Sun N1 Service Provisioning System 5.2 Installation Guide

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

Part No: 819–4445–10
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 is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. JVM is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. 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. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. Netscape Navigator is a trademark or registered trademark of Netscape Communications Corporation in the United States and other countries. Mozilla is a trademark or registered trademark of Netscape Communications Corporation in the United States and other countries.

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 sont des marques déposées ou enregistrées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. JVM sont des marques déposées ou enregistrées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. 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. UNIX est une marque enregistree aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd. Netscape Navigator est une marque de Netscape Communications Corporation aux Etats-Unis et dans d'autres pays. Mozilla est une marque de Netscape Communications Corporation aux Etats-Unis et ? d'autres pays.

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.

060331@14558

Contents

Preface 13

1 Sun N1 Service Provisioning System 5.2 Overview 17
Installing the Sun N1 Service Provisioning System 5.2 – Process Overview 17
Overview of Sun N1 Service Provisioning System 5.2 Applications 18
Master Server 19
Local Distributor 19
Remote Agent 19
Command Line Interface Client 20
Network Protocols 20
Raw (TCP/IP) 21
Secure Shell 21
Secure Sockets Layer 21
Introduction to Plug-Ins 22
Acquiring Plug-Ins 22
Security Features in the Sun N1 Service Provisioning System 22

2 System Requirements for the Sun N1 Service Provisioning System 5.2 25
Operating System Requirements 25
Operating System Configurations 26
Required Operating System Patches 28
Hardware Requirements 32
Master Server Hardware Requirements 32
Local Distributor, Remote Agent, and CLI Client Hardware Requirements 32
Directory Size Requirements 33
General System Requirements 33
Supported Web Browsers 33
Requirements for SSH 34

Requirements for Jython 34
Requirements for Locales 34

3 Gathering Information Before Installation 35
General Configuration Decisions 35
The Java Runtime Environment 35
User Ownership of Applications 36
Host Names and IP Addresses 37
Jython 37
Security Configuration Decisions 37
Network Protocol – Raw, SSH, SSL 37
HTTP or HTTPS 38
Worksheet for All Applications 39
Worksheet for the Master Server 40
Worksheet for Local Distributors 41
Worksheet for Remote Agents 41
Worksheet for CLI Clients 42

4 Installing the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems 43
Installing the Sun N1 Service Provisioning System 5.2 43
▼ How to Install the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems 43
Non-Interactive Installation of a Remote Agent on Linux and UNIX Systems 46
▼ How to Non-Interactively Install a Remote Agent on Linux and UNIX Systems 46
Remote Installation of Remote Agents on Linux and UNIX Systems 48
▼ How to Remotely Install Remote Agents on Linux and UNIX Systems 48
Starting Applications on Linux and UNIX Systems 51

5 Installing the Sun N1 Service Provisioning System 5.2 on Windows Systems 53
Installing the Master Server 53
▼ How to Install the Sun N1 Service Provisioning System 5.2 Master Server on Windows 53
▼ How to Create a Scheduled Task to Optimize the Database 55
Installing the Remote Agent, Local Distributor, and CLI Client 56
▼ How to Install the Remote Agent, Local Distributor, and CLI Client on Windows 56
Non-Interactive Installation of a Remote Agent on Windows 57
▼ How to Non-Interactively Install Remote Agents on Windows 57
Remote Installation of Remote Agents on Windows 59

▼ How to Remotely Install Remote Agents on Windows 59
Remote Agent Variable Values 61
Starting Applications on Windows Systems 62

6 Configuring the Sun N1 Service Provisioning System 5.2 for HTTPS 63
Creating a Keystore File and Keystore Password for HTTPS Connections 64
▼ How to Generate SSL Certificates 64
▼ How to Obtain a Signature for an SSL Certificate 65
Configuring HTTPS After Installation 65
▼ How to Copy the Keystore File 66
▼ How to Create and Configure an Encoded Keystore Password 66
Configuring HTTPS After Selecting HTTP During Installation 67
▼ How to Enable HTTPS Connections from the Master Server Browser Interface to the Web Interface of the Master Server 67
Reverting to HTTP 68
▼ How to Revert to HTTP 69

7 Configuring the Sun N1 Service Provisioning System 5.2 to Use Secure Shell 71
Overview of SSH and Requirements 72
`ssh-agent` or Empty Password Keys 72
SSH Requirements 73
Additional SSH Security 74
Configuring SSH – Process Overview 75
Preparing the Keys 75
▼ How to Generate Key Pairs 75
▼ How to Set Up Keys for the `ssh-agent` 76
▼ How to Set Up Keys for Empty Password Files When Using One Key Pair 77
▼ How to Set Up Keys for Empty Password Files When Using Multiple Key Pairs 78
Setting Up and Testing the Connectivity on the Master Server 78
▼ How to Start the `ssh-agent` on the Master Server 78
▼ How to Set Up and Test the Connectivity on the Master Server 79
Configuring SSH for the Applications 81
▼ How to Configure SSH for Local Distributors and Remote Agents 81
▼ How to Configure SSH for the CLI Client With the `ssh-agent` 82
▼ How to Configure SSH for the CLI Client With Empty Passwords 83
SSH Advanced Parameters and Command Reference 84

Advanced Parameters Reference 84
OpenSSH 2.0 Command Reference 86

8 Configuring the Sun N1 Service Provisioning System 5.2 for SSL 87
Overview of SSL Support in the Sun N1 Service Provisioning System 5.2 87
Cipher Suites: Encryption and Authentication Overview 87
Authentication Keystores 88
Using Passwords With SSL 90
Limitations of SSL on the Sun N1 Service Provisioning System 5.2 90
Configuring SSL – Process Overview 91
▼ How to Create Keystores 91
▼ How to Edit the `config.properties` File to Configure SSL 94
Sample Configuration Scenarios 95
SSL Cipher Suites 99
Cipher Suites 99
Cipher Suites for IBM AIX 100

9 Configuring the Java Virtual Machine Security Policy 101
Configuring the JVM Security Policy 101
▼ How to Configure the JVM Policy for the Master Server 102
▼ How to Configure the JVM Policy for the Remote Agent 102
▼ How to Configure the JVM Policy for the Local Distributor 103
Postgres Security 103

10 Upgrading to the Sun N1 Service Provisioning System 5.2 105
Upgrading Overview 105
Upgrading Requirements 105
Upgrading – Process Overview 106
Upgrading the Master Server 106
▼ How to Migrate Master Server Data 106
Master Server Data Migration Details 109
Upgrading Remote Agents and Local Distributors 111
▼ How to Upgrade Remote Agents and Local Distributors 111

11 Uninstalling the Sun N1 Service Provisioning System 5.2 113
Uninstalling Applications on Linux and UNIX Systems 113
▼ How to Uninstall a Solaris OS Master Server or CLI Client 113
▼ How to Uninstall File-Based Applications on Linux and UNIX Systems 114
▼ How to Disable Automatic Database Optimization 115
Uninstalling Applications on Windows Systems 115

A Installation and Configuration Reference 117
Reference Data for the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems 117
Directory Structure of the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems 117
Database Optimization on Linux and UNIX Systems 120
Sample Remote Agent Parameters File for Linux and UNIX Systems 120
Reference Data for the Sun N1 Service Provisioning System 5.2 on Windows 123
Directory Structure of the Sun N1 Service Provisioning System 5.2 on Windows 123
Cygwin 125
Actions Performed by the Windows Installation Scripts 126

B Troubleshooting 129
Issues During Installation on Linux and UNIX Systems 129
Package Corrupt Error When Installing on IBM AIX (6279820) 129
Warning When Installing the JRE on IBM AIX 130
Issues on Solaris 130
Incomplete Installation on Solaris 130
▼ To Identify and Remove Solaris Packages 130
Issues During Installation on Microsoft Windows 131
Error When Installing on Windows 131
SSH Connectivity 131
Master Server Unable to Connect to Local Distributor Through an Intermediate Local Distributor 131
Unable to Connect to an Application Using SSH 131
▼ How to Troubleshoot SSH Connectivity Issues 132

Glossary 135

Index 141

Tables

TABLE 2–1 Supported Operating Systems for the Sun N1 Service Provisioning System 5.2 25
TABLE 2–2 Solaris `/etc/system` Settings 26
TABLE 2–3 Solaris 10 Resource Controls Settings 27
TABLE 2–4 Linux Master Server System Settings (32–bit, 64–bit) 28
TABLE 2–5 Required Patches for Supported Operating Systems 29
TABLE 2–6 Hardware Requirements for the Master Server 32
TABLE 2–7 Hardware Requirements for the Local Distributor, Remote Agent, and CLI Client 32
TABLE 2–8 Directory Size Minimums for `/var` or `/var/tmp` 33
TABLE 2–9 Web Browser Requirements for the Browser Interface 33
TABLE 3–1 Information Requested for All Applications 39
TABLE 3–2 Information Requested for the Master Server 40
TABLE 3–3 Information Requested for Local Distributors 41
TABLE 3–4 Information Requested for Remote Agents 41
TABLE 3–5 Information Requested for CLI Clients 42
TABLE 4–1 Start Commands for Linux and UNIX System Applications 51
TABLE 5–1 Remote Agent Variable Values 61
TABLE 5–2 Names of Services to Start for the Windows Master Server, Local Distributor, and Remote Agent 62
TABLE 5–3 Start Commands for the Windows CLI Client 62
TABLE 7–1 OpenSSH 2.0 Commands 86
TABLE 10–1 Migration Overview 110
TABLE A–1 Directories Common to All Applications 118
TABLE A–2 Directories Installed for the Master Server 118
TABLE A–3 Directories Installed for the Local Distributor 119
TABLE A–4 Directories Installed for the Remote Agent 119
TABLE A–5 Directories Installed for the CLI Client 119
TABLE A–6 Directories Common to All Applications 124
TABLE A–7 Directories Installed for the Master Server 124
TABLE A–8 Directories Installed for the Local Distributor 124

TABLE A–9 Directories Installed for the Remote Agent 125
TABLE A–10 Directories Installed for the CLI Client 125

Examples

EXAMPLE 5–1 Non-Interactive Installation of a Remote Agent on Windows 58

EXAMPLE 5–2 Remote Installation of a Remote Agent On Windows 60

EXAMPLE 7–1 cprefix Example 85

EXAMPLE 7–2 sshpath Example 85

EXAMPLE 7–3 sshargs Example 85

EXAMPLE 8–1 `crkeys` Command Syntax 93

EXAMPLE 8–2 How to Configure SSL Without Authentication Between the Master Server, Local Distributor, and Remote Agent 95

EXAMPLE 8–3 How to Configure SSL Server Authentication 95

EXAMPLE 8–4 How to Configure SSL Server and Client Authentication 97

EXAMPLE 8–5 How to Configure SSL Authentication Between a CLI Client and Master Server 98

Preface

The *Sun Sun N1 Service Provisioning System 5.2 Installation Guide* describes how to install and upgrade the Sun N1™ Service Provisioning System 5.2.

Note – In this document the term "x86" refers to the Intel 32–bit family of microprocessors and compatible 64–bit and 32–bit microprocessors made by AMD.

Note – The Solaris, IBM AIX, and HP-UX operating systems are based on the UNIX™ platform. These operating systems are generically called "UNIX systems." Red Hat Linux and SUSE Linux are generically called "Linux systems."

Who Should Use This Book

This book is intended for system administrators responsible for installing and configuring the Sun N1 Service Provisioning System 5.2.

How This Book Is Organized

The Sun N1 Service Provisioning System 5.2 Installation Guide describes the following topics.

- Chapter 1 provides an overview of the tasks required to install and configure the software. This chapter also contains an overview of the software and supported network protocols.
- Chapter 2 describes the system requirements for installing and using the software.
- Chapter 3 contains worksheets to help you gather the information you need to install the software.
- Chapter 4 describes the steps to install the software on Linux and UNIX servers.
- Chapter 5 describes the steps to install the software on Windows.
- Chapter 6 describes the steps to configure the browser interface to use HTTPS to connect to the web interface of the Master Server.
- Chapter 7 describes the tasks necessary to configure the software to communicate using SSH.

- Chapter 8 describes the tasks necessary to configure the software to communicate using SSL.
- Chapter 9 describes how to configure the JVM™ [1] security policy.
- Chapter 10 describes the steps to upgrade the software.
- Chapter 11 describes the steps to uninstall the software.
- Appendix A contains reference material related to installing and configuring the software.
- Appendix B describes steps to troubleshoot installation and configuration issues.

Related Books

You might need to refer to the following manuals when you install and use the Sun N1 Service Provisioning System 5.2.

- *Sun N1 Service Provisioning System 5.2 Release Notes*
- *Sun N1 Service Provisioning System 5.2 System Administration Guide*
- *Sun N1 Service Provisioning System 5.2 Command-Line Interface Reference Manual*

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.`

1 The terms "Java Virtual Machine" and "JVM" mean a Virtual Machine for the Java™ platform.

TABLE P–1 Typographic Conventions *(Continued)*

Typeface	Meaning	Example
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

Sun N1 Service Provisioning System 5.2 Overview

This chapter provides an overview of the tasks required to install and configure the Sun N1 Service Provisioning System 5.2. This chapter also contains an overview of the applications included in the Sun N1 Service Provisioning System 5.2 and the types of network protocols that you can use for additional security.

This chapter discusses the following topics:

- "Installing the Sun N1 Service Provisioning System 5.2 – Process Overview" on page 17
- "Overview of Sun N1 Service Provisioning System 5.2 Applications" on page 18
- "Network Protocols" on page 20
- "Introduction to Plug-Ins" on page 22

Installing the Sun N1 Service Provisioning System 5.2 – Process Overview

The process overview below describes the tasks necessary to properly install and configure the Sun N1 Service Provisioning System 5.2.

1. Determine whether your server meets the minimum requirements to install.

   See Chapter 2.

2. Make configuration decisions and gather the information that you need to install the product.

   See Chapter 3.

3. (Optional) You can create a special operating system group and user account to be used by Sun N1 Service Provisioning System 5.2.

   If you create a new user and a new group, be sure to include the new user in the group. For more information about creating user accounts, see the documentation for your operating system.

4. (Optional) Install Jython on CLI Client machines.

   You might choose to install Jython on any machine from which you want to run the CLI Client. Jython is not required to run the CLI Client. Jython is available from `http://www.jython.org`.

For more information about using the CLI CLient with Jython, see Chapter 1, "Using the Command-Line Interface," in *Sun N1 Service Provisioning System 5.2 Command-Line Interface Reference Manual*.

5. Install each of the Sun N1 Service Provisioning System 5.2 applications individually using the appropriate installation script provided on the product media.

   For installation instructions, see Chapter 4 or Chapter 5.

6. (Optional) If you plan to access the Master Server on the Internet, you can increase the Master Server security by configuring the Sun N1 Service Provisioning System 5.2 to use SSH to communicate with that server.

   See Chapter 7.

7. (Optional) If you want to provide the maximum security for communication among the applications, configure the applications to use SSL when communicating.

   See Chapter 8.

8. (Optional) If you do not use SSL to provide security for communication among applications, you can configure the JVM security policy so that the applications accept only connections from `localhost`. This setup provides a minimum level of security.

   See Chapter 9.

9. (Optional) Start the applications.

   The installation program prompts you to start the applications upon successful installation. If you choose not to start the applications at that time, start the applications by following the instructions in "Starting Applications on Linux and UNIX Systems" on page 51 or "Starting Applications on Windows Systems" on page 62.

10. Complete the initial setup.

    See "Configuring the Sun N1 Service Provisioning System – Process Overview" in *Sun N1 Service Provisioning System 5.2 System Administration Guide* for more initial setup instructions.

Overview of Sun N1 Service Provisioning System 5.2 Applications

The Sun N1 Service Provisioning System 5.2 is a distributed software platform. The provisioning system includes the following special-purpose applications that you install on the servers in your network. These applications interact to allow you to deploy software to the servers in your network.

- "Master Server" on page 19 – A central server that stores components and plans, and provides an interface for managing application deployments.
- "Local Distributor" on page 19 – Optional servers that act as a proxy for the Master Server to optimize network communications across data centers and through firewalls.
- "Remote Agent" on page 19 – A management application that performs operations on a host. Every server that you want to be controlled by the Sun N1 Service Provisioning System 5.2 must have the Remote Agent application.

- "Command Line Interface Client" on page 20 – Optional applications that accept commands to be executed on the Master Server.

Master Server

The Master Server runs on Linux, UNIX, and Windows based servers. The Master Server is a central server that does the following:

- Manages a database that identifies all of the hosts that are registered in the provisioning software
- Stores components and plans in a repository
- Performs version control on the objects that are stored in the repository
- Authenticates provisioning system users and ensures that only authorized users perform specific operations
- Includes special-purpose engines for performing tasks such as dependency tracking and deployments
- Provides both a browser interface and a command-line interface for users

Local Distributor

A Local Distributor is a proxy that optimizes the distribution and management of Remote Agents. Data centers can use Local Distributors to do the following:

- Minimize network traffic during deployments. The Master Server sends one copy of a component to a Local Distributor, which replicates the component for installation on other servers.
- Minimize firewall reconfigurations. If a firewall stands between the Master Server and a collection of servers, administrators can open the firewall only for the Local Distributors, rather than for every server involved in a deployment.
- Minimize the load to the Master Server during large scale deployments.

Remote Agent

The Remote Agent is an application that runs on every server being managed by the Sun N1 Service Provisioning System 5.2. Remote Agents perform the tasks requested by the Master Server. Remote agents can do the following:

- Report server hardware and software configurations to the Master Server
- Start and stop services
- Manage directory contents and properties
- Install and uninstall software
- Run operating system commands and native scripts specified by components and plans

Command Line Interface Client

The Command Line Interface (CLI) Client provides a communication path to the Master Server to enable the execution of commands from local and remote servers. The CLI Client enables commands to be executed in the following environments:

- Windows command line
- UNIX shell such as bash

To execute these commands, the CLI Client establishes a connection to the Master Server through TCP/IP or securely using SSL, or SSH.

The CLI Client operates in the following two modes:

- Single-command mode, which enables you to submit one command at a time
- Interactive mode, which prompts you for commands, maintains a command history and allows for Jython scripting

When operating in interactive mode, the CLI Client uses the Jython programming language. Jython is a Java implementation of the high-level, dynamic, object-oriented language Python.

Note – Install Jython on any server on which you plan to run the CLI Client in interactive mode. For more information about Jython and to download Jython, visit `http://www.jython.org`.

Network Protocols

The Sun N1 Service Provisioning System 5.2 supports a variety of network protocols for communication among the software applications. You select the protocol to apply to each of the following types of network communication:

- Communication between the Master Server and Local Distributors or Remote Agents
- Communication between a particular Local Distributor and Remote Agents
- Communication between the Master Server and a CLI Client

The Sun N1 Service Provisioning System 5.2 supports the following protocols:

- Raw (TCP/IP)
- Secure Shell
- Secure Sockets Layer (SSL)

You can tailor your network security to meet the needs of your particular network topology. For example, the communication within each of your data centers is secure, but your network connection to a remote data center passes through the public Internet. You might configure the Master Server to use SSL when communicating with a Local Distributor that is installed inside the firewall of the remote data center. Consequently, the communication over the Internet to the remote data center is secure. The Local Distributor might use raw TCP/IP to communicate with the Remote

Agents because the communication over the local network is secure. For more information about the different protocols and about configuring the protocols, read Chapter 7 and Chapter 8.

Raw (TCP/IP)

Raw (TCP/IP) is standard TCP/IP without additional encryption or authentication. The advantage of raw is that it requires no additional set-up and configuration. If your data center network is protected by a firewall, using raw provides a convenient method for communication among Sun N1 Service Provisioning System 5.2 applications.

Secure Shell

Secure Shell (SSH) is a UNIX command suite and protocol for securely accessing a remote computer. SSH secures network client/server communications by authenticating both endpoints with a digital certificate and by encrypting passwords. SSH uses RSA public key cryptography to manage connections and authentication. SSH is more secure than telnet or other shell-based communication methods.

You can configure the Sun N1 Service Provisioning System 5.2 applications to communicate using SSH. The Sun N1 Service Provisioning System 5.2 supports OpenSSH which is a free version of SSH that has been primarily developed by the OpenBSD Project. For more details about OpenSSH, see `http://www.openssh.com`. The software can be configured to support other versions of SSH as well.

Secure Sockets Layer

Secure Sockets Layer (SSL) is a protocol for securing communication over IP networks. SSL uses TCP/IP sockets technology to exchange messages between a client and a server while protecting the message with a public-and-private key encryption system developed by RSA. Support for SSL is included in most web server products, as well as in the Netscape Navigator™ browser and Microsoft web browsers.

You can configure the Sun N1 Service Provisioning System 5.2 applications to use SSL for network communications to help prevent the software messages from being read or altered. Optionally, the applications can be configured to use SSL to authenticate each other before communicating, thereby increasing network security.

Introduction to Plug-Ins

In general usage, plug-in applications are programs that can easily be installed and used as part of your web browser. A plug-in application is recognized automatically by the browser and its function is integrated into the main HTML file that is being presented. Web browser plug-in applications generally play sound or motion video or perform some other functions.

In the Sun N1 Service Provisioning System environment, a plug-in differs only slightly in concept from the general usage. A plug-in for the Sun N1 Service Provisioning System product is a packaged solution that extends the provisioning capability of the product for a specific platform, application, or environment. For example, you might create a plug-in solution for a specific application, such as Oracle 8i, or for some feature of an operating system, such as Solaris Zones.

A plug-in includes all of the relevant data that is needed to support a new custom application. The contents of the plug– in are described in the plug-in descriptor file. This file is located in a standard place within the plug–in packaging structure.

Acquiring Plug-Ins

Several plug–ins have been created for use with the Sun N1 Service Provisioning System. The plug-ins are available on the Sun N1 Service Provisioning System 5.2: Supplement CD and in the image downloaded from the Sun Download Center.

The plug-ins are packaged in Java archive files (.jar files). To make a given plug-in known to the Sun N1 Service Provisioning System product, you need to import the plug-in. For instructions to import a plug-in, see the user's guide associated with the plug-in that you want to import in the Plug-In User's Guide document collection at `http://docs.sun.com/db/coll/1329.1`.

Security Features in the Sun N1 Service Provisioning System

The Sun N1 Service Provisioning System provides the following features to ensure the security of your data, applications, and network.

- Secure HTTP (HTTPS) support –You can configure your SPS environment to use HTTPS to ensure the security of your connection to the master server. By using Secure Sockets Layer (SSL) digital certificates and keystores, you can protect your connection to the Master Server through the web browser interface.

  For more information, see Chapter 6.

- Secure Shell (SSH) support –SSH is a UNIX-based command suite and protocol for securely accessing a remote computer. SSH secures network client/server communications by authenticating both endpoints with a digital certificate and by encrypting passwords. SSH uses RSA public key cryptography to manage connections and authentication.

  For more information, see Chapter 7.

- Secure Sockets Layer (SSL) support – SSL is a protocol for securing communication over IP networks. SSL uses TCP/IP sockets technology to exchange messages between a client and a server, while protecting the message with a public and private key encryption system developed by RSA. Support for SSL is included in most web server products and most web browser software.

  Sun N1 Service Provisioning System 5.2 applications can be configured to use SSL for their network communications, preventing messages from being read or tampered with.

  For more information, see Chapter 8.

- Java Virtual Machine (JVM) security policies – Each Sun N1 Service Provisioning System 5.2 application has a Java Virtual Machine (JVM) security policy file. You can modify this policy file to restrict access to the Master Server, Remote Agents, and Local Distributors. You can configure these applications to only accept connections from a specific IP Address and Port range or to allow them only to connect to a specific IP Address and Port range.

  For more information, see Chapter 9.

- User authentication - The Sun N1 Service Provisioning System supports the Java Authentication and Authorization Service (JAAS) for user authentication. By editing the `jaas.config` file, you can configure your SPS environment to use the following authentication services:

  - LDAP
  - Sun Directory Server
  - Microsoft Windows 2000 Active Directory server

  For more information, see Appendix A, "Authentication Methods," in *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

- User groups - You can control the access to the different features of the Sun N1 Service Provisioning System software by assigning users to specific user groups. You can then assign certain permissions to each user group to limit the actions available to the users in that group.

  For more information, see Chapter 4, "Managing Users," in *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

- Permissions - You can assign permissions to user groups to restrict user access to either your entire provisioning environment or to specific plans and components. You can designate permission to run provisioning plans or to create components to a specific user group, and limit the permissions of a different user group to enable those users to manage user accounts.

  For more information, see Chapter 3, "Controlling Access Using Permissions," in *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

  You can also assign file system permissions to the files and directories assigned to a specific resource. By editing the XML in a resource descriptor file, you can override the file system permissions that are created when you check in a resource to SPS. For more information, see "Using a Resource Descriptor File" in *Sun N1 Service Provisioning System 5.2 XML Schema Reference Guide*.

- User-Specific Steps – The SPS XML schema enables you to specify alternate users to run specific plans or components. The `UserToRunAs` attribute of the execNative element allows you designate specific users to run commands native to the operating system on a target host.

For more information, see "<execNative> Step" in *Sun N1 Service Provisioning System 5.2 XML Schema Reference Guide* Step in Sun N1 Service Provisioning System 5.2 XML Schema Reference Guide.

