N1 Service Provisioning System 4.1 Installation Guide

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

Part No: 817–4820–10
February 2004

040205@7940

Contents

Tables

Examples

Preface

The N1 Service Provisioning System 4.1 Installation Guide describes how to install and upgrade the N1™ Service Provisioning System 4.1 on the Solaris™ Operating System (OS), Red Hat Linux, IBM AIX, and Windows 2000.

Who Should Use This Book

This book is intended for system administrators responsible for installing and configuring the N1 Service Provisioning System 4.1.

How This Book Is Organized

The N1 Service Provisioning System 4.1 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 Solaris OS, Red Hat Linux, and IBM AIX systems.
- Chapter 5 describes the steps to install the software on Windows.

- Chapter 6 describes the tasks necessary to configure the software to communicate using SSH.
- Chapter 7 describes the tasks necessary to configure the software to communicate using SSL.
- Chapter 8 describes how to configure the JVM™ [1] security policy.
- Chapter 9 describes the steps to upgrade the software.
- Chapter 10 describes the steps to uninstall the software.
- Chapter 11 describes the steps to back up and restore 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 N1 Service Provisioning System 4.1.

- *N1 Service Provisioning System 4.1 Release Notes*
- *N1 Service Provisioning System 4.1 User's Guide*
- *N1 Service Provisioning System 4.1 Reference Guide*

Accessing Sun Documentation Online

The docs.sun.com℠ Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Ordering Sun Documentation

Sun Microsystems offers select product documentation in print. For a list of documents and how to order them, see "Buy printed documentation" at `http://docs.sun.com`.

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

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new terms, or terms to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

Shell Prompts in Command Examples

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

TABLE P–2 Shell Prompts

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

CHAPTER 1

N1 Service Provisioning System 4.1 Overview

This chapter provides an overview of the tasks required to install and configure the N1 Service Provisioning System 4.1. This chapter also contains an overview of the applications included in the N1 Service Provisioning System 4.1 and the types of network protocols that you can use for additional security.

This chapter discusses the following topics:

Overview of the N1 Service Provisioning System 4.1 Installation Tasks

The task map below describes the tasks necessary to properly install and configure the N1 Service Provisioning System 4.1.

TABLE 1–1 Task Map: Installing the N1 Service Provisioning System 4.1

Task	Description	For Instructions
Review system requirements.	Determine whether your system meets the minimum requirements to install.	Chapter 2
Gather information for installation.	Before installing, gather the information that you will need to install the product.	Chapter 3

TABLE 1–1 Task Map: Installing the N1 Service Provisioning System 4.1 *(Continued)*

Task	Description	For Instructions
(Optional) Create a user account.	You can create a special, operating system user account to be used by N1 Service Provisioning System 4.1.	The documentation for your operating system.
(Optional) Install Jython on CLI Client machines.	You may 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`.	The Jython web site.
Install the applications.	You will install each of the N1 Service Provisioning System 4.1 applications individually using the appropriate installation script provided on the product media.	Chapter 4 Chapter 5
(Optional) Configure SSH.	If you plan to access the Master Server on the Internet, you can increase the Master Server security by configuring the N1 Service Provisioning System 4.1 to use SSH to communicate with that server.	Chapter 6
(Optional) Configure SSL.	If you want to provide the maximum security for communication among the applications, configure the applications to use SSL when communicating. SSL support is based on self-signed digital certificates that your organization can issue to itself.	Chapter 7
(Optional) Configure the JVM Security Policy.	If you do not use SSL to provide security for communication among application, you can configure the JVM security policy so that the applications accept only connections from localhost. This setup provides a minimum level of security.	Chapter 8

Overview of N1 Service Provisioning System 4.1 Applications

The N1 Service Provisioning System 4.1 is a distributed software platform that includes the following special-purpose applications:

- "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 – One or more small management applications that perform operations on individual hosts. Every host that you want to be controlled by the N1 Service Provisioning System 4.1 must have the Remote Agent application.

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

Master Server

The Master Server runs on Solaris OS, Red Hat Linux, or Microsoft Windows 2000 Server and Microsoft Windows 2000 Advanced Server systems. The Master Server is a central server that does the following.

- Manages a database that identifies all hosts registered in the provisioning software
- Stores components and plans in a repository
- Performs version control on the objects stored in the repository
- Authenticates IT operators and ensures that only authorized users perform specific operations
- Includes special-purpose engines for performing tasks such as dependency tracking and deployments
- Provides both an HTML 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 can send one copy of a component to a Local Distributor, which then replicates the component for installation on a collection of systems.
- Minimize firewall reconfigurations. If a firewall stands between the Master Server and a collection of systems, administrators can open the firewall only for the systems running Local Distributors, rather than for every system involved in a deployment.
- Minimize the load to the Master Server during large scale deployments.

Remote Agent

A Remote Agent is an application that runs on every system being managed by the N1 Service Provisioning System 4.1. Remote Agents perform the tasks requested by the Master Server. The Remote Agents are supported on the Solaris OS, Red Hat Linux, IBM AIX, and Microsoft Windows 2000 platforms. 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 in component models

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 systems. 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 system 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 N1 Service Provisioning System 4.1 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 N1 Service Provisioning System 4.1 supports the following protocols:

- Raw TCP/IP
- Secure Shell
- Secure Sockets Layer

You can tailor your network security to meet the needs of your particular network topology. For example, say 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 for the remote data center, so that all communication over the Internet is secure. The Local Distributor might use raw TCP/IP to communicate with the Remote Agents because all the communication over the local network is secure. For more information about how to configure the different protocols, read Chapter 6 and Chapter 7.

