false`

Description: Does the network management interface provide boot service (boot order disk/network)

Default value: `true`

`console_service_true_false`

Description: Does the network management interface provide console

Default value: `false`

Sun Mid-Range Target Host Variables

The following list shows 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.

`installPath`

Description: Physical host name for the target system

Example: `sunfire-4800-A-host`

`target_host`

Description: Virtual host name for the target system

Default value: `:[installPath]-target`

`description`

Description: Optional description of the target

Example: `test 4800 domain`

`ethernet_mac_address`

Description: Mac address

Example: `00:0E:0C:07:E9:31`

`ethernet_ip_address`

Description: IP address

Example: `10.42.42.77`

`ethernet_netmask`

Description: Network mask

Default value: `255.255.255.0`

`architecture`

Description: Target architecture (for example, sun4u, x86)

Default value: `sun4u`

`kernel_arch`

Description: Kernel architecture (sun4u, i86pc)

Default value: `sun4u`

`disk_size`

Description: Size of disk

Default value: `20GB`

`default_router`

Description: Default router to use for network connectivity

Example: 10.42.42.1

`sysidcfg_default_route`

Description: Default router IP address for Solaris deployments.

Example: 10.42.42.1

`sysidcfg_network_interface`

Description: Network interface to configure for Solaris deployments.

Default value: PRIMARY

`networkifs_base_config`

Description: Additional network interfaces to configure for Solaris deployments.

Example: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

Description: Configures IP Multipathing for Solaris deployments.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`osp_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)

Default value: `TRUE`

`sps_ra_parent`
Description: Name of parent for RA

Default value: Name of parent for RA

`sps_ra_parent_type`
Description: Whether RA parent is an MS or an LD

Default value: `MS`

`sps_ra_host`
Description: Name or IP address for RA host

`sps_ra_conn`
Description: Type of RA connection (raw, ssl, ssh)

Default value: `raw`

`sps_ra_ssl_auth`
Description: Whether RA has SSL authorization (y,n)

Default value: `N`

`sps_ra_port`
Description: Port for RA

Default value: `1131`

`sps_ra_adv_parms`
Description: Advanced parameters for RA

`domain_name`
Description: Domain

`domainsc_access_userid`
Description: User id to access domain SC

`domainsc_access_password`
Description: Encrypted password to access domain SC

`platformsc_ip_address`
Description: IP address of platform SC

`platformsc_access_userid`
Description: User id to access platform SC

`platformsc_access_password`
Description: Encrypted password to access platform SC

HP ILO Target Host Variables

The following list shows 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.

`installPath`

Description: Physical host name for the target system

Example: `hp-proliant1-host`

`target_host`

Description: Virtual host name for the target system

Default value: `:[installPath]-target`

`description`

Description: Optional description of the target

Example: `test hp proliant1 host`

`ethernet_mac_address`

Description: Mac address

Example: `00:0E:0C:07:E9:31`

`ethernet_ip_address`

Description: IP address

Example: `10.42.42.77`

`ethernet_netmask`

Description: Network mask

Default value: `255.255.255.0`

`architecture`

Description: Target architecture (for example, sun4u, x86)

Default value: `x86`

`kernel_arch`

Description: Kernel architecture (sun4u, i86pc)

Default value: `i86pc`

`disk_size`

Description: Size of disk

Default value: `20GB`

`default_router`

Description: Default router to use for network connectivity

Example: 10.42.42.1

`sysidcfg_default_route`

Description: Default router IP address for Solaris deployments.

Example: 10.42.42.1

`sysidcfg_network_interface`

Description: Network interface to configure for Solaris deployments.

Default value: PRIMARY

`networkifs_base_config`

Description: Additional network interfaces to configure for Solaris deployments.

Example: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

Description: Configures IP Multipathing for Solaris deployments.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`osp_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)

Default value: `TRUE`

`sps_ra_parent`

Description: Name of parent for RA

Default value: `masterserver`

`sps_ra_parent_type`

Description: Whether RA parent is an MS or an LD

Default value: `MS`

`sps_ra_host`

Description: Name or IP address of RA host

Default value: `:[installPath]`

`sps_ra_conn`

Description: Type of RA connection (raw, ssl, ssh)

Default value: `raw`

`sps_ra_ssl_auth`

Description: Whether RA has SSL authorization (y,n)

Default value: `N`

`sps_ra_port`

Description: Port for RA

Default value: `1131`

`sps_ra_adv_parms`

Description: Advanced parameters for RA

`linux_network_interface`

Description: Network interfaces to use to provision Linux to target hosts. Use this variable for Linux OS profiles created with JET.

Default value: `eth0`

`networkifs_redhat_base_config`

Description: Additional network interfaces to configure for RedHat Linux deployments. Use this variable for RedHat Linux OS profiles created with JET.

Example: `eth1!--ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

`networkifs_suse_base_config`

Description: Additional network interfaces to configure for SuSe Linux deployments. Use this variable for SuSe Linux OS profiles created with JET.

Example: `th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255`

`guid`

Description: GUID of the target specified within {}

Note – The GUID must be in the form {*dddddddd-dddd-dddd-dddd-dddddddddddd*}, where *d* is a hexadecimal text digit.

Example: `{3AE45192-159D-11D8-AEF3-001083FDFE8A}`

`ilo_ip_address`

Description: ILO IP address

Example: `10.5.133.135`

`ilo_version`

Description: ILO version

Default value: `1.6.4`

`ilo_access_userid`

Description: User id to access ILO

`ilo_access_password`

Description: Encrypted password to access ILO

Other IPMI Target Host Variables

The following list shows 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.

`installPath`
: **Description:** Physical host name for the target system

  Example: `sunfire-v60x-host`

`target_host`
: **Description:** Virtual host name for the target system

  Default value: `:[installPath]-target`

`description`
: **Description:** Optional description of the target

  Example: `test v60x`

`ethernet_mac_address`
: **Description:** Mac address

  Example: `00:0E:0C:07:E9:31`

`ethernet_ip_address`
: **Description:** IP address

  Example: `10.42.42.77`

`ethernet_netmask`
: **Description:** Network mask

  Default value: `255.255.255.0`

`architecture`
: **Description:** Target architecture (for example, sun4u, x86)

  Default value: `x86`

`kernel_arch`
: **Description:** Kernel architecture (sun4u, i86pc)

  Default value: `i86pc`

`disk_size`
: **Description:** Size of disk

  Default value: `20GB`

`default_router`
: **Description:** Default router to use for network connectivity

Example: 10.42.42.1

`sysidcfg_default_route`

Description: Default router IP address for Solaris deployments.

Example: 10.42.42.1

`sysidcfg_network_interface`

Description: Network interface to configure for Solaris deployments.

Default value: PRIMARY

`networkifs_base_config`

Description: Additional network interfaces to configure for Solaris deployments.

Example: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

Description: Configures IP Multipathing for Solaris deployments.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`osp_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)

Default value: `TRUE`

`sps_ra_parent`

Description: Name of parent for RA

Default value: `masterserver`

`sps_ra_parent_type`

Description: Whether RA parent is an MS or an LD

Default value: `MS`

`sps_ra_host`

Description: Name or IP address of RA host

Default value: `:[installPath]`

`sps_ra_conn`

Description: Type of RA connection (raw, ssl, ssh)

Default value: `raw`

`sps_ra_ssl_auth`

Description: Whether RA has SSL authorization (y,n)

Default value: `N`

`sps_ra_port`

Description: Port for RA

Default value: `1131`

`sps_ra_adv_parms`

Description: Advanced parameters for RA

`linux_network_interface`

Description: Network interfaces to use to provision Linux to target hosts. Use this variable for Linux OS profiles created with JET.

Default value: `eth0`

`networkifs_redhat_base_config`

Description: Additional network interfaces to configure for RedHat Linux deployments. Use this variable for RedHat Linux OS profiles created with JET.

Example: `eth1!--ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

`networkifs_suse_base_config`

Description: Additional network interfaces to configure for SuSe Linux deployments. Use this variable for SuSe Linux OS profiles created with JET.

Example: `th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255`

`guid`

Description: GUID of the target specified within {}

Note – The GUID must be in the form {*dddddddd-dddd-dddd-dddd-dddddddddddd*}, where *d* is a hexadecimal text digit.

Example: `{3AE45192-159D-11D8-AEF3-001083FDFE8A}`

`disk_type`

Description: Disk type (for example, SCSI, IDE)

Default value: `scsi`

`ipmi_ip_address`

Description: IPMI IP address

Example: `10.5.133.135`

`ipmi_access_userid`

Description: User id to access IPMI

`ipmi_access_password`

Description: Encrypted password to access IPMI

`ipmi_hal_jar`

Description: Path to IPMI HAL JAR file

Default value:
`file://:[targetableComponent:installPath]/SUNWn1osp/drivers/sun/ipmi.jar`

`ipmi_version`
Description: IPMI version

Default value: `1.5`

`power_service_true_false`
Description: Does the network management interface provide power on/off service?

Default value: `true`

`boot_service_true_false`
Description: Does the network management interface provide boot service (disk, then network)?

Default value: `false`

`console_service_true_false`
Description: Does the network management interface provide console service?

Default value: `false`

Other Remotely-Managed Target Host Variables

The following list shows 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, and for which you have HAL implementations that you can bind to target definitions to automate power, boot, and console functions. Several HAL implementations are included in the N1 SPS 5.2 product. For example, the Sun Fire B100s blade server uses the `file://:[targetableComponent:installPath]/SUNWn1osp/drivers/sun/b1600sc.jar`, which supports power, boot, and console service.

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.

`installPath`
Description: Physical host name for the target system

Example: `sunfire-b100s-host`

`target_host`
Description: Virtual host name for the target system

Default value: `:[installPath]-target`

`description`
Description: Optional description of the target

Example: `test b100s blade server`

`ethernet_mac_address`
Description: Mac address

Example: `00:0E:0C:07:E9:31`

`ethernet_ip_address`
Description: IP address

Example: `10.42.42.77`

`ethernet_netmask`
Description: Network mask

Default value: `255.255.255.0`

`architecture`
Description: Target architecture (for example, sun4u, x86)

Default value: `x86`

`kernel_arch`
Description: Kernel architecture (sun4u, i86pc)it literal

Default value: `i86pc`

`disk_size`
Description: Size of disk

Default value: `20GB`

`default_router`
Description: Default router to use for network connectivity

Example: 10.42.42.1

`sysidcfg_default_route`
Description: Default router IP address for Solaris deployments.

Example: 10.42.42.1

`sysidcfg_network_interface`
Description: Network interface to configure for Solaris deployments.

Default value: PRIMARY

`networkifs_base_config`
Description: Additional network interfaces to configure for Solaris deployments.

Example: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0` `ipmp_networkifs_base_config`

Description: Configures IP Multipathing for Solaris deployments.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`osp_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False)

Default value: `TRUE`

`sps_ra_parent`

Description: Name of parent for RA

Default value: `masterserver`

`sps_ra_parent_type`

Description: Whether RA parent is an MS or an LD

Default value: `MS`

`sps_ra_host`

Description: Name or IP address of RA host

Default value: `:[ethernet_ip_address]`

`sps_ra_conn`

Description: Type of RA connection (raw, ssl, ssh)

Default value: `RAW`

`sps_ra_ssl_auth`

Description: Whether RA has SSL authorization (y,n)

Default value: `N`

`sps_ra_port`

Description: Port for RA

Default value: `1131`

`sps_ra_adv_parms`

Description: Advanced parameters for RA

`linux_network_interface`

Description: Network interfaces to use to provision Linux to target hosts. Use this variable for Linux OS profiles created with JET.

Default value: `eth0`

`networkifs_redhat_base_config`

Description: Additional network interfaces to configure for RedHat Linux deployments. Use this variable for RedHat Linux OS profiles created with JET.

Example: `eth1!--ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

`networkifs_suse_base_config`

Description: Additional network interfaces to configure for SuSe Linux deployments. Use this variable for SuSe Linux OS profiles created with JET.

Example: `th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255`

`guid`

Description: GUID of the target specified within {}

Note – The GUID must be in the form {*dddddddd-dddd-dddd-dddd-dddddddddddd*}, where *d* is a hexadecimal text digit.

Example: `{3AE45192-159D-11D8-AEF3-001083FDFE8A}`

`disk_type`

Description: Disk type (for example, SCSI, IDE)

Default value: `scsi`

`netmgmt_type_1`

Description: Network management interface type (ipmi/alom/lom)

`netmgmt_version_1`

Description: Network management version

`netmgmt_hal_jar_1`

Description: Path to network management HAL JAR file

Default value:
`file://:[targetableComponent:installPath]/SUNWn1osp/drivers/sun/ipmi.jar`

`netmgmt_ip_address_1`

Description: Network management IP address

`netmgmt_protocol_1`

Description: Network management protocol (ipmi/telnet)

Default value: `telnet`

`netmgmt_port_1`

Description: Network management port

`power_service_true_false_1`

Description: Does the network management interface provide power on/off service?

Default value: `true`

`boot_service_true_false_1`

Description: Does the network management interface provide boot service (disk, then network)?

Default value: `true`

`console_service_true_false_1`
Description: Does the network management interface provide console service?