- Access to Component Elements - The XML schema for Sun N1 SPS components allows you to control the access to specific elements of a component, including the following items:
  - Component variables
  - Steps to install, uninstall, or control steps on the component

  The access attribute enables you to limit the accessibility of specific component elements to varying degrees. For more information, see Chapter 3, "Component Schema," in *Sun N1 Service Provisioning System 5.2 XML Schema Reference Guide*.
- User Sessions - Sun N1 SPS can use user sessions to authenticate users and their credentials to perform provisioning tasks. These sessions can be used to identify users throughout a series of related requests without reauthentication. User sessions use session variables to preserve session-related information, such as user authentication and other credentials.

  For more information, see "Session Variable Concepts" in *Sun N1 Service Provisioning System 5.2 Plan and Component Developer's Guide*.

  You can modify the session variables in a current session without affecting the session variables that are saved in the database, or you can save your modifications. You can use the either the browser interface or the command-line interface to manage session variables. For more information, see Chapter 5, "Session Variables," in *Sun N1 Service Provisioning System 5.2 Plan and Component Developer's Guide*.

  You can also supply session IDs with the command-line interface to authenticate specific commands. For more information, see Chapter 1, "Using the Command-Line Interface," in *Sun N1 Service Provisioning System 5.2 Command-Line Interface Reference Manual*.

  You can set user session duration and timeout policies for your SPS environment with a series of configuration variables in the `config.properties` file. For more information, see Appendix B, "Commonly Updated Configuration Variables," in *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

◆ ◆ ◆ CHAPTER 2

System Requirements for the Sun N1 Service Provisioning System 5.2

This chapter lists the system requirements for installing and using the Sun N1 Service Provisioning System 5.2. This chapter discusses the following topics:

- "Operating System Requirements" on page 25
- "Hardware Requirements" on page 32
- "General System Requirements" on page 33

Operating System Requirements

You can install the Sun N1 Service Provisioning System 5.2 Master Server, Remote Agent, Local Distributor, and CLI Client on servers that are running the following operating systems:

TABLE 2–1 Supported Operating Systems for the Sun N1 Service Provisioning System 5.2

Sun N1 Service Provisioning System Application	Supported Operating Systems
Master Server	■ Solaris 8 OS running on SPARC® based servers ■ Solaris 9 and Solaris 10 OS running on SPARC and x86 based servers ■ Red Hat Linux Advanced Server 2.1 (32-bit) ■ Red Hat Linux Advanced Server 3 (32-bit, 64-bit) ■ Red Hat Linux Advanced Server 4 (32–bit, 64–bit) ■ Microsoft Windows 2000 Server and Microsoft Windows 2000 Advanced Server

TABLE 2–1 Supported Operating Systems for the Sun N1 Service Provisioning System 5.2 *(Continued)*

Sun N1 Service Provisioning System Application	Supported Operating Systems
Local Distributor, Remote Agent, CLI Client	■ Solaris 7 and Solaris 8 OS running on SPARC based servers ■ Solaris 9 and Solaris 10 OS running on SPARC and x86 based servers ■ Red Hat Linux Advanced Server 2.1 (32-bit) ■ Red Hat Linux Advanced Server 3 (32–bit, 64–bit) ■ Red Hat Linux Advanced Server 4 (32–bit, 64–bit) ■ IBM AIX 5.1, 5.2, and 5.3 ■ SUSE Linux Enterprise Server 8 and SUSE Linux Enterprise Server 9 (32-bit) ■ HP-UX 11i V1 on PA-RISC based servers ■ Microsoft Windows 2000 Server and Microsoft Windows 2000 Advanced Server ■ Microsoft Windows Server 2003: Standard, Enterprise, or Web Edition ■ Microsoft Windows Standard 2003 x64 Edition or Microsoft Windows Enterprise x64 Edition

Operating System Configurations

To install and run the provisioning system, you must configure your systems as described in the following sections.

Solaris System Configuration Requirements

A Solaris system that is running the Master Server requires the following `/etc/system` settings.

Note – If you are using the Solaris 9 or Solaris 10 OS, you cannot change the values for `shmsys:shminfo_shmmin` and `shmsys:shminfo_shmseg`. If you are using the Solaris 10 OS, you cannot change the values for `shmsys:shminfo_shmmax`, `shmsys:shminfo_shmmni`, `semsys:seminfo_semmns`, and `semsys:seminfo_semvmx`.

The default values for these settings are acceptable.

TABLE 2–2 Solaris `/etc/system` Settings

Variable	Minimum Value
`shmsys:shminfo_shmmax`	0x20000000[1]
`shmsys:shminfo_shmmin`	1
`shmsys:shminfo_shmmni`	2

[1] 536870912 in decimal (512Mb), but this number must be specified in hex for the Solaris 8 Operating System.

TABLE 2–2 Solaris `/etc/system` Settings *(Continued)*

Variable	Minimum Value
`shmsys:shminfo_shmseg`	1
`semsys:seminfo_semmni`	32
`semsys:seminfo_semmns`	512
`semsys:seminfo_semmsl`	17
`semsys:seminfo_semvmx`	537

For more instructions to change the `/etc/system` settings, see *Solaris Tunable Parameters Reference Manual*.

Solaris 10 Configuration Requirements

In the Solaris 10 environment, the `/etc/system` settings are replaced by resource controls in the context of the user who owns the installation. The following table defines the minimum values:

TABLE 2–3 Solaris 10 Resource Controls Settings

Variable	Minimum Value
`project.max-shm-memory`	512
`project.max-sem-ids`	32
`process.max-sem-nsems`	17

▼ To Determine If the Resource Controls Need to be Reset

1 Determine the project ID.

```
$ id -p
```

2 Verify the current values of `project.max-shm-memory`, `project.max-sem-ids`, and `process.max-sem-nsems` assuming a project ID value of *1*.

```
# prctl -P -n project.max-shm-memory -i project 1
# prctl -P -n project.max-sem-ids -i project 1
# prctl -P -n process.max-sem-nsems -i process $$
```

3 Set resource controls.

If the current values do not meet the minimum settings required, you can use either the `prctl` command or the `projmod` command. Changes made with the `prctl` command are good only until system reboot. The `projmod` command should be used to make changes that persist across reboot.

- Using the `prctl` command:

```
# prctl -n project.max-shm-memory -v 536870912 -r -i project 1
# prctl -n project.max-sem-ids -v 32 -r -i project 1
# prctl -n process.max-sem-nsems -v 17 -r -i process $$
```

- Using the `projmod` command:

```
# projmod -a -K "project.max-shm-memory=(priv,512mb,deny)" default
# projmod -a -K "project.max-sem-ids=(priv,32,deny)" default
# projmod -a -K "process.max-sem-nsems=(priv,17,deny)" default
```

Linux System Configuration Requirements

The `bc` command must be in the user's path when you install the Sun N1 Service Provisioning System. Without the `bc` command, the installation exits and requests that `bc` be installed. Install the `bc-1.06-5.rpm` package or a later version of the package.

When you install the Sun N1 Service Provisioning System on an NFS mounted directory on a SUSE Linux 8 server and you want to use SSH or SSL for secure connections, configure the NFS client with `nolocks` or the NFS server to allow locks. If neither the NFS client nor the NFS server is configured properly, the Sun N1 Service Provisioning System application will not start.

The Linux Master Server installation program checks the following system parameters and exits with an error if the minimum values are not met.

TABLE 2–4 Linux Master Server System Settings (32–bit, 64–bit)

System Parameter	Minimum Value
`shmall` in `/proc/sys/kernel/shmall`	536870912 (512Mb)
`shmmax` in `/proc/sys/kernel/shmmax`	536870912 (512Mb)

Additionally, for the Linux 64–bit platform the semaphore settings should be changed in the `/proc/sys/kernel/sem` file to the following values:

```
4000  512000  512  4096
```

These numbers equate to the following four values:

- SEMMSL – number of semaphores in a semaphore set
- SEMMNS – number of semaphores in the system
- SEMOPM – number of operations per semop call
- SEMMNI – number of semaphore identifiers

Required Operating System Patches

The following table lists the required patches for each supported operating system.

TABLE 2–5 Required Patches for Supported Operating Systems

OS Version	Required Patches
Solaris 7	106980-16
	106541-16
	107544-03
	106950-13
	106327-08
	106300-09
Solaris 8, SPARC based servers	111310-01
	109147-28
	111308-04
	112438-03
	108434-15
	108435-15
	111111-04
	112396-02
	110386-03
	111023-03
	111317-05
	113648-03
	115827-01
	116602-01
	108987-13
	108528-29
	108989-02
	108993-33
	109326-14
	110615-10
Solaris 9, SPARC based servers	114356-06
Solaris 9, x86 based servers	114357–06
Solaris 10, SPARC based servers	None

TABLE 2–5 Required Patches for Supported Operating Systems *(Continued)*

OS Version	Required Patches
Solaris 10, x86 based servers	None
IBM AIX 5.1	AIX 5.1–5.1.4.0 maintenance level: APAR IY44478
IBM AIX 5.2	AIX 5.2–5.2.1.0 maintenance level: APAR IY44479
IBM AIX 5.3	AIX 5.3.1.0 maintenance level: APAR IY58143
Red Hat Linux Advanced Server 2.1 (32-bit)	None
Red Hat Linux Advanced Server 3 (32-bit, 64-bit)	None
Red Hat Linux Advanced Server 4 (32-bit, 64-bit)	None
SUSE Linux Enterprise Server 8 (32-bit)	None
SUSE Linux Enterprise Server 9 (32-bit)	None

TABLE 2–5 Required Patches for Supported Operating Systems *(Continued)*

OS Version	Required Patches
HP-UX 11i V1 on PA-RISC based systems	PHNE_23502
	PHKL_24253
	PHKL_24254
	PHKL_24255
	PHKL_24256
	PHKL_24257
	PHKL_24751
	PHNE_25084
	PHCO_25226
	PHKL_25227
	PHKL_25367
	PHCO_25452
	PHKL_25468
	PHKL_25614
	PHKL_25728
	PHKL_25729
	PHKL_25840
	PHKL_25842
	PHKL_25871
	PHKL_27091
	PHKL_27092
	PHKL_28489
	PHNE_29887
	PHCO_29960
	PHSS_30049
Windows 2000 Server or Windows 2000 Advanced Server	Service Pack 3
Windows Server 2003	none
Windows Server 2003 x64	none

Hardware Requirements

Master Server Hardware Requirements

The following table lists the hardware requirements for installing the Master Server on the supported operating systems.

TABLE 2–6 Hardware Requirements for the Master Server

	Solaris	Red Hat Linux	Windows
Hardware	SPARC or x86 based[1]	x86 based	x86 based
CPU	450 MHz single or multiple CPU	1 GHz single or multiple CPU	1 GHz single or multiple CPU
RAM	At least 1 GB RAM	At least 1 GB RAM	At least 1 GB RAM
Free Hard Disk Space	2 GBs	2 GBs	2 GBs

[1] SPARC only for the Solaris 8 OS

Local Distributor, Remote Agent, and CLI Client Hardware Requirements

The following table lists the hardware requirements for installing the Local Distributor, Remote Agent, and CLI Client on the supported operating systems.

TABLE 2–7 Hardware Requirements for the Local Distributor, Remote Agent, and CLI Client

	Solaris	Red Hat Linux	IBM AIX	Suse Linux	HP-UX	Windows
Hardware	SPARC or x86 based[1]	x86 based	pSeries	x86 based	PA-RISC based	x86 based
CPU	400 MHz single or multiple CPU	1 GHz single or multiple CPU	400 MHz single or multiple CPU	400 MHz single or multiple CPU	400 MHz single or multiple CPU	1 GHz single or multiple CPU
RAM	At least 256 MBs RAM	At least 1 GB RAM	At least 256 MBs RAM	At least 1 GB RAM	At least 256 MBs RAM	At least 1 GB RAM
Free Hard Disk Space	1 GB	1 GB	1 GB	1 GB	1 GB	1 GB HD free space

[1] SPARC only for the Solaris 7 and Solaris 8 OS

Directory Size Requirements

The following table lists the minimum directory sizes for the `/var` or `/var/tmp` directories.

TABLE 2–8 Directory Size Minimums for `/var` or `/var/tmp`

	Solaris	Red Hat Linux	IBM AIX	HP UX
Master Server	500 MB	300 MB	N/A	N/A
CLI	350 MB	100 MB	100 MB	300MB
Remote Agent	100 MB	100 MB	100 MB	300 MB
Local Distributor	100 MB	100 MB	100 MB	300 MB

General System Requirements

This section lists general system requirements for installing and using the Sun N1 Service Provisioning System 5.2.

Supported Web Browsers

The following table lists the web browser requirements for the Sun N1 Service Provisioning System 5.2 browser interface.

TABLE 2–9 Web Browser Requirements for the Browser Interface

Platform	Browser
Solaris, Red Hat, SUSE, HP-UX	Netscape Navigator™ 7.1, Mozilla™ 1.4
Windows	Netscape Navigator 7.1, Mozilla 1.4, Internet Explorer 5.5, Internet Explorer 6.0
AIX	Mozilla 1.4

Note – Some web proxy servers are configured to block popup windows. The Sun N1 Service Provisioning System 5.2 relies on the ability to present popup windows to run properly. Do not run web proxy servers that block popup windows or set your browser to block popup windows.

To run properly, the Sun N1 Service Provisioning System 5.2 relies on the ability to use cookies. Set your browser to allow the use of cookies.

Requirements for SSH

If you want to use SSH for secure connections on Linux and UNIX systems, you must have SSH protocol version 2 installed on each server that you want to use SSH.

Requirements for Jython

If you want to use Jython with the CLI Client, install Jython version 2.0 or higher. For more information about Jython, see `http://www.jython.org`.

Requirements for Locales

The Sun N1 Service Provisioning System 5.2 has been internationalized to install and run in localized environment. Also, the Sun N1 Service Provisioning System 5.2 accepts non–ASCII characters. You will need to adhere to the following requirements if you require that the software support non-ASCII characters:

- All applications must be run in the same locale or in locales that are equivalent. The Remote Agent, Local Distributors, and CLI Client must run a locale that is compatible with the locale in which the Master Server is running.
- You must use Internet Explorer 5.5 or 6.0, or Netscape 7.0.
- Set the web browser Character Interface to use UTF-8, which is also known as Unicode or Universal Alphabet.
- In the configuration files, such as the `config.properties` file, all non-ASCII characters must be Unicode-encoded. You can create configuration files in any encoding. Then, use the `native2ascii` command that is available in the Java™ Development Kit (JDK) package to convert the file to Unicode-encoded ASCII characters.

◆ ◆ ◆ CHAPTER 3

Gathering Information Before Installation

This chapter contains information and worksheets to help you make decisions and gather all of the information that you need to install the Sun N1 Service Provisioning System 5.2. This chapter covers the following topics:

- "General Configuration Decisions" on page 35
- "Security Configuration Decisions" on page 37
- "Worksheet for All Applications" on page 39
- "Worksheet for the Master Server" on page 40
- "Worksheet for Local Distributors" on page 41
- "Worksheet for Remote Agents" on page 41
- "Worksheet for CLI Clients" on page 42

General Configuration Decisions

The installation program prompts you for configuration information for the Sun N1 Service Provisioning System 5.2. Use the sections below to make configuration decisions before you begin the installation.

The Java Runtime Environment

When installing on Linux and UNIX servers, the installation program prompts you to install the JRE or to provide a valid path to a JRE. When installing on Windows, the installation program automatically installs the JRE without prompting you.

Note – On Solaris 10 servers, the installation script prompts you to use JRE v1.4.2 that is already installed in `/usr/j2se`. You can choose to use this version of the JRE rather than installing a new version during an interactive installation.

If you are installing on a Red Hat Linux or a SUSE Linux server, the installation script searches your machine for an instance of the JRE in the default location.

- If the installation program finds the JRE in the default location, you can choose whether or not to reinstall the JRE.

If you are installing on a Solaris OS, IBM AIX, or HP-UX server and you chose not to install the JRE, the installation script prompts you to provide a path to a valid JRE. Then the installation script verifies that the JRE is supported.

- If the JRE is not supported but has a higher version number than the versions that are supported, the installer warns you that the JRE is not supported and asks if you want to continue.
- If you specified a version of the JRE that is supported by the Sun N1 Service Provisioning System 5.2, the installation script sets the `JRE_HOME` variable to the JRE that you specified. The installation script also creates a symbolic link, *N1SPS5.2-home*`/common/jre`, which points to the JRE directory. *N1SPS5.2-home* is the home directory of the Sun N1 Service Provisioning System 5.2. By creating a symbolic link, the Sun N1 Service Provisioning System 5.2 applications use the JRE without changing its location, which other applications might depend upon.

Note – You should install the bundled JRE only once for each machine. For example, if you are installing the Master Server, a Local Distributor, and the CLI Client on the same machine, you should install the JRE with the Master Server, but not with the Local Distributor or the CLI Client.

User Ownership of Applications

The installation program prompts you to select a user and group to own the application that you are installing. If you want to configure the applications to communicate using SSH, install the Master Server, Local Distributors, and Remote Agents as the same user.

The root user cannot own the Master Server. You can install the Master Server as the user that owns the Master Server or you can install the application as root and, when you are prompted, specify which user owns the Master Server .

Note – If you are installing the Master Server or the CLI Client on a Solaris server, you must login as root.

If you want the Remote Agent to have root privileges on the machine where it is running, then you must run the installation program as the root user. Even though you may specify a user other than root to own the Remote Agent, if you want the Remote Agent to have root privileges on the machine where it is running, start the installation program as the root user.

Host Names and IP Addresses

The Sun N1 Service Provisioning System 5.2 applications require all servers to have a static IP address because the Sun N1 Service Provisioning System 5.2 application installed on a server uses the IP address to listen for network requests. The installation program prompts you to supply either a host name or an IP address. If the host name on a server does not resolve to that server's IP address, you will not be able to configure that server to connect within the Sun N1 Service Provisioning System.

If you supply a host name during the installation, the host name must resolve to the actual IP address of the server. Some servers are configured so that the host name does not resolve to the IP address or so that the host name resolves to the loopback address, 127.0.0.1. If the Sun N1 Service Provisioning System application is configured with the host name on a server with this configuration, the application might fail to start. Or, connections to this server from other Sun N1 Service Provisioning System applications also might fail.

When installing an Sun N1 Service Provisioning System application, specify the IP address of the server, not the host name. If you choose to specify the host name, ensure that the host name resolves to the actual IP address of the server.

Note – The installation program prompts you to choose network protocols for communication among the software applications. If you choose to use SSH to communicate between the Master Server and the CLI Clients, the installation program sets the IP address for the Master Server to 127.0.0.1

Jython

When you install the CLI Client, the installation program prompts you to specify whether or not Jython is installed on the machine. The CLI Client uses the Jython programming language to run in interactive mode. However, Jython is not required to use the CLI Client. For more information about Jython and the CLI Client, see "Command Line Interface Client" on page 20.

Security Configuration Decisions

Network Protocol – Raw, SSH, SSL

The installation program prompts you to choose a network protocol for communication among the software applications. For the Master Server, you can choose raw (TCP/IP) or SSL. For Local Distributors, Remote Agents, and CLI Clients, you can choose raw (TCP/IP), SSH, or SSL.

Raw (TCP/IP) is an insecure communication protocol. When using this connection protocol with the provisioning system, anyone with network access to a server that has an Sun N1 Service

Provisioning System 5.2 application installed on it can connect to the provisioning system and issue commands. If you choose raw, you can secure the provisioning system by configuring the security policy file to only accept connections from servers that have Sun N1 Service Provisioning System 5.2 applications. For more details, see Chapter 9.

SSL is more secure than raw. If you select SSL, you must also specify which cipher suite to use, encryption with no authentication or encryption with authentication. Encryption with no authentication is similar to using raw in that anyone with network access to a server that has a provisioning system application installed on it can connect to the provisioning system and issue commands. The encryption with authentication mode is the most secure choice when using SSL. You can further secure the provisioning system by configuring the security policy file to only accept connections from servers that have Sun N1 Service Provisioning System 5.2 applications. For more details, see Chapter 9. For more information about SSL, see Chapter 8.

Note – When you use SSL with a Local Distributor on an AIX server, the SSL cipher suite is set to encryption with authentication. Encryption with no authentication is not available for Local Distributors that are running on AIX servers.

SSH is the most secure network protocol and supported on only Linux and UNIX based platforms. To use SSH with the Sun N1 Service Provisioning System 5.2, you must install SSH software on your servers. For more information, see Chapter 7.

Note – If you choose to use SSH as the network protocol for communication between the Master Server and the CLI Clients, the IP address of the Master Server is set to 127.0.0.1. The communication protocol for the Master Server is set to raw. You must configure the CLI Client to connect to the Master Server using SSH.

HTTP or HTTPS

You can choose for the browser interface to use Hypertext Transmission Protocol (HTTP) or Hypertext Transmission Protocol, Secure (HTTPS) to connect to the Master Server. If you select HTTP, anyone on the network can intercept data that is transmitted between the browser interface and the Master Server. Also, users might attempt to act as a Master Server to obtain secure data from the browser interface, such as user passwords.

HTTPS transfers data more securely than HTTP. HTTPS requires a keystore file and a keystore password. You must create a keystore file and a keystore password for the provisioning system to use.

The installation program prompts you to select HTTP or HTTPS. If you choose HTTPS, the installation program prompts you to enter the keystore file and keystore password. The installation program offers two options for providing the keystore file and the keystore password:

- **Create keystore file and keystore password before installation:** If you create a keystore file and a keystore password before you install the provisioning system, then you can provide the keystore file location and keystore password information during installation. For instructions on how to create a keystore file and a keystore password before you install the provisioning system, see "Creating a Keystore File and Keystore Password for HTTPS Connections" on page 64.

  Note – The installation program does not verify that the keystore password is valid. If you enter an invalid password, you might not be able to start the Master Server.

- **Create keystore file and keystore password after installation:** If you do not create a keystore file and a keystore password before installation, you cannot provide the keystore file and keystore password information during the installation. Consequently, the installation script creates an empty keystore file in the *N1SPS5.2-home*`/server/tomcat` directory. You must manually replace the empty keystore file and keystore password information before you can start the Master Server. For instructions, see "Configuring HTTPS After Installation" on page 65.

If you select HTTP during installation, you can manually configure the Master Server to use HTTPS. For instructions, see "Configuring HTTPS After Selecting HTTP During Installation" on page 67.

Worksheet for All Applications

The installation scripts for each of the Sun N1 Service Provisioning System 5.2 applications begin by performing the same set of preparatory tasks and asking the same questions about directories and files. Use the following worksheet to gather the information that you need to install each of the Sun N1 Service Provisioning System 5.2 applications.

TABLE 3–1 Information Requested for All Applications

Description	Value
The base directory in which to install the software. Example: `/opt/SUNWn1sps` **Note –** The installation path must not exceed 37 characters for Solaris and 35 characters for Linux.	_____
If the JRE is already installed on the machine, the path to the JRE. Example: `/usr/local/jre` or the value of your `JAVA_HOME` environment variable	_____
The user that you want to own the application that you are installing.	_____
On Linux and UNIX systems, the group that you want to own the application that you are installing.	_____

Worksheet for the Master Server

Use the following worksheet to gather the information that you need to install the Master Server.

TABLE 3–2 Information Requested for the Master Server

Description	Value
IP address or host name for the Master Server machine. If you chose to use SSH as the communication protocol between the Master Server and the CLI Clients, the IP address for the Master Server is automatically set to 127.0.0.1.	_____
IP port number that the CLI Client should use to connect to the Master Server. Example: 1130	_____
IP address or host name of the SMTP mail server for the software to use to send notification mail messages.	_____
The subject line of email notifications from the software. Example: Sun N1 Service Provisioning System notification	_____
The name of the user account (user name) from which email notifications are sent. The installation program does not verify the validity of the user account name that you type.	_____
The name of the user account that the software should use when executing native commands. The installation program does not verify the validity of the user account name that you type.	_____
The port number on which the Postgres database will listen. Example: 5432	_____
The password for the admin user to access the Master Server browser interface after installation is complete.	_____
The port number on which the browser interface will be available. Example: 8080 for HTTP or 8443 for HTTPS	_____
Whether you want to automate the optimization of your Postgres database. If yes, specify the time of day you want the Master Server database to be optimized by using the HH:MM format. Example: 23:00 An entry will be made in your `crontab` file to optimize the database every day. Before installing, verify that a `crontab` file exists. If not, create one.	_____

TABLE 3–2 Information Requested for the Master Server *(Continued)*

Description	Value
For HTTPS only:	
Path to keystore file and keystore password.	_____
For more information, see "Creating a Keystore File and Keystore Password for HTTPS Connections" on page 64	_____

Worksheet for Local Distributors

Use the following worksheet to gather the information that you need to install Local Distributors.

TABLE 3–3 Information Requested for Local Distributors

Description	Value
IP address or host name for the Local Distributor machine.	_____
The port number on which this Local Distributor will listen. If you chose to use SSH as the communication protocol between the parent application and the Local Distributor, the port number was automatically set to 70001. Example: 1132	_____

Worksheet for Remote Agents

Use the following worksheet to gather the information that you need to install Remote Agents.