Raw TCP/IP

Raw TCP/IP is standard TCP/IP without additional encryption or authentication. The advantage of raw TCP/IP is that it requires no additional set-up and configuration. If your data center network is protected by a firewall and secured from intrusion, using raw TCP/IP provides a convenient method for communication among N1 Service Provisioning System 4.1 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 N1 Service Provisioning System 4.1 applications to communicate using ssh. The N1 Service Provisioning System 4.1 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 N1 Service Provisioning System 4.1 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.

CHAPTER 2

System Requirements for the N1 Service Provisioning System 4.1

This chapter lists the system requirements for installing and using the N1 Service Provisioning System 4.1. This chapter discusses the following topics:

General System Requirements

This section lists requirements for installing and using the N1 Service Provisioning System 4.1:

Supported Operating Systems

You can install the N1 Service Provisioning System 4.1 Master Server on systems that are running the following operating systems:

- Solaris 8 or Solaris 9 systems
- Red Hat Linux 7.2, 7.3, 8.0 and Red Hat Advanced Server 2.1
- Microsoft Windows 2000 Server and Microsoft Windows 2000 Advanced Server

You can install the N1 Service Provisioning System 4.1 Remote Agent, Local Distributor, and CLI Client on systems that are running the following operating systems:

- Solaris 2.6, Solaris 7, Solaris 8, or Solaris 9 systems
- Red Hat Linux 7.2, 7.3, 8.0 and Red Hat Advanced Server 2.1
- IBM AIX 4.3.3, 5.1, 5.2
- Microsoft Windows 2000 Server and Microsoft Windows 2000 Advanced Server

Supported Web Browsers

The following table lists the web browser requirements for the N1 Service Provisioning System 4.1 Web Interface.

TABLE 2–1 Web Browser Requirements for the HTML User Interface

Platform	Browser
Solaris	Netscape Navigator 6.2.2, Netscape Navigator 7.0
Red Hat	Netscape Navigator 6, Netscape Navigator 7.1
Windows	Internet Explorer 5.5 and 6, Netscape Navigator 6, Netscape Navigator 7.1

Required Operating System Patches

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

TABLE 2–2 Required Patches for Supported Operating Systems

OS Version	Required Patches
Solaris 2.6	105633-56
	107733-09
	105568-23
	105210-38
	108091-03
	106842-09
	106841-01
	105181-33
	105591-09
	106125-11
	112542–01

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

OS Version	Required Patches
Solaris 7	106980-16
	106541-16
	107544-03
	106950-13
	106327-08
	106300-09
Solaris 8	None
Solaris 9	None
IBM AIX 4.3.3.0	AIX 4330-09 maintenance level (APAR IY22024)
IBM AIX 5.1.0.0	AIX 5100-01 maintenance level (APAR IY21957)
	APAR IY19375
Red Hat Linux 7.2, 7.3, or 8.0	None
Red Hat Linux Advanced Server 2.1	None
Windows 200 Server or Windows 2000 Advanced Server	Service Pack 3

Requirements for SSH

If you want to use SSH for secure connections on Solaris OS, Red Hat Linux, or IBM AIX systems, you must have SSH protocol version 2 installed on each machine that you want to use SSH.

Requirement 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 N1 Service Provisioning System 4.1 has been internationalized to install and run in localized environments. You will need to adhere to the following requirements if you want to run the software in a localized environment.

- 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 in the same locale as the Master Server.

- The software accepts only ASCII characters for file names, directory names, and other input.

Application Requirements

This section lists the requirements for installing and using each of the N1 Service Provisioning System 4.1 applications:

System Requirements for Applications on Solaris OS Systems

Solaris Master Server

The Solaris Master Server requires the Solaris 8 or Solaris 9 operating system. The system must meet the following hardware requirements:

- 450 MHz single or multiple CPU, SPARC® hardware only.
- At least 1 GByte RAM.
- 2 GBytes HD free space. The repository space requirement is determined by the size of your deployed applications.

The following table lists the `/etc/system` settings required for a Solaris system running the Master Server.

Note – If you are using the Solaris 9 Operating System, you cannot change the values for `shmsys:shminfo_shmmin` and `shmsys:shminfo_shmseg`. The default values for these settings are acceptable.

TABLE 2–3 Solaris `/etc/system` Settings

Variable	Minimum Value	Recommended Value
`shmsys:shminfo_shmmax`	0x20000000[1]	0x20000000
`shmsys:shminfo_shmmin`	1	1
`shmsys:shminfo_shmmni`	2	256
`shmsys:shminfo_shmseg`	1	256
`semsys:seminfo_semmni`	32	512
`semsys:seminfo_semmns`	512	512
`semsys:seminfo_semmsl`	17	32

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

Solaris Local Distributor, Remote Agent, and CLI Client

The Solaris Local Distributor requires the Solaris 6, Solaris 7, Solaris 8, or Solaris 9 operating system. The system must meet the following hardware requirements:

- 400 MHz single or multiple CPU, SPARC hardware only.
- At least 256 MBytes RAM.
- 1 GBytes HD free space. The cache space requirement is determined by the size of your deployed applications.

System Requirements for Applications on Red Hat Linux

The `bc` command must be in the user's path when the Red Hat Linux Master Server installation begins. 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.

Red Hat Linux Master Server

The Red Hat Linux Master Server requires one of the following versions of Red Hat Linux:

- Red Hat Linux 7.2, 7.3, or 8.0
- Red Hat Advanced Server 2.1

The system must meet the