Default value: `false`

`netmgmt_userid_1`
Description: User ID to access network management interface

`netmgmt_password_1`
Description: Encrypted password to access network management interface

`terminal_server_type_1`
Description: Terminal server type

Default value: `Cisco2621`

`terminal_server_version_1`
Description: Terminal server version

`terminal_server_hal_jar_1`
Description: Path to terminal server HAL JAR file

If you are using a Cyclades terminal server, set this value to `file:///opt/n1osp/drivers/cyclades/cycladesTS.jar`.

Default value:
`file://:[targetableComponent:installPath]/SUNWn1osp/drivers/cisco/c2621.jar`

`terminal_server_ip_address_1`
Description: IP address of terminal server connected to this system

Example: `10.5.133.135`

`terminal_server_protocol_1`
Description: Terminal server access protocol

Default value: `telnet`

`terminal_server_port_1`
Description: Port of terminal server

`terminal_server_userid_1`
Description: Terminal server user id

`terminal_server_password_1`
Description: Encrypted password for terminal server

`netmgmt_type_2`
Description: Network management interface type (ipmi/alom/lom)

Default value: `ipmi`

`netmgmt_version_2`
Description: Network management version

`netmgmt_hal_jar_2`
Description: Path to network management HAL JAR file

Default value: `file://:[targetableComponent:installPath]/SUNWn1osp/drivers/sun/ipmi.jar`

`netmgmt_ip_address_2`
Description: Network management IP address

`netmgmt_protocol_2`
Description: Network management protocol (ipmi/telnet)

Default value: `telnet`

`netmgmt_port_2`
Description: Network management port

`power_service_true_false_2`
Description: Does the network management interface provide power on/off service?

Default value: `false`

`boot_service_true_false_2`
Description: Does the network management interface provide boot service (disk, then network)?

Default value: `false`

`console_service_true_false_2`
Description: Does the network management interface provide console service?

Default value: `false`

`netmgmt_userid_2`
Description: User ID to access network management interface

`netmgmt_password_2`
Description: Encrypted password to access network management interface

`terminal_server_type_2`
Description: Terminal server type

Default value: `Cisco2621`

`terminal_server_version_2`
Description: Terminal server version

`terminal_server_hal_jar_2`
Description: Path to terminal server HAL JAR file

Default value: `file://:[targetableComponent:installPath]/SUNWn1osp/drivers/cisco/c2621.jar`

`terminal_server_ip_address_2`
Description: IP address of terminal server connected to this system

Example: `10.5.133.135`

`terminal_server_protocol_2`
Description: Terminal server access protocol

`terminal_server_port_2`
Description: Port of terminal server

`terminal_server_userid_2`
Description: Terminal server user id

`terminal_server_password_2`
Description: Encrypted password for terminal server

Generic Target Host Variables

The following list shows 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, and for which you do not have HAL implementations that automate power, boot, and console operations. 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.

`installPath`
Description: Physical host name for the target system

`target_host`
Description: Virtual host name for the target system

Default value: `:[installPath]-target`

`description`
Description: Optional description of the target

Example: `test ultra 10`

`ethernet_mac_address`
Description: Mac address

Example: `00:0E:0C:07:E9:31`

`ethernet_ip_address`
Description: IP address

Example: `10.42.42.77`

`ethernet_netmask`
Description: Network mask

Default value: `255.255.255.0`

`architecture`

Description: Target architecture (for example, sun4u, x86)

Default value: `sun4u`

`kernel_arch`

Description: Kernel architecture (sun4u, i86pc)

Default value: `sun4u`

`disk_size`

Description: Size of disk

Default value: `20GB`

`default_router`

Description: Default router to use for network connectivity

Example: 10.42.42.1

`sysidcfg_default_route`

Description: Default router IP address for Solaris deployments.

Example: 10.42.42.1

`sysidcfg_network_interface`

Description: Network interface to configure for Solaris deployments.

Default value: PRIMARY

`networkifs_base_config`

Description: Additional network interfaces to configure for Solaris deployments.

Example: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

Description: Configures IP Multipathing for Solaris deployments.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`osp_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions

Default value: `TRUE`

`sps_ra_parent`

Description: Name of parent for RA

Default value: `masterserver`

`sps_ra_parent_type`

Description: Whether RA parent is an MS or an LD

Default value: `MS`

`sps_ra_host`

Description: Name or IP address of RA host

Default value: `:[ethernet_ip_address]`

`sps_ra_conn`

Description: Type of RA connection (raw, ssl, ssh)

Default value: `raw`

`sps_ra_ssl_auth`

Description: Whether RA has SSL authorization (y,n)

Default value: `N`

`sps_ra_port`

Description: Port for RA

Default value: `1131`

`sps_ra_adv_parms`

Description: Advanced parameters for RA

`linux_network_interface`

Description: Network interfaces to use to provision Linux to target hosts. Use this variable for Linux OS profiles created with JET.

Default value: `eth0`

`networkifs_redhat_base_config`

Description: Additional network interfaces to configure for RedHat Linux deployments. Use this variable for RedHat Linux OS profiles created with JET.

Example: `eth1!--ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

`networkifs_suse_base_config`

Description: Additional network interfaces to configure for SuSe Linux deployments. Use this variable for SuSe Linux OS profiles created with JET.

Example: `th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255`

`guid`

Description: GUID of the target specified within {}

Note – The GUID must be in the form {*dddddddd-dddd-dddd-dddd-dddddddddddd*}, where *d* is a hexadecimal text digit.

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 Solaris OS Configuration Variables" on page 207
- "Component Variables for Solaris Remote Agents" on page 219
- "General Solaris OS Provisioning Plug-In Variables" on page 221

Basic Solaris OS Configuration Variables

Component Variables for `base_config` Module

The variables described in the following list provide basic operating system configuration. These variables directly equate to the JET `base_config` module. In the list 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.

`ClientArch_base_config`

Description: Kernel architecture, such as sun4u or x86. By default, this is set to the kernel architecture of the targetable component.

Default value: `:[targetableComponent:kernel_arch]`

`ClientEther_base_config`

Description: Ethernet MAC address. By default, this is set to the Ethernet MAC address of the targetable component.

Default value: `:[targetableComponent:ethernet_mac_address]`

`ClientOS_base_config`

Description: Version of the OS to be provisioned.

Example: `Solaris9_u7_sparc`

`client_allocation_base_config`

Description: 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.

If you are provisioning the Solaris 10 1/06 x86 release, set the value of this variable to `grub` to enable Grub-based booting and installation.

Note – If you set the `client_allocation_base_config` to `bootp`, set the `osp_profile_control_service` to `false`.

`grub_append_base_config`

Description: For Solaris 10 1/06 x86 systems, specifies additional options or arguments to pass to the GRUB bootloader.

`products_base_config`

Description: JET modules to provision.

Example: `base_config spsra`

`sysidcfg_nameservice_base_config`

Description: Name service to configure at initial boot.

Default value: `NIS`

`sysidcfg_network_interface_base_config`

Description: Network interface to use at initial boot.

Default value: `PRIMARY`

`sysidcfg_ip_address_base_config`

Description: IP address to use at initial boot. By default, this is set to the IP address of the targetable component.

Default value: `:[targetableComponent:ethernet_ip_address]`

`sysidcfg_netmask_base_config`

Description: Netmask to use at initial boot. By default, this is set to the netmask of the targetable component.

Default value: `:[targetableComponent:ethernet_netmask]`

`sysidcfg_root_password_base_config`

Description: Encrypted root password.

Example: `boajrOmU7GFmY` is the encrypted value for the password "newroot."

`sysidcfg_system_locale_base_config`

Description: System locale to use at initial boot.

Example: `en_US.ISO8859-1`

`sysidcfg_timeserver_base_config`

Description: Where to get system time for initial boot. If blank, system time comes from the JET server. Alternatively, you can set this variable to `localhost` to get the system time from the hardware clock on the client.

`sysidcfg_timezone_base_config`

Description: System time zone to use for initial boot.

Example: `US/Pacific`

`sysidcfg_terminal_base_config`

Description: Terminal emulator to set at initial boot.

Default value: `vt100`

`sysidcfg_security_policy_base_config`

Description: Kerberos security policy to use at initial boot.

Default value: `NONE`

`sysidcfg_protocol_ipv6_base_config`

Description: Whether to use IPv6 protocol at initial boot.

Default value: `no`

`sysidcfg_default_route_base_config`

Description: Router IP address to use during JumpStart for Solaris 9 or later environments. If blank, will try to use value from the `defaultrouter_base_config` variable. If that is also blank, or for another net interface, JumpStart sysidcfg will get a router IP from the Jet server.

`x86_nowin_base_config`

Description: For x86 systems, prevents Solaris from trying to run Windows during the install.

Default value: `yes`

`x86_console_base_config`

Description: 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`

Description: 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`

Description: 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`.

`x86_disable_kdmconfig_base_config`

Description: For Solaris x86 systems, disables the `kdmconfig` interactive utility for configuring the keyboard, display, and mouse of the target host. If you are installing a Solaris OS with the GRUB bootloader, set this variable value to `yes`.

`x86_confflags_base_config`

Description: For Solaris 9 x86 systems, specifies arguments to be used with the confflags attribute of the `add_install_client` command.

Example: `-f -P /boot/solaris/dca`.

`sysidcfg_x86_kdmfile_base_config`

Description: For Solaris x86 systems, specifies the name of a keyboard, display, and mouse configuration file to append to the `sysidcfg` file.

Example: `/sysidcfg-addon-file`.

`profile_base_config`

Description: Create your own custom JumpStart profile.

By default, if you leave this variable blank, the OS provisioning plug-in creates the `/opt/SUNWjet/Clients/`*hostname*`/profile` for you based on the following 21 `profile_*` variables. Alternatively, you can create your own custom JumpStart profile as explained in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*. To use the profile that you created manually, set the `profile_base_config` variable to the name of the created profile. By default, the OS provisioning plug-in looks for the profile in the `/opt/SUNWjet/Clients/`*hostname* directory. To direct the plug-in to a profile in another directory, provide an absolute path name in the `profile_base_config` variable.

Note – If you are provisioning the Solaris OS on x86 target hosts, you must create a custom JumpStart profile that deletes any existing partitions, and point to that profile in the `profile_base_config` variable. For instructions about how to create a JumpStart profile that deletes existing partitions with the `fdisk` keyword, see *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

`profile_cluster_base_config`

Description: 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

Default value: `SUNWCreq`

`profile_usedisk_base_config`

Description: 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.

Default value: `rootdisk`

`profile_dontuse_base_config`

Description: 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`

Description: Root space (free, or size in Megabytes)

Default value: `free`

`profile_swap_base_config`

Description: Swap space (in Megabytes).

Default value: `256`

`profile_s3_mtpt_base_config`

Description: Mount path to the slice 3 partition.

Note – 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`

Description: Size of the slice 3 partition (in Megabytes).

`profile_s4_mtpt_base_config`

Description: Mount path to the slice 4 partition.

`profile_s4_size_base_config`

Description: Size of the slice 4 partition (in Megabytes).

`profile_s5_mtpt_base_config`

Description: Mount path to the slice 5 partition.

Default value: `/var`

`profile_s5_size_base_config`

Description: Size of the slice 5 partition (in Megabytes).

`profile_s6_mtpt_base_config`

Description: Mount path to the slice 6 partition.

Default value: `/usr`

`profile_s6_size_base_config`

Description: Size of the slice 6 partition (in Megabytes).

`profile_s7_mtpt_base_config`

Description: Mount path to the slice 7 partition.

Note – 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.

Default value: `/opt`

`profile_s7_size_base_config`

Description: Size of the slice 7 partition (in Megabytes).

`profile_additional_disks_base_config`

Description: A list of disks to use and configure in addition to the boot disk. 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. For more information, see "Configuring Additional Disks" on page 217.

`profile_add_locales_base_config`

Description: Comma-separated list of locales to add.

Example: `fr_FR, ja_JP.UTF-8`

`profile_del_locales_base_config`

Description: Comma-separated list of locales to delete.

`profile_add_geos_base_config`

Description: Comma-separated list of geographical regions to add.

Example: `N_Europe, C_Europe`

`profile_del_geos_base_config`

Description: Comma-separated list of geographical regions to delete.

`ufs_logging_filesys_base_config`

Description: 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`.

Note – 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.

Default value: `all`

`profile_add_packages_base_config`

Description: Space-separated list of packages to add.

`profile_del_packages_base_config`

Description: Space-separated list of packages to remove.

To prevent interactive installations on Solaris x86 headless target hosts, set this value to `SUNWxwssu SUNWxwscf`.

`profile_add_clusters_base_config`

Description: Space-separated list of cluster packages to add.

`profile_del_clusters_base_config`

Description: Space-separated list of cluster packages to remove.

Example: `SUNWCpm SUNWCpmx SUNWCdial SUNWCdialx`

`nfs_mounts_base_config`

Description: Space-separated list of remote NFS mount points. Use ? to separate the mount source from the mount target, as shown in the example.

Example: `fs?1.1.1.1:/fs`

`nodename_base_config`

Description: Value to use for `/etc/nodename` if not the default hostname.

`defaultrouter_base_config`

Description: Value to use for `/etc/defaultrouter`.

`notrouter_base_config`

Description: If set to `y`, then disable IPv4 forwarding and create `/etc/notrouter` file.

`dns_domain_base_config`

Description: DNS domain entry for `/etc/resolv.conf` file.

`dns_nameservers_base_config`

Description: Space-separated list of IP addresses to use for DNS nameserver entries in `/etc/resolv.conf` file.