TABLE 3–4 Information Requested for Remote Agents

Description	Value
IP address or host name on which the Remote Agent will run.	_____
The port number on which this Remote Agent will listen. If you chose to use SSH as the communication protocol between the parent application and the Remote Agent the port number was automatically set to 70000. Example: 1131	_____

Worksheet for CLI Clients

Use the following worksheet to gather the information that you need to install CLI Clients.

TABLE 3–5 Information Requested for CLI Clients

Description	Value
IP address or host name of the Master Server for the command line user interface. If you chose to use SSH as the communication protocol between the Master Server and the CLI Client, the IP address for the Master Server is 127.0.0.1.	_____
The IP port number of the Master Server. If you chose to use SSH as the communication protocol between the Master Server and the CLI Client, the port number was automatically set to 80001. Example: 1130	_____
If you chose to use SSH as the communication protocol between the Master Server and the CLI Client, you must enter the installation directory of the Master Server. Example: `/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/server`	_____
If Jython is already installed on this machine, the path to Jython. Default Value: `/usr/local/jython`	_____

◆ ◆ ◆ CHAPTER 4

Installing the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems

This chapter describes the steps to install the Sun N1 Service Provisioning System 5.2 on Linux and UNIX systems. This chapter discusses the following topics:

- "Installing the Sun N1 Service Provisioning System 5.2" on page 43
- "Non-Interactive Installation of a Remote Agent on Linux and UNIX Systems" on page 46
- "Remote Installation of Remote Agents on Linux and UNIX Systems" on page 48
- "Starting Applications on Linux and UNIX Systems" on page 51

Installing the Sun N1 Service Provisioning System 5.2

You will install each of the applications separately by using the appropriate installation script on the product media. The installation scripts for each of the Sun N1 Service Provisioning System 5.2 applications begin by performing the same set of preparatory tasks and asking the same questions about directories, files, and installing the Java™ runtime environment (JRE). Each script then asks specific configuration questions about the application that it will install.

▼ How to Install the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems

Before You Begin Review the installation process overview in "Installing the Sun N1 Service Provisioning System 5.2 – Process Overview" on page 17. Complete any necessary tasks prior to installing the applications.

1 Log in as the user that you want to own the application.

You can log in as root and install the software as the root user. If necessary, the installation program prompts you for information about which user should own the software.

Note – If you are installing the Master Server or the CLI Client on a Solaris server, you must login as root.

2 Access the installation scripts.

- If you are installing from the DVD, insert the DVD: :
- If you are installing from the image that you downloaded, change to the directory where you saved the downloaded image.

3 Change to the directory on the software DVD or within the downloaded image where the installation script is located.

```
# cd /script-directory
```

script-directory is one of the following values:

- `solaris_sparc`
- `solaris_x86`
- `aix`
- `linux`
- `hpux`

4 Start the installation script for the application that you want to install.

```
# cr_app_opsystem_5.2.sh [-allowForwardVersion]
```

app is one of the following values:

- ms – installs the Master Server
- ra – installs the Remote Agent
- ld – installs the Local Distributor
- cli – installs the CLI Client

opsystem is one of the following values:

- solaris_sparc – installs the application on SPARC based hardware that is running the Solaris OS. To install the Master Server or CLI client, use `solaris_sparc_pkg`.
- solaris_x86 – installs the application on x86 based hardware that is running the Solaris OS. To install the Master Server or CLI client, use `solaris_x86_pkg`.
- aix – installs the application on IBM AIX
- linux_i386 – installs the application on Red Hat Linux and SUSE Linux Enterprise Server (32–bit)
- linux_x86_64 – installs the application on Red Hat Linux and SUSE Linux Enterprise Server (64–bit)
- hpux – installs the application on PA-RISC based systems that are running HP-UX

The `-allowForwardVersion` option enables you to install an Sun N1 Service Provisioning System 5.2 application on a version of an operating system that is numerically higher than the highest version

the Sun N1 Service Provisioning System 5.2 supports for that operating system. If you use the `-allowForwardVersion` option, the installation program does not verify that the operating system on which you are installing the application is supported. There is no standard Sun Services support for use of the Sun N1 Service Provisioning System 5.2 on unsupported operating systems.

Caution – Installing the Sun N1 Service Provisioning System 5.2 on unsupported operating systems might result in undefined and unexpected behavior. Install the Sun N1 Service Provisioning System 5.2 on unsupported operating systems only for testing purposes. Do not use the Sun N1 Service Provisioning System 5.2 on unsupported operating systems in a production environment.

5 Answer the configuration questions when prompted by the installation program.

A worksheet is available at "Worksheet for the Master Server" on page 40 outlining the sequence of questions, example values, and options to be gathered prior to installation.

Note –

- The Master Server installation path (BASEDIR) is limited to 37 characters on Solaris.
- The Master Server installation path (BASEDIR) is limited to 35 characters on Linux.

The installation program completes the installation and asks if you want to start the application.

The installation program saves a log of events in the `/tmp/N1SPSInstaller.log.`*pid* file.

pid is the process ID of the installer.

At this point, the following packages have been installed:

- Master Server
  - SUNWspsms
  - SUNWspsc1
  - SUNWspsj1
- CLI
  - SUNWspsc1
  - SUNWspscl
  - SUNWspsj1

Non-Interactive Installation of a Remote Agent on Linux and UNIX Systems

You can install the Remote Agent non-interactively by providing a parameters file to indicate your configuration selections. When you provide a parameters file to the installation program, the installation program does not prompt you for configuration selections during the installation. Instead, the installation program uses the configuration information that is provided in the parameters file.

▼ How to Non-Interactively Install a Remote Agent on Linux and UNIX Systems

Before You Begin You must install a Master Server before you install a Remote Agent. The Master Server does not need to be installed on the machine on which you want to install the Remote Agent.

1 On the machine where you want to install the Remote Agent, log in as the user that you want to own the Remote Agent.

You can log in as root and install the software as the root user. If necessary, the installation program prompts you for information about which user should own the software.

2 Access the installation scripts.

- If you are installing from a DVD, insert the DVD:
- If you are installing from the image that you downloaded, change to the directory where you saved the downloaded image.

3 Change to the directory on the software CD or within the downloaded image where the installation script is located.

`% cd` */script-directory*

script-directory is one of the following values:

- `solaris_sparc`
- `solaris_x86`
- `aix`
- `linux`
- `hpux`

4 Copy the installation script to the machine on which you want to install the Remote Agent.

`% cp cr_ra_`*opsystem*`_5.2.sh` *RA-machine/*

RA-machine is a directory on the machine on which you want to install the Remote Agent. *opsystem* is one of the following values:

- solaris_sparc – installs the Remote Agent on SPARC based hardware running the Solaris OS

- solaris_x86 – installs the Remote Agent on x86 based hardware running the Solaris OS
- aix – installs the Remote Agent on IBM AIX
- linux_i386 – installs the Remote Agent on Red Hat Linux and SUSE Linux Enterprise Server (32–bit)
- linux_x86_64 – installs the Remote Agent on Red Hat Linux and SUSE Linux Enterprise Server (64–bit)
- hpux – installs the application on PA-RISC based systems that are running HP-UX

5 Copy a parameters file into the same directory as the installation script.

A sample parameters file is installed on the Master Server in the *N1SPS5.2-MasterServer-home*`/server/bin` directory when you install the Master Server. You can use the default values that are provided in this file or edit the file and add your custom values. The contents of the `cr_ra_remote_params.sh` sample parameters file and descriptions of the variables that you can set are in "Sample Remote Agent Parameters File for Linux and UNIX Systems" on page 120.

You can also create a new parameters file to use. The parameters file must be an executable file.

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

Caution – The values for the `CR_RA_SUID` and `CR_RA_CTYPE` variables are not set in the sample parameters file. You must set values for these variables or the installation script will abort. Also, if you choose to use an insecure connection type, such as `raw` or SSL encryption with no authentication, you must set the `CR_RA_CTYPE_CONFIRM` variable value to `true`.

The value of the `CR_RA_SUID` variable can be `y`, if and only if the connection type is ssh. If this condition is not met, the installer terminates.

6 Start the installation script.

```
% cr_ra_opsystem_5.2.sh -paramfile parameters-file.sh [-allowForwardVersion]
```

opsystem is one of the following values:

- solaris_sparc – installs the Remote Agent on SPARC based hardware running the Solaris OS
- solaris_x86 – installs the Remote Agent on x86 based hardware running the Solaris OS
- aix – installs the Remote Agent on IBM AIX
- linux_i386 – installs the Remote Agent on Red Hat Linux and SUSE Linux Enterprise Server (32–bit)
- linux_x86_64 – installs the Remote Agent on Red Hat Linux and SUSE Linux Enterprise Server (64–bit)
- hpux – installs the application on PA-RISC based systems that are running HP-UX

parameters-file is the name of the parameters file that you want the installation program to use to obtain the configuration information. The parameters file must be an executable file.

The `-allowForwardVersion` option enables you to install an Sun N1 Service Provisioning System 5.2 Remote Agent on a version of an operating system that is numerically higher than the highest version the Sun N1 Service Provisioning System 5.2 supports for that operating system. If you use the `-allowForwardVersion` option, the installation program does not verify that the operating system on which you are installing the Remote Agent is supported. There is no standard Sun Services support for use of the Sun N1 Service Provisioning System 5.2 on unsupported operating systems.

Caution – Installing the Sun N1 Service Provisioning System 5.2 on unsupported operating systems might result in undefined and unexpected behavior. Install the Sun N1 Service Provisioning System 5.2 on unsupported operating systems only for testing purposes. Do not use the Sun N1 Service Provisioning System 5.2 on unsupported operating systems in a production environment.

Remote Installation of Remote Agents on Linux and UNIX Systems

You can install a Remote Agent remotely from another machine across the network. When you install the Master Server, the scripts needed to remotely install a Remote Agent are installed in the *N1SPS5.2-MasterServer-home*`/server/bin` directory. The installation is a non-interactive installation and uses environment variables to manage the installation and configuration of the Remote Agents. You can set the environment variables in a parameters file, at the command line, or use the default values provided by the installation script.

▼ How to Remotely Install Remote Agents on Linux and UNIX Systems

The installation scripts for the Remote Agent are specific to the operating system of the target machine. You need to complete these steps for each operating system running on your target machines.

Before You Begin The target machine must meet the following requirements:

- The UNIX utility `sshd` must be running and have direct IP connectivity to the source machine.
- Support for the UNIX `hostname` command must exist, so that the remote installation script can call this command. The Remote Agent must be configured to listen on the IP address of the host name returned by the `hostname` command.

The Master Server machine must have the UNIX utilities `ssh` and `scp` installed and in the path at the time of execution.

The remote installation program uses environment variables to manage the installation and configuration of the Remote Agent. You can set the environment variables in a parameters file or at the command line. You must declare a value for the following environment variables:

- `CR_RA_CTYPE=raw` – The Remote Agent connects to the Master Server or Local Distributor using no encryption. Valid values are `raw`, `ssh`, and `ssl`.
- `CR_RA_SUID` – Installs the Remote Agent with setuid root privilege. To specify a value of yes, you must run the installation script as the root user. Valid values are `y` and `n`.

  The value of the `CR_RA_SUID` variable can be `y`, if and only if the connection type is ssh. If this condition is not met, the installer terminates.
- `CR_RA_INSTALLER_HOSTS` – If you do not supply host names on the command line or as an environment variable, the installation script exits with an error. An example of a valid value is: `CR_RA_INSTALLER_HOSTS=host1,host3.enterprise.com,10.10.0.207`.

1 On the Master Server machine, access the installation scripts.

- If you are installing from a DVD, insert the DVD:
- If you are installing from the image that you downloaded, change to the directory where you saved the downloaded image.

2 Change to the directory on the software DVD or within the downloaded image where the installation script is located.

`% cd /`*script-directory*

script-directory is one of the following values:

- `solaris_sparc`
- `solaris_x86`
- `aix`
- `linux`
- `hpux`

3 Copy the installation script to the Master Server.

`% cp cr_ra_`*opsystem*`_5.2.sh` *N1SPS5.2-MasterServer-home*`/server/bin`

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server and *opsystem* is one of the following values:

- solaris_sparc – installs the Remote Agent on SPARC based hardware running the Solaris OS
- solaris_x86 – installs the Remote Agent on x86 based hardware running the Solaris OS
- aix – installs the Remote Agent on IBM AIX
- linux_i386 – installs the Remote Agent on Red Hat Linux and SUSE Linux Enterprise Server (32–bit)
- linux_x86_64 – installs the Remote Agent on Red Hat Linux and SUSE Linux Enterprise Server (64–bit)
- hpux – installs the application on PA-RISC based systems that are running HP-UX

4 Change directories to where the scripts are located.

`% cd` *N1SPS5.2-MasterServer-home*`/server/bin`

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

5 Determine how to provide configuration information to the installation script.

- Create a new parameters file or edit the sample parameters file that was installed by the Sun N1 Service Provisioning System 5.2. When you install the Master Server, a parameters file is installed. The file is named *N1SPS5.2-MasterServer-home*`/server/bin/cr_ra_remote_params.sh`. You can use the default values that are provided in this file or edit the file and add your custom values. You can also create a new parameters file to use. The contents of the sample parameters file and descriptions of the variables that you can set are in "Sample Remote Agent Parameters File for Linux and UNIX Systems" on page 120. The parameters file must be an executable file.

Caution – The values for the `CR_RA_SUID` and `CR_RA_CTYPE` variables are not set in the sample parameters file. You must set values for these variables or the installation script will abort. Also, if you choose to use an insecure connection type, such as `raw` or SSL encryption with no authentication, you must set the `CR_RA_CTYPE_CONFIRM` variable value to `true`.

The value of the `CR_RA_SUID` variable can be `y`, if and only if the connection type is ssh. If this condition is not met, the installer terminates.

- Set the environment variables.

```
% export CR_RA_INSTALLER_USER=username
% export CR_RA_INSTALLER_WORKDIR=/working_directory
% export CR_RA_INSTALLER_LEAVEFILES=yes, no
% export CR_RA_CTYPE=raw, ssh, ssl
% export CR_RA_SUID=y, n
% export CR_RA_INSTALLER_HOSTS=hostnames.enterprise.com,10.10.0.207
```

6 Start the remote installation.

```
% cr_ra_remote.sh -paramfile path-to-file/parameters-file.sh –f
cr_ra_opsystem_5.2.sh hostnames
```

- `cr_ra_`*opsystem*`_5.2.sh` is the installation script that you copied from the Sun N1 Service Provisioning System 5.2 CD or downloaded image.
- *path-to-file/parameters-file* is the path to the parameters file and the name of the parameters file that you want the installation program to use to obtain the configuration information. If you set the environment variables or you want the installation script to use the default values, you do not need to specify a parameters file.
- *hostnames* are the host names of the machines on which to perform the installation. Separate the hostnames by a space. If you specified the host names in the `CR_RA_INSTALLER_HOSTS` parameter, either in the parameters file or as an environment variable, you do not need to specify the hos tnames on the command line. If you specify host names on the command line, those hosts are installed and any hosts specified in the `CR_RA_INSTALLER_HOSTS` parameter are not be installed.

7 **Make a note of the location of the log file.**

The installation program notifies you that it is creating a log file and displays the location of the log file. Note the location of the file so that you can view it later.

8 **If prompted by the installation program, provide passwords for the remote machine.**

The installation script generates log files on the remote machine.

Starting Applications on Linux and UNIX Systems

The following table lists the commands to start the Sun N1 Service Provisioning System 5.2 applications on Linux and UNIX systems. *N1SPS5.2-home* is the home directory of the application.

Caution – Do not use the Bourne shell to start the Master Server or other Sun N1 Service Provisioning System 5.2 applications. If you start the Master Server process using the `cr_server start` command in a Bourne shell, and if a `^C` command is issued to any subsequent command in the same shell that started the Master Server, then the database and Master Server processes stop.

In the *N1SPS5.2-home*`/server/bin/roxdb.out` file, the following messages appear as the most recent entries:

```
DEBUG:  fast shutdown request
DEBUG:  aborting any active transactions
```

TABLE 4–1 Start Commands for Linux and UNIX System Applications

Application	Path to Command	Command to Start
Master Server	*N1SPS5.2-home*`/server/bin/`	`cr_server start`
Local Distributor	*N1SPS5.2-home*`/ld/bin/`	`cr_ld start`
Remote Agent	*N1SPS5.2-home*`/agent/bin/`	`cr_agent start`
CLI Client	*N1SPS5.2-home*`/cli/bin/`	`cr_cli` *CLI-command*
Jython version of CLI Client	*N1SPS5.2-home*`/cli/bin/`	`cr_clij` *CLI-command*

◆ ◆ ◆ CHAPTER 5

Installing the Sun N1 Service Provisioning System 5.2 on Windows Systems

This chapter describes the steps to install the Sun N1 Service Provisioning System 5.2 on servers running Windows. You will install each of the applications separately by using the appropriate Microsoft Installer (MSI) package on the product media. The MSI packages for each of the Sun N1 Service Provisioning System 5.2 applications begin by performing the same set of preparatory tasks and asking the same questions about directories and files. Each MSI package then asks specific configuration questions about the application that it will install.

This chapter discusses the following topics:

- "Installing the Master Server" on page 53
- "Installing the Remote Agent, Local Distributor, and CLI Client" on page 56
- "Non-Interactive Installation of a Remote Agent on Windows" on page 57
- "Remote Installation of Remote Agents on Windows" on page 59
- "Remote Agent Variable Values" on page 61
- "Starting Applications on Windows Systems" on page 62

Installing the Master Server

▼ How to Install the Sun N1 Service Provisioning System 5.2 Master Server on Windows

Before You Begin Review the installation process overview in "Installing the Sun N1 Service Provisioning System 5.2 – Process Overview" on page 17. Complete any necessary tasks prior to installing the Master Server.

If you want to install the Master Server remotely, use the Virtual Network Computing (VNC) software to access the server. If you use Terminal Client Services to access the server, the installation fails.

Ensure that you have write permissions on the folder in which the MSI packages are saved.

1. **Access the MSI packages.**

   - If you are installing from a CD, insert the Sun N1 Service Provisioning System 5.2: Windows CD.
   - If you are installing from the image that you downloaded, change to the directory where you saved the downloaded image.

2. **Use the Windows File Manager or a Command Prompt to access the `windows` directory on the CD or in the directory where you saved the downloaded image.**

3. **Start the Master Server installation.**

   - If you are using the File Manager, double-click the `cr_ms_win32_5.2.msi` file.
   - If you are at a Command Prompt, type the name of the installation file at the prompt.

     E:\N1GSPS5.2\windows> **`cr_ms_win32_5.2.msi [ALLOWFORWARDVERSION=true]`**

     The `ALLOWFORWARDVERSION=true` option enables you to install an Sun N1 Service Provisioning System 5.2 Master Server on a version of an operating system that is numerically higher than the highest version the Sun N1 Service Provisioning System 5.2 supports for that operating system. If you use the `ALLOWFORWARDVERSION=true` option, the installation program does not verify that the operating system on which you are installing the Master Server is supported. There is no standard Sun Services support for use of the Sun N1 Service Provisioning System 5.2 on unsupported operating systems.

Caution – Installing the Sun N1 Service Provisioning System 5.2 on unsupported operating systems might result in undefined and unexpected behavior. Install the Sun N1 Service Provisioning System 5.2 on unsupported operating systems only for testing purposes. Do not use the Sun N1 Service Provisioning System 5.2 on unsupported operating systems in a production environment.

4. **Answer the configuration questions when prompted by the installation program.**

   The installation program prompts you to answer a series of configuration questions and then displays the Ready to Install screen.

5. **Click Install to begin the installation.**

   The installation program installs the program files. When the installation completes, the installation program prompts you to restart the machine.

6. **Restart the machine to complete the installation.**

   You must restart the machine to complete the installation of the Sun N1 Service Provisioning System 5.2.

7. **Log in to the server.**

   After you log in, the installation program displays a Welcome screen.

8 **Click Next to complete the installation.**

Note – The installer opens Command Prompt windows and executes commands. Some of the commands might take several minutes to run. Do not close the Command Prompt windows or cancel the operations. The operations complete automatically after a few minutes.

---

9 **Click Finish to exit the installation program.**

The Master Server is installed. Access the Master Server by using your web browser and the browser interface address that you specified during the installation.

10 **(Optional) Create a scheduled task to optimize the database.**

To optimize the performance of your database, create a scheduled task that runs the `vacuumdb` utility daily. To create the scheduled task, follow the instructions in "How to Create a Scheduled Task to Optimize the Database" on page 55.

▼ How to Create a Scheduled Task to Optimize the Database

1 **Open the Windows Scheduled Tasks Folder.**

You can open the Scheduled tasks folder by clicking the Start menu, then clicking Programs -> Accessories -> System Tools -> Scheduled Tasks.

2 **To create a new task, right click in the folder and select New -> Scheduled Task.**

3 **Name the task.**

4 **Double–click on the task to edit it.**

5 **In the Run field, type the following command on a single line:**

`bash -c "/cygdrive/`*`c/Program\ Files/N1\ Service\ Provisioning\ System/5.2`***`/server/bin/roxdbcmd vacuumdb -h localhost -a -z"`**

c/Program\ Files/N1\ Service\ Provisioning\ System/5.2 is the directory in which you installed the Master Server.

6 **In the Schedule tab, configure the task to run once a day.**

Installing the Remote Agent, Local Distributor, and CLI Client

▼ How to Install the Remote Agent, Local Distributor, and CLI Client on Windows

Before You Begin Review the installation process overview in "Installing the Sun N1 Service Provisioning System 5.2 – Process Overview" on page 17. Complete any necessary tasks prior to installing the Master Server.

Ensure that you have write permissions on the folder in which the MSI packages are saved.

1 Access the installation MSI packages.

- If you are installing from a CD, insert the Sun N1 Service Provisioning System 5.2: Windows CD.
- If you are installing from the image that you downloaded, change to the directory where you saved the downloaded image.

2 Use the Windows File Manager or a Command Prompt to access the `windows` directory on the CD or in the directory where you saved the downloaded image.

3 Start the installation for the application you want to install.

- If you are using the File Manager, double-click the `cr_`*app*`_win32_5.2.msi` file.
- If you are in a Command Prompt, type the name of the installation file at the prompt.

  E:\N1GSPS5.2\windows> **`cr_`*app*`_win32_5.2.msi [ALLOWFORWARDVERSION=true]`**

  app is one of the following values:

  - ra – installs the Remote Agent
  - ld – installs the Local Distributor
  - cli – installs the CLI Client

  The `ALLOWFORWARDVERSION=true` option enables you to install an Sun N1 Service Provisioning System 5.2 application on a version of an operating system that is numerically higher than the highest version the Sun N1 Service Provisioning System 5.2 supports for that operating system. If you use the `ALLOWFORWARDVERSION=true` option, the installation program does not verify that the operating system on which you are installing the application is supported. There is no standard Sun Services support for use of the Sun N1 Service Provisioning System 5.2 on unsupported operating systems.

Caution – Installing the Sun N1 Service Provisioning System 5.2 on unsupported operating systems might result in undefined and unexpected behavior. Install the Sun N1 Service Provisioning System 5.2 on unsupported operating systems only for testing purposes. Do not use the Sun N1 Service Provisioning System 5.2 on unsupported operating systems in a production environment.

4. **Answer the configuration questions when prompted by the installation program.**

   The installation program prompts you to answer a series of configuration questions and then displays the Ready to Install screen.

5. **Click Install to begin the installation.**

   The installation program installs the program files.

6. **Click Finish to exit the installation program.**

Non-Interactive Installation of a Remote Agent on Windows

You can install the Remote Agent by using variables on a command line to indicate your configuration selections. The non-interactive installation for Remote Agents is accomplished by using the `msiexec` command that is installed as part of the Windows Installer Service.

▼ How to Non-Interactively Install Remote Agents on Windows

Before You Begin Ensure that you have write permissions on the folder in which the MSI packages are saved.

1. **On the machine where you want to install the Remote Agent, open a Command Prompt window.**

2. **Insert the Sun N1 Service Provisioning System 5.2: Windows CD.**

3. **Change to the directory where the MSI package is located, either on the software CD or from the directory where you saved the downloaded image.**

4. **Copy the installation MSI package to the machine on which you want to install the Remote Agent.**

   `% copy cr_ra_win`*address-space*`_5.2.msi` *RA-machine*`\`

   RA-machine is a directory on the machine on which you want to install the Remote Agent.

5. **Start the installation.**

`C:`*RA-machine*`\>` **`msiexec /i cr_ra_win`***address-space*`_5.2.msi /qn`**
`RA_PARENT_CONNECTION=`*value* **`CR_RA_CTYPE_CONFIRM=`***value* *VARIABLE*`=`*value*`[ALLOWFORWARDVERSION=true]`**

address-space is one of the following values:

- 32
- 64

You may include as many variables as necessary. Variable values that contain spaces, such as directory names, must be included in quotation marks. For the variables and values accepted by the non-interactive installation program, refer to Table 5–1. For variables for which you do not specify a value, the installation program installs the Remote Agent using the default value.

Caution – You must set a value for the `RA_PARENT_CONNECTION` variable or the installation script will abort. Also, if you choose to use an insecure connection type, such as `raw` or SSL encryption with no authentication, you must set the `CR_RA_CTYPE_CONFIRM` variable value to `true`.