`dns_searchpath_base_config`

Description: List of entries to go in the DNS search line in `/etc/resolv.conf` file.

`dns_disableforbuild_base_config`

Description: Delay DNS configuration until later. If DNS is not available in the build environment, set this variable to `yes`.

`ntp_servers_base_config`

Description: Space-separated list of names or IP addresses for the NTP servers. The first server will be given a "prefer" tag. This section places 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.

`networkifs_base_config`

Description: Space-separated list of additional network interfaces to be defined. For logical interfaces, use underscores (_) rather than colons (`:`). Use the format `c`*n*`t`*n*`d`*n*. For each interface listed, define sets of variables to provide the netname, netmask, hostname, and IP address for the interface. For more information about this variable, see "Configuring Additional Network Interfaces" on page 215.

Example: `le1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

Description: 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. For more information about this variable, see "Configuring IP Network Multipathing (IPMP)" on page 216.

Example: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`update_terminal_base_config`

Description: If set, put the `sysidcfg` terminal type into `inittab`.

Default value: `yes`

`enable_savecore_base_config`

Description: If set to any value, enable save core for Solaris 2.6 systems.

Default value: `yes`

`dumpadm_minfree_base_config`

Description: Set a limit so that crash dumps do not fill up the `dump` filesystem. See dumpadm(1M) -m option for possible values.

Example: `20000k`

`noautoshutdown_base_config`

Description: If set to any value, disable power management.

Default value: `pm_disabled`

`enable_rootlogin_base_config`

Description: If set to any value, enable network root login from both telnet/rsh and ssh.

`enable_rootftp_base_config`

Description: If set to any value, enable root FTP access.

`shutup_sendmail_base_config`

Description: If set, create an alias hostname to shut up sendmail.

Default value: `yes`

`poweroff_afterbuild_base_config`

Description: If set, shut down the machine once the build completes.

`dedicated_dump_device_base_config`

Description: If set, the `dumpadm` utility configures the partition as a Dedicated Dump Device. See dumpadm(1M) for supported Operating Environments.

enable_altbreak_base_config
: **Description:** If set, enable alternate break sequence.

disable_sysid_probe_base_config
: **Description:** If set, skip the sysid stuff on the first reboot. This can significantly increase provisioning efficiency on systems that have many unused network adapters.

: **Default value:** yes

nfsv4_domain_base_config
: **Description:** 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
: **Description:** 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
: **Description:** 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 Network Interfaces

The networkifs_base_config variable provides a way to configure additional network interfaces, in addition to the first network interface configured through the sysidcfg_network_interface_basevariable.

For each additional interface device, you provide a group of four variables that define the netname, netmask, hostname, and IP address for the interface.

The example below shows how to define the networkifs_base_config variable for two interfaces:

- le1
  - netname — netB
  - netmask — 255.255.255.0
  - hostname — myhost-netB
  - interface IP — 192.168.1.0
- hme0
  - netname — netC
  - netmask — 255.255.255.0
  - hostname — myhost-netC
  - interface IP — 192.162.2.0

You would use the following syntax to define the single networkifs_base_config variable for the two additional interfaces:

```
le1!netB 255.255.255.0 myhost-netB 192.168.1.0|hme0!netC 255.255.255.0 myhost-netC 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_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4
```

Configuring Additional Disks

The profile_additional_disks_base_config variable provides the ability to format and configure disks in addition to the root disk. Use the format c*n*t*n*d*n* to identify each additional disk.

The following example illustrates a three-disk configuration. The root disk is c0t0d0 and is defined using the variable profile_usedisk_base_config. The profile_additional_disks_base_config variable identifies two additional disks: c0t1d0 and c0t1d1 For each disk slice, a directory name is specified along with a size for that directory.

```
profile_usedisk_base_config = c0t0d0
profile_additional_disks_base_config = c0t1d0s0!/export/mydisk1!5000MB|c0t1d1s0!/export/mydisk2!6000MB
```

When you run the provision_start plan on the Solaris profile, appropriate files will be created to configure the two disks, with appropriate mount points and slices.

Automating Solaris 9 Installations on x86 Based Systems

To fully automate Solaris 9 installations on x86 based systems, you need to create additional configuration files to supply information to the Solaris Device Configuration Assistant (DCA) and kdmconfig utility.

▼ How to Automate Solaris 9 x86 Installations

1 Create a file named dca with the following contents.

```
IF 0,2="Solaris Device Configuration" THEN
F2
FI
```

```
IF *,0="Enter_Continue" THEN
ENTER
FI
IF *,0="Enter_Continue" THEN
ENTER
FI
IF *,0="Enter_Continue" THEN
ENTER
FI
F2 F2
```

2. **Save the `dca` file to the `boot/solaris` directory in the Solaris 9 installation image.**

   For example, save this file to the `/export/s9_x86/Solaris_9/Tools/Boot/boot/solaris` directory.

3. **Edit the `bootrc` file in the `Solaris_9/Tools/Boot/etc` directory of the Solaris 9 installation image to read as follows.**

```
#       @(#)bootrc 1.10 01/04/18
#
# bootrc script for installations
#
# ER - This script has been reduced to just start up the kernel boot.
# No user interaction.

# definitions
set DEFAULT_BOOTFILE kernel/unix

# retrieve boot-path constructed by MDB
set bootfile $DEFAULT_BOOTFILE
getprop boot-path bp
getprop boot-args ba
set bootargs "${ba}"

# display current defaults
echo "                        <<< Current Boot Parameters >>>
Boot path: ${bp}
Boot args: ${ba}

"

        setprop boot-args '- install nowin'
        echo '
<<< starting automatic installation >>>
'
echo "Booting $DEFAULT_BOOTFILE..."
run $DEFAULT_BOOTFILE
```

4 **Create a file named `/sysidcfg-addon-file` that contains the following text.**

```
keyboard=workaround:Unknown
display=workaround:Unknown
pointer=workaround:Unknown
monitor=workaround:Unknown
```

5 **Edit the following basic Solaris OS configuration variables in the N1 SPS browser interface.**

`x86_confflags_base_config`	Set this variable to point to the `dca` file in the Solaris installation image directory. For example, `-f -P/boot/solaris/dca`.
`sysidcfg_x86_kdmfile_base_config`	Set this variable to point to the `/sysidcfg-addon-file` you created.

For instructions about how to create or edit Solaris OS Provisioning profile variables, see "How to Create a Non-JET Solaris Provisioning Profile (Browser Interface)" on page 69.

Component Variables for Solaris Remote Agents

The variables described in the following list provide Remote Agent (RA) configuration for the N1 SPS software. These variables directly equate to the JET `spsra` module.

`install_spsra`

Description: Whether to install the N1 SPS RA software.

Default value: `y`

`product_version_spsra`

Description: Version of RA to install.

Default value: `5.2`

If you are using the N1 SPS 5.1 release to provision with the OS Provisioning Plug-In 3.0, set this variable value to `5.1`.

`basedir_spsra`

Description: Base directory in which RA files are installed.

Default value: `/opt/SUNWn1sps`

`install_ra_from_snapshot_spsra`

Note – The `install_ra_from_snapshot_spsra` is deprecated in the N1 SPS 5.2 release.

Default value: `n`

`username_spsra`

Description: When using the RA installer, specify user name to own the RA filesystem.

Example: `jdoe`

`groupname_spsra`

Description: When using the RA installer, specify group name to own the RA filesystem.

Example: `staff`

`setup_user_acct_spsra`

Description: Determines whether to set up user and group account for `username_spsra` and `groupname_spsra`.

Note – If the user account is not available to the target after provisioning, set this variable value to `y` to start an N1 SPS Remote Agent on the target host on reboot after provisioning.

Default value: `n`

`user_passwd_spsra`

Description: Encrypted password for `username_spsra`.

Example: `6GR9gOi1vRVFw` shows the encrypted value for the password "n1spspwd."

`userid_spsra`

Description: User account ID for `username_spsra`.

Example: `505`

`groupid_spsra`

Description: Group account ID for `groupname_spsra`.

Example: `505`

`homedir_spsra`

Description: Home directory for RA files.

Default value: `/opt/SUNWn1sps`

`jre_path_spsra`

Description: 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`

Description: Enable RA root access.

Default value: `y`

`setuid_spsra`

Description: Host name or IP address of the Remote Agent.

Example: `solaris10-target`

`conn_spsra`
Description: Determines method to use for connecting to RA. By default, uses the connection method defined for the targetable component.

Default value: `:[targetableComponent:sps_ra_conn]`

`ssl_auth_spsra`
Description: SSL authorization to use for network connection.

Default value: `1`

`port_spsra`
Description: Port to use for network connection. By default, uses the port defined for the targetable component.

Note – To use an SSH connection, set the port number to 70000.

Default value: `:[targetableComponent:sps_ra_port]`

`adv_params_spsra`
Description: Advanced parameters to use for network connection. By default, uses the advanced parameters defined for the targetable component.

Default value: `:[targetableComponent:sps_ra_adv_parms]`

General Solaris OS Provisioning Plug-In Variables

The variables in the following list 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 SPS software.

`installPath`
Description: Jumpstart Enterprise Toolkit (JET) virtual host.

Example: `myhost-jet`

`osp_target_host`
Description: Target host name.

Default value: `:[targetableComponent:installPath]`

`osp_profile_control_service`
Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False).

Note – If you set the `client_allocation_base_config` to `bootp`, set the `osp_profile_control_service` to `false`.

Default value: `:[targetableComponent:osp_control_service]`

`sps_ra_parent_host`

Description: Host name of the N1 SPS RA parent.

Default value: `:[targetableComponent:sps_ra_parent]`

`sps_ra_parent_host_type`

Description: Host type of the N1 SPS RA parent (MS or LD).

Default value: `:[targetableComponent:sps_ra_parent_type]`

`osp_jet_server`

Description: Host name of the JET server.

Example: `myserver`

`osp_profile_architecture`

Description: Target host architecture (sun4u, x86).

Default value: `:[targetableComponent:architecture]`

`osp_profile_min_disk_size`

Description: Minimum disk size.

Example: `5GB`

`osp_profile_description`

Description: Profile description.

`osp_profile_boot_server_name`

Description: Solaris boot server name.

Default value: `:[osp_jet_server]`

`osp_profile_BS_subnet_address`

Description: Subnet served by boot server.

`osp_profile_BS_subnet_ip_address`

Description: IP address of boot server on this subnet.

Example: `199.199.8.6`

`osp_profile_product_ip_address`

Description: IP address of Solaris image server.

Example: `199.199.8.6`

`osp_profile_product_directory`
Description: Path to image directory on image server.

Example: `/osp_image/Solaris9_u7_sparc`

`osp_profile_boot_ip_address`
Description: IP address of Solaris boot server.

Example: `199.199.8.6`

`osp_profile_boot_directory`
Description: Solaris boot directory.

Default value: `/osp_image/Solaris9_u7_sparc/Solaris_9/Tools/Boot`

`osp_profile_sysidcfg_ip_address`
Description: IP address of system ID configuration.

Example: `199.199.8.6`

`osp_profile_sysidcfg_directory`
Description: Directory of system ID configuration.

Default value: `/opt/SUNWjet/Clients/:[osp_target_host]`

`osp_profile_jumpstart_ip_address`
Description: JumpStart IP address

Example: `199.199.8.6`

`osp_profile_jumpstart_directory`
Description: JumpStart directory

Default value: `/opt/SUNWjet`

`osp_profile_boot_file_name`
Description: Solaris boot file name.

Default value: `:[osp_target_host]`

`osp_profile_nbp_boot_file_name`
Description: Network boot file name.

Default value: `:[osp_target_host]`

APPENDIX C

RedHat Linux Profile Component Variables

The RedHat Linux Profile component includes variables for each JET module. These variables are separated into the following categories:

- "Basic RedHat OS Configuration Variables" on page 225
- "Component Variables for RedHat Remote Agents" on page 239
- "General RedHat Linux OS Provisioning Plug-In Variables" on page 241

Basic RedHat OS Configuration Variables

Component Variables for `base_config` Module

The variables described in the following list provide basic operating system configuration. These variables directly equate to the JET `base_config` module. In the list you see several variables that use the value of the *targetable component*. For purposes of the RedHat Linux Profile variables, the targetable component is the target host.

`ClientArch_base_config`

Description: Kernel architecture, such as x86. By default, this is set to the kernel architecture of the targetable component.

Default value: `:[targetableComponent:kernel_arch]`

`ClientEther_base_config`

Description: Ethernet MAC address. By default, this is set to the Ethernet MAC address of the targetable component.

Default value: `:[targetableComponent:ethernet_mac_address]`

`ClientOSName_base_config`

Description: Name of the OS to be provisioned.

Example: `redhat`

`ClientOS_base_config`

Description: Version of the OS to be provisioned.

Example: `LinuxRedHat3.0_x86`

`products_base_config`

Description: JET modules to provision.

Example: `base_config spsra`

`client_allocation_base_config`

Description: 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.

`productdir_base_config`

Description: Path to the RedHat Linux distribution. To use the information from the `jumpstart.conf` file and the IP address of the JET server, leave this value blank. 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`

Description: Path to the patches you want to apply. To use the information from the `jumpstart.conf` file and the IP address of the JET server, leave this value blank. If your patches are not stored on the JET server, then provide an NFS-style path to the location of the patches.

`kickstart_media_base_config`

Description: Location of the RedHat Linux product media.

`pxelinux_console_base_config`

Description: Console parameters that are passed to the PXE boot environment.

Default value: `ttyS0,9600`

`lang_base_config`

Description: Languages to install on the target host.

Example: `en_US`

`lang_additional_base_conf`

Description: Additional languages to install on the target host.

Example: `fr_FR`

`keyboard_base_conf`

Description: Keyboard mapping to use on the target host.

Default value: `us`

`mouse_base_config`

Description: Mouse type for the target host.

Default value: `generic3`

`auth_md5_base_config`

Description: Specifies whether to use MD5 authentication..

Default value: `yes`

`auth_shadow_base_config`

Description: Specifies whether to use the shadow password.

Default value: `yes`

`auth_cache_base_config`

Description: Specifies whether to use the authentication cache.

Default value: `no`

`auth_nis_enable_base_config`

Description: Specifies whether to enable NIS (yes/no).

`auth_nis_server_base_config`

Description: Name of the NIS server, if `auth_nis_enable_base_config` is set to `yes`.

Example: `nis-1.west.example.com`

`auth_nis_domain_base_config`

Description: Name of the NIS domain..

Example: `west.example.com`

`auth_ldap_enable_base_config`

Description: Specifies whether to enable LDAP (yes/no).

`auth_ldap_enable_auth_base_config`

Description: Specifies whether to use LDAP for user authentication (yes/no).

`auth_ldap_server_base_config`

Description: Name of the LDAP server.

Example: `ldap.west.example.com`.

`auth_ldap_basedn_base_config`

Description: Base domain name for the LDAP server.

Example: `west.example.com`.

`auth_ldap_tls_base_config`

Description: Specifies whether to use the Transport Layer Security protocol (TLS) for LDAP (yes/no).

`auth_kerberos5_kdcs_base_config`

Description: List of Kerberos key distribution centers (KDCs). Include entries in a space separated list.

`auth_kerberos5_realm_base_config`

Description: Name of the Kerberos default realm.

`auth_kerberos5_adminserver_base_config`
Description: Name of the Kerberos administration server.

`auth_hesiod_enabled_base_config`
Description: Specifies whether to enable the Hesiod name service (yes/no).

`auth_hesiod_right_side_base_config`
Description: Specifies whether to use the Hesiod right side option, set in `/etc/hesiod.conf`(yes/no).

This option is used by the Hesiod library to determine the name to search DNS for when looking up information, similar to LDAP's use of a base domain name.

`auth_hesiod_left_side_base_config`
Description: Specifies whether to use the Hesiod left side option, set in `/etc/hesiod.conf`(yes/no).

This option is used by the Hesiod library to determine the name to search DNS for when looking up information, similar to LDAP's use of a base domain name.

`auth_smb_servers_base_config`
Description: Name of the Server Message Block server. SMB is is a protocol for sharing files, printers, serial ports, and communications abstractions such as named pipes and mail slots between computers.

`auth_smb_workgroup_base_config`
Description: Name of the SMB workgroup.

`root_password_encrypted_base_config`
Description: Specifies whether the root password for the target host is encrypted in the `root_password_base_config` variable (yes/no).

Default value: `no`

`root_password_base_config`
Description: Root password for the target host. If the `root_password_encrypted_base_config` variable is set to `yes`, set this value to the encrypted password.

Default value: `newroot`

`timezone_base_config`
Description: Time zone for the target host.

Default value: `US/Pacific`

`timezone_utc_base_config`
Description: Specifies whether the time zone is set in Coordinated Universal Time (yes/no).

Default value: `no`

`firewall_base_config`
Description: Specifies whether to enable a firewall on the target host (enabled/disabled).

Default value: `disabled`

`firewall_trusted_base_config`

Description: Space-separated list of network interfaces for which all traffic is allowed to pass through the firewall.

Example: `eth0 eth1`

`firewall_incoming_base_config`

Description: Space-separated list of port numbers and services, such as ssh, http, dhcp, telnet, smtp, or ftp. You can also specify service types and the associated protocol in the format `service:protocol`.

Example: `imap:tcp 1234:udp ssh ftp`

`bootloader_name_base_config`

Description: Name of the bootloader to boot and install the target host (grub/lilo).

Default value: `grub`

`bootloader_location_base_config`

Description: Location of the bootloader.

Default value: `mbr`

`bootloader_append_base_config`

Description: Space-separated list of kernel parameters for the bootloader to pass to the kernel.

Example: `hdd=ide-scsi ide=nodma`

`bootloader_lilo_lba32_base_config`

Description: If the `bootloader_name_base_config` variable is set to `lilo`, this variable specifies whether the bootloader should use lba32.

Default value: `yes`

`bootloader_lilo_linear_base_config`

Description: If the `bootloader_name_base_config` variable is set to `lilo`, this variable specifies whether the bootloader should use linnear.

Default value: `yes`

`bootloader_grub_password_encrypted_base_config`

Description: If the `bootloader_name_base_config` variable is set to `grub`, this variable specifies whether the GRUB root password is encrypted in the `root_password_base_config` variable (yes/no).

Example: `no`

`bootloader_grub_password_base_config`

Description: GRUB root password for the target host. If the `bootloader_grub_password_encrypted_base_config` variable is set to `yes`, set this value to the encrypted password.

`xconfig_card_base_config`
Description: Video card for the target host.

`xconfig_videoram_base_config`
Description: Video RAM on the target host.

`xconfig_monitor_base_config`
Description: Monitor model for the target host.

`xconfig_resolution_base_config`
Description: Resolution of the monitor on the target host.

`xconfig_depth_base_config`
Description: Color depth of the monitor on the target host.

`xconfig_desktop_base_config`
Description: Specifies the desktop software to use on the target host.

`bootproto_base_config`
Description: Boot protocol for the target host.

Default value: `dhcp`

`network_interface_base_config`
Description: Network interface to use during provisioning.

Default value: `:[targetableComponent:linux_network_interface]`

`ip_base_config`
Description: IP address of the target host.

Default value: `:[targetableComponent:ethernet_ip_address]`

`hostname_base_config`
Description: Host name of the target host.

Default value: `:[targetableComponent:installPath]`

`netmask_base_config`
Description: Netmask to use during provisioning.

Default value: `:[targetableComponent:ethernet_netmask]`

`gateway_base_config`
Description: Gateway to use during provisioning.

Default value: `:[targetableComponent:default_router]`

`nameservers_base_config`
Description: Space-separated list of name servers to use during provisioning.

`networkifs_base_config`
Description: Specifies additional network interfaces to configure. This variable value is derived from the target host variable `networkifs_redhat_base_config`.

Example: `l eth1!--ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

`partition_clear_base_config`

Description: Specifies partitions to clear during provisioning.

Example: `all`

`partition_zerombr_base_config`

Description: Specifies whether to clear the master boot record (MBR).

Default value: `yes`

`partition_1_disk_base_config`

Description: Specifies the device for the first partition.

Default value: `sda`

Note – Be sure to define the partition variables sequentially, starting with partition 1. If you do not start with partition 1, the installation switches to interactive mode.

`partition_1_mountpoint_base_config`

Description: Specifies the mount point for the first partition.

Default value: `/diag`

`partition_1_fstype_base_config`

Description: Specifies the file system type for the first partition.

Default value: `vfat`

`partition_1_size_base_config`

Description: Specifies the size in MB for the first partition.

Default value: `60`

`partition_1_maxsize_base_config`

Description: Specifies the maximum size in MB for the first partition.

`partition_1_grow_base_config`

Description: Specifies whether to allow the partition to grow (yes/no).

`partition_1_primary_base_config`

Description: Specifies whether to force a primary partition (yes/no).

Default value: `yes`

`partition_2_disk_base_config`

Description: Specifies the device for the second partition.

Default value: `sda`

`partition_2_mountpoint_base_config`
Description: Specifies the mount point for the second partition.

Default value: `/boot`

`partition_2_fstype_base_config`
Description: Specifies the file system type for the second partition.

Default value: `ext3`

`partition_2_size_base_config`
Description: Specifies the size in MB for the second partition.

Default value: `100`

`partition_2_maxsize_base_config`
Description: Specifies the maximum size in MB for the second partition.

`partition_2_grow_base_config`
Description: Specifies whether to allow the partition to grow (yes/no).

`partition_2_primary_base_config`
Description: Specifies whether to force a primary partition (yes/no).

Default value: `yes`

`partition_3_disk_base_config`
Description: Specifies the device for the third partition.

Default value: `sda`

`partition_3_mountpoint_base_config`
Description: Specifies the mount point for the third partition.

Default value: `/`

`partition_3_fstype_base_config`
Description: Specifies the file system type for the third partition.

Default value: `ext3`

`partition_3_size_base_config`
Description: Specifies the size in MB for the third partition.

Default value: `10000`

`partition_3_maxsize_base_config`
Description: Specifies the maximum size in MB for the first partition.

`partition_3_grow_base_config`
Description: Specifies whether to allow the partition to grow (yes/no).

`partition_3_primary_base_config`
Description: Specifies whether to force a primary partition (yes/no).

`partition_4_disk_base_config`
Description: Specifies the device for the fourth partition.

Default value: `sda`

`partition_4_mountpoint_base_config`
Description: Specifies the mount point for the fourth partition.

Default value: `swap`

`partition_4_fstype_base_config`
Description: Specifies the file system type for the fourth partition.

Default value: `swap`

`partition_4_size_base_config`
Description: Specifies the size in MB for the fourth partition.

Default value: `1024`

`partition_4_maxsize_base_config`
Description: Specifies the maximum size in MB for the fourth partition.

`partition_1_grow_base_config`
Description: Specifies whether to allow the partition to grow (yes/no).

`partition_4_primary_base_config`
Description: Specifies whether to force a primary partition (yes/no).

`partition_5_disk_base_config`
Description: Specifies the device for the fifth partition.

`partition_5_mountpoint_base_config`
Description: Specifies the mount point for the fifth partition.

`partition_5_fstype_base_config`
Description: Specifies the file system type for the fifth partition.

`partition_5_size_base_config`
Description: Specifies the size in MB for the fifth partition.

`partition_5_maxsize_base_config`
Description: Specifies the maximum size in MB for the fifth partition.

`partition_5_grow_base_config`
Description: Specifies whether to allow the partition to grow (yes/no).

`partition_5_primary_base_config`
Description: Specifies whether to force a primary partition (yes/no).

`partition_6_disk_base_config`
Description: Specifies the device for the sixth partition.

`partition_6_mountpoint_base_config`
Description: Specifies the mount point for the sixth partition.

`partition_6_fstype_base_config`
Description: Specifies the file system type for the sixth partition.

`partition_6_size_base_config`
Description: Specifies the size in MB for the sixth partition.

`partition_6_maxsize_base_config`
Description: Specifies the maximum size in MB for the sixth partition.

`partition_6_grow_base_config`
Description: Specifies whether to allow the partition to grow (yes/no).

`partition_6_primary_base_config`
Description: Specifies whether to force a primary partition (yes/no).

`partition_7_disk_base_config`
Description: Specifies the device for the seventh partition.

`partition_7_mountpoint_base_config`
Description: Specifies the mount point for the seventh partition.

`partition_7_fstype_base_config`
Description: Specifies the file system type for the seventh partition.

`partition_7_size_base_config`
Description: Specifies the size in MB for the seventh partition.

`partition_7_maxsize_base_config`
Description: Specifies the maximum size in MB for the seventh partition.

`partition_7_grow_base_config`
Description: Specifies whether to allow the partition to grow (yes/no).

`partition_7_primary_base_config`
Description: Specifies whether to force a primary partition (yes/no).

`partition_8_disk_base_config`
Description: Specifies the device for the eighth partition.

`partition_8_mountpoint_base_config`
Description: Specifies the mount point for the eighth partition.

`partition_8_fstype_base_config`
Description: Specifies the file system type for the eighth partition.

`partition_8_size_base_config`
Description: Specifies the size in MB for the eighth partition.

`partition_8_maxsize_base_config`
Description: Specifies the maximum size in MB for the eighth partition.

`partition_8_grow_base_config`
Description: Specifies whether to allow the partition to grow (yes/no).

`partition_8_primary_base_config`

Description: Specifies whether to force a primary partition (yes/no).

`raid_1_device_base_config`

Description: Specifies the name of the RAID device to use (such as md0 or md1). RAID devices range from md0 to md7, and each may only be used once.

Example: `md0`

`raid_1_level_base_config`

Description: Specifies the level (0,1, or 5) for the first RAID device.

Example: 1

`raid_1_mountpoint_base_config`

Description: Specifies the mount point of the first RAID device.

Example: `raid.01`

`raid_1_meta_devices_base_config`

Description: Space-separated list of metadevices for the first RAID device. These metadevices are created in the `partition_` variables by specifying `raid.`*number* for the `partition _`*n*`_mountpoint_base_config` variable, for example, `partition_1_mountpoint_base_config=raid.01`

The `raid.01` and `raid.02` partition names are partition names which can be created with the JET `base_config` module. If you set the RAID level to 1 with the `raid_1_level_base_config` variable, you must to specify a minimum of two partitions of equal size. If you set the RAID level to 5 with the `raid_1_level_base_config` variable, you must to specify a minimum of three partitions of equal size.

`raid_2_device_base_config`

Description: Specifies the name of the RAID device to use (such as md0 or md1). RAID devices range from md0 to md7, and each may only be used once.

Example: `md1`

`raid_2_level_base_config`

Description: Specifies the level (0, 1, or 5) for the second RAID device.