The `ALLOWFORWARDVERSION=true` option enables you to install an Sun N1 Service Provisioning System 5.2 Remote Agent on a version of an operating system that is numerically higher than the highest version the Sun N1 Service Provisioning System 5.2 supports for that operating system. If you use the `ALLOWFORWARDVERSION=true` option, the installation program does not verify that the operating system on which you are installing the Remote Agent is supported. There is no standard Sun Services support for use of the Sun N1 Service Provisioning System 5.2 on unsupported operating systems.

Caution – Installing the Sun N1 Service Provisioning System 5.2 on unsupported operating systems might result in undefined and unexpected behavior. Install the Sun N1 Service Provisioning System 5.2 on unsupported operating systems only for testing purposes. Do not use the Sun N1 Service Provisioning System 5.2 on unsupported operating systems in a production environment.

Example 5–1 Non-Interactive Installation of a Remote Agent on Windows

The following example is a sample of the command to install the Remote Agent non-interactively on Windows.

```
C:\> msiexec /i cr_ra_winaddress-space_5.2.msi /qn
INSTALLDIR="C:\Program Files\N1 Service Provisioning System\"
RA_PARENT_CONNECTION=false CR_RA_CTYPE_CONFIRM=true
```

address-space is one of the following values:

- 32
- 64

Remote Installation of Remote Agents on Windows

The Remote Agent MSI package facilitates remote installation in a non-interacitve mode. The installation is accomplished by using a `.wsh` script used by the Windows Scripting Host. The script file contains VB script code that does the following:

- Attaches to the Remote Systems WMI DCOM interface
- Uses WMI to create a temporary Windows file share on the target server
- Copies `cr_ra_win32_5.2.msi` from the local location to the target share
- Uses WMI remotely to run the silent MSI on the target machine

▼ How to Remotely Install Remote Agents on Windows

Before You Begin Ensure that you have write permissions on the folder in which the MSI packages are saved.

1 On the Master Server machine, open a Command Prompt window.

2 Access the MSI packages.

- If you are installing from a CD, insert the Sun N1 Service Provisioning System 5.2: Windows CD.
- If you are installing from the image that you downloaded, change to the directory where you saved the downloaded image.

3 Change to the directory where the MSI package is located, either on the software CD or from the directory where you saved the downloaded image.

4 Copy the MSI package to the Master Server.

C:\> **copy cr_ra_win***address-space*_**5.2.msi** *MS-machine*\

MS-machine is a directory on the Master Server machine.

address-space is one of the following values:

- 32
- 64

5 Change to the Master Server home directory.

C:\> **cd** *N1SPS5.2-home*\server\bin\WinInstaller

N1SPS5.2-home is the directory in which you installed the Master Server.

6 Start the installation.

C:*MS-machine*> **cscript WinInstaller.wsf [ALLOWFORWARDVERSION=true]** *parameters Hostname*

The `ALLOWFORWARDVERSION=true` option enables you to install an Sun N1 Service Provisioning System 5.2 Remote Agent on a version of an operating system that is numerically higher than the highest version the Sun N1 Service Provisioning System 5.2 supports for that operating system. If

you use the `ALLOWFORWARDVERSION=true` option, the installation program does not verify that the operating system on which you are installing the Remote Agent is supported. There is no standard Sun Services support for use of the Sun N1 Service Provisioning System 5.2 on unsupported operating systems.

Caution – Installing the Sun N1 Service Provisioning System 5.2 on unsupported operating systems might result in undefined and unexpected behavior. Install the Sun N1 Service Provisioning System 5.2 on unsupported operating systems only for testing purposes. Do not use the Sun N1 Service Provisioning System 5.2 on unsupported operating systems in a production environment.

Hostname is the hostname of the machine on which to install the Remote Agent.

If you do not include values for any of the following *parameters* on the command line, the installation program installs the Remote Agent using the default values as shown below.

The Remote Agent non-interactive installation program accepts the parameters that are listed in the following table.

Parameter	Description	Default
`-user`	User to connect to WMI on the target machine.	None
`-password`	Password to connect to the WMI on the target machine.	None
variables	The Windows variable for the `cscript WinInstaller.wsf` command found in Table 5–1. All variables and values must be contained in a string that is enclosed by quotation marks. **Caution –** You must set a value for the `RA_PARENT_CONNECTION`variable or the installation script will abort. Also, if you choose to use an insecure connection type, such as `raw` or SSL encryption with no authentication, you must set the `CR_RA_CTYPE_CONFIRM` variable value to `true`.	None
`-msiLocation`	Paths to the `.msi\.input` files to install.	Current working directory
`-shareLocation`	An existing directory on the target machine in which to create a temporary Windows file share. The file share directory must be at least the size of the MSI package.	`C:\WINNT\Temp`

The exit code is 0 for a successful installation and 1 for a failure.

Example 5–2 Remote Installation of a Remote Agent On Windows

The following example is a sample of the command to remotely install a Remote Agent on Windows.

```
C:\> cscript WinInstaller.wsf -shareLocation C:\installs -options
"RA_PARENT_CONNECTION=false CR_RA_CTYPE_CONFIRM=true" targetHost
```

Remote Agent Variable Values

The Remote Agent non-interactive and remote installation programs accepts the following variables.

TABLE 5–1 Remote Agent Variable Values

Variable Name	Description	Default	Values
`INSTALLDIR`	Specify the directory in which to install the Remote Agent.	`C:\Program Files\N1 Service Provisioning System`	Any valid directory.
`REMOTE_AGENT_HOSTNAME`	Specify the hostname or IP address for the machine on which to install the Remote Agent.	The Windows computer name	Any valid hostname or IP address.
`RA_PORT_NUMBER`	Specify the IP Port number to use for this Remote Agent.	1131	Any valid port number.
`RA_PARENT_CONNECTION`	Specify that the parent application connects to this Remote Agent by using unencrypted (raw) or SSL connections.	none	`true` specifies to use SSL. `false` specifies to use raw.
`RA_SSL_CIPHER`	If you selected SSL, specify the type of SSL cipher suite to use.	1	`0` specifies to use SSL encryption with authentication. `1` specifies to use SSL encryption without authentication.
`CR_RA_CTYPE_CONFIRM`	Confirm the type of encryption to use.	none	`true` if you selected raw or SSLencryption without authentication. `false` if you selected SSL encryption with authentication.
`RA_SERVICE_USERNAME` `RA_SERVICE_PASSWORD`	Specify which user account the Remote Agent is to run as.	system user	Use a prefix of `.\` for local user names. If you define these variables, you must set `RA_SERVICE_CONTROL` to `other`.

TABLE 5–1 Remote Agent Variable Values *(Continued)*

Variable Name	Description	Default	Values
RA_SERVICE_AUTOSTART	Specify whether to start the Remote Agent automatically on server restart. The variable also determines whether the Remote Agent is started at the time of the installation.	1	1 specifies to start automatically. 0 specifies to not start automatically.

Starting Applications on Windows Systems

On Windows servers, you start the Master Server, Local Distributor, and Remote agent in the Services Panel. You start the CLI Client from a Command Prompt window.

To start the Master Server, Local Distributor, or Remote Agent, click the Start menu, then Programs -> Administrative Tools -> Services. In the Services panel, find the name of the application and start it.

TABLE 5–2 Names of Services to Start for the Windows Master Server, Local Distributor, and Remote Agent

Application	Name of Service to Start
Master Server	Sun N1 Service Provisioning System 5.2 Server
	Sun N1 Service Provisioning System 5.2 PostgresSQL Server
	Sun N1 Service Provisioning System 5.2 IPC Daemon
	Sun N1 Service Provisioning System 5.2 Database Preparer
Local Distributor	Sun N1 Service Provisioning System 5.2 Distributor
Remote Agent	Sun N1 Service Provisioning System 5.2 Agent

To start the CLI Client on a Windows server, type one of the following commands at a Command Prompt. *N1SPS5.2-home* is the home directory of the application.

TABLE 5–3 Start Commands for the Windows CLI Client

Application	Path to Command	Command to Start
CLI Client	*N1SPS5.2-home*\cli\bin\	cr_cli.cmd *CLI-command*
Jython version of CLI Client	*N1SPS5.2-home*\cli\bin\	cr_clij.cmd *CLI-command*

◆ ◆ ◆ CHAPTER 6

Configuring the Sun N1 Service Provisioning System 5.2 for HTTPS

The Sun N1 Service Provisioning System installation program prompts you to select for the browser interface to use HTTP or HTTPS to connect to the Master Server. If you choose HTTPS, the installation program prompts you to enter the keystore file and keystore password. The installation program offers two options for providing the keystore file and the keystore password:

- **Create keystore file and keystore password before installation:** If you create a keystore file and a keystore password before you install the provisioning system, then you can provide the keystore file location and keystore password information during installation. For instructions on how to create a keystore file and a keystore password before you install the provisioning system, see "Creating a Keystore File and Keystore Password for HTTPS Connections" on page 64.

  Note – The installation program does not verify that the keystore password is valid. If you enter an invalid password, you might not be able to start the Master Server.

- **Create keystore file and keystore password after installation:** If you do not create a keystore file and a keystore password before installation, you cannot provide the keystore file and keystore password information during the installation. Consequently, the installation script creates an empty keystore file in the *N1SPS5.2-home*`/server/tomcat` directory. You must manually replace the empty keystore file and keystore password information before you can start the Master Server. For instructions, see "Configuring HTTPS After Installation" on page 65.

If you select HTTP during installation, you can manually configure the Master Server to use HTTPS. For instructions, see "Configuring HTTPS After Selecting HTTP During Installation" on page 67.

If you have configured the provisioning system to use HTTPS, you can manually reconfigure the provisioning system to use HTTP. For instructions see "Reverting to HTTP" on page 68.

Creating a Keystore File and Keystore Password for HTTPS Connections

HTTPS requires an SSL Certificate. When you generate an SSL Certificate, you are creating a keystore file and a keystore password for use when the browser interface connects to the Master Server.

You can choose whether to have a Certifying Authority sign the certificate or you can use a self-signed certificate. A certificate signed by a Certifying Authority is trusted by browsers, therefore the browser does not issue a warning when a user connects to the browser interface on the Master Server. Generally, Certifying Authorities charge a fee to sign a certificate. A self-signed certificate is available for use immediately after you generate the certificate because you do not have to wait for the Certifying Authority to sign it. However, a self-signed certificate is not trusted by the browser, so the browser issues a warning each time a user connects to the Master Server.

▼ How to Generate SSL Certificates

To enable the browser interface to use SSL, you must first generate an SSL Certificate. You create a keystore file and a keystore password while you generate an SSL Certificate.

Before You Begin To create a keystore file and a keystore password you will use `keytool`. `keytool` is a security tool available with the JRE. If you do not have `keytool` installed, you must install `keytool` before you can configure the provisioning system to use HTTPS. The Sun N1 Service Provisioning System installs the JRE. If you are configuring HTTPS after you have installed the provisioning system, `keytool` is installed on the system.

1 Change to the directory in which you installed the JRE.

```
% cd JAVA-HOME/bin
```

JAVA-HOME is the directory where you installed the JRE. If you installed the JRE with the Sun N1 Service Provisioning System 5.2, the JRE is installed in the *N1SPS5.2-home*`/common/jre/bin` directory.

2 Generate the certificate.

```
% keytool -genkey -alias tomcat -keyalg RSA -keystore /keystore-location
-storepass password
```

Set */keystore-location* to the location and filename of the keystore file where you want to store the generated key.

Set *password* to whatever password that you want to use as the keystore password.

3 Follow the prompts to complete.

Do not include any punctuation in the name of your organization, otherwise the Java Certificate tool fails when attempting to generate the request. The Common Name (CN) must be set to the fully qualified host name, including the domain name, component of the URI.

▼ How to Obtain a Signature for an SSL Certificate

If you want to use a Certificate signed by a Certifying Authority, follow this procedure to submit the Certificate to the Certifying Authority to be signed.

1 Generate the Certificate Request.

```
% keytool -certreq -v -alias tomcat -keyalg RSA -keystore /keystore-location
```

/keystore-location is the location and filename where you stored the generated key.

2 Send the Certificate Request to the Certifying Authority.

Follow the instructions provided by the Certifying Authority. The Certifying Authority returns a Certificate Reply.

3 Save the Certificate Reply to a file.

4 Verify the Certificate Reply.

```
% keytool -printcert -file certificate-reply-file
```

certificate-reply-file is the filename of the Certificate Reply that you received from the Certifying Authority.

5 Import the Certificate Reply file to the `keystore` file.

```
% keytool -v -import -trustcacerts -keystore /keystore-location
-file certificate-reply-file -alias tomcat
```

/keystore-location is the location and filename where you stored the generated key.
certificate-reply-file is the filename of the Certificate Reply that you received from the Certifying Authority.

6 Verify the imported Certificate Reply.

```
% keytool -v -list -keystore /keystore-location
```

/keystore-location is the location and filename where you stored the generated key.

Configuring HTTPS After Installation

The Sun N1 Service Provisioning System installation program prompts you to choose to select HTTPS or HTTP for the browser interface to use to connect to the web interface of the Master Server. If you selected HTTPS during the installation, but you did not supply a keystore file and a keystore password during the installation you must manually copy the keystore file to the provisioning system keystore directory. You must also create an encoded keystore password and configure the password in the `server.xml` file.

How to Copy the Keystore File

Complete the following steps to copy the keystore file to the *N1SPS5.2-MasterServer-home*`/server/tomcat/keystore` file.

Before You Begin Generate an SSL Certificate, and obtain a signature from a Certifying Authority if necessary. For instructions, see "Creating a Keystore File and Keystore Password for HTTPS Connections" on page 64.

1 Move the keystore file to the Master Server home directory.

```
%mv /keystore-location N1SPS5.2-MasterServer-home/server/tomcat/keystore
```

/keystore-location is the location and filename where you stored the generated key. *N1SPS5.2-MasterServer-home* is the directory where you installed the Master Server.

2 Change to the directory where you moved the keystore file.

```
% cd N1SPS5.2-MasterServer-home/server/tomcat/
```

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

3 Set the ownership and permissions on the keystore file:

```
%chmod 600 keystore
```

```
%chown MS_user:MS_group keystore
```

MS_user is the user that owns the Master Server application. *MS_group* is the group that owns the Master Server application.

How to Create and Configure an Encoded Keystore Password

When you generate an SSL certificate, you create a keystore file and you supply a keystore password. The provisioning system must store the keystore password in the `server.xml` file. For added security, the provisioning system requires an encoded version of the password to store in the `server.xml` file.

1 Change to the directory on the Master Server that contains the `crkeys` command.

```
%cd N1SPS5.2-MasterServer-home/server/bin
```

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

2 Create an encoded version of the password that you created when you created the keystore file.

```
% crkeys -epass -password password
```

password is the password that you created while generating the SSL certificate.

The `crkeys` tool prints the encoded password on the screen. Make note of the encoded password.

3 **Change to the directory where the Tomcat configuration files are located.**

```
% cd /N1SPS5.2-MasterServer-home/server/tomcat/conf
```

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

4 **Edit the Factory element to include the encoded password.**

```
<Factory className="com.raplix.rolloutexpress.ui.web.EncodedPasswordSSLFactory"
          clientAuth="false" protocol="TLS"
    keystoreFile="/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/server/tomcat/keystore"
 keystorePass="ADD_ENCODED_PASSWORD_HERE"/>
```

Change `ADD_ENCODED_PASSWORD_HERE` to the encoded version of the password that was generated by using the `crkeys` tool.

Configuring HTTPS After Selecting HTTP During Installation

The Sun N1 Service Provisioning System installation program prompts you to choose to select HTTPS or HTTP for the browser interface to use to connect to the Master Server. If you selected HTTP during the installation, you can manually reconfigure the Sun N1 Service Provisioning System to use HTTPS.

▼ How to Enable HTTPS Connections from the Master Server Browser Interface to the Web Interface of the Master Server

Complete the following steps to configure the browser interface to use HTTPS instead of HTTP to connect to the Master Server.

Before You Begin Generate an SSL Certificate and obtain a signature from a Certifying Authority if necessary. For instructions, see "Creating a Keystore File and Keystore Password for HTTPS Connections" on page 64.

1 **Stop the Master Server.**

```
% N1SPS5.2-MasterServer-home/server/bin/cr_server stop
```

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

2 **Move the keystore file to the Master Server home directory.**

```
%mv /keystore-location N1SPS5.2-MasterServer-home/server/tomcat/keystore
```

/keystore-location is the location and filename where you stored the generated key.
N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

3. **Change to the directory where you moved the keystore file.**

   ```
   % cd N1SPS5.2-MasterServer-home/server/tomcat/
   ```

   N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

4. **Set the ownership and permissions on the keystore file:**

   ```
   %chmod 600 keystore
   ```

   ```
   %chown MS_user:MS_group keystore
   ```

   MS_user is the user that owns the Master Server application. *MS_group* is the group that owns the Master Server application. */keystore-location* is the filename where you stored the generated key.

5. **Change to the directory where the Tomcat configuration files are located.**

   ```
   % cd /N1SPS5.2-MasterServer-home/server/tomcat/conf
   ```

   N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

6. **In the `server.xml` file, uncomment the following lines. XML comments begin with `<!--` and end with `-->`.**

```
<Connector className="org.apache.catalina.connector.http.HttpConnector"
               port="8443" minProcessors="5" maxProcessors="75"
               enableLookups="true"
               acceptCount="10" debug="0" scheme="https" secure="true">
          <Factory className="com.raplix.rolloutexpress.ui.web.EncodedPasswordSSLFactory"
               clientAuth="false" protocol="TLS"/>
     </Connector>
```

7. **Edit the Factory element as follows.**

   ```
   <Factory className="com.raplix.rolloutexpress.ui.web.EncodedPasswordSSLFactory"
             clientAuth="false" protocol="TLS"
      keystoreFile="N1SPS5.2-MasterServer-home/server/tomcat/keystore"
    keystorePass="password"/>
   ```

   N1SPS5.2-MasterServer-home is the directory in which you installed the Master Server. *password* is the encoded version of the password.

Reverting to HTTP

If you have configured the provisioning system to use HTTPS, you can manually reconfigure the provisioning system to use HTTP.

▼ How to Revert to HTTP

1 Stop the Master Server

```
% N1SPS5.2-MasterServer-home/server/bin/cr_server stop
```

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

2 To return to the original configuration, replace the secure `web.xml` file with the default `web.xml` file.

```
% cd /N1SPS5.2-MasterServer-home/server/webapp/WEB-INF
% cp web.xml.default web.xml
```

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

3 Restart the Master Server.

```
% N1SPS5.2-MasterServer-home/server/bin/cr_server start
```

N1SPS5.2-MasterServer-home is the directory where you installed the Master Server.

◆ ◆ ◆ CHAPTER 7

Configuring the Sun N1 Service Provisioning System 5.2 to Use Secure Shell

This chapter contains instructions for configuring the Sun N1 Service Provisioning System 5.2 to communicate using Secure Shell (SSH).

The Sun N1 Service Provisioning System 5.2 supports OpenSSH 2.0 explicitly. OpenSSH 2.0 is a free version of SSH that has primarily been developed by the OpenBSD Project. For more details, see `http://www.openssh.com`. The software can be configured to support other versions of SSH.

Using the same implementation of SSH on each server in your network ensures that the keys are compatible and that the servers communicate properly. You may choose to use different implementations of SSH on servers in your network, but you must test to verify that the various implementations are compatible and interoperable.

Note – The commands and interface examples in this chapter apply to OpenSSH 2.0. If you are using a different version of SSH, refer to the documentation provided with that version of SSH to determine the commands and options that are equivalent to the commands used in OpenSSH 2.0. For details about the OpenSSH 2.0 commands and options used, refer to "OpenSSH 2.0 Command Reference" on page 86.

This chapter discusses the following topics:

- "Overview of SSH and Requirements" on page 72
- "Configuring SSH – Process Overview" on page 75
- "Preparing the Keys" on page 75
- "Setting Up and Testing the Connectivity on the Master Server" on page 78
- "Configuring SSH for the Applications" on page 81
- "SSH Advanced Parameters and Command Reference" on page 84

Overview of SSH and Requirements

SSH is a UNIX-based command suite and protocol for securely accessing a remote computer. SSH secures network client/server communications by authenticating both endpoints with a digital certificate and by encrypting passwords. SSH uses RSA public key cryptography to manage connections and authentication. SSH is more secure than telnet and other shell based communication methods, and is used to manage web servers and other remote servers.

Unlike the other connection types, when an SSH connection is set up between two Sun N1 Service Provisioning System 5.2 applications, the downstream application does not need to be manually started. The upstream application automatically starts the downstream application when it is needed. The downstream application remains running for the duration necessary and shuts down automatically when it is not used for a configurable period of time.

Do not manually start the downstream application for an SSH connection. For example, if you set up a Local Distributor to connect to an Remote Agent using SSH, do not manually start the Remote Agent. The Local Distributor automatically starts the Remote Agent when necessary. The Remote Agent continues to run for as long as it is being used. The Local Distributor will automatically shut down the Remote Agent when it has not been used for a configurable period of time.

`ssh-agent` or Empty Password Keys

You can configure SSH to use the `ssh-agent` or to use empty password keys . If you use empty password keys, the generated SSH private key is stored with an empty password. As a result, you do not need a password to access the key. When you use SSH to communicate with another machine that trusts its public key, you are not prompted for a password. When using the `ssh-agent`, the generated private key is stored with a secure password and saved on secure media. You communicate with another machine by starting the `ssh-agent`, uploading the private key from the secure media, and supplying the password. The private key is not stored on the file system, but is stored in the memory of the `ssh-agent` process.

- `Empty Password Keys`

  If you use empty password keys, the generated SSH private key is stored with an empty password. As a result, you do not need a password to access the key. When you use SSH to communicate with another machine that trusts its public key, you are not prompted for a password.

- ssh-agent

When using the `ssh-agent`, the private key is stored with the `ssh-agent` that is running only on the Master Server. The public key is distributed to other machines on the network. When an SSH application requires authentication, it communicates with the `ssh-agent` to authenticate. You must turn on `ssh-agent` forwarding when making intermediate SSH connections to enable Local Distributors to proxy to the `ssh-agent` that is running on the Master Server for authentication. `ssh-agent` forwarding allows Local Distributors to authenticate to Local Distributors and Remote Agents that are downstream. This approach provides more security. Also, configuring the `ssh-agent` for use with the Sun N1 Service Provisioning System 5.2 is less complicated than configuring empty passwords.

When using empty passwords, the private key is stored on the file system of the machine without a password. Also, the private key must be present on all machines that initiate SSH communications. In the case of the Sun N1 Service Provisioning System 5.2, all Master Servers and Local Distributors that are connecting to applications downstream using SSH are required to have a private key. This approach provides less security.

SSH Requirements

The Sun N1 Service Provisioning System 5.2 requires the following SSH capabilities:

- Remote command invocation through `ssh`
- Public-private key authentication
- Support for `BatchMode yes` interaction, which is the ability to invoke the `ssh` command without interaction from an operator

If you are using the `ssh-agent`, the following SSH capabilities are required:

- Support for `ssh-agent`.
- Support for `ssh-agent` forwarding in SSH. Use the `-A` option in Open SSH.

The following capabilities are helpful when configuring machines for SSH connectivity, but are not requirements:

- Force allocate a `tty` when doing remote command invocation. Use the `-t` option in OpenSSH.
- Kill the `ssh` agent. Use the `-k` option for the `ssh-agent` command in OpenSSH.
- Generate an RSA key for higher security. Use the `-t rsa` in OpenSSH.

Review the following checklist to determine whether an implementation of SSH meets the requirements of the Sun N1 Service Provisioning System 5.2.

- The `ssh-keygen` command must generate a public-private key pair that can be used for authenticating SSH invocations.
- On the specified host, without prompting for any extra information to exchange host keys, obtain a password, etc., when the private key used for authentication was created without a password or with an empty password, the `ssh` command must be able to execute the following:

  ```
  % ssh –o 'BatchMode yes' hostname
  ```

- After hopping from the current host to host1 to host2 to host3, on host3 with the `ssh-agent` running on the current host, uploaded with a private key created with a non-empty password, without prompting for any extra information to exchange host keys, obtain a password, etc., the `ssh` command must be able to execute the following:

  ```
  % ssh –o 'BatchMode yes' –A host1 ssh  –o 'BatchMode yes' –A host2
  ssh  –o 'BatchMode yes' host3
  ```

- The `ssh` command must be able to correctly pipe its own standard input, output, and error streams to the command being executed on the remote machine.

- The `ssh-add` command must be able to upload private keys with non-empty passwords into the `ssh-agent` so that the private keys can be used for authentication.

Additional SSH Security

When you invoke the Remote Agent through SSH, the Remote Agent uses the jexec wrapper to invoke the Java Virtual Machine. This wrapper is a native executable that is owned by root and that has the `setuid` bit set. This file has the same group ID as the user that you used to install the Remote Agent and it gives execute permission to the group. Additionally, the file is stored in a directory that is called `protect` that is owned by the user you used to install the Remote Agent. The file gives execute permission only to the user that owns the Remote Agent. This prevents any other user from being able to execute the jexec wrapper.