Example: 1

`raid_2_mountpoint_base_config`

Description: Specifies the mount point of the second RAID device.

Example: `raid.02`

`raid_2_meta_devices_base_config`

Description: Space-separated list of metadevices for the second RAID device. These metadevices are created in the `partition_` variables by specifying `raid.`*number* for the `partition _`*n*`_mountpoint_base_config` variable, for example, `partition_2_mountpoint_base_config=raid.02`

The raid.01 and raid.02 partition names are partition names which can be created with the JET base_config module. If you set the RAID level to 1 with the raid_1_level_base_config variable, you must to specify a minimum of two partitions of equal size. If you set the RAID level to 5 with the raid_1_level_base_config variable, you must to specify a minimum of three partitions of equal size.

append_kernel_cmdline_base_config
Description: Additional command line options for the kernel at PXE boot time.

clusters_base_config
Description: Specifies the package groups to install.

Default value: Everything

packages_add_base_config
Description: Space-separated list that includes additional packages to install on the target host.

packages_remove_base_config
Description: Space-separated list that includes additional packages to remove from the target host.

packages_resolvedeps_base_config
Description: Specifies whether to resolve package dependencies during provisioning (yes/no).

packages_ignoredeps_base_config
Description: Specifies whether to ignore package dependencies during provisioning (yes/no).

packages_ignoremissing_base_config
Description: Specifies whether to ignore missing packages and continue provisioning (yes/no).

enable_services_base_config
Description: Space-separated list of Xinetd services to enable on the target host.

postscript_chroot_base_config
Description: Specifies whether to run JET postscripts in a different root directory (yes/no).

bootloader_grub_redirect_fix_base_config
Description: Specifies whether to redirect installation output to the console instead of a serial console(yes/no).

On certain hardware, the installer does not correctly set the serial or console display in the bootloader. If your hardware does not properly set the console redirection in the GRUB bootloader, set this value to yes.

no_autoreboot_base_config
Description: Specifies whether to reboot after the installation (yes/no).

If you set this value to no, any postinstall scripts are not run until the system is rebooted.

script_1_pre_base_config
Description: Specifies the location of a preinstall script on the JET server.

script_1_pre_interpreter_base_config
Description: Specifies the interpreter for the preinstall script.

`script_1_post_base_config`

Description: Specifies the location of a postinstall script on the JET server.

`script_1_post_interpreter_base_config`

Description: Specifies the interpreter for the postinstall script.

`script_1_post_nochroot_base_config`

Description: Specifies whether to run the postinstall script in a jail in an alternate root directory.

`partitionadd_base_config`

Description: Adds partitions to the target host, starting with partition 9. The variable value requires the following syntax.

n!*disk*!*mountpoint*!*fstype*!*size*!*maxsize*!*grow*!*primary*

The previous summary includes the following elements.

n specifies the partition number.

disk specifies the device for the partition.

mountpoint specifies the mount point for the partition.

fstype specifies the file system type for the partition.

size specifies the size in MBs of the partition.

maxsize specifies the maximum size for the partition.

grow specifies whether to enable the partition to grow.

primary specifies whether to force this partition as a primary partition.

Include the information for all the additional partitions you want to create in the variable value. For example, if you want to create a ninth and tenth partition, set this variable to a value similar to this example.

```
9!sda!/export1!ext3!1000!!!!|10!sda!/export10!jfs!100!2000!yes!!
```

This value creates the following variable set.

```
base_config_partition_9_disk=sda
base_config_partition_9_mountpoint=/export1
base_config_partition_9_fstype=ext3
base_config_partition_9_size=1000
base_config_partition_9_maxsize=
base_config_partition_9_grow=
base_config_partition_9_primary=

base_config_partition_10_disk=sda
```

```
base_config_partition_10_mountpoint=/export10
base_config_partition_10_fstype=jfs
base_config_partition_10_size=100
base_config_partition_10_maxsize=2000
base_config_partition_10_grow=yes
base_config_partition_10_primary=
```

`raidadd_base_config`

Description: Adds RAID devices to the target host, starting with device 3. The variable value requires the following syntax.

n!*device*!*level*!*mountpoint*!*metadevices*

The previous summary includes the following elements.

n specifies the RAID volume number.

device specifies the device for the RAID volume.

level specifies the RAID level for the volume.

mountpoint specifies the mount point for the RAID volume.

metadevices specifies metadevices to create for the RAID volume.

Include the information for all the additional partitions you want to create in the variable value. The `raidadd_base_config` variable uses the same basic syntax as the `partitionadd_base_config` variable.

`scriptpreadd_base_config`

Description: Adds additional preinstall scripts to the OS provisioning profile, starting with the number 2. The variable value requires the following syntax.

n!*pre*!*pre_interpreter*

The previous summary includes the following elements.

n specifies the preinstall script number.

pre specifies the name of the preinstall script.

pre_interpreter specifies the interpreter to use with the preinstall script.

Include the information for all the additional preinstall scripts you want to use in the variable value. The `scriptpreadd_base_config` variable uses the same basic syntax as the `partitionadd_base_config` variable.

`scriptpostadd_base_config`

Description: Adds additional postinstall scripts to the OS provisioning profile, starting with the number 2. The variable value requires the following syntax.

n!*post*!*post_interpreter*!*post_nochroot*

The previous summary includes the following elements.

n specifies the postinstall script number.

post specifies the name of the postinstall script.

post_interpreter specifies the interpreter to use with the postinstall script.

post_nochroot specifies whether to run the postinstall script in a jail in an alternate root directory (yes/no).

Include the information for all the additional postinstall scripts you want to use in the variable value. The `scriptpostadd_base_config` variable uses the same basic syntax as the `partitionadd_base_config` variable.

Component Variables for RedHat Remote Agents

The variables described in the following list provide Remote Agent (RA) configuration for the N1 SPS software. These variables directly equate to the JET `spsra` module.

`install_spsra`

Description: Whether to install the N1 SPS RA software.

Default value: `y`

`product_version_spsra`

Description: Version of RA to install.

Default value: `5.2`

If you are using the N1 SPS 5.1 release to provision with the OS Provisioning Plug-In 3.0, set this variable value to `5.1`.

`basedir_spsra`

Description: Base directory in which RA files are installed.

Default value: `/opt/SUNWn1sps`

`username_spsra`

Description: When using the RA installer, specify user name to own the RA filesystem.

Example: `jdoe`

`groupname_spsra`

Description: When using the RA installer, specify group name to own the RA filesystem.

Example: `staff`

`setup_user_acct_spsra`

Description: Determines whether to set up user and group account for `username_spsra` and `groupname_spsra`.

Note – If the user account is not available to the target after provisioning, set this variable value to y to start an N1 SPS Remote Agent on the target host on reboot after provisioning.

Default value: n

user_passwd_spsra
Description: Encrypted password for username_spsra.

Example: 6GR9gOi1vRVFw shows the encrypted value for the password "n1spspwd."

userid_spsra
Description: User account ID for username_spsra.

Example: 505

groupid_spsra
Description: Group account ID for groupname_spsra.

Example: 505

homedir_spsra
Description: Home directory for RA files.

Default value: /opt/SUNWn1sps

jre_path_spsra
Description: 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
Description: Enable RA root access.

Default value: y

host_spsra
Description: Host name or IP address of the Remote Agent.

Default value: :[targetableComponent:sps_ra_host]

conn_spsra
Description: Determines method to use for connecting to RA. By default, uses the connection method defined for the targetable component.

Default value: :[targetableComponent:sps_ra_conn]

ssl_auth_spsra
Description: SSL authorization to use for network connection.

Default value: :[targetableComponent:sps_ra_ssl_auth] `port_spsra`

Description: Port to use for network connection. By default, uses the port defined for the targetable component.

Note – To use an SSH connection, set the port number to 70000.

Default value: `:[targetableComponent:sps_ra_port]`

`adv_params_spsra`

Description: Advanced parameters to use for network connection. By default, uses the advanced parameters defined for the targetable component.

Default value: `:[targetableComponent:sps_ra_adv_parms]`

General RedHat Linux OS Provisioning Plug-In Variables

The variables in the following list 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 SPS software.

`installPath`

Description: Jumpstart Enterprise Toolkit (JET) virtual host.

Example: `myhost-jet`

`osp_target_host`

Description: Target host name.

Default value: `:[targetableComponent:installPath]`

`osp_profile_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False).

Default value: `:[targetableComponent:osp_control_service]`

`osp_jet_server`

Description: Host name of the JET server.

Example: `myserver`

`osp_profile_architecture`

Description: Target host architecture (x86).

Default value: `:[targetableComponent:architecture]`

`osp_profile_min_disk_size`

Description: Minimum disk size.

Example: `5GB`

`osp_profile_description`
Description: Profile description.

`osp_profile_boot_server_name`
Description: RedHat Linux boot server name.

Default value: `:[osp_jet_server]`

`osp_profile_BS_subnet_address`
Description: Subnet served by boot server.

`osp_profile_BS_subnet_ip_address`
Description: IP address of boot server on this subnet.

Example: `199.199.8.6`

`osp_profile_product_ip_address`
Description: IP address of RedHat Linux image server.

Example: `199.199.8.6`

`osp_profile_product_directory`
Description: Path to image directory on image server.

Example: `/osp_image/RedHat/RHEL3-U4`

`osp_profile_boot_ip_address`
Description: IP address of RedHat Linux boot server.

Example: `199.199.8.6`

APPENDIX D

SuSE Linux Profile Component Variables

The SuSe Linux Profile component includes variables for each JET module. These variables are separated into the following categories:

- "Basic SuSe Linux OS Configuration Variables" on page 243
- "Component Variables for SuSe Linux Remote Agents" on page 264
- "General SuSe Linux OS Provisioning Plug-In Variables" on page 266

Basic SuSe Linux OS Configuration Variables

Component Variables for `base_config` Module

The variables described in the following list provide basic operating system configuration. These variables directly equate to the JET `base_config` module. In the list you see several variables that use the value of the *targetable component*. For purposes of the SuSe Linux Profile variables, the targetable component is the target host.

`ClientArch_base_config`

Description: Kernel architecture, such as x86. By default, this is set to the kernel architecture of the targetable component.

Default value: `:[targetableComponent:kernel_arch]`

`ClientEther_base_config`

Description: Ethernet MAC address. By default, this is set to the Ethernet MAC address of the targetable component.

Default value: `:[targetableComponent:ethernet_mac_address]`

`ClientOSName_base_config`

Description: Name of the OS to be provisioned.

Example: `redhat`

`ClientOS_base_config`
Description: Version of the OS to be provisioned.

Example: `LinuxRedHat3.0_x86`

`products_base_config`
Description: JET modules to provision.

Example: `base_config spsra`

`client_allocation_base_config`
Description: 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.

`productdir_base_config`
Description: Path to the SuSe Linux distribution. To use the information from the `jumpstart.conf` file and the IP address of the JET server, leave this value blank. 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`
Description: Path to the patches you want to provision to the target host. If the patches you want to install are stored on the JET SuSe image server, leave this value blank. If the patches are not located on the JET SuSe image server, provide the absolute path to the location of these patches on an NFS-accessible system.

Default value: null

`suse_media_base_config`
Description: Location of the SuSe Linux product media.

`pxelinux_console_base_config`
Description: Console parameters that are passed to the PXE boot environment.

Example: `ttyS0,9600`

`append_kernel_cmdline_base_config`
Description: Additional command line options to pass to the kernel at PXE boot time.

`nis_domain_base_config`
Description: Name of the NIS domain.

`nis_broadcast_base_config`
Description: Specifies whether to broadcast name service data from the NIS server (yes/no).

Default value: `no`

`nis_broken_server_base_config`
Description: Corresponds to the `nis_broken_server` variable in the `<nis>` section for SuSe Linux installations (yes/no).

`nis_local_only_base_config`
Description: Corresponds to the `nis_local_only` variable in the `<nis>` section for SuSe Linux installations (yes/no).

`nis_options_base_config`
Description: Corresponds to the `nis_options` variable in the `<nis>` section for SuSe Linux installations (yes/no).

`nis_servers_base_config`
Description:. Space-separated list of NIS servers.

`nis_start_autofs_base_config`
Description: Corresponds to the `start_autofs` variable in the `<nis>` section for SuSe Linux installations (yes/no).

`nis_start_nis_config`
Description: Specifies whether to start the NIS service (yes/no).

Default value: `no`

`nis_other_1_domain_base_config`
Description: Specifies an additional NIS domain.

`nis_other_1_broadcast_base_config`
Description: Specifies whether to broadcast name service data from the additional NIS server (yes/no).

`nis_other_1_servers_base_config`
Description: Space-separated list that specifies additional NIS servers.

`ldap_v2_base_config`
Description: Specifies whether to use LDAP v2 authentication (true/false).

Default value: `false`

`ldap_start_base_config`
Description: Specifies whether to start the LDAP service (true/false).

Default value: `false`

`ldap_server_base_config`
Description: Name of the LDAP server.

`ldap_basedn_base_config`
Description: Base domain name for the LDAP server.

`ldap_tls_base_config`
Description: Specifies whether to use the Transport Layer Security protocol (TLS) for LDAP (true/false).

Default value: `false`

`runlevel_base_config`
Description: Specifies the run level for the target host.

Default value: `3`

`security_console_shutdown_base_config`
Description: Specifies the shutdown behavior for the console.

Default value: `reboot`

`security_cwd_in_root_path_base_config`
Description: Specifies whether to allow `cwd` (change working directory) in the root path (yes/no).

Default value: `no`

`security_cwd_in_user_path_base_config`
Description: Specifies whether to allow `cwd` (change working directory) in the user's path (yes/no).

Default value: `no`

`security_displaymanager_remote_access_base_config`
Description: Enable remote access to the Displaymanager.

Default value: `no`

`security_encryption_base_config`
Description: Enable encryption in communications from the target host.

Default value: `md5`

`security_fail_delay_base_config`
Description: Specifies the time in seconds after each login attempt.

Default value: `1`

`security_faillog_enab_base_config`
Description: Enable logging of failures in `/var/log/faillog`

Default value: `yes`

`security_gid_max_base_config`
Description: Maximum number group IDs

Default value: `60000`

`security_gid_min_base_config`
Description: Minimum number of group IDs

Default value: `101`

`security_kdm_shutdown_base_config`
Description: KDM shutdown

Default value: `all`

`security_lastlog_enab_base_config`

Description: lastlog enable

Default value: `yes`

`security_obscure_checks_enab_base_config`

Description: Enable obscure checks

Default value: `no`

`security_pass_max_days_base_config`

Description: Maximum number of days for lifespan of a password

Default value: `99999`

`security_pass_max_len_base_config`

Description: Maximum length in characters of user passwords

Default value: `8`

`security_pass_min_days_base_config`

Description: Minimum number of days for lifespan of user passwords

Default value: `0`

`security_pass_min_len_base_config`

Description: Minimum length in characters of user passwords

Default value: `5`

`security_pass_warn_age_base_config`

Description: Specifies the number of days left before a user password expires to warn the user

Default value: `7`

`security_passwd_use_crackilib_base_config`

Description: Specifies whether to use the Cracklib utility to encrypt the password

Default value: `no`

`security_permission_security_base_config`

Description: Specifies the level of permission security

Default value: `easy`

`security_run_updatedb_as_base_config`

Description: Specifies the user to run the `updatedb` job

Default value: `root`

`security_uid_max_base_config`

Description: Maximum number of user IDs

Default value: `60000`

`security_uid_min_base_config`
Description: Minimum number of user IDs

Default value: `500`

`users_base_config`
Description: Specifies the root user

Default value: `root`

`user_root_encrypted_base_config`
Description: Specifies whether to pass an encrypted root password during the installation

Default value: `true`

`user_root_password_base_config`
Description: Encrypted root password

Default value: `JgkyZIu7eRxPU`

`user_root_fullname_base_config`
Description: Full name of the root user

`user_root_gid_base_config`
Description: Group ID for the root user

`user_root_grouplist_base_config`
Description: Space-separated list of additional groups of which the root user is a member

`user_root_home_base_config`
Description: Home directory for the root user

`user_root_shell_base_config`
Description: Default shell for the root user

`user_root_uid_base_config`
Description: User ID for the root user

`x11_color_depth_base_config`
Description: Color depth of the display device

`x11_configure_x11_base_config`
Description: Specifies whether to configure the X11 window system (true, false)

Default value: `false`

`x11_start_x11_base_config`
Description: Specifies whether to start the X11 windowing system (true, false)

Default value: `false`

`x11_display_manager_base_config`
Description: Specifies the display manager

`x11_enable_3d_base_config`
Description: Enable 3D in X11 windowing system

`x11_window_manager_base_config`
Description: Specifies the window manager software

`x11_resolution_base_config`
Description: Specifies the resolution to use for the display device

`x11_monitor_display_frequency_base_config`
Description: Monitor frequency for the display device

`x11_monitor_display_bandwidth_base_config`
Description: Monitor bandwidth

`x11_monitor_display_height_base_config`
Description: Monitor height

`x11_monitor_display_max_hsync_base_config`
Description: Maximum horizontal sync for the monitor

`x11_monitor_display_max_vsync_base_config`
Description: Maximum vertical sync for the monitor

`x11_monitor_display_min_hsync_base_config`
Description: Minimum horizontal sync for the monitor

`x11_monitor_display_min_vsync_base_config`
Description: Minimum vertical sync for the monitor

`x11_monitor_display_width_base_config`
Description: Monitor display width

`x11_monitor_device_base_config`
Description: Monitor device

`x11_monitor_vendor_base_config`
Description: Monitor vendor

`bootloader_location_base_config`
Description: Location of the bootloader

Default value: `mbr`

`bootloader_repl_mbr_base_config`
Description: Specifies whether to clear the Master Boot Record (mbr)

Default value: `true`

`bootloader_initrd_modules_base_config`
Description: Specifies the initrd bootloader modules

`bootloader_kernel_parameters_base_config`
Description: Specifies kernel parameters for the bootloader

`bootloader_loader_type_base_config`

Description: Specifies the type of bootloader

Default value: `grub`

`bootloader_loader_device_base_config`

Description: Specifies the bootloader device

`general_keyboard_base_config`

Description: Specifies the keyboard map

Default value: `english-us`

`general_mouse_id_base_config`

Description: Specifies the mouse ID

Default value: `none`

`general_mouse_device_base_config`

Description: Specifies the mouse device

`general_clock_hwclock_base_config`

Description: Specifies the hardware clock (UTC, localtime)

Default value: `localtime`

`general_timezone_base_config`

Description: Specifies the time zone

Default value: `US/Pacific`

`general_language_base_config`

Description: Specifies the language to install on the system

Default value: `en_US`

`general_mode_confirm_base_config`

Description: Confirm mode (true, false)

Default value: `false`

`general_mode_forceboot_base_config`

Description: Force boot (true, false)

Default value: `true`

`general_mode_interactive_boot_base_config`

Description: Interactive boot (true, false)

Default value: `false`

`general_mode_reboot_base_config`

Description: Reboot (true, false)

Default value: `true`

`dhclient_additional_options_base_config`
Description: Specifies additional DHCP client options

`dhclient_client_id_base_config`
Description: DHCP client ID

`dhclient_hostname_option_base_config`
Description: DHCP host name options

`drive_list_base_config`
Description: Pipe-separated list of hard drives on the target host. The variable value requires the following syntax.

name`!`*init*`!`*use*`!`*device*

For example:

```
sda!true!all!/dev/sda|sdb!true!linux!/dev/sdb
```

The previous example identifies two hard drives with the following set of variable and value pairs

```
base_config_drive_list="sda sdb"

base_config_drive_sda_init=true
base_config_drive_sda_use=all
base_config_drive_sda_device=/dev/sda

base_config_drive_sdb_init=true
base_config_drive_sdb_use=linux
base_config_drive_sdb_device=/dev/sdb
```

Default value: `sda!true!all!/dev/sda`

`partition_1_drive_base_config`
Description: Drive name for the partition. The value of this variable must be one of the drives you declared in the `drive_list_base_config`.

The `partition` variables must start at 1 and mus increase sequentially without skipping any numbers.

Default value: `sda`

`partition_1_id_base_config`
Description: Partition ID for the partition

Example:

`partition_1_nr_base_config`
Description: Partition number for the partition

Example:

`partition_1_mount_base_config`

Description: Mount point for the partition

Default value: `/boot`

`partition_1_filesystem_base_config`

Description: File system type for the partition

Default value: `ext3`

`partition_1_format_base_config`

Description: Specifies whether to format the partition (true, false)

Default value: `true`

`partition_1_size_base_config`

Description: Partition size of the partition (MB number, auto, or max)

Default value: `100M`

`partition_1_raid_device_base_config`

Description: RAID device for the partition

Example:

`partition_1_primary_base_config`

Description: Force the partition as a primary partition (true, false)

Default value: `true`

`partition_2_drive_base_config`

Description: Drive name for the partition

Default value: `sda`

`partition_2_id_base_config`

Description: Partition ID for the partition

Example:

`partition_2_nr_base_config`

Description: Partition number for the partition

Example:

`partition_2_mount_base_config`

Description: Mount point for the partition

Default value: `/`

`partition_2_filesystem_base_config`

Description: File system type for the partition

Default value: `ext3`

`partition_2_format_base_config`

Description: Specifies whether to format the partition (true, false)

Default value: `true`

`partition_2_size_base_config`

Description: Partition size of the partition (MB number, auto, or max)

Default value: `10GB`

`partition_2_raid_device_base_config`

Description: RAID device for the partition

Example:

`partition_2_primary_base_config`

Description: Force the partition as a primary partition (true, false)

Default value: `true`

`partition_3_drive_base_config`

Description: Drive name for the partition

Default value: `sda`

`partition_3_id_base_config`

Description: Partition ID for the partition

Example:

`partition_3_nr_base_config`

Description: Partition number for the partition

Example:

`partition_3_mount_base_config`

Description: Mount point for the partition

Default value: `swap`

`partition_3_filesystem_base_config`

Description: File system type for the partition

Default value: `swap`

`partition_3_format_base_config`

Description: Specifies whether to format the partition (true, false)

Default value: `true`

`partition_3_size_base_config`

Description: Partition size of the partition (MB number, auto, or max)

Default value: `4096M`

`partition_3_raid_device_base_config`
Description: RAID device for the partition

Example:

`partition_3_primary_base_config`
Description: Force the partition as a primary partition (true, false)

Default value: `true`

`partition_4_drive_base_config`
Description: Drive name for the partition

`partition_4_id_base_config`
Description: Partition ID for the partition

`partition_4_nr_base_config`
Description: Partition number for the partition

`partition_4_mount_base_config`
Description: Mount point for the partition

`partition_4_filesystem_base_config`
Description: File system type for the partition

Default value: `ext3`

`partition_4_format_base_config`
Description: Specifies whether to format the partition (true, false)

Default value: `true`

`partition_4_size_base_config`
Description: Partition size of the partition (MB number, auto, or max)

`partition_4_raid_device_base_config`
Description: RAID device for the partition

`partition_4_primary_base_config`
Description: Force the partition as a primary partition (true, false)

Default value: `true`

`partition_5_drive_base_config`
Description: Drive name for the partition

Default value: `sda`

`partition_5_id_base_config`
Description: Partition ID for the partition

Example:

`partition_5_nr_base_config`
Description: Partition number for the partition

`partition_5_mount_base_config`
Description: Mount point for the partition

`partition_5_filesystem_base_config`
Description: File system type for the partition

`partition_5_format_base_config`
Description: Specifies whether to format the partition (true, false)

`partition_5_size_base_config`
Description: Partition size of the partition (MB number, auto, or max)

`partition_5_raid_device_base_config`
Description: RAID device for the partition

`partition_51_primary_base_config`
Description: Force the partition as a primary partition (true, false)

`partition_6_drive_base_config`
Description: Drive name for the partition

`partition_6_id_base_config`
Description: Partition ID for the partition

`partition_6_nr_base_config`
Description: Partition number for the partition

`partition_6_mount_base_config`
Description: Mount point for the partition

`partition_6_filesystem_base_config`
Description: File system type for the partition

`partition_6_format_base_config`
Description: Specifies whether to format the partition (true, false)

`partition_6_size_base_config`
Description: Partition size of the partition (MB number, auto, or max)

`partition_6_raid_device_base_config`
Description: RAID device for the partition

`partition_6_primary_base_config`
Description: Force the partition as a primary partition (true, false)

`partition_7_drive_base_config`
Description: Drive name for the partition

Default value: `sda`

`partition_7_id_base_config`
Description: Partition ID for the partition

`partition_7_nr_base_config`
Description: Partition number for the partition

`partition_7_mount_base_config`
Description: Mount point for the partition

`partition_7_filesystem_base_config`
Description: File system type for the partition

`partition_7_format_base_config`
Description: Specifies whether to format the partition (true, false)

`partition_7_size_base_config`
Description: Partition size of the partition (MB number, auto, or max)

`partition_7_raid_device_base_config`
Description: RAID device for the partition

`partition_7_primary_base_config`
Description: Force the partition as a primary partition (true, false)

`partition_8_drive_base_config`
Description: Drive name for the partition

`partition_8_id_base_config`
Description: Partition ID for the partition

`partition_8_nr_base_config`
Description: Partition number for the partition

`partition_8_mount_base_config`
Description: Mount point for the partition

`partition_8_filesystem_base_config`
Description: File system type for the partition

`partition_8_format_base_config`
Description: Specifies whether to format the partition (true, false)

`partition_8_size_base_config`
Description: Partition size of the partition (MB number, auto, or max)

`partition_8_raid_device_base_config`
Description: RAID device for the partition

`partition_8_primary_base_config`
Description: Force the partition as a primary partition (true, false)

`raid_1_device_base_config`
Description: Specifies the device for a RAID volume.

Example: `/dev/sda1`

`raid_1_chunk_size_base_config`
Description: Specifies size of a RAID strip in bytes.

`raid_1_filesystem_base_config`
Description: File system to use on the RAID volume

`raid_1_format_base_config`

Description: Specifies the format of the RAID volume.

`raid_1_parity_algorithm_base_config`

Description: The parity algorithm to use with RAID5. The value must be one of left-asymmetric, right-asymmetric, left-symmetric, or right-symmetric.

`raid_1_partition_id_base_config`

Description:

Default value:

`raid_1_persistent_superblock_base_config`

Description: Specifies whether to use a persistent superblock (true/false). A persistent superblock is a small disk area allocated at the end of each RAID device, enabling the kernel to safely detect RAID devices even if disks have been moved between SCSI controllers.