You must ensure that the file permissions on `jexec` and `protect` are not accidentally changed at any point.

To further tighten security for jexec, make any or all of the following changes:

- The JVM executables, usually shell scripts, must be owned by root or the user that owns the application and do not give write permissions to any other users or groups. If you install the JRE with the Sun N1 Service Provisioning System 5.2, ensure that all the files in *N1SPS5.2-home*`/common/jre` are owned by the user that owns the application and do not give write access to any other users or groups.
- The user ID of the user that owns the application must only be allowed to log in using SSH. When logging in using SSH, only public-key authentication should be allowed. The */N1SPS5.2-home*`/.ssh` directory should not give any permissions to any other users or groups.
- The SSH server can be configured to allow only public key authentication by ensuring that the `etc/sshd_config` file contains the following line to disable password authentication.

  ```
  PasswordAuthentication no
  ```

- Ensure that the `etc/sshd_config` file does not have lines that contain RhostsRSAAuthentication, because this is not allowed by default. Also, ensure that RSAAuthentication, if present, is set to `yes`, the default.
- You can further tighten security on the Remote Agent by editing the */N1SPS5.2-home*`/.ssh/authorized_keys2` file and prefixing the following text to the line that contains the public key of the Master Server.

  ```
  no-port-forwarding,no-X11-forwarding,no-agent-forwarding,no-pty
  ```

The `sshd(1M)` man page offers additional details.

Configuring SSH – Process Overview

The following process overview describes the tasks necessary to configure the Sun N1 Service Provisioning System 5.2 to use SSH.

1. Determine whether you want to use empty password keys or the `ssh-agent`.

   For more information about choosing an SSH security level, see "`ssh-agent` or Empty Password Keys" on page 72.

2. Generate the keys on the applications that initiate SSH connections.

   See "How to Generate Key Pairs" on page 75.

3. Copy the generated keys to the Local Distributors and the Remote Agents.

   From the following list, choose the appropriate task based on whether you are using empty password keys or the `ssh-agent`:

   - "How to Set Up Keys for the `ssh-agent`" on page 76
   - "How to Set Up Keys for Empty Password Files When Using One Key Pair" on page 77
   - "How to Set Up Keys for Empty Password Files When Using Multiple Key Pairs" on page 78

4. Set up the SSH connectivity. Then, test the connectivity before you start the Master Server.

   See "Setting Up and Testing the Connectivity on the Master Server" on page 78.

5. Configure the Local Distributors and Remote Agents to use SSH.

   See "How to Configure SSH for Local Distributors and Remote Agents" on page 81.

6. (Optional) If you have any CLI Clients, configure the clients to use SSH.

   See "How to Configure SSH for the CLI Client With the `ssh-agent`" on page 82.

Preparing the Keys

Generate the public-private key pair that will be used to authenticate communication from the Master Server to the Local Distributors and the Remote Agents. Then, copy the generated keys to the Local Distributors and Remote Agents. Choose the appropriate task to complete based on whether you are using the `ssh-agent` or empty password keys.

Note – The following instructions describe how to create keys with the default key lengths. For maximum security, create keys with the longest possible key lengths.

▼ How to Generate Key Pairs

If you are using the `ssh-agent`, you only need to generate one key pair. If you are using empty passwords, you may generate a key pair for each SSH connection that the software makes between two machines. Alternatively, you may generate one single key pair for use by all the connections. Complete this task for each key pair that you want to generate.

Before You Begin Ensure that the user ID and group ID that you are using for the Sun N1 Service Provisioning System 5.2 are the same on all of the servers in the network.

1 **On the Master Server, or if you are using empty passwords and are generating key pairs for each connection, on the machine that is upstream, generate the keys.**

```
% ssh-keygen –t rsa
```

The server prompts you to save the keys.

2 **Save the keys in the default locations by pressing Return.**

The private key is saved in */User-home/*`.ssh/id_rsa`. The public key is saved in */HOME/*`.ssh/id_rsa.pub`.

User-home is the home directory of the currently logged in user on the Master Server machine.

The server prompts you for a password.

3 **Determine whether you need to supply a password.**

- If you are using empty password keys, do not supply a password. Press Return to continue.
- If you are using the `ssh-agent`, supply a password for the keys.

▼ How to Set Up Keys for the `ssh-agent`

If you are using the `ssh-agent`, complete this task to copy the keys to the Local Distributors and Remote Agents.

1 **On the Master Server, copy the private key file, `~/.ssh/id_rsa`, to a secure media.**

```
% cp /User-home/.ssh/id_rsa path_to_file/
```

User-home is the home directory of the currently logged in user on the Master Server machine. *path_to_file/* is the path to the secure media where you want to save the private key file.

2 **Delete the private key file from the local file system.**

```
% rm /User-home/.ssh/id_rsa
```

3 **Copy the public key to each Local Distributor and Remote Agent that you want to set up to use SSH. Save the key in the `~/.ssh/authorized_keys2` file.**

```
% cp /User-home.ssh/id_rsa.pub /User-home-APP/.ssh/authorized_keys2
```

User-home is the home directory on the Master Server machine. *User-home-APP* is the home directory of the currently logged in user on the Local Distributor or the Remote Agent machine.

4 **Ensure that the `.ssh/` directory and any parent directories are not world writable.**

5 **Change the permissions for the `.ssh/authorized_keys2` file to 600.**

6 **Edit the following line in the `config.properties` files on the Master Server and the Local Distributors to enable `ssh-agent` forwarding.**

Current configuration:

```
net.ssh.args=-o|BatchMode yes
```

Edit the line to include the `-A` option:

```
net.ssh.args=-o|BatchMode yes|-A
```

How to Set Up Keys for Empty Password Files When Using One Key Pair

If you are using empty password files and you generated only one key pair, complete this task to copy the keys to the Local Distributors and Remote Agents.

1 **From the Master Server, copy the private key to each machine that is upstream. Save the key in the home directory.**

```
% cp /User-home/.ssh/id_rsa /User-home-upstream/.ssh/id_rsa
```

User-home is the home directory of the currently logged in user on the Master Server machine. *User-home-upstream* is the home directory on the machine that is upstream. The upstream machine is the machine that initiates the SSH connection with the machine that is downstream.

Each Local Distributor can have a unique private key, or you can use the same private key for all Local Distributors.

2 **Copy the public key to each machine that is downstream. Save the key in the `/.ssh/authorized_keys2` file.**

```
% cp /HOME-MS/.ssh/id_rsa.pub /HOME-downstream/.ssh/authorized_keys2
```

User-home is the home directory on the Master Server machine. *User-home-downstream* is the home directory on the Local Distributor or the Remote Agent machine to which the machine that you set up in the previous step will connect. Copy the public key to all Local Distributors and Remote Agents that connect using SSH.

3 **Ensure that the `.ssh/` directory and any parent directories are not world writable.**

4 **Ensure that the private key file, `.ssh/id_rsa`, is not accessible by other users or groups.**

5 **Change the permissions for the `.ssh/authorized_keys2` file to 600.**

▼ How to Set Up Keys for Empty Password Files When Using Multiple Key Pairs

If you are using empty password files and you generated a key pair for each SSH connection, complete this task to copy the keys to the Local Distributors and Remote Agents.

Before You Begin Complete this task for every SSH connection, therefore every key pair, that is made on the network.

1. **From the machine that is upstream, copy the public key to each machine that is downstream. Save the key in the** *User-home*`/.ssh/authorized_keys2` **file.**

   ```
   % cp /User-home-upstream/.ssh/id_rsa.pub /User-home-downstream/.ssh/authorized_keys2
   ```

   User-home-upstream is the home directory on the machine that is upstream. *Use-home-downstream* is the home directory on the Local Distributor or the Remote Agent machine to which the upstream machine will connect.

2. **Ensure that the** `.ssh/` **directory and any parent directories are not world writable.**

3. **Ensure that the private key file,** `.ssh/id_rsa`**, is not accessible by other users or groups.**

4. **Change the permissions for the** `.ssh/authorized_keys2` **file to 600.**

Setting Up and Testing the Connectivity on the Master Server

This section describes the initial setup and testing of SSH that must be done before you use SSH with the Sun N1 Service Provisioning System 5.2. If you are using the `ssh-agent`, you will need to start the `ssh-agent` before you begin the setup and testing task.

▼ How to Start the `ssh-agent` on the Master Server

Complete this task only if you are using the `ssh-agent`. Complete this task before you start the Master Server

---

Note – You must execute all the SSH setup commands in the following setup tasks, `ssh`, `ssh-add`, and `cr_server start`, in the same session as the session that you used to start the `ssh-agent`. If this session is terminated, you must kill the `ssh-agent` program that is running and start a new `ssh-agent` program.

---

1. **Start the** `ssh-agent`**.**

   ```
   % eval `ssh-agent`
   ```

The `ssh-agent` starts and sets two environment variables. `SSH_AUTH_SOCK` and `SSH_AGENT_PID` are used by `ssh` and `ssh-add` to connect to the `ssh-agent`.

2 Upload the private key that you generated.

```
% ssh-add path-to-file/
```

path-to-file/ is the path to the secure media where you saved the private key file.

You are prompted to provide a password.

3 Provide the password that you created when you generated the keys.

More Information

Shutting Down the `ssh-agent`

You can shut down the `ssh-agent` by running the command `eval `ssh-agent –k``.

This command uses the `SSH_AGENT_PID` variable to send a signal to the `ssh-agent` process to shut it down. The command also unsets the environment variables that were set when you started the `ssh-agent`.

▼ How to Set Up and Test the Connectivity on the Master Server

Before You Begin If you are using the `ssh-agent`, be sure to start the `ssh-agent` by following the instructions in "How to Start the `ssh-agent` on the Master Server" on page 78.

Caution – The setup is session sensitive, so you must execute all the SSH commands, `ssh`, `ssh-add`, and `cr_server start`, in the same session as the session that you used to start the `ssh-agent`. If this session is terminated, you must kill the `ssh-agent` program that is running and start a new `ssh-agent` program. You will also need to upload the private key.

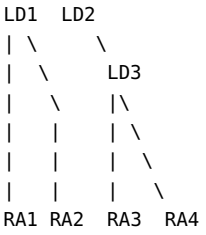
1 Test the SSH connection paths.

```
% ssh target-host-IP set
% ssh -A -t target-host-IP ls -l
```

Use the `-A` option only if you are using the `ssh-agent`. *target-host-IP* is the IP address for the machine to which this machine will connect.

For example, you might have a network setup with the following Master Server (MS), Local Distributors (LD1, LD2, and LD3), and Remote Agents (RA1, RA2, RA3, and RA4).

```
MS
 | \
 |  \
 |   \
```



For this example network, executing the following commands on the Master Server, substituting the IP addresses of the Local Distributors and Remote Agents on the network for LD1, LD2, RA1, RA2, RA3, and RA4 to test the SSH connection paths.

```
% ssh -A -t LD1 ssh -t RA1 set
% ssh -A -t LD1 ssh -t RA2 set
% ssh -A -t LD2 ssh -A -t LD3 ssh -t RA3 set
% ssh -A -t LD2 ssh -A -t LD3 ssh -t RA4 set
```

These commands follow the paths that the Master Server uses when using SSH to connect to the machines that are downstream. Each command enables SSH to exchange the host keys required for communicating to the machines specified as arguments.

SSH prompts you to allow the host key exchange.

2 Answer yes to each of the prompts.

3 Verify the output of all of the commands to ensure that the environment variables are correctly set up.

The `PATH` variable should have `/bin`, `/usr/bin`, and any other directories that are part of your environments.

4 Test the SSH connection paths again.

Use the same command that you used in Step 1 to test the connection paths again to ensure that the server does not prompt you for any information.

More Information

Repeating Set Up and Testing

If you change any of the keys, you might need to perform this task again. Depending upon your server setup, you also might need to complete this task again whenever you reboot any of the machines.

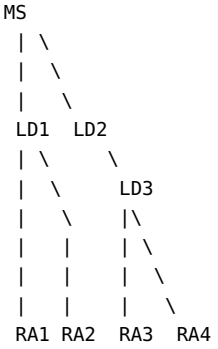
Configuring SSH for the Applications

After you set up and test SSH on the Master Server, configure the other machines in the Sun N1 Service Provisioning System 5.2 so that the Master Server can connect to them using SSH.

▼ How to Configure SSH for Local Distributors and Remote Agents

The SSH configuration has must be completed by following the Sun N1 Service Provisioning System 5.2 network from the Master Server to the Remote Agents and configuring the intermediate Local Distributors in the order in which you encounter them. Essentially, this is a preorder traversal of the tree network.

For example, you might have a network setup with the following Master Server (MS), Local Distributors (LD1, LD2, and LD3), and Remote Agents (RA1, RA2, RA3, and RA4).



Configure your network in the following order: LD1, RA1, RA2, LD2, LD3, RA3, RA4. Follow this order strictly and complete the configuration of one machine before moving on to the next machine.

1. **Use the Master Server browser interface to view the Host Details page for the machine you want to configure.**

2. **Add the connection details in the Local Distributor or the Remote Agent section depending on what application you are configuring on that machine.**

3. **Specify the connection type as ssh.**

4. **Add the following text in the Advanced Parameters field.**

   `cprefix=`*`/N1SPS5.2-Home/application`*

   N1SPS5.2–Home is the home directory of the application. *application* is agent if you are configuring a Remote Agent or ld if you are configuring a Local Distributor.

For example, if the Sun N1 Service Provisioning System 5.2 is installed in `/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/` and you are configuring a Remote Agent, the text you add to the Advanced Parameters field is:

```
cprefix=/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/agent
```

5 **Save the Host Details.**

6 **Ensure that you do not have a Remote Agent or Local Distributor instance running on this machine.**

7 **Click Test Connection on the Host Details page for this application instance.**

8 **Repeat this task for each machine in your network.**

▼ How to Configure SSH for the CLI Client With the `ssh-agent`

Complete this task if you want to use SSH connectivity for the CLI Client with the `ssh-agent`.

1 **Create a new operating system user account on the Master Server and the machine on which the CLI Client is installed.**

This account should be different from the account that you specified during the installation of the Master Server, Local Distributor, or Remote Agent.

2 **Log in to the Master Server as the new user that you created in the previous step.**

3 **Generate public and private keys for the new user by following the instructions in "How to Generate Key Pairs" on page 75.**

Do not reuse the keys that you generated for communication between the Master Server, Local Distributors, and Remote Agents.

4 **On the Master Server, copy the private key file to a secure media.**

```
% cp /User-home/.ssh/id_rsa path-to-file/.ssh/id_rsa
```

User-home is the home directory of the currently logged in user on the Master Server machine. *path-to-file/* is the path to the secure media where you want to save the private key file.

5 **Delete the private key file from the local file system.**

```
% rm /User-home/.ssh/id_rsa
```

6 **On the Master Server, concatenate the public key to the `/.ssh/authorized_keys2` file for that user.**

```
% cat /User-home/.ssh/id_rsa.pub >> /HOME-MS/.ssh/authorized_keys2
```

User-home is the home directory on the Master Server machine.

7 **Log in to the CLI Client machine as the new user that you created.**

8 **Start the** `ssh-agent`**.**

```
% ssh-agent > /User-home/.ssh/agent_vars
```

User-home is the home directory of the currently logged in user on the CLI Client machine.

9 **Add the following line to the** `.profile`**, the** `.cshrc`**, or the** `.bash_profile` **file.**

```
. /User-home/.ssh/agent_vars
```

User-home is the home directory on the CLI Client machine.

10 **Log out of the Master Server and log back in.**

11 **Upload the private key that you generated.**

```
% ssh-add path-to-file/
```

path-to-file/ is the path to the secure media where you saved the private key file.

The CLI Client now uses SSH and the `ssh-agent` for authentication when connecting to the Master Server.

12 **Configure the Master Server to accept only connections from localhost. For instructions, see** **"Configuring the JVM Security Policy" on page 101****.**

More Information

Stopping the `ssh-agent`

Note – If you want to stop the `ssh-agent`, on the CLI Client, use the following command:

```
% eval `ssh-agent –k >User-home/.ssh/agent_vars`
```

User-home is the home directory of the currently logged in user on the CLI Client machine.

▼ How to Configure SSH for the CLI Client With Empty Passwords

Complete this task if you want to use SSH connectivity for the CLI Client with empty passwords.

1 **Create a new operating system user account on the Master Server and the machine on which the CLI Client is installed.**

This account should be different from the account that you specified during the installation of the Master Server, Local Distributor, or Remote Agent.

2 **Log in to the CLI Client machine as the new user that you created in the previous Step.**

3. **Generate public and private keys for the new user by following the instructions in "How to Generate Key Pairs" on page 75.**

   Do not reuse the keys that you generated for communication between the Master Server, Local Distributors, and Remote Agents.

4. **On the CLI Client, copy the public key file to the new user's `authorized_keys2` file on the Master Server machine.**

   `% cp` *User-home-CLI*`/.ssh/id_rsa.pub` *User-home-MS*`/.ssh/id_rsa.pub`

   User-home-CLI is the home directory on the CLI Client machine. *User-home-MS* is the home directory on the Master Server machine.

5. **On the Master Server, concatenate the public key to the `/.ssh/authorized_keys2` file for that user.**

   `% cat /`*User-home*`/.ssh/id_rsa.pub >> /`*User-home*`/.ssh/authorized_keys2`

   User-home is the home directory of the currently logged in user on the Master Server machine.

6. **Log in to the CLI Client machine as the new user that you created.**

7. **Test the SSH connection.**

   `% ssh` *IP-Address-MS* `set`

   IP-Address-MS is the IP address of the Master Server machine.

   You might be prompted to exchange keys.

8. **If you are prompted to exchange keys, answer yes.**

9. **Verify that the `PATH` variable is set correctly.**

   The `PATH` variable must contain `/bin`, `/usr/bin`, and any other directories that are part of your environment.

10. **Configure the Master Server to accept only connections from localhost. For instructions, see "Configuring the JVM Security Policy" on page 101.**

SSH Advanced Parameters and Command Reference

Advanced Parameters Reference

On the Host Details page in the Advanced Parameters field, you can specify additional SSH configuration information. The Advanced Parameters that are accepted are as follows. If you want to use more than one parameter, separate the parameters with a comma. Do not add any unnecessary spaces in the Advanced Parameters field.

cprefix

The cprefix parameter is required for all Local Distributors and Remote Agents. The syntax of the parameter in the Advanced Parameters field is as follows:

`cprefix=`*/N1SPS5.2-Home/application*

N1SPS5.2–Home is the home directory of the application. *application* is agent if you are configuring a Remote Agent or ld if you are configuring a Local Distributor.

EXAMPLE 7–1 cprefix Example

If the Sun N1 Service Provisioning System 5.2 is installed in `/opt/SUNWn1sps/` and you are configuring a Remote Agent, the text you add to the Advanced Parameters field is:

```
cprefix=/opt/SUNWn1sps/N1_Service_Provisioning_System/agent
```

sshpath

If you have not added and do not want to add the path to the SSH executable to the PATH on the target server, you can specify the path to the SSH executable with this parameter. The syntax of the parameter in the Advanced Parameters field is as follows:

`sshpath=`*/path-to-SSH*

path-to-SSH is the directory in which the SSH executable is installed.

EXAMPLE 7–2 sshpath Example

If the SSH executable is installed in the `/usr/local/bin/ssh` directory, the text that you add to the Advanced Parameters field is:

```
sshpath=/usr/local/bin/ssh
```

sshargs

If you want to supply additional command line arguments to the `ssh` command when it runs on a specific Host, you can supply those arguments in the Advanced Parameters field for that Host. The syntax of the parameter in the Advanced Parameters field is as follows:

`sshargs=-`*option*`|-`*option*

option is the command line option that you want supplied to the `ssh` command. Add more than one option by separating the options with a `|`.

EXAMPLE 7–3 sshargs Example

If you want the `ssh` command to enable authentication agent forwarding, supply the `-A` option in the Advanced Parameters field:

```
sshargs=-o|BatchMode yes|-A
```

OpenSSH 2.0 Command Reference

This section describes the OpenSSH 2.0 commands and options that are used in the instructions in this chapter. If you are using a different version of SSH, determine the commands and options that are available in that version of SSH that are equivalent to the following commands. Then, use the equivalent commands when following the instructions to configure SSH.

TABLE 7–1 OpenSSH 2.0 Commands

Tool	Description
`ssh`	Enables the calling application to invoke another application remotely. When configured to use SSH for communications, the software uses the `ssh` command to invoke the remote application, either a Remote Agent or a Local Distributor, and uses the standard input and output streams of SSH to communicate with it.
`ssh-agent`	Used when you want to use private keys with passwords. Upload your keys with the `ssh-agent` so that SSH invocations of the applications communicate with the `ssh-agent` for authentication.
`ssh-add`	Uploads private keys into `ssh-agent`.
`ssh-keygen`	Generates the public-private key pair to secure an SSH connection.

The following options can be used with the `ssh` command:

`-A` Enables authentication agent forwarding

`-o 'BatchMode yes'` Disables passphrase querying

`-t` Allocates a `tty` even if a command is given

The following option can be used with the `ssh-keygen` command:

`-t rsa` Specifies RSA as the type of key to generate.

The following option can be used with the `ssh-agent` command:

`-k` Kills the agent using the `pid` set in the environment variable `SSH_AGENT_PID`. Other implementations might use a different environment variable.

◆ ◆ ◆ CHAPTER 8

Configuring the Sun N1 Service Provisioning System 5.2 for SSL

This chapter contains instructions for configuring the Sun N1 Service Provisioning System 5.2 to communicate using Secure Socket Layer (SSL). This chapter discusses the following topics:

- "Overview of SSL Support in the Sun N1 Service Provisioning System 5.2" on page 87
- "Configuring SSL – Process Overview" on page 91
- "Sample Configuration Scenarios" on page 95
- "SSL Cipher Suites" on page 99

Overview of SSL Support in the Sun N1 Service Provisioning System 5.2

SSL is a protocol for securing communication over IP networks. SSL uses TCP/IP sockets technology to exchange messages between a client and a server, while protecting the message with a public and private key encryption system developed by RSA. Support for SSL is included in most web server products, as well as in the Netscape Navigator and Microsoft web browsers.

Sun N1 Service Provisioning System 5.2 applications can be configured to use SSL for their network communications, preventing messages from being read or tampered with. Optionally, applications can be configured to use SSL to authenticate before communicating, further increasing network security.

Cipher Suites: Encryption and Authentication Overview

The SSL protocol supports a variety of different cryptographic algorithms, or ciphers, for use in operations such as authenticating the server and client to each other, transmitting certificates, and establishing session keys. The cipher suite that SSL uses to connect determines whether any authentication takes place.

Exercise caution when selecting cipher suites. Each application must enable only those cipher suites that provide the minimum security needed by the node. SSL uses the most secure cipher suites supported by both the client and server. If low security cipher suites are enabled, a third party client can force the server to use the less secure cipher suites by publishing support for only the least secure cipher suite during cipher suite negotiation.

SSL can be operated in the following modes:

- Encryption only, no authentication – Connections are encrypted. However, SSL does not authenticate the applications that are connecting.
- Server Authentication – Clients authenticate the server to which they are connecting.
- Server and Client Authentication – Both the client and server authenticate each other.

During the installation, when you select to use SSL to secure communications between applications, you are prompted to select the cipher suite to use. The cipher suite value is stored as the value of `net.ssl.cipher.suites` in the `config.properties` file. The cipher suite value is set to the following value based on the selection you make:

- If you select encryption, no authentication, the cipher suite is set to `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA`.
- If you select encryption, with authentication, the cipher suite is set to `SSL_RSA_WITH_3DES_EDE_CBC_SHA`.

When you use SSL with a Local Distributor on an AIX server, the SSL cipher suite is set to encryption with authentication. Encryption with no authentication is not available for Local Distributors that are running on AIX servers.

For lists of SSL cipher suites that do and do not require server authentication, see "SSL Cipher Suites" on page 99. You can configure client authentication only for cipher suites that require server authentication.

Note – The Sun N1 Service Provisioning System 5.2 applications allow you to configure SSL connections with encryption, no authentication or encryption with authentication. Encryption with authentication uses client and server authentication. Although the configurations described above are possible, encryption, with authentication is the most secure.

Authentication Keystores

The Sun N1 Service Provisioning System 5.2 supports self-signed certificates and certificates signed by a Certifying Authority. Two types of keystores exist:

- Private Keystore – The private keystore contains the public-private key pairs that the application uses to authenticate itself when connecting to other applications.
- Trust Keystore – The trust keystore contains the public key, in self-signed certificates, of other applications that the keystore trusts and allows them to connect to the application.

When enabling SSL for client-server authentication, each enabled application needs to be configured with two keystores that SSL will use to authenticate itself to other applications and to authenticate other applications.

When enabling SSL for server-only authentication, the application acting as the SSL server requires a private keystore and the application acting as the SSL client requires a public, or trusted, keystore. The public keystores are in the proprietary JKS format provided by the Java Secure Sockets Extension (JSSE) v1.0.3.

You must specify a password for both of the keystores. The password for both of the keystores must be the same.

For example, application A, an SSL client, and application B, an SSL server, want to connect with each other using SSL. Both are configured to use a cipher suite that requires server authentication. Application B must have a public-private key pair in its private keystore, and application A must have application B's public key in its trust keystore. When application A attempts to connect to application B, application B sends its public key down to application A. Application A is able to verify the public key by finding it in its trust keystore.

If application B is configured to require client authentication, application A must have a public-private key pair in its private keystore. Also, application B must have application A's public key in its trust keystore. After application A has authenticated application B, application B is able to verify application A's public key, as it finds the public key in its trust keystore.

When you generate a keystore using the `crkeys` command, you use the `-mode upstream|downstream` option to specify the location of the machine that is being authenticated in relationship to the machine that is performing the authentication. When attempting an SSL connection, the server that is validating the connection verifies that the certificate it received contains a mode that matches the relative location of the server that transmitted the certificate. For example, a Local Distributor receives an SSL connection request from a Master Server. The Local Distributor verifies that the Master Server certificate does not have a `downstream` annotation.

Note – The server does not verify that a certificate contains an `upstream` annotation. The server only verifies that the certificate does not have a `downstream` annotation. Consequently, any servers that were originally configured to use SSL without an `upstream` annotation will continue to connect using SSL.

A CLI client cannot validate a connection with a server that transmits a certificate with an `upstream` annotation because no server can be upstream from a CLI client. A Remote Agent cannot validate a connection with a server that transmits a certificate with a `downstream` annotation because no server can be downstream from a Remote Agent.

Using Passwords With SSL

If you supply a password for trust keystore operations, the password is only used to verify the integrity of the keystore. The password does not prevent access to the contents of the trust keystore, but it does protect updates to the keystore. Users are not able to change the contents of the keystore without supplying the password.

If you supply a password for private keystore operations, the password is used to verify the integrity of the keystore, protect against modifications of the keystore contents, and to encrypt and protect access to the private key.

The `crkeys` command validates that you specified the same password for both the keystores. When creating a trust store for the first time by importing certificates, the `crkeys` script ensures that the trust store has the same password as the private store, if one exists. Similarly, when creating a private store for the first time, the `crkeys` script ensures that the private store has the same password as the trust store, if one exists.

The `crkeys` command enables you to create an encoded version of the keystore password. You can use the encoded version of the password in any properties files in which you intend to save the keystore password. Saving an encoded version of the password in a properties file is more secure than saving the plaintext version of a password in a properties file.

Limitations of SSL on the Sun N1 Service Provisioning System 5.2

The SSL implementation on the Sun N1 Service Provisioning System 5.2 has the following limitations:

- Both the trust and the private keystores must be configured with the same password. Also, within the private keystore, the key password for each key in the store must be the same as the store password. The `crkeys` script used to create keys enforces this limitation.
- Although enabling client authentication for CLI Client applications is possible, this setup is not supported due to security limitations. The CLI Client applications do not prompt the user for keystore passwords. If the keystores have been created, the encoded password must be provided in the CLI Client properties file.
- The Sun N1 Service Provisioning System 5.2 uses single trust keystore for both incoming and outgoing connections. Therefore, if a Master Server connects to a Remote Agent and trusts its public key and if that Remote Agent becomes compromised, that Remote Agent's keys could be used to authenticate the CLI Client to the Master Server, if the CLI Client were to use client authentication. Similarly, if a Local Distributor connects to a Remote Agent and the Remote Agent becomes compromised, the Local Distributor can be used to issue commands to the Master Server.

To secure the Master Server and the Local Distributor against such issues, configure the applications to accept connections only from servers that are expected to connect to them. Permit a Local Distributor to accept connections only from its parent node. Permit the Master Server to accept connections only from the designated CLI hosts. For instructions, see Chapter 9.

- For SSH connections, the remote application, the Local Distributor or Remote Agent, is automatically started. The server does not prompt you for the keystore passwords to start these applications. If the applications are initialized with keystores, the encoded passwords to their keystores must be specified in their properties file.
- When you configure the CLI Client to connect to the Master Server using SSH, the CLI Client connects to the Master Server using an SshProxy application that connects to the Master Server through sockets. The SshProxy can connect to the Master Server through SSL, but this configuration is not supported.
- For windows applications, the encoded keystore password must be supplied in the properties file.

Configuring SSL – Process Overview

The following process overview describes the tasks necessary to configure the Sun N1 Service Provisioning System 5.2 to use SSL.

1. Determine the SSL connectivity that you want to use.

   For more information, see "Overview of SSL Support in the Sun N1 Service Provisioning System 5.2" on page 87.

2. Use the `crkeys` command to create keystores.

   See "How to Create Keystores" on page 91.

3. Edit the `config.properties` file to configure SSL.

   See "How to Edit the `config.properties` File to Configure SSL" on page 94.

▼ How to Create Keystores

The Sun N1 Service Provisioning System 5.2 uses the keytool utility provided with the JRE. The keytool utility is wrapped in a shell script, `crkeys`, to enable you to create keystores. The script ensures that the correct parameters are supplied to the `keytool` utility.

When you create a keystore, the X.509 Distinguished Name in the self-signed certificate is set to the following:

`CN=`*application_name* `OU=Engineering O=Sun Microsystems Inc L=Menlo Park ST=CA C=US`

◗ **Generate the keys.**

`%` **`crkeys -`***options*

Use the following options to create keystores based on the type of SSL connectivity you want to use.

`-alias` *application_hostname*	Specifies an alias for the certificate or the key pair. Use the host name of the application as the alias. The alias names must be unique within a keystore.
`-mode` upstream|downstream	Specifies the location of the machine that is being authenticated in relationship to the machine that is performing the authentication. For example, you are generating certificates for a Remote Agent that is downstream form a Local Distributor. Specify downstream as the mode for the Remote Agent. Specify upstream as the mode for the Local Distributor.
`-cpass`	Changes the password of the keystore and all the keys within the keystore.
`-delete`	Specifies that the key pair or certificate for the specified entity should be deleted from keystore.
`-epass`	Converts and prints the encoded version of the plaintext password. Create an encoded version of a password if you plan to store the password in a file. For example, if you choose to store the keystore password in the `config.properties` file, you must supply an encoded version of the password.
`-export`	Exports a self-signed certificate of the specified entity to the specified file.
`-file` *cert_file*	Specifies the name of the file that the certificate is to be imported from or exported to.
`-generate`	Generates a new key pair for the specified alias.
`-help`	Lists all the options.
`-import`	Imports a self-signed certificate of an entity that is allowed to connect to this node. When importing the certificate, the host name of the node that this certificate represents should be used as the alias.
`-keyalg` *keyalg*	The key generation algorithm. Defaults to RSA. Can be either RSA or DSA.
`-keysize` *keysize*	The key size. Defaults to 1024. Can be any multiple of 64 in the range 512-1024 for DSA keys, and range 512-2048 for RSA keys.
`-list`	Lists all the entities contained in the keystore.
`-new` *newpassword*	Specifies the new password for the keystore and all the keys in the keystore. The password must contain at least six characters.
`-password` *password*	Specifies the password for the keystore. If a password is not specified, the user is prompted for a password. The password must contain at least six characters.

`-private`	Specifies the private keystore as the target of the operation.
`-validity` *days_valid*	Number of days the self-signed certificate is valid.
`-trust`	Specifies the trust keystore as the target of the operation.

Example 8–1 `crkeys` Command Syntax

The following examples show how to use the `crkeys` command.

To generate a public-private key pair:

```
crkeys -private –generate -mode {upstream|downstream} –alias application_hostname [-keyalg keyalg]
[-keysize keysize] [-validity days_valid] [–password password]
```

To export the self signed public key for a key pair to a file:

```
crkeys -private –export –file cert_file
 –alias application_hostname [–password password]
```

To import an exported, as shown in the previous example, self signed public key into the trust store:

```
crkeys –trust –import –file cert_file
 –alias application_hostname [-password password]
```

To delete a key or key pair:

```
crkeys {-private|–trust} -delete
 –alias application_hostname [-password password]
```

To list all of the public keys:

```
crkeys {-private|–trust} –list [-password password]
```

To change the SSL keystore, both the trust and the private store, password:

```
crkeys –cpass -password oldpassword
-new newpassword
```

To convert and print the encoded version of the plaintext pasword:

```
crkeys -epass -password password
```

To print instructions for using the `crkeys` command:

```
crkeys -help
```

▼ How to Edit the `config.properties` File to Configure SSL

During the installation, each application is configured to do the following:

- Support cipher suites that require server authentication.
- Do not require client authentication.
- Find the private keystore in the *N1SPS5.2-home*/*app*/`data/private.store` file.
- Find the trust keystore in the *N1SPS5.2-home*/*app*/`data/trust.store` file.
- Supply empty passwords for each keystore.

You can change the SSL configuration of each application to perform the following security checks:

- Selectively enable cipher suites on each application

  You can explicitly specify which cipher suites to enable. If unspecified, the reference implementation uses the cipher suites that are enabled by default. The default cipher suites enabled by the reference implementation require server authentication. For the list of supported cipher suites, see "SSL Cipher Suites" on page 99.
- Specify that the application authenticates the SSL clients that are connecting to it
- Specify the location and password of the private and trust keystores

---

Note – To enable authentication, you must initialize the keystores after installation of the application.

◗ **(Optional) Manually edit the `config.properties` file to change the SSL configuration.**

The following table lists the settings in the `config.properties` file that are related to SSL configurations. Change the parameters based on the type of SSL connectivity you want to use.

Parameter	Default Value	Description
`net.ssl.cipher.suites`	`SSL_RSA_WITH_3DES_EDE_CBC_SHA`	A comma separated list of SSL cipher suites to enable. For a list of supported SSL Cipher suite, see "SSL Cipher Suites" on page 99.
`net.ssl.client.auth`	`false`	Specifies whether the SSL server should authenticate clients that are connecting to it.
`net.ssl.key.store.pass`		The keystore password. Required in some instances. See the following for more information.

Note – The `net.ssl.key.store.pass` parameter specifies the SSL keystore password for an Sun N1 Service Provisioning System 5.2 application. Use this parameter when you configure an application with SSL keystores and you do not want to be prompted for the passwords to the keystore when you start the application. You must specify this parameter in the following instances:

- When you setup the Sun N1 Service Provisioning System applications to start automatically when the system boots
- On Windows servers, Sun N1 Service Provisioning System applications do not prompt for keystore passwords, so this parameter must be specified for any applications configured to use SSL on Windows servers.
- The CLI application does not prompt for keystore passwords, so this parameter must be specified for any CLI Clients that you configure to use SSL.
- If a Local Distributor is connected to its parent through an SSH connection, the Local Distributor cannot prompt for passwords.

Sample Configuration Scenarios

EXAMPLE 8–2 How to Configure SSL Without Authentication Between the Master Server, Local Distributor, and Remote Agent

1. Install the Master Server, Local Distributor, and Remote Agent and select SSL when the installation program prompts you to select a connection type. When prompted to select a cipher suite, select encryption with no authentication.
2. Add the following property to the `config.properties` file for each application.

   ```
   net.ssl.cipher.suites=SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
   ```

   More than one cipher suite or a different cipher suite can be enabled. To enable multiple cipher suites, set the parameter to a comma separated list of cipher suites.
3. From the browser interface, create a new host.
4. On the host that you just created, add a Local Distributor with the connection type SSL.
5. Test the connection to the Local Distributor.
6. Create a new host.
7. On the host that you just created, add a Remote Agent with the connection type SSL.
8. Test the connection to the Remote Agent.

EXAMPLE 8–3 How to Configure SSL Server Authentication

By default, cipher suites requiring server authentication are enabled, so no change is required in the `config.properties` file to enable cipher suites.

EXAMPLE 8–3 How to Configure SSL Server Authentication *(Continued)*

1. Generate a key pair for the Local Distributor and store it in the private keystore for the Local Distributor. Specify `-mode downstream`.

```
% ld/bin/crkeys –private –generate -mode downstream –alias ldhostname-downstream.cr.com –validity 365
```

2. Export the self-signed certificate from the private keystore on the Local Distributor into a file.

```
% ld/bin/crkeys –private –export –file ld-downstream.cert –alias ldhostname-downstream.cr.com
```

3. Copy the self-signed certificate for Local Distributor to the Master Server.
4. Import the self-signed certificate into the Master Server trust keystore.

```
% server/bin/crkeys –trust –import –file ld-downstream.cert –alias ldhostname-downstream.cr.com
```

5. Create a new host.
6. On the new host, add a Local Distributor with the connection type SSL.
7. For the Local Distributor, use the CLI `net.gencfg` command to manually generate the `transport.config` file.
8. Copy the `transport.config` file to the Local Distributor.
9. If already running, stop and start the Master Server and the Local Distributor.
10. Start the Master Server and the Local Distributor.
11. Provide the keystore password for the Master Server and Local Distributor.
12. Test the connection to the Local Distributor.
13. Generate a key pair for the Remote Agent and store it in the private store for the Remote Agent. Specify `-mode downstream`.

```
% agent/bin/crkeys –private –generate -mode downstream –alias rahostname-downstream.cr.com –validity 365
```

14. Export the self-signed certificate from the private store on the Remote Agent into a file.

```
% agent/bin/crkeys –private –export –file ra-downstream.cert –alias rahostname-downstream.cr.com
```

15. Copy the self-signed certificate for the Remote Agent to the Local Distributor.
16. Import the self-signed certificate into the Local Distributor trust store.

```
% ld/bin/crkeys –trust –import –file ra-downstream.cert –alias rahostname-downstream.cr.com
```

17. Create a new host.
18. On the new host, add a Remote Agent with the connection type SSL.
19. For the Remote Agent, use the CLI `net.gencfg` command to manually generate the `transport.config` file.
20. Copy the `transport.config` file to the Remote Agent.
21. If already running, stop the Local Distributor and Remote Agent.
22. Start the Local Distributor and the Remote Agent.
23. Provide the keystore password for the Local Distributor and Remote Agent.

EXAMPLE 8–3 How to Configure SSL Server Authentication *(Continued)*

24. Test the connection to the Remote Agent.

EXAMPLE 8–4 How to Configure SSL Server and Client Authentication

1. Install the Master Server, Local Distributor, and Remote Agent and select SSL when the installation program prompts you to select a connection type. When prompted to select a cipher suite, select encryption with authentication.
2. Edit the `config.properties` file to include the cipher suite you want to use and the encoded keystore password.

   You must use the same keystore password for all of the hosts.
3. Generate a key pair for the Local Distributor and store it in the private store for the Local Distributor. Specify `-mode downstream`.

```
% ld/bin/crkeys –private –generate -mode downstream –alias ldhostname-downstream.cr.com –validity 365
```

4. Generate a key pair for the Master Server and store it in the private store for the Master Server. Specify `-mode upstream`.

```
% server/bin/crkeys –private –generate -mode upstream –alias mshostname-upstream.cr.com –validity 365
```

5. Export the self-signed certificate from the private store for the Local Distributor into a file.

```
% ld/bin/crkeys –private –export –file ld-downstream.cert –alias ldhostname-downstream.cr.com
```

6. Copy the self-signed certificate for the Local Distributor to the Master Server.
7. Import the self-signed certificate into the Master Server trust store.

```
% server/bin/crkeys –trust –import –file ld-downstream.cert –alias ldhostname-downstream.cr.com
```

8. Export the self-signed certificate from the private store for the Master Server into a file.

```
% server/bin/crkeys –private –export –file ms-upstream.cert –alias mshostname-upstream.cr.com
```

9. Copy the self-signed certificate for the Master Server to the Local Distributor.
10. Import the self-signed certificate into the Local Distributor trust store.

    ```
    % ld/bin/crkeys –trust –import –file ms.cert –alias mshostname.cr.com
    ```
11. If already running, stop the Master Server and the Local Distributor.
12. Start the Master Server and the Local Distributor.
13. Provide the keystore password for the Master Server and Local Distributor.
14. Create a new host.
15. On the new host, add a Local Distributor with the connection type SSL.
16. Test the connection to the Local Distributor.
17. Generate a key pair for the Remote Agent and store it in the private store for the Remote Agent. Specify `-mode downstream`.

```
% agent/bin/crkeys –private –generate -mode downstream –alias rahostname-downstream.cr.com –validity 365
```

EXAMPLE 8–4 How to Configure SSL Server and Client Authentication *(Continued)*

18. Export the self-signed certificate from private store for the Remote Agent into a file.

```
% agent/bin/crkeys –private –export –file ra-downstream.cert –alias rahostname-downstream.cr.com
```

19. Copy the self-signed certificate for the Remote Agent to the Local Distributor.
20. Import the self-signed certificate into the Local Distributor trust store.

```
% ld/bin/crkeys –trust –import –file ra-downstream.cert –alias rahostname-downstream.cr.com
```

21. Generate a key pair for the Local Distributor and store it in the private store for the Local Distributor. Specify `-mode upstream`.

```
% ls/bin/crkeys -private -generate -mode upstream - alias ldhostname-upstream.cr.com -validity 365
```

22. Export the self-signed certificate from the private store for the Local Distributor into a file.

```
% ld/bin/crkeys -private -export -file ld-upstream.cert -alias ldhostname-upstream.cr.com
```

23. Copy the self-signed certificate for the Local Distributor, that you exported in Step 21, to the Remote Agent machine.
24. Import the self-signed certificate into the Remote Agent trust store.

```
% agent/bin/crkeys –trust –import –file ld-upstream.cert –alias ldhostname-upstream.cr.com
```

25. If already running, stop the Local Distributor and Remote Agent.
26. Start the Local Distributor and the Remote Agent.
27. Provide the keystore password for the Local Distributor and Remote Agent.
28. Create a new host.
29. On the new host, add a Remote Agent with the connection type SSL.
30. Test the connection to the Remote Agent.

EXAMPLE 8–5 How to Configure SSL Authentication Between a CLI Client and Master Server

1. Install the Master Server and the CLI Client and select SSL when the installation program prompts you to select a connection type. When prompted to select a cipher suite, select encryption with authentication.
2. Generate a key pair for the Master Server and store it in the private store for the Master Server.

   ```
   % server/bin/crkeys –private –generate –alias mshostname.cr.com –validity 365
   ```

3. Generate a key pair for the CLI Client and store it in the private store for the CLI Client.

   ```
   % cli/bin/crkeys -private -generate -alias clihostname.cr.com.cr.com -validity 365
   ```

4. Export the self-signed certificate from the private store for Master Server private store into a file.

   ```
   % server/bin/crkeys –private –export –file ms.cert –alias mshostname.cr.com
   ```

5. Copy the Master Server self-signed certificate to the CLI Client.
6. Import the self-signed certificate into CLI Client trust store.

EXAMPLE 8–5 How to Configure SSL Authentication Between a CLI Client and Master Server *(Continued)*

```
% cli/bin/crkeys –trust –import –file ms.cert –alias mshostname.cr.com
```

7. Export the self-signed certificate from the private store for CLI Client into a file.

```
% cli/bin/crkeys -private -export -file cli.cert -alias clihostname.cr.com
```

8. Copy the CLI Client self-signed certificate to the Master Server.
9. Import the self-signed certificate into the Master Server trust store.

```
% server/bin/crkeys -trust -import -file cli.cert -alias clihostname.cr.com
```

10. If the Master Server is running, stop the Master Server.
11. Start the Master Server.
12. Provide the keystore password for the Master Server.
13. On the CLI Client, edit the `config.properties` file to include the following line:

```
net.ssl.key.store.pass=trust-store-password
```

14. Run a CLI Client command to verify the connection.

SSL Cipher Suites

Cipher Suites

The following lists describe the supported SSL cipher suites for all supported operating systems except IBM AIX.

The following suites require server authentication:

```
SSL_DHE_DSS_WITH_DES_CBC_SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_DES_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA
```

The following suites do not require server authentication:

```
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
```

```
SSL_DH_anon_WITH_DES_CBC_SHA
SSL_DH_anon_WITH_RC4_128_MD5
TLS_DH_anon_WITH_AES_128_CBC_SHA
```

The following suites require server authentication with no encryption:

```
SSL_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA
```

Cipher Suites for IBM AIX

The following lists describe the supported SSL cipher suites for IBM AIX servers.

All of the following cipher suites are available for use with Remote Agents. Cipher suites that do not require server authentication cannot be used for Local Distributors or CLI.

The following suites require server authentication:

```
SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_DES_CBC_SHA
SSL_RSA_FIPS_WITH_DES_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
```

The following suites do not require server authentication:

Note – Cipher suites that do not require server authentication cannot be used for Local Distributors or CLI.

```
SSL_DH_anon_WITH_RC4_128_MD5
SSL_DH_anon_WITH_DES_CBC_SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
```

The following suites require server authentication with no encryption:

```
SSL_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA
```

◆ ◆ ◆ CHAPTER 9

Configuring the Java Virtual Machine Security Policy

This chapter describes how to configure the security policy of the Sun N1 Service Provisioning System 5.2 applications to allow them only to accept connections from a specific IP Address and Port range or to allow them only to connect to a specific IP Address and Port range.

Configuring the JVM Security Policy

Each Sun N1 Service Provisioning System 5.2 application has a Java Virtual Machine (JVM) security policy file located in `lib/security/rox.policy`. This file specifies the permissions assigned to the application. As installed, the policy file allows the application to connect to and accept connections from any host. If you are using the CLI Client with SSH, change the policy file to restrict the connection to only the localhost.

The following line in the `lib/security/rox.policy` file grants these permissions.

```
permission java.net.SocketPermission "*", "connect,accept,listen";
```

If you want to restrict the network access abilities of the application, delete this line and add more restrictive permissions.

The host parameter for SocketPermission is as follows:

`host =` *hostname*|*IPaddress* `:`*portrange*

hostname is the host name of the machine. *IPaddress* is the IP address of the machine. *portrange* is the following:

```
portrange = portnumber | -portnumber | portnumber-[portnumber]
```

For more information about the syntax for the security policy file, see `http://java.sun.com/j2se/1.4.2/docs/guide/security/PolicyFiles.html` and click on the Policy File Syntax link.

How to Configure the JVM Policy for the Master Server

1. **Edit the `lib/security/rox.policy` file.**
2. **Delete the line that allows the application to connect to or accept connections from all hosts.**
3. **Add the following lines to give the application permission selectively.**

```
permission java.net.SocketPermission "localhost:localport", "accept";
permission java.net.SocketPermission "localhost:dbport", "connect";
permission java.net.SocketPermission "<domain>:httpport", "connect";
permission java.net.SocketPermission "ipAddress1:port1", "connect";
permission java.net.SocketPermission "ipAddress2:port2", "connect"; ...
```

- *localport* is the port that the CLI Client uses to connect to the Master Server. The first line restricts the Master Server to allow CLI Clients to connect only locally or through `ssh-proxy`.
- *dbport* is the port number for the Postgres database server.
- *domain* is the domain of the hosts that are to be allowed to connect to the browser interface. *httpport* is the port number the browser interface.
- *ipAddress1:port1* and *ipAddress2:port2* are the IP address and port numbers of the Remote Agents or Local Distributors that are connected directly to the Master Server.

How to Configure the JVM Policy for the Remote Agent

1. **Edit the `lib/security/rox.policy` file.**
2. **Delete the line that allows the application to connect to or accept connections from all hosts.**
3. **Add the following line to give the application permission.**

```
permission java.net.SocketPermission "ipAddress", "accept";
```

ipAddress is the IP address of the Local Distributor or the Master Server to which this Remote Agent is connected.

More Information

Adding Permissions to Connect to a Host

If you plan to execute plans containing steps that require network access, such as `urltest`, you might want to add permissions for this Remote Agent to connect to a particular host.

▼ How to Configure the JVM Policy for the Local Distributor

1. **Edit the `lib/security/rox.policy` file.**
2. **Delete the line that allows the application to connect to or accept connections from all hosts.**
3. **Add the following lines to give the application permission selectively.**

```
permission java.net.SocketPermission "ipAddress", "accept";
permission java.net.SocketPermission "ipAddress1:port1", "connect";
permission java.net.SocketPermission "ipAddress2:port2", "connect"; ...
```

   - *ipAddress* is the IP address of the Local Distributor or Master Server that is the parent of this Local Distributor.
   - *ipAddress1*:*port1* and *ipAddress2*:*port2* are the IP address and port numbers of the Remote Agents or Local Distributors for which this Local Distributor is the parent.

Postgres Security

Ensure that the Postgres database does not accept connections from other hosts. The default configuration of the Postgres database is to accept connections from UNIX sockets and `localhost`. Change this default setting in the `server/postgres/data/pg_hba.conf` configuration file. Also, change the database password after installation using the alter user username with password 'password' query. If you make these changes to the Postgres configuration file, in the *N1SPS5.2-MasterServer-home*`/config/config.properties` file, you must change the value of `db.password`.

◆ ◆ ◆ CHAPTER 10

Upgrading to the Sun N1 Service Provisioning System 5.2

This chapter contains instructions for upgrading from the 5.1 version of product to the Sun N1 Service Provisioning System 5.2.

Note – If you have a version of the software that was released prior to version 5.1, you must upgrade to version 5.1 before you upgrade to the Sun N1 Service Provisioning System 5.2. If you have a version of the software that was released prior to version 4.0, you must upgrade to version 4.0 before you upgrade to version 4.1.

- "Upgrading Overview" on page 105
- "Upgrading the Master Server" on page 106
- "Upgrading Remote Agents and Local Distributors" on page 111

Upgrading Overview

The upgrade procedure that you complete depends on whether the N1 Service Provisioning System 5.1 is running on an operating system version that is supported for the Sun N1 Service Provisioning System 5.2. Occasionally, new versions of the Sun N1 Service Provisioning System no longer contain support for operating systems that were previously supported.

Upgrading Requirements

When upgrading any of the Sun N1 Service Provisioning System applications, the new server on which you install the 5.2 application must meet the following requirements:

- Operating System – The operating system on which you run the 5.2 application must be the same type of operating system on which you were running the 5.1 application. For example, you can migrate a Master Server running on Red Hat Linux Advanced Server 2.1 to a Master Server running on Red Hat Linux Advanced Server 3.0. You cannot migrate a Master Server running on Red Hat Linux Advanced Server 2.1 to a Master Server running on any version of the Solaris OS.

- Hardware Architecture – The hardware architecture of the server on which you run the 5.2 application must be the same as the architecture on which you were running the 5.1 application. For example, you can upgrade a Remote Agent from a SPARC based server running on the Solaris OS to another SPARC based server running on the Solaris OS. You cannot upgrade a Remote Agent from a SPARC based server running on the Solaris OS to an x86 based server running on the Solaris OS.
- User Ownership of Application – The 5.2 application must be owned by the same user that owns the 5.1 application. For example, if you installed the 5.1 Master Server and assigned ownership to user `foo`, then the 5.2 Master Server must have user `foo` assigned as its owner.

Upgrading – Process Overview

The following process overview describes the tasks necessary to properly upgrade from the Sun N1 Service Provisioning System 5.1 to the Sun N1 Service Provisioning System 5.2.

1. Determine whether the servers that you are upgrading meet the minimum requirements to run the Sun N1 Service Provisioning System 5.2.

   See Chapter 2.
2. Complete the instructions to upgrade the Master Server.
3. Complete the instructions to upgrade the Remote Agents and Local Distributors.

   See, "Upgrading Remote Agents and Local Distributors" on page 111
4. CLI Clients do not need to be upgraded. Simply install the 5.2 version of the CLI Client and uninstall the 5.1 version.

Upgrading the Master Server

The Master Server application is not upgraded like most software is upgraded. Rather, the new version of the Master Server is installed on the same server as the previous version of the Master Server. Then, the data is migrated from the previous version of the Master Server to the new version of the Master Server.

▼ How to Migrate Master Server Data

Migrating data from the 5.1 version of the Master Server to the 5.2 version of the Master Server deletes any data in the 5.2 version of the Master Server. The migration script stops both versions of the Master Server until the script completes the migration. The Master Servers will be unavailable for the duration of the migration.

Before You Begin Verify that the operating system version on the server you want to upgrade is supported by the Sun N1 Service Provisioning System 5.2. For a list of supported operating systems, see "Operating System Requirements" on page 25.

Back up your data before you migrate. See Chapter 9, "Backing Up and Restoring," in *Sun N1 Service Provisioning System 5.2 System Administration Guide*.

1 Log into the machine as the user who installed the 5.1 version of the Master Server.

2 Install the 5.2 version of the Master Server on the same server where the 5.1 version of the Master Server is installed.

- If you are running a Linux or UNIX system, follow the instructions in "How to Install the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems" on page 43 to install the Sun N1 Service Provisioning System 5.2 Master Server.
- If you are running Windows, complete the following steps.

  a. Stop the 5.1 Master Server by using the Service application in the Windows Administrative Tools to stop the IPC Daemon service.

  b. Set the 5.1 Master Services to start manually, specifically the IPC Daemon and Server.

  c. Install the 5.2 version of the Master Server by following the instructions in "How to Install the Sun N1 Service Provisioning System 5.2 Master Server on Windows" on page 53.

  Install the 5.2 version of the Master Server with ownership by the same user and group that owns the 5.1 version of the Master Server.

3 For Linux and UNIX Master Servers, verify that the database will not be optimized during the migration process.

Check that you do not have any `cron` jobs scheduled that would start a database optimization while you are migrating the data.

4 Access a command prompt.

- On Linux and UNIX servers, open a shell window and log in as the user that owns the Master Server.
- On Windows, open a Command Prompt window.

5 Change to the directory that contains the migration script.

- On Linux and UNIX servers, type:

  ```
  % cd /N1SPS5.2-home/server/bin/migrate
  ```

  N1SPS5.2-home is the directory where you installed the Master Server.
- On Windows, type:

  ```
  C:\> cd C:\N1SPS5.2-home\server\bin\migrate
  ```

  C:\N1SPS5.2-home is the directory in which you installed the Master Server.

6 Start the migration script.

- On Linux and UNIX servers, type:

```
% ./migrateMS_5.1-5.2.sh
```

The migration script has the following command line options:

`-a` *archive_file* or `-archive` *archive_file*	Specify *archive_file* to which Master Server is archived
`-bg`	Run migration in the background

Note – If you use the `-bg` option, you need to review the `MS-Migration-`*number*`.log` file to make sure that the migration completed successfully before starting the Master Server.

```
Migration started : date_and_time
...
Master server migration completed successfully: date_and_time

Stopping 5.2 Database
stop Database in /opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/server succeeded
Issuing sync for flushing buffer data to disk.

A log of migration can be found in
/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/server/bin/migration/MS-Migration-number.log

A log of output of database commands can be found in
/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/server/bin/migration/MS-Migration-DB-CMD-number.log
```

`-d` *logfile*	Write database migration log output to *logfile* instead of the default log file
`-dbfsyncon`	The migration script now turns the `fsync()` system call off while restoring the database for performance improvement. This option can be used to keep the `fsync()` system call on.
`-dbtextdump`	The migration script now uses a new method to migrate the database. The new method uses binary dumps instead of textual dumps for a significant performance improvement. The `-dbtextdump` option is provided if you want to revert back to the previous method.
`-l` *logfile*	Write migration log output to *logfile* instead of the default log file
`-m` *archive_file* or `-migrate` *archive_file*	Specify *archive_file* from which to migrate Master Server

`-n` *new_ms_path*	Specify the path of the new Master Server installation.
`-o` *old_ms_path*	Specify the path of the old Master Server installation.
`-sj`	Skip Java migration
`-y`	Default to yes

- On Windows, type:

  C:\N1SPS5.2-home\5.2`\server\bin\migrate\>`**`.\migrateMS_5.1-5.2.cmd`**

7 Follow the instructions on the screen to complete the migration.

When the migration is complete, the following message appears.

```
Master Server migration completed successfully.
```

Note – The listener port numbers for the Postgres database, browser interface, and the Master Server are not migrated. The Sun N1 Service Provisioning System 5.2 Master Server uses the port numbers that you supplied during installation.

---

8 Check the log file for any errors that might have occurred during migration.

The migration script displays the location of the log file.

9 Clear the cache on the browser that you use to access the Master Server browser interface.

If you started a browser session before upgrading the Master Server, graphics and style sheets that are cached in the browser might prevent you from seeing the upgraded browser interface.

10 Back up the data that you migrated to the new Master Server by using the instructions in Chapter 9, "Backing Up and Restoring," in *Sun N1 Service Provisioning System 5.2 System Administration Guide***.**

You cannot restore data from the 5.1 Master Server to the 5.2 Master Server. Back up the data on the 5.2 Master Server so that you have a complete and accurate backup of the data to use if necessary.

11 (Optional) Uninstall the 5.1 version of the Master Server.

You can uninstall the 5.1 version of the Master Server if you do not want to use it by following the instructions in Chapter 11.

Master Server Data Migration Details

This section describes details about the data that is migrated from the previous version of the Master Server to the new version of the Master Server.

Master Server Data Migration

The following table details the types of data that are migrated on the Master Server.

TABLE 10–1 Migration Overview

Data on the Master Server	Is the Data Migrated?	Mechanism for Migration
PostgresSQL data	Yes	SQL scripts
CLI Client script for changes to existing commands	No	
Migration of objects serialized through CLI Client	No	
Migration of changes to the `config.properties` file on each host	Yes	Properties listed in the file are migrated using the details found in "Migration Details for the Properties File" on page 110.
Resource migration	Yes	Copy the resources directory.
Logger Configuration file	No	
User interface customizations	No	

Migration Details for the Properties File

The 5.1 `config.properties` file is migrated to the 5.2 `config.properties` file. During the migration, the value of each property in the 5.1 file is compared to the value of the property in the 5.2 `config.properties` file. If the value is the same, the property is ignored. If the value is different, then the 5.1 value is copied to the 5.2 `config.properties` file. If a property exists in the 5.1 file and is absent in the 5.2 file, then the 5.1 value is added to the 5.2 file. Values for the following properties are not migrated to the 5.2 `config.properties` file:

- `webserver.TomcatHome`
- `rsrc.localrepo`
- `db.port`
- `db.password`
- `db.hostname`
- `db.instancename`
- `db.username`
- `db.type`
- `db.maxconnections`
- `hostdb.ms.ipaddress`
- `hostdb.ms.port`
- `note.mailsubject`
- `net.server.nconn`
- `net.server.type.1`
- `net.server.ip.1`
- `net.server.port.1`

- `net.server.parms.1`
- `note.url`
- `pe.defaultUserToRunAs`
- `hostdb.ms.connectiontype`
- `pe.maxSimulPlans`

If you have changed the values for any of these properties in the 5.1 properties file, you need to manually change the value in the 5.2 `config.properties` file.

Upgrading Remote Agents and Local Distributors

The Sun N1 Service Provisioning System 5.2 provides an automatic upgrade utility that allows you to upgrade Remote Agents and Local Distributors by using the browser interface to the Master Server.

For information on upgrading by using a CLI command, see "Overview of the node Commands" in *Sun N1 Service Provisioning System 5.2 Command-Line Interface Reference Manual*.

▼ How to Upgrade Remote Agents and Local Distributors

Before You Begin Migrate the Master Server before upgrading Remote Agents and Local Distributors.

1 Log in to the browser interface of the Sun N1 Service Provisioning System 5.2 Master Server.

2 Click Hosts.

3 Click masterserver.

4 Click the Update Entire N1 SPS network... button.

A window opens and displays a list of hosts that are being upgraded. The progress of the upgrade is also displayed. When the process completes, the window displays the following message.

```
Host Update not yet complete.
```

5 Click the Close button.

6 To complete the second phase of the upgrade, click the Update Entire N1 SPS network... button again.

A window displays a list of hosts that are being upgraded. The progress of the upgrade is also displayed. When the process completes, the status for each of the hosts displays as Updated.

7 Click the Close button.

The upgrade is complete.

8 Prepare the Hosts that you upgraded.

Before you can run a Plan on a Host that you upgraded, you must Prepare the Host. To Prepare Hosts, follow the instructions in "How to Prepare a Physical Host" in *Sun N1 Service Provisioning System 5.2 System Administration Guide* .

◆ ◆ ◆ CHAPTER 11

Uninstalling the Sun N1 Service Provisioning System 5.2

This chapter describes procedures for uninstalling the Sun N1 Service Provisioning System 5.2 in the following sections:

- "Uninstalling Applications on Linux and UNIX Systems" on page 113
- "Uninstalling Applications on Windows Systems" on page 115

Uninstalling Applications on Linux and UNIX Systems

The procedure to uninstall the Sun N1 Service Provisioning System 5.2 depends upon the application you want to uninstall and on which operating system it is running.

- If you want to uninstall the Solaris OS Master Server or CLI Client, use the instructions in "How to Uninstall a Solaris OS Master Server or CLI Client" on page 113.
- If you want to uninstall the Solaris OS Remote Agent or Local Distributor or any of the applications on other Linux or UNIX servers, use the instructions in "How to Uninstall File-Based Applications on Linux and UNIX Systems" on page 114.

▼ How to Uninstall a Solaris OS Master Server or CLI Client

The Solaris OS Master Server and CLI Client are installed as packages. The uninstall script removes only 5.2 versions of the Master Server or CLI Client.

1 On the server that you want to uninstall the application, verify that you are not in the directory of the application that you want to uninstall.

2 Stop the Master Server and CLI Client.

3 Begin the uninstallation.

```
# /N1SPS5.2-home/app_directory/bin/cr_uninstall_app.sh
```

N1SPS5.2-home is the directory where you installed the application. The default directory is `/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2`. *app_directory* is one of the following values:

- server – Master Server
- cli – CLI Client

app is one of the following values:

- ms – uninstalls the Master Server
- cli – uninstalls the CLI Client

The following message appears when the uninstallation is complete.

```
Successfully removed SUNWsps app
Successfully removed SUNWspsc1
Successfully removed SUNWspsj1
```

app is ms when uninstalling a Master Server and cl when uninstalling a CLI Client.

Note – The `SUNWspsc1` and `SUNWspsj1` packages are not removed if another application is installed on this server. For example, if you have a Master Server and a CLI Client both installed on the same server, when you uninstall only the Master Server, the `SUNWspsc1` and `SUNWspsj1` packages remain on the server until you uninstall the CLI Client.

▼ How to Uninstall File-Based Applications on Linux and UNIX Systems

1 On the server that you want to uninstall the application, verify that you are not in the directory of the application you want to uninstall.

2 Stop the application that you want to uninstall.

3 If you are uninstalling a Remote Agent, change the permissions on files in the `/protect` directory.

```
% chmod -R 755 /N1SPS5.2-home/agent/bin/protect
```

N1SPS5.2-home is the directory where you installed the Remote Agent.

4 Delete the directory that contains the application that you want to uninstall.

```
# rm -r /N1SPS5.2-home/app-directory
```

N1SPS5.2-home is the directory where you installed the application. The default directory on UNIX systems is `/opt/SUNWn1sps/`. The default directory on Linux systems is `/opt/sun`. The value for *app-directory* is one of the following:

- server – uninstalls a Master Server

- agent – uninstalls a Remote Agent
- cli – uninstalls a CLI Client
- ld – uninstalls a Local Distributor

5 **If you are uninstalling all of the applications from the machine, when the *N1SPS5.2–home* directory contains no more application directories, delete the `common/` directory.**

```
# rm -r N1SPS5.2-home/common
```

The uninstallation is complete.

How to Disable Automatic Database Optimization

If you uninstall a Red Hat Linux Master Server, you must manually remove the entry from the `crontab` file that instructs the system to automatically optimize the database. The uninstall script for Solaris Master Servers automatically removes this entry from the `cronjob` file.

1 **As the user that owns the Master Server, list the current `crontab` and direct the output to a file.**

```
# crontab -l > newcrontabfile
```

2 **Open the `newcrontab` file in a text editor.**

3 **Remove the following line from the `newcrontab` file.**

```
MM HH * * * N1SPS5.2-home/server/bin/roxdbcmd vacuumdb -d rox > /dev/null 2> /dev/null
```

N1SPS5.2-home is the home directory of the Master Server.

4 **Save the `newcrontab` file.**

5 **Update the `crontab`.**

```
# crontab newcrontabfile
```

Uninstalling Applications on Windows Systems

To uninstall applications on Windows servers, use the Add and Remove Programs function available in the Windows Control Panel. When you perform an uninstallation, ensure that the Microsoft Management Console with Services snap-in, also know as the Services console, is not open. Otherwise, the Master Server, Remote Agent, or Local Distributor might not uninstall properly.

◆ ◆ ◆ APPENDIX A

Installation and Configuration Reference

This appendix contains details about the installation of the Sun N1 Service Provisioning System 5.2 in the following sections:

- "Reference Data for the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems" on page 117
- "Reference Data for the Sun N1 Service Provisioning System 5.2 on Windows" on page 123

Reference Data for the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems

This section contains details about the installation of the Sun N1 Service Provisioning System 5.2 on Linux and UNIX systems. The topics include the following sections:

- "Directory Structure of the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems" on page 117
- "Database Optimization on Linux and UNIX Systems" on page 120
- "Sample Remote Agent Parameters File for Linux and UNIX Systems" on page 120

Directory Structure of the Sun N1 Service Provisioning System 5.2 on Linux and UNIX Systems

When installing the Sun N1 Service Provisioning System 5.2, you are prompted to select a home directory for the software. The default directory on UNIX systems is `/opt/SUNWn1sps`. The default directory on Linux systems is `/opt/sun`. The installation program creates the following directory tree within the home directory:

- `N1_Service_Provisioning_System_5.2` is the directory created for the Master Server and CLI Client that contains the software.
- `N1_Service_Provisioning_System` is the directory created for the Local Distributor and Remote Agent that contains the software.

The installation scripts install the Sun N1 Service Provisioning System 5.2 software into default destination directories that are subdirectories of the home directory for the software. All directories are created with the permissions set to 755, `rwxr-xr-x`, except when noted in the tables below. Most files are assigned with the permissions set to 644, `rw-r--r`, except for executable files and scripts, which are set to 755.

The following table lists the directories that are installed for every Sun N1 Service Provisioning System 5.2 application, the Master Server, Local Distributor, Remote Agent, and CLI Client.

TABLE A–1 Directories Common to All Applications

Directory	Contents
`/common`	Common files for all applications
`/common/jre`	Bundled copy of platform-specific JRE
`/common/lib`	Library files common for some or all applications

The following table lists the directories installed for the Master Server.

TABLE A–2 Directories Installed for the Master Server

Directory	Contents
`/server/config`	Master Server configuration files
`/server/custom`	User interface customization files
`/server/data`	Master Server data files
`/server/bin`	Master Server executable files
`/server/lib`	Master Server-specific library files
`/server/postgres`	Bundled copy of Postgres
`/server/tomcat`	Bundled copy of Apache Tomcat
`/server/webapp`	Browser Interface Web Application
`/server/setup`	Miscellaneous files used to initialize the Master Server
`/server/config/proxy/config`	Command line user interface SSH proxy properties file
`/server/data/tmp`	Master Server temporary directory with permissions set to 777
`/server/README`	Text license agreement

The following table lists the directories installed for the Local Distributor.

TABLE A–3 Directories Installed for the Local Distributor

Directory	Contents
`/ld/config`	Local Distributor configuration files
`/ld/bin`	Local Distributor executable files
`/ld/lib`	Local Distributor library files
`/ld/data`	Local Distributor specific data
`/ld/data/tmp`	Local Distributor temporary directory with permissions set to 777
`/ld/jvm/jre/bin`	Local Distributor JRE proxy
`/ld/README`	Text license agreement

The following table lists the directories installed for the Remote Agent.

TABLE A–4 Directories Installed for the Remote Agent

Directory	Contents
`/agent/config`	Remote Agent configuration files
`/agent/bin`	Remote Agent executable files
`/agent/bin/protect`	Jexec directory with permissions set to 100, `--x------`
`/agent/bin/protect/jexec`	Jexec is used when the agent needs root permissions with permissions set to 4110
`/agent/lib`	Remote Agent library files
`/agent/data`	Remote Agent specific data
`/agent/work`	Default directory for execution of execNatives.
`/agent/data/tmp`	Remote Agent temporary directory with permissions set to 777
`/agent/jvm/jre/bin`	Remote Agent JRE proxy
`/agent/README`	Text license agreement

The following table lists the directories installed for the CLI Client.

TABLE A–5 Directories Installed for the CLI Client

Directory	Contents
`/cli/config`	CLI configuration files

TABLE A–5 Directories Installed for the CLI Client *(Continued)*

Directory	Contents
`/cli/bin`	CLI executable files
`/cli/lib`	CLI library files
`/cli/data`	CLI specific data
`/cli/data/tmp`	CLI temporary directory with permissions set to 777
`/cli/README`	Text license agreement

Database Optimization on Linux and UNIX Systems

The installation program prompts you to set up database optimization daily. If you select to optimize the database daily, the installation script adds the following command to the `cronjob` file. You can add this command to the `cronjob` file at any time to begin daily optimization of the database.

`MM HH * * *` *N1SPS5.2-home*`/server/bin/roxdbcmd vacuumdb -d rox > /dev/null 2> /dev/null`

N1SPS5.2-home is the home directory of the Master Server.

Sample Remote Agent Parameters File for Linux and UNIX Systems

A sample parameters file is installed on the Master Server in the `/server/bin` directory, along with other scripts, when you install the Master Server. You can use this file to indicate configuration selections so that you can non-interactively install the Remote Agent. The contents of the sample parameters file are shown below.

```
# This is a sample file that sets the parameters required
# for the remote installation of Remote Agents.
#
# This file must be uncommented and edited with the correct
# values before it can be used.
# $Id: cr_ra_remote_params.sh,v 1.4 2005/05/19 23:52:07 echiquet Exp $

# CR_RA_INSTALLBASE - the base directory where the
# Remote Agent will be installed.  If the directory
# does not exist, the installer will attempt to create it.
# Defaults to /opt/SUNWn1sps
#
CR_RA_INSTALLBASE=/opt/SUNWn1sps

# CR_RA_OWNER - The owner of the distribution. A pre-existing
# user must be specified.  Defaults to 'n1sps'.
```

```
#
CR_RA_OWNER=n1sps

# CR_RA_GROUP - The group owner of the distribution.  A
# pre-existing group name must be specified.   Defaults to 'n1sps'.
#
CR_RA_GROUP=n1sps

# CR_RA_PORT - Port number that the Remote Agent will listen on.
# An integer value between 1024 and 65535 must be specified.  Defaults
# to 1131.
#
CR_RA_PORT=1131

# CR_RA_CTYPE - Parent connection type.  How the parent connects to
# this RA.  One of 'raw' (unencrypted), 'ssh', or 'ssl'.  There is no default.
# This parameter is required.
#
#CR_RA_CTYPE=raw

# CR_RA_CIPHER_TYPE - SSL cipher suite type.  One of '1' (encryption,
# no authentication) or '2' (encryption, with authentication).
# Default is 1, but has no effect for parent connection type of raw or
# ssh.
#
CR_RA_CIPHER_TYPE=1

# CR_RA_CTYPE_CONFIRM - Selection confirmation for an insecure
# connection type, such as 'raw' or 'ssl with no authentication'. If set to
# anything else than 'true', the installation will fail for such connection
# types.
CR_RA_CTYPE_CONFIRM=false

# CR_RA_INSTALL_JRE - Directive of whether or not a JRE should be
# installed with the Remote Agent for it's use.  Defaults to 'y'.  Valid
# values are 'y' or 'n'.
#
CR_RA_INSTALL_JRE=y

# JRE_HOME - Directive for the location of the JRE installation.  If
# the CR_RA_INSTALL_JRE directive is set to 'y', the installer will
# install the JRE.  In this case, the JRE_HOME value will be
# $CR_RA_INSTALLBASE/common/jre.  If the installer is not going to
# install the JRE, the JRE_HOME should point to where the pre-existing JRE
# is installed.
#
JRE_HOME=$CR_RA_INSTALLBASE/N1_Service_Provisioning_System/common/jre
```

```
# CR_RA_SUID - Directive of whether or not the RA should be installed
# with the setuid root privileges.  Valid values are 'y' or 'n'.  This
# only works when the remote installer is run as the root user and
# the connection type is ssh.
# There is no default. This parameter is required.
#
#CR_RA_SUID=y

# CR_RA_INSTALLER_USER - The user that should perform this install.  This
# is what the remote installer will use to ssh into the remote hosts
# and run the commands as.  It is highly recommended that this be set to
# root, although, it doesn't have to be.  Defaults to the current user.
#
CR_RA_INSTALLER_USER=root

# CR_RA_INSTALLER_WORKDIR - The directory to use to store temporary files.
# The distribution will be copied into this directory so make sure
# that this it has enough space to store the distribution file.  Defaults to
# /tmp
#
CR_RA_INSTALLER_WORKDIR=/tmp

# CR_RA_INSTALLER_LEAVEFILES - Directive of whether or not the temporary
# files should be preserved on the remote host.  Defaults to 'n'.
#
CR_RA_INSTALLER_LEAVEFILES=n

# CR_RA_INSTALLER_HOSTS - List of remote hosts on which the Remote Agent is
# to be installed.  This must contain at least one host name.  This host list
# can also be set in the  environment  variable 'CR_RA_INSTALLER_HOSTS', or
# specified on the command line.  Check the remote agent installer script
# usage message for exactly how this can be done.
#
# Note : The format of the list of hosts is critical.  The list of hosts
# must be separated by a comma (',') and cannot have any spaces in between.
# It must be in one contiguous string.
#
CR_RA_INSTALLER_HOSTS=""

export CR_RA_INSTALLBASE CR_RA_PORT CR_RA_GROUP CR_RA_OWNER CR_RA_INSTALL_JRE CR_RA_SUID
export CR_RA_CTYPE CR_RA_CIPHER_TYPE
export CR_RA_INSTALLER_USER CR_RA_INSTALLER_WORKDIR CR_RA_INSTALLER_LEAVEFILES
export CR_RA_INSTALLER_HOSTS  JRE_HOME
```

CR_RA_ALLOWFORWARDVERSION Parameter

If you want to install an Sun N1 Service Provisioning System 5.2 Remote Agent on a version of an operating system that is numerically higher than the highest version the Sun N1 Service Provisioning System 5.2 supports for that operating system, add the following parameter to the parameters file:

```
CR_RA_ALLOWFORWARDVERSION=y
```

If you use the `CR_RA_ALLOWFORWARDVERSION=y` parameter, the installation program does not verify that the operating system on which you are installing the Remote Agent is supported. There is no standard Sun Services support for use of the Sun N1 Service Provisioning System 5.2 on unsupported operating systems.

Caution – Installing the Sun N1 Service Provisioning System 5.2 on unsupported operating systems might result in undefined and unexpected behavior. Install the Sun N1 Service Provisioning System 5.2 on unsupported operating systems only for testing purposes. Do not use the Sun N1 Service Provisioning System 5.2 on unsupported operating systems in a production environment.

Reference Data for the Sun N1 Service Provisioning System 5.2 on Windows

This section contains details about the installation of the Sun N1 Service Provisioning System 5.2 on Windows in the following sections:

- "Directory Structure of the Sun N1 Service Provisioning System 5.2 on Windows" on page 123
- "Cygwin" on page 125
- "Actions Performed by the Windows Installation Scripts" on page 126

Directory Structure of the Sun N1 Service Provisioning System 5.2 on Windows

When installing the Sun N1 Service Provisioning System 5.2, you are prompted to select a home directory for the software. The default directory is one of the following.

- `C:\Program Files\N1 Service Provisioning System\5.2` is the directory created for the Master Server and CLI Client that contains the software.
- `C:\Program Files\N1 Service Provisioning System` is the directory created for the Local Distributor and Remote Agent that contains the software.

The installation scripts install the Sun N1 Service Provisioning System 5.2 software into default destination directories that are subdirectories of the home directory for the software. The following table lists the directories that are installed for every Sun N1 Service Provisioning System 5.2 application, the Master Server, Local Distributor, Remote Agent, and CLI Client.

TABLE A–6 Directories Common to All Applications

Directory	Contents
\common	Common files for all applications
\common\jre	Bundled copy of the JRE for Windows
\common\lib	Library files common for some or all applications

The following table lists the directories installed for the Master Server.

TABLE A–7 Directories Installed for the Master Server

Directory	Contents
\server\config	Master Server Configuration files
\server\data	Master Server data files
\server\bin	Master Server Executable files
\server\lib	Master Server-specific library files
\server\postgres	Bundled copy of Postgres
\server\cygwin	Bundled subset of Red Hat cygwin
\server\tomcat	Bundled copy of Apache Tomcat
\server\webapp	Browser Interface Web Application
\server\setup	Miscellaneous files used to initialize the Master Server
\server\data\tmp	Master Server temporary directory with permissions set to 777
\server\README	Text license agreement

The following table lists the directories installed for the Local Distributor.

TABLE A–8 Directories Installed for the Local Distributor

Directory	Contents
\ld\config	Local Distributor configuration files
\ld\bin	Local Distributor executable files
\ld\lib	Local Distributor library files
\ld\data	Local Distributor-specific data
\ld\data\tmp	Local Distributor temporary directory

TABLE A–8 Directories Installed for the Local Distributor *(Continued)*

Directory	Contents
\ld\jvm\jre\bin	Local Distributor JRE proxy
\ld\README	Text license agreement

The following table lists the directories installed for the Remote Agent.

TABLE A–9 Directories Installed for the Remote Agent

Directory	Contents
\agent\config	Remote Agent configuration files
\agent\bin	Remote Agent executable files
\agent\lib	Remote Agent library files
\agent\data	Remote Agent-specific data
\agent\work	Default directory for execution of execNatives
\agent\data\tmp	Remote Agent temporary directory
\agent\jvm\jre\bin	Remote Agent JRE proxy
\agent\README	Text license agreement

The following table lists the directories installed for the CLI Client.

TABLE A–10 Directories Installed for the CLI Client

Directory	Contents
\cli\config	CLI configuration files
\cli\bin	CLI executable files
\cli\lib	CLI library files
\cli\data	CLI specific data
\cli\data\tmp	CLI temporary directory with permissions set to 777
\cli\README	Text license agreement

Cygwin

To facilitate interoperability with applications running on UNIX and Linux systems, the Windows version of the software includes a subset of the Red Hat `cygwin` UNIX environment. The following description of `cygwin` comes from the official Cygwin web site at `http://www.cygwin.com`.

Cygwin is a UNIX environment, developed by Red Hat, for Windows. It consists of two parts: - A DLL (cygwin1.dll) which acts as a UNIX emulation layer providing substantial UNIX API functionality. - A collection of tools, ported from UNIX, which provide UNIX/Linux look and feel. The Cygwin DLL works with all non-beta, non "release candidate", ix86 versions of Windows since Windows 95, with the exception of Windows CE.

Actions Performed by the Windows Installation Scripts

The Windows Master Server installation script performs the following actions:

- Copies all installation contents to the directories you specified.
- Sets up the registry entries for the proper mount points for `cygwin`.
- Registers the `cygipc` service.
- Registers the postmaster service with a dependency on the `cygipc` service.
- Registers the Master Server service with a dependency on the `postmaster` service.
- Creates a Start menu shortcut.
- If you selected SSL as a communications protocol, it runs scripts to generate the configuration files that are needed for SSL.

The Windows Local Distributor installation script performs the following actions:

- Copies the installation contents to the directories you specified.
- If you selected SSL as a communications protocol, it runs scripts to generate the configuration files that are needed for SSL.
- Registers the Local Distributor service.
- Creates a Start menu shortcut.
- If you requested that the installation script start the Local Distributor, it starts the Local Distributor.

The Windows Remote Agent installation script performs the following actions:

- Copies the installation contents to the directories you specified.
- If you selected SSL as a communications protocol, it runs scripts to generate the configuration files that are needed for SSL.
- Registers the Remote Agent service.
- Creates a Start menu shortcut.

The Windows CLI Client installation script performs the following actions:

- Copies the installation contents to the directories you specified.
- If you selected SSL as a communications protocol, it runs scripts to generate the configuration files that are needed for SSL.

- Creates a Start menu shortcut.

◆ ◆ ◆ APPENDIX B

Troubleshooting

This appendix provides troubleshooting information for installation and configuration of the Sun N1 Service Provisioning System 5.2.

- "Issues During Installation on Linux and UNIX Systems" on page 129
- "Issues on Solaris" on page 130
- "Issues During Installation on Microsoft Windows" on page 131
- "SSH Connectivity" on page 131

Issues During Installation on Linux and UNIX Systems

Package Corrupt Error When Installing on IBM AIX (6279820)

When installing the provisioning system on IBM AIX 5.2 or 5.3 systems, the following error might appear:

```
Checking the integrity of the package.
Error! This package seems to be corrupted.
Use another copy of the package.
```

Workaround: An issue in the IBM AIX operating system causes the installation to fail. Install one of the following patches to correct the problem:

- On IBM AIX 5.2 systems, install the IBM APAR IY68428 patch. For more information about this patch, see `http://www-1.ibm.com/support/docview.wss?uid=isg1IY68428`.
- On IBM AIX 5.3 systems, install the IBM APAR IY67843 patch. For more information about this patch, see `http://www-1.ibm.com/support/docview.wss?uid=isg1IY67843`.

Warning When Installing the JRE on IBM AIX

If the installation script detects any JRE instances already installed in the common directory on an AIX machine, the following warning appears:

```
WARNING: Overwriting the JRE can result in installation
problems when libraries from this JRE are cached by the
OS. If you have used, or are running another CenterRun
module that uses this JRE, you should stop that other
module, and run /usr/sbin/slibclean as root.

     Do you wish to continue installation?
(default: y) [y,n]
```

When a JRE is installed on an AIX machine, AIX caches native libraries from the JRE in memory. When these libraries are cached, they are locked on disk. Trying to install a new JRE over the locked libraries creates errors.

Do not install a new version of the JRE. When prompted to install the JRE, choose no. Then, provide a path to the JRE that is already installed on the machine.

Issues on Solaris

Incomplete Installation on Solaris

If the installation did not complete normally, you might need to manually remove packages in order to perform another installation.

▼ To Identify and Remove Solaris Packages

The packages for the Master Server are:

- SUNWspsms
- SUNWspsc1
- SUNWspsj1

The packages for the CLI are:

- SUNWspsc1
- SUNWspscl
- SUNWspsj1

1 List all Sun N1 Service Provisioning System packages installed on the system.

```
# pkginfo -c n1sps
```

Note – There might be multiple versions of packages installed.

2. **Determine which packages need to be removed.**

   `# pkginfo -l` *package_name*

3. **Remove the appropriate packages.**

   `# pkgrm` *package_name*

Issues During Installation on Microsoft Windows

Error When Installing on Windows

When installing on a Windows server, the following message appears:

```
/!\ Internal Error 2755.
```

The error occurs when you do not have write permissions on the directory in which the MSI packages are saved. To complete the installation, change the permissions on the directory to include write permissions for the user that is running the installation program. Restart the installation.

SSH Connectivity

Master Server Unable to Connect to Local Distributor Through an Intermediate Local Distributor

If the Master Server is unable to connect to another machine and displays a TTL expiry error after you use the Host Details page to update the configuration of that machine or any machine upstream, you might need to manually generate the `transport.config` file for some or all of the intermediate Local Distributors between that machine and the Master Server. Test the connection to each of the upstream Local Distributors of the problem machine by moving from the problem machine to the Master Server. For the Local Distributor to which you can successfully connect that is closest to the problem machine, regenerate the `transport.config` file and all of its downstream Local Distributors. Use the CLI Client `net.gencfg` command to generate `transport.config` files.

Unable to Connect to an Application Using SSH

If you are experiencing problems connecting to a machine after configuring the Sun N1 Service Provisioning System 5.2 to use SSH, follow the steps below to troubleshoot the problem.

How to Troubleshoot SSH Connectivity Issues

Before You Begin If you are using `ssh-agent`, complete this task from the same session as the session that you used to start the `ssh-agent`.

1 On the upstream machine, test the connection to the downstream machine.

- To test the machine immediately downstream from the upstream machine, use the following command:

  ```
  # ssh target-IPaddress ls -l
  ```

  target-IPaddress is the IP address of the machine that is the furthest downstream that you want to test.

- If you are using `ssh-agent`, to test a machine that is more than one other machine downstream from the machine on which you are running the `ssh-agent`, use the following command:

  ```
  # ssh -A target-IPadress-parentmachine
  ssh -A target-IPadress-parentmachine ssh -A target-IPaddress ls -l

  # ssh -A ssh -A target-machine-n-IPaddress ssh -A target-machine-2-IPaddress
  ssh -A target-machine-1-IPaddress ssh -A target-IPaddress ls -l
  ```

  target-machine-n-IPaddress are the IP addresses of the upstream Local Distributor machines of the machine being tested in the specified in order. For example, `1` is the machine that is closest to the machine being tested and `n` is the machine that is right before the Master Server. *target-IPaddress* is the IP address of the machine that is the furthest downstream that you want to test.

  target-IPadress-parentmachine is the IP address of any machine that is between the upstream machine and the downstream machine for which you are testing connectivity.

  If you are prompted for information, supply the information. Try the test again.

  If you are not prompted for information, continue to the next step.

2 On the upstream machine, in the `logger_config.xml` file, before the `<root>` section, insert the following lines to enable logging with `priority="debug"`:

```
<category name="SSH.STDERR">
<priority value="debug" />
</category>
<category name="com.raplix.rolloutexpress.net.transport.SshClientConnectionHandler">
<priority value="debug" />
</category>
```

Wait for the upstream machine to read the log file updates.

3 Test the connection again using the command that you used in Step 1.

Examine the log output on the command line and in the `SSH.STDERR` log. Correct any problems found in the log files and try the test again.

Examine the application log output on the upstream machine for the SSH command line you used to invoke the downstream application and the `stderr` output of the SSH command. Correct any problems identified by the logged messages and try the test again.

If you do not find any problems in the log files, the upstream machine might be connecting properly to the downstream machine, but the application is not starting properly. Continue to the next step.

4 Examine the ROX log file for errors starting the application on the downstream machine.

- On Solaris OS systems, examine the `/var/tmp/ROX`*appnumbers*`.log` file.
- On all other systems, examine the `/tmp/ROX`*appnumbers*`.log` file.

app is the application on the downstream machine that you are testing. Use Agent for a Remote Agent, Dist for a Local Distributor, and Proxy for a CLI Client. *numbers* are randomly generated numbers that are included in the file name.

5 Correct any errors found in the log file.

Glossary

abstract component A component that serves only as a base component for other components to extend. An abstract component cannot be installed and only an abstract component is permitted to declare abstract child elements.

call compatibility A compatibility type for system service components. This compatibility is also called API compatibility or interface compatibility.

category A general class in which you can group objects that are stored in multiple folders.

child component A component that is referenced by a container component. Also called *contained component*.

See also *container component*.

comparison A feature that searches for and identifies differences between hosts and component models. The Sun N1 Service Provisioning System supports these three types of comparisons:

- **Model to model –** Examines the deployment repository and history that is stored on the master server for two hosts and reports any differences
- **Model to install –** Compares what the master server reports is installed on a host to what is actually on the host and reports any differences
- **Install to install –** Examines the contents of two hosts' file systems and reports any differences

component A logical grouping of source information that defines an application. A component also includes a set of instructions that specifies how to manage the source information.

The XML representation of a component includes the following:

- List of resources used by the application
- Installation steps
- Uninstallation steps
- Dependencies

component compatibility A situation where a component can be safely replaced by another. The Sun N1 Service Provisioning System supports two kinds of component compatibility: call compatibility and install compatibility.

component inheritance The means by which a component obtains attributes and behavior from another component. When you create a component, it inherits any variables, snapshots, and procedures from the associated component type.

component procedure A program in a component that controls deployment of the component, such as installation, uninstallation, management, and capturing snapshots. Management procedures are defined in the control block.

component repository A location on the master server where components and their resources are checked in.

component type A special kind of component that encapsulates behavior that can be reused by other components. A component can inherit the behavior of a component type by extending from it.

component variable A user-definable name-value pair that is used to make parts of a component accessible and configurable by objects that are external to the component.

composite component A component that contains only references to other components, both simple and composite. A composite component cannot contain any resources.

composite plan A plan that is composed solely of subplans, which can be simple or composite subplans. A composite plan is not directly targeted, as each subplan can run on a different set of targets.

configuration generation engine A software engine on the master server that replaces substitution variable references with the appropriate variable setting values. The engine interacts with the host repository and component repository to resolve values any time that you run a plan to deploy a component.

contained component A component that is referenced by other components.

container component A component that contains references to other components.

control A procedure defined by a component that can be used to manage the deployed applications. For example, a control might be used to start or stop an application. Also called *control service*.

deployment Using a plan or component procedure to act on a component. The component's lifecycle includes installation, uninstallation, and application management.

direct-run procedure A component procedure that can be run directly from the component by using the browser interface.

downstream In the Sun N1 Service Provisioning System 5.2 network hierarchy, the server that is further from the master server. For example, the master server connects downstream to a local distributor. Any remote agents connected to the local distributor are downstream from the local distributor.

`execNative` **call** An optional call out to custom scripts from the XML of a plan or component.

execution plan See *plan*.

extend To base a component on a component type so that the component inherits variables and procedures that are defined by the component type. The component can override variable values and procedure definitions defined by its associated component type.

final component A component that cannot be extended by another component.

folder Directory-like containers that enable you to apply permissions to and organize components, plans, and subfolders.

gold server A reference server that contains files, directories, and other resources that make up an application and that checks in these resources to the master server.

host A server that is managed by the Sun N1 Service Provisioning System.

host search A query run on the host repository that yields a list of hosts whose attributes match those specified by the query. For example, you can use host searches to create a list of hosts that have the same host type, that run the same applications, and that are configured with the same subnet masks.

host set A user-defined, logical grouping of hosts that share one or more common attributes, such as physical location or functional group. Use a host set to quickly and easily update applications on all hosts in the set. You can also use a host set to perform model-to-install comparisons between two hosts.

host type A base class of servers that is bound by a set of common attributes, all of which are user-defined. You can use host types to categorize hosts into logical groupings and to facilitate host searches.

install compatibility A compatibility type for component types. This compatibility is also called structural compatibility.

Java Runtime Environment (JRE) A subset of the Java Development Kit (JDK®) for users and developers who want to redistribute the runtime environment. The Java runtime environment consists of the Java virtual machine (JVM), the Java core classes, and supporting files.

Java Virtual Machine (JVM) The part of the Java runtime environment (JRE) responsible for interpreting bytecodes.

Jython An implementation of the high-level, dynamic, object-oriented language, Python, seamlessly integrated with the Java platform. The predecessor to Jython, JPython, is certified as 100% pure Java.

label A means of marking a component version beyond the Sun N1 Service Provisioning System version number. For example, a component version number describes the version of the component. A label can describe the version of the application that the component represents.

local distributor The application that is installed on a server. The Local Distributor application acts as a link between other servers in the Sun N1 Service Provisioning System in the following ways:

- master server to remote agents

- master server to other local distributors
- local distributor to remote agents

Local distributors maximize bandwidth efficiency and speed, and can also provide secure network connections for navigating restricted environments.

master server The application that is installed on a server that manages the Sun N1 Service Provisioning System. The Master Server application can connect to any of the data center environments managed by the Sun N1 Service Provisioning System. The master server provides centralized data storage, data processing, and user interfaces.

modeling To create components and plans that represent an application that you want to deploy with the Sun N1 Service Provisioning System.

nested component A contained component that, when installed, can provide its services only to its container component. A nested contained component defines a finer-grained unit of functionality required by the container component, but is not otherwise useful to other components.

network protocol A way to transmit data between devices on a network. The Sun N1 Service Provisioning System 5.2 uses TCP/IP, SSH, and SSL.

notification email An email sent by the Sun N1 Service Provisioning System to advise that a system, administrative, or custom event has occurred. The system administrator specifies the rules used to determine when notification emails are sent and the email addresses to which the email is sent.

notification rule The criteria used by the Sun N1 Service Provisioning System to determine whether an email notification is sent. The system administrator defines the criteria that that is used to determine when an email notification is sent.

parent component A component that contains references to other components. Also called *container component*.

See also *contained component*.

physical host A physical server that is connected to the network. Within the provisioning system, a physical host can act as a remote agent or a local distributor.

plan A sequence of instructions that is used to manipulate one or more components. A plan can also be a sequence of other plans, which enables common instruction sequences to be shared between one or more plans.

plan executor The software engine on the master server that runs preflights and deployments.

preflight The simulated execution of a plan to a simulated UNIX environment that finds and reports any errors or potential errors that might affect the deployment. A preflight always precedes a deployment, but you can run a preflight as a standalone operation.

procedure See *component procedure*.

provisioning system The software applications that, when installed on servers, form the Sun N1 Service Provisioning System 5.2.

remote agent The application that is installed on any server in the Sun N1 Service Provisioning System to which components are deployed. The Remote Agent application manages tasks, such as installing software, controlling services, and collecting information to deliver to the master server.

resource A file that is deployed to a host when a plan is executed. The file might be a directory, a symbolic link, or another kind of file.

resource descriptor file An XML file that specifies the owner, group, and permission settings to use for the files and directories that comprise the resource of a simple component. By using a resource descriptor file, you can override the permissions that are determined at component check-in time.

server A computer that manages resources and supplies services to a client. In the Sun N1 Service Provisioning System 5.2, a server is a computer on which one of the Sun N1 Service Provisioning System applications has been installed.

session A period of time that is initiated when you log in. A session persists until you log out or inactivity causes the session to expire. Logically, a session represents the authenticated credentials of a particular user. A session is used to identify the user throughout a series of related requests without reauthentication.

session variable A variable that is associated with a user session. The user can change session variable values for each login session. Session variable values can also be securely saved for reuse in subsequent sessions.

simple component A component that contains a single resource. A simple component cannot contain references to other components.

simple plan A sequential list of steps that are executed on a particular set of target servers. A simple plan does not contain or call other plans.

snapshot A capture of the resources that are stored on a host during a deployment. The snapshot is used when performing comparisons between a host and its model on the master server (model-to-install).

step An instruction that can be part of a plan or a component.

substitution variable A variable that appears in plans, components, or configuration files that is substituted by the configuration generation engine during deployment.

system service A component that is automatically deployed to all applicable hosts when the hosts are prepared. System services define utility controls and resources that can be used by other components.

targetable component A component that creates a host that serves as a deployment target for other components when it is installed. When a targetable component is uninstalled, the host it created is automatically deleted.

top-level component A contained component that, when installed, can be used by any component just as if it had been directly installed by a plan. A top-level contained component defines services that will be used by the container component as well as by other components.

upstream In the Sun N1 Service Provisioning System network hierarchy, the server that is closer to the master server. For example, the master server is upstream from the local distributor. The docal distributor is upstream from any remote agents that are connected to that local distributor.

variable See *component variable*.

variable settings A collection of variable values that can be used to override the default values of one or more component variables. Based on the variable settings that you use, you can specify different values for component variables. You specify the variable settings to use when you run a plan.

virtual host Services that act as a host for other services. For example, a virtual host can represent an application server that acts as a host for web applications.

XML schema The language used by the Sun N1 Service Provisioning System to create plans and components.

Index

A

acquiring, plug-ins, 22

C

cipher suites
  description, 87-88
  list, 99-100
CLI (Command Line Interface) Client
  application directory structure, 119-120, 125
  description, 20
  hardware requirements, 32-33
  installing, 43-45, 56-57
  operating system requirements, 25-32
  running, 51
  uninstalling, 113-114, 114-115, 115
  upgrading, 106
CLI Clients, worksheets, 42
`config.properties` file, edit for SSL, 94-95
creating keystore file, 64-65
creating keystore password, 64-65

D

directory structure
  Linux and UNIX systems, 117-120
  Windows, 123-125

H

hardware requirements, 32-33
HTTP, selecting during installation, 38-39
HTTPS
  creating keystore file, 64-65
  creating keystore password, 64-65
  encoding keystore password, 65-67
  installing, 38-39
  reconfiguring from HTTP, 67-68
  selecting during installation, 38-39
  updating keystore file after installation, 65-67

I

installating, security configuration during, 37-39
installing
  HTTPS, 38-39
  interactively, 43-45, 53-55, 56-57
  JRE (Java Runtime Environment), 35-36
  overview, 17-18
  pre-install decisions, 35-42
introduction, plug-ins, 22

J

Java Virtual Machine, 74
jexec, 74
JRE (Java Runtime Environment), 35-36
  troubleshooting, 130
Jython, 37
  requirements, 34

K

keystore
  creating, 91-93
  description, 88-89
keystore file
  creating, 64-65
  manually updating after installation, 65-67
keystore password
  configuring password after installation, 65-67
  creating, 64-65

L

Linux, system settings, 28
Local Distributor
  application directory structure, 119, 124-125
  description, 19
  hardware requirements, 32-33
  installing, 43-45, 56-57
  operating system requirements, 25-32
  running, 51
  uninstalling, 114-115, 115
  upgrading, 111-112
Local Distributors, worksheets, 41
locales, requirements, 34

M

Master Server
  application directory structure, 118, 124
  description, 19
  installing, 43-45, 53-55
  operating system requirements, 25-32
  running, 51
  system requirements, 32
  uninstalling, 113-114, 114-115, 115
  upgrading, 106-111
  worksheets, 40-41
migrating, *See* upgrading

N

`no locks available`, 28
non-interactive installing
  instructions, 57-58
  Remote Agent, 46-48
  Remote Agent sample parameters file, 120-123
  Windows variable values, 61-62

O

operating system configuration, 26-28
operating system requirements, 25-32
  CLI (Command Line Interface) Client, 25-32
  Local Distributor, 25-32
  Master Server, 25-32

P

plug-ins
  acquiring, 22
  introduction, 22
product overview, 18-20

R

raw (TCP/IP), selecting during installation, 37-38
Remote Agent, 25-32
  application directory structure, 119, 125
  description, 19
  hardware requirements, 32-33
  installing
    interactively, 43-45, 56-57
    non-interactively, 46-48, 57-58
    remotely, 48-51, 59-61
  operating system requirements, 25-32
  running, 51
  sample parameters file, 120-123
  uninstalling, 114-115, 115
  upgrading, 111-112
  Windows variable values, 61-62
Remote Agents, worksheets, 41
remotely installing
  instructions, 59-61
  Remote Agent, 48-51
  Windows variable values, 61-62

requirements
Jython, 34
locales, 34
running applications, 51

S

security
database, 103
JVM security policy, 101-103
overview, 20-21
Solaris, system settings, 26-27
SSH (Secure Shell)
additional security, 74
commands, 86
configuration process, 75
description, 21
jexec, 74
overview, 72-74
selecting during installation, 37-38
system requirements, 34
troubleshooting, 131-133
SSL (Secure Socket Layer)
cipher suites
description, 87-88
list, 99-100
configuration examples, 95-99
configuration process, 91-95
creating keystores, 91-93
description, 21
editing `config.properties` file, 94-95
overview, 87-91
selecting during installation, 37-38
starting applications, *See* running applications
SUSE Linux, `nolocks`, 28
system requirements
hardware, 32-33
Linux system settings, 28
operating system configuration, 26-28
operating system versions, 25-32
operating systems
patches, 28-32
Solaris system settings, 26-27
SSH (Secure Shell), 34
web browsers, 33

U

uninstalling, 113-115, 115
upgrading
instructions, 106-111, 111-112
overview, 106

W

web browser requirements, 33
worksheets
all applications, 39-40
CLI Clinets, 42
Local Distributors, 41
Master Server, 40-41
Remote Agents, 41