`raid_1_mount_base_config`

Description: Specifies the mount point for the RAID device.

`raid_1_format_base_config`

Description: Specifies the type of RAID volume (0, 1, or 5) to create

`raid_2_device_base_config`

Description: Specifies the device for a RAID volume.

Example: `/dev/sda2`

`raid_2_chunk_size_base_config`

Description: Specifies size of a RAID strip in bytes.

`raid_2_filesystem_base_config`

Description: File system to use on the RAID volume

`raid_2_format_base_config`

Description: Specifies the format of the RAID volume.

`raid_2_parity_algorithm_base_config`

Description: The parity algorithm to use with RAID5. The value must be one of left-asymmetric, right-asymmetric, left-symmetric, or right-symmetric.

`raid_2_partition_id_base_config`

Description:

Default value:

`raid_2_persistent_superblock_base_config`

Description: Specifies whether to use a persistent superblock (true/false). A persistent superblock is a small disk area allocated at the end of each RAID device, enabling the kernel to safely detect RAID devices even if disks have been moved between SCSI controllers.

`raid_2_mount_base_config`

Description: Specifies the mount point for the RAID device.

`raid_2_format_base_config`
Description: Specifies the type of RAID volume (0, 1, or 5) to create

`software_base_base_config`
Description: Specifies the software base to install (Minimal, default, Minimal+X11, Full-Installation.sel)

Default value: `Minimal`

`software_addons_base_config`
Description: Specifies addon software to install

`software_kernel_base_config`
Description: Kernel to install

`software_packages_base_config`
Description: Space-separated list of additional packages to install

`software_remove_base_config`
Description: Space-separated list of packages to remove

`bootproto_base_config`
Description: Boot protocol for the target host.

Default value: `dhcp`

`network_interface_base_config`
Description: Network interface to use during provisioning.

Default value: `:[targetableComponent:linux_network_interface]`

`ip_base_config`
Description: IP address of the target host.

Default value: `:[targetableComponent:ethernet_ip_address]`

`netmask_base_config`
Description: Netmask to use during provisioning.

Default value: `:[targetableComponent:ethernet_netmask]`

`gateway_base_config`
Description: Gateway to use during provisioning.

Default value: `:[targetableComponent:default_router]`

`nameservers_base_config`
Description: Space-separated list of name servers to use during provisioning.

`startmode_base_config`
Description: Start mode

Example:

`broadcast_base_config`

Description: Broadcast

Example:

`networkifs_base_config`

Description: Specifies additional network interfaces to configure. This variable value is derived from the target host variable `networkifs_redhat_base_config`

Example: `eth1!--ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

`dns_dhcp_hostname_base_config`

Description: Specifies whether to acquire a host name from DHCP (true, false)

Default value: `false`

`dns_dhcp_resolv_base_config`

Description: Specifies whether to modify the DNS resolve file

Default value: `false`

`hostname_base_config`

Description: Host name of the target host.

Default value: `:[targetableComponent:installPath]`

`domain_base_config`

Description: DNS domain name

Example: `west.example.com`

`ip_forward_base_config`

Description: Specifies whether to enable IP forwarding (true, false)

Example: `false`

`fw_allow_traceroute_base_config`

Description: Allow traceroute

`fw_autoprotect_services_base_config`

Description: Autoprotect services

`fw_ipsec_trust_base_config`

Description: Trust IPSEC

`fw_log_accept_all_base_config`

Description: Accept all

`fw_log_accept_crit_base_config`

Description: Accept critical

`fw_log_drop_all_base_config`

Description: Drop all

`fw_log_drop_crit_base_config`
Description: Drop critical

`fw_masquerade_base_config`
Description: Masquerade

`fw_protect_from_internal_base_config`
Description: Protect from internal network

`fw_route_base_config`
Description: Firewall route

`fw_start_firewall_base_config`
Description: Specifies whether to start the firewall

`fw_dev_dmz_base_config`
Description: Specifies the interface for the DMZ

`fw_dev_ext_base_config`
Description: Specifies the external interface

`fw_dev_int_base_config`
Description: Specifies the internal interface

`fw_masq_nets_base_config`
Description: Specifies the masquerade networks

`fw_services_dmz_ip_base_config`
Description: IP address of DMZ

`fw_services_dmz_tcp_base_config`
Description: List of DMZ TCP services. Specify a name list, a port number, or minimum and maximum range.

`fw_services_dmz_udp_base_config`
Description: List of DMZ UDP services. Specify a name list, a port number, or minimum and maximum range.

`fw_services_ext_ip_base_config`
Description: External IP address

`fw_services_ext_tcp_base_config`
Description: List of external TCP services. Specify a name list, a port number, or minimum and maximum range.

`fw_services_ext_udp_base_config`
Description: List of external UDP services. Specify a name list, a port number, or minimum and maximum range.

`fw_services_int_ip_base_config`
Description: Internal IP address

`fw_services_int_tcp_base_config`

Description: List of internal TCP services. Specify a name list, a port number, or minimum and maximum range.

`fw_services_int_udp_base_config`

Description: List of internal UDP services. Specify a name list, a port number, or minimum and maximum range.

`script_pre_1_base_config`

Description: Specifies the location of a preinstall script on the JET server.

`script_pre_1_interpreter_base_config`

Description: Specifies the interpreter for the preinstall script.

`script_post_1_base_config`

Description: Specifies the location of a postinstall script on the JET server.

`script_post_1_interpreter_base_config`

Description: Specifies the interpreter for the postinstall script.

`script_chroot_1_base_config`

Description: Specifies whether to run the postinstall script in a jail in an alternate root directory.

`script_chroot_1_interpreter_base_config`

Description: Interpreter to use for the postinstall script.

`script_chroot_1_chrooted_base_config`

Description: Specifies whether to run the script in a different root directory.

`partitionadd_base_config`

Description: Adds partitions to the target host, starting with partition 9. The variable value requires the following syntax.

n!*drive*!*id*!*nr*!*mount*!*filesystem*!*format*!*size*!*raid_device*!*primary*

The previous summary includes the following elements.

n specifies the partition number.

drive specifies the device for the partition.

id specifies the ID for the partition.

nr specifies the partition number.

mount specifies the mount point for the partition.

filesystem specifies the file system type for the partition.

format specifies whether to format the partition.

size specifies the size in MBs of the partition.

maxsize specifies the maximum size for the partition.

grow specifies whether to enable the partition to grow.

primary specifies whether to force this partition as a primary partition.

Include the information for all the additional partitions you want to create in the variable value. For example, if you want to create a ninth and tenth partition, set this variable to a value similar to this example.

```
9!sda!!!/export9!ext3!true!10GB!!!
```

This value creates the following variable set.

```
base_config_partition_9_drive=sda
base_config_partition_9_id=
base_config_partition_9_nr=
base_config_partition_9_mount=/export9
base_config_partition_9_filesystem=ext3
base_config_partition_9_format=true
base_config_partition_9_size=10GB
base_config_partition_9_raid_device=
base_config_partition_9_primary=
```

`raidadd_base_config`

Description: Adds RAID devices to the target host, starting with device 3. The variable value requires the following syntax.

n!*device*!*chunk_size*!*filesystem*!*format*!*parity_algorithm*!*partition_id*!*persistent_superblock*!*mount*!*raid_type*

The previous summary includes the following elements.

n specifies the RAID volume number.

device specifies the device for the RAID volume.

chunk_size specifies the RAID level for the volume.

filesystem specifies the file system to use on the RAID volume..

format specifies whether to format the disk to be used as the RAID volume.

parity_algorithm specifies the algorithm to use to check the integrity of the RAID volumes.

partition_id specifies the partition ID for the RAID volume.

persistent_superblock specifies whether to create a persistent superblock on the RAID device.

mount specifies the mount point for the RAID volume.

raid_type specifies the type of RAID volume.

Include the information for all the additional partitions you want to create in the variable value. The `raidadd_base_config` variable uses the same basic syntax as the `partitionadd_base_config` variable.

`scriptpreadd_base_config`

Description: Adds additional preinstall scripts to the OS provisioning profile, starting with the number 2. The variable value requires the following syntax.

n`!`*pre*`!`*pre_interpreter*

The previous summary includes the following elements.

n specifies the script index number. Start with the index number 2.

pre specifies the name of the preinstall script.

pre_interpreter specifies the interpreter to use with the preinstall script.

Include the information for all the additional preinstall scripts you want to use in the variable value. The `scriptpreadd_base_config` variable uses the same basic syntax as the `partitionadd_base_config` variable.

`scriptpostadd_base_config`

Description: Adds additional postinstall scripts to the OS provisioning profile, starting with the number 2. The variable value requires the following syntax.

n`!`*post*`!`*post_interpreter*

The previous summary includes the following elements.

n specifies the script index number. Start with the index number 2.

post specifies the name of the postinstall script.

post_interpreter specifies the interpreter to use with the postinstall script.

Include the information for all the additional postinstall scripts you want to use in the variable value. The `scriptpostadd_base_config` variable uses the same basic syntax as the `partitionadd_base_config` variable.

`scriptchrootadd_base_config`

Description: Adds additional `chroot` scripts to the OS provisioning profile, starting with the number 2. The variable value requires the following syntax.

n`!`*chroot*`!`*interpreter*`!`*chrooted*

The previous summary includes the following elements.

n specifies the script index number. Start with the index number 2.

chroot specifies the name of the `chroot`script.

interpreter specifies the interpreter to use with the `chroot` script.

chrooted specifies whether to run the script in an different root directory.

Include the information for all the additional postinstall scripts you want to use in the variable value. The `scriptpostadd_base_config` variable uses the same basic syntax as the `partitionadd_base_config` variable.

Configuring Additional Network Interfaces

The `networkifs_base_config` variable provides a way to configure additional network interfaces, in addition to the first network interface configured through the `network_interface_base`variable.

For each additional interface device, you provide a group of four variables that define the netname, netmask, hostname, and IP address for the interface.

The example below shows how to define the `networkifs_base_config` variable for two interfaces:

- `le1`
  - netname — `netB`
  - netmask — `255.255.255.0`
  - hostname — `myhost-netB`
  - interface IP — `192.168.1.0`
- `hme0`
  - netname — `netC`
  - netmask — `255.255.255.0`
  - hostname — `myhost-netC`
  - interface IP — `192.162.2.0`

You would use the following syntax to define the single `networkifs_base_config` variable for the two additional interfaces:

```
le1!netB 255.255.255.0 myhost-netB 192.168.1.0|hme0!netC 255.255.255.0 myhost-netC 192.168.2.0
```

Component Variables for SuSe Linux Remote Agents

The variables described in the following list provide Remote Agent (RA) configuration for the N1 SPS software. These variables directly equate to the JET `spsra` module.

`install_spsra`

Description: Whether to install the N1 SPS RA software.

Default value: `y`

`product_version_spsra`

Description: Version of RA to install.

Default value: `5.2`

If you are using the N1 SPS 5.1 release to provision with the OS Provisioning Plug-In 3.0, set this variable value to `5.1`.

`basedir_spsra`

Description: Base directory in which RA files are installed.

Default value: `/opt/SUNWn1sps`

`username_spsra`

Description: When using the RA installer, specify user name to own the RA filesystem.

Example: `jdoe`

`groupname_spsra`

Description: When using the RA installer, specify group name to own the RA filesystem.

Example: `staff`

`setup_user_acct_spsra`

Description: Determines whether to set up user and group account for `username_spsra` and `groupname_spsra`.

Note – If the user account is not available to the target after provisioning, set this variable value to `y` to start an N1 SPS Remote Agent on the target host on reboot after provisioning.

Default value: `n`

`user_passwd_spsra`

Description: Encrypted password for `username_spsra`.

Example: `6GR9gOi1vRVFw` shows the encrypted value for the password “n1spspwd.”

`userid_spsra`

Description: User account ID for `username_spsra`.

Example: `505`

`groupid_spsra`

Description: Group account ID for `groupname_spsra`.

Example: `505`

`homedir_spsra`

Description: Home directory for RA files.

Default value: `/opt/SUNWn1sps`

`jre_path_spsra`

Description: 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`

Description: Enable RA root access.

Default value: `y`

`setuid_spsra`

Description: Host name or IP address of the Remote Agent.

Example: `solaris10-target`

`conn_spsra`

Description: Determines method to use for connecting to RA. By default, uses the connection method defined for the targetable component.

Default value: `:[targetableComponent:sps_ra_conn]`

`ssl_auth_spsra`

Description: SSL authorization to use for network connection.

Default value: `1`

`port_spsra`

Description: Port to use for network connection. By default, uses the port defined for the targetable component.

Note – To use an SSH connection, set the port number to 70000.

Default value: `:[targetableComponent:sps_ra_port]`

`adv_params_spsra`

Description: Advanced parameters to use for network connection. By default, uses the advanced parameters defined for the targetable component.

Default value: `:[targetableComponent:sps_ra_adv_parms]`

General SuSe Linux OS Provisioning Plug-In Variables

The variables in the following list 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 SPS software.

`installPath`

Description: Jumpstart Enterprise Toolkit (JET) virtual host.

Example: `myhost-jet`

`osp_target_host`

Description: Target host name.

Default value: `:[targetableComponent:installPath]` `osp_profile_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False).

Default value: `:[targetableComponent:osp_control_service]`

`osp_jet_server`

Description: Host name of the JET server.

Example: `myserver`

`osp_profile_architecture`

Description: Target host architecture (x86).

Default value: `:[targetableComponent:architecture]`

`osp_profile_min_disk_size`

Description: Minimum disk size.

Example: `5GB`

`osp_profile_description`

Description: Profile description.

`osp_profile_boot_server_name`

Description: SuSe Linux boot server name.

Default value: `:[osp_jet_server]`

`osp_profile_BS_subnet_address`

Description: Subnet served by boot server.

`osp_profile_BS_subnet_ip_address`

Description: IP address of boot server on this subnet.

Example: `199.199.8.6`

`osp_profile_product_ip_address`

Description: IP address of SuSe Linux image server.

Example: `199.199.8.6`

`osp_profile_product_directory`

Description: Path to image directory on image server.

Example: `/osp_image/SUSE/SLES-9-x86-64/installroot`

`osp_profile_boot_ip_address`

Description: IP address of SuSe Linux boot server.

Example: `199.199.8.6`

APPENDIX E

Microsoft Windows Profile Component Variables

The Microsoft Windows Profile component includes variables for Windows OS configuration and Windows Remote Agents.

Windows OS Component Variables

The variables described in the following list provide all the installation and configuration parameters for the Windows OS. In the list you see several variables that use the value of the *targetable component*. For purposes of the Windows Profile variables, the targetable component is the target host.

`installPath`

Description: Name of the Windows RIS image server.

Example: `risserver-windows`

`osp_target_host`

Description: Target host name.

Default value: `:[targetableComponent:installPath]`

`osp_profile_control_service`

Description: Whether to call the OSP Control Service, which automates power off, reboot, and power on functions (True, False).

Note – To provision the Windows OS on target hosts, the value of this variable must be `True`.

`description`

Description: Description of the component.

Example: `Windows 2003 Profile`

`version`

Description: Windows version to be installed with this profile (2000, 2003).

Default: `2003`

`architecture`

Description: Processor architecture of system to be installed with this profile (x86, x64).

Default: `x86`

`min_disk_size`

Description: Minimum disk size for this profile to be deployed.

Default: `10GB`

`boot_server_name`

Description: Name of the Windows boot server.

Example: `biwindows1`

`boot_server_subnet_address`

Description: Subnet served by boot server.

Example: `192.168.5.0`

`boot_server_subnet_ip_address`

Description: IP address of boot server on this subnet.

Example: `192.168.5.0`

`boot_srv_active_directory_domain`

Description: Windows active directory domain for this subnet.

Example: `winprov.n1lab.sun.com`

`windows_product_directory`

Description: Windows product directory name on boot server.

Example: `D:\RemoteInstall\Setup\English\Images\Win2003`

`sif_file`

Description: Location of software information file (SIF).

Example:`D:\RemoteInstall\Setup\English\Images\Win2003\i386\templates\ristndrd.sif`

`active_directory_userid`

Description: Active directory user ID.

Example: `n1ospadmin`

`boot_server_access_protocol`

Description: Protocol to access the boot server, either RSH or SSH.

Note – The RSH protocol is not supported with custom Windows profiles. You must use SSH as the communication protocol with custom Windows profiles.

Example: `ssh`

`boot_server_access_userid`

Description: User ID to access the boot server.

Example: `n1opspssh`

`boot_server_access_password`

Description: Password to access the boot server. Type the encrypted password as the value for this variable. For more information about how to encrypt passwords, see "Password Encryption" on page 159.

Example: `Clkjgfw==`

`temporary_sif_directory`

Description: Location of the temporary software information files (sif) in the RemInst share..

Example: `D:\RemoteInstall\N1ospsif`

`ciw_directory`

Description: Location of the CIW and OS Chooser files.

Example: `D:\RemoteInstall\OSChooser\English`

`isp_scripts_directory`

Description: Location of the OS Provisioning Plug-In scripts.

Example: `C:\N1ospRisUtil\N1ospScripts`

`computer_name`

Description: Host name of the target host.

Example: `mysystem`

`user_full_name`

Description: Full name of the user for the target host.

Example: `Default user`

`organization_name`

Description: Name of the organization to which the target host belongs.

Example: `Default organization`

`product_key`

Description: Product key for the Windows OS that you want to install. The product key is available with the software distribution of the OS, and follows the format xxxxx-xxxxx-xxxxx-xxxxx-xxxxx.

`repartition`

Description: Specifies whether to reformat all partitions on the first drive of the target host with an NTFS file system.

Default: `Yes`

`administrator_password`

Description: Clear text password for the administrator user on the target host.

Default: `default123`

`timezone`

Description: Time zone for the target host.

Default value: `%TIMEZONE%`

`profile_language`

Description: Language to install on the target host.For more information, see “Supported Unattended RIS Variables” at theMicrosoft web site.

Example: `0409`

`profile_language_group`

Description: Language group to install on the system.

Example: `1`

`ClientAccessLicense_automode_setting`

Description: Specifies whether the Windows OS license applies on a per-seat basis or a per-server basis.

Example: `PerSeat`

`ClientAccessLicense_autousers_setting`

Description: Specifies the number of client licenses purchased for the server. You must set a value for this variable if `ClientAccessLicense_automode_setting` is set to `PerServer`.

Example: `100`

`dns_server_ip_address`

Description: IP address of the DNS server on this subnet.

Example: `192.168.5.167`

`domain_name`

Description: Name of the domain for the target host

Note – If you specify this value, do not specify a value for the `workgroup_name` variable.

Example: `n1lab.west.example.com`

`domain_administrator_name`

Description: Name of the domain administrator user account.

Example: `domadmin`

`domain_administrator_password`

Description: Password for the domain administrator user account.

Example: `domadmin567`

`workgroup_name`

Description: Name of the workgroup in which the target host participates.

Note – If you specify this value, do not specify a value for the `domain_name`, `domain_administrator_name`, or `domain_administrator_password` variables.

Example: `n1labs`

`install_terminal_server`

Description: Specifies whether to install a terminal server on the target host. Set this value to `On` to install the terminal server, or `Off` to not install the terminal server.

Example: `On`

`terminal_server_allow_connections`

Description: Specifies whether to allow remote connections to this target host through the terminal server.

Example: `1`

`terminal_server_licensing_mode`

Description: Specifies how the Terminal Services manage the Client Access Licenses. Valid values are PerDevice and PerUser.

Example: `PerDevice`

`terminal_server_permissions_setting`

Description: Specifies the permissions for terminal server users. If this variable is set to 0, terminal server users have the same permissions as users of Windows 2000, Windows XP, and Windows 2003. If this variable is set to 1, terminal server users have the same permissions as users of terminal server 4.0.

Example: `0`

`pnp_drivers_path`

Description: Location of Plug and Play drivers under the distribution share. To include multiple paths, separate each path with a semicolon (;).

Example: `Drivers\Nic;Drivers\Lsi`

`mass_storage_drivers_entry1`

Description: Specifies OEM SCSI drivers to use with the Setup Loader.

Example: `LSI Logic PCI SCSI/FC MPI Driver (Server 2003 32-bit) = OEM`

`mass_storage_drivers_entry`*n*

Description: Specifies additional OEM or retail SCSI drivers to use with the Setup Loader. *n* specifies a number between 2 and 5. For each additional driver, increase the value of *n* for each additional driver, up to 5. For example, `mass_storage_drivers_entry2`.

`mass_storage_drivers_entry6`

Description: Specifies retail SCSI drivers to use with the Setup Loader.

Note – The retail driver must be listed at the end of the `mass_storage_drivers_entry_` settings.

Example: `IDE CD-ROM (ATAPI 1.2)/PCI IDE Controller = RETAIL`

`oem_boot_file`*n*

Description: Specifies mass storage SCSI driver files, as listed under `$OEM$\Textmode` folder. For each driver file, increase the value of *n* for each additional driver file, starting with the number 1, up to 10. For example, `oem_boot_file1`.

Example: `lsi_sas.sys`

`gui_run_once_cmd`*n*

Description: Specifies individual commands to run after GUI mode of Setup completes. For each command, increase the value of *n* for each additional command file, starting with the number 1, up to 10. For example, `gui_run_once_cmd1`

Example: `dcpromo /answer:%SYSTEMDRIVE%\N1OSP\dc_answerfile`

up to cmd5

`help_page_url`

Description: Specifies the URL for HTML-based help.

Example: `http://n1labs.west.example.com/help.html`

`home_page_url`

Description: Specifies the URL for the browser home page.

Example: `http://www.example.com`

APPENDIX F

Creating Additional 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 unusual dependencies.

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

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.**

   a. **To create a variable set for the Solaris 10 Flash image, type a command similar to the following example:**

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/n1osp/untyped/SolarisImage \
-name "solaris10sparc" -u admin -p admin -vars "version=10;release=ga;architecture=sparc; \
image_path=/export/install/s10ga-sparc;image_subnet_addr=10.42.42.2; \
image_subnet_mask=255.255.255.0;media_src="
```

b. **To provide Flash information, create a file `/tmp/flash-profile` that contains the following entries:**

```
flash-with-ra
Solaris10 Flash Archive With SPS RA
base_config flash 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(s) field in the plan variables section of the browser interface.

c. **To create the component, type a command similar to the following example:**

```
# cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/untyped/SolarisImage-create-profile -tar H:NM:biss1-jet \
-comp - -vs solaris10sparc -pto 30 -nto 10 -f /tmp/flash-profile
```

Tip – You can also create the Profile component through the N1 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.

APPENDIX G

Advanced Configuration Properties

This appendix describes how to set up advanced configuration properties for your OS provisioning operations. This appendix explains the following topics.

- "Disabling the Plug-In Provided DHCP" on page 283
- "Running Deployment Jobs in the Foreground" on page 284
- "Validating New Target Hosts" on page 284

Disabling the Plug-In Provided DHCP

By default, the OS Provisioning Plug-In creates and starts an ISC DHCP server to enable target hosts to boot and install over the network. If, however, you want to use your own DHCP server to respond to DHCP requests on your network, you can disable the provisioning software DHCP server.

Note – If you disable the ISC DHCP server that is provided with the provisioning software, you must ensure that your DHCP server is correctly configured to respond to DHCP requests on your network, and responds to network boot requests.

To disable the ISC DHCP server that is provided with the provisioning software, set the `com.sun.n1.isp.dhcp.DHCPManager.serverBinding` property in the `n1osp-dhcp.properties` file as follows.

```
com.sun.n1.isp.dhcp.DHCPManager.serverBinding=com.sun.n1.isp.dhcp.NoDHCP
```

The `n1osp-dhcp.properties` file os located in the `/opt/SUNWn1osp/etc/n1osp-dhcp.properties`on the OS Provisioning server.

Running Deployment Jobs in the Foreground

By default, provisioning plans are run in the background. If you prefer to run your deployments in the foreground to easily check the status of your deployment, perform the following tasks.

1. Disable physical locking in the N1 SPS software.

   Caution – Disabling physical locking of hosts is not supported in the N1 SPS software.

   Set the `pe.lockPhysicalHost` property in the `config.properties` file as follows.

   ```
   pe.lockPhysicalHost=false
   ```

   The location of the `config.properties` file on the Master Server varies depending on the OS that is running. The following list indicates the location of the file by OS.

   - **Solaris OS** – `/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/server/config/config.properties`
   - **Linux OS** – `/opt/sun/N1_Service_Provisioning_System_5.2/server/config/config.properties`
   - **Windows OS** – `c:\Program Files\N1 Service Provisioning System 5.2\server\config\config.properties`
2. Restart the Master Server.
3. Create the OS Provisioning server.

   When you create the OS Provisioning server, set the `run_provisioning_foreground` variable to `true`.

Validating New Target Hosts

By default, the OS Provisioning server validates the target hosts before deploying an OS to the target host. To disable target validation, edit the `/opt/SUNWn1osp/etc/n1osp-service.properties` file on the OS Provisioning server to include one or more of the following entries.

- `com.sun.n1.isp.osservice.validate.Target.skipValidation=true`

  Skipstarget host validation. The default value for this property is `false`.
- `com.sun.n1.isp.osservice.validate.Target.skipNMValidation=true`

  Skips Network Management validation for all Service Controllers or network management controllers of the target host. The default value for this property is `false`.
- `com.sun.n1.isp.osservice.validate.Target.skipNMValidation.`*hostname*`=true`

  Skips Network Management validation for all Service Controllers or network management controllers of a specific target instance (*hostname*). The default value for this property is `false`.

- `com.sun.n1.isp.osservice.validate.Target.skipNM-ICMP-Validation=true`

  Skips the ICMP echo validation of the Network Management IP or Terminal server IP connected to the Network Management controller. The default value for this property is `false`.
- `com.sun.n1.isp.osservice.validate.Target.skipNM-ICMP-Validation.`*hostname*`=true`

  Skips the ICMP echo validation of the Network Management IP or Terminal server IP connected to the Network Management controller for a specific target instance (*hostname*). The default value for this property is `false`.
- `com.sun.n1.isp.osservice.validate.Target.NMValidation.`*NMtype*`=ip,TS`

  Enables you to specify the Network Management validation scheme for a custom Network Management type (*NMtype*). Valid values are `ip`, `ts`, `ip,ts`, or `none`, where `ip` is IP validation and `ts` is Terminal Server validation.

Note – You cannot override the default validation scheme of the built-in network management types, for example, `alom`, `lom`, `rsc`, or `ilo`.

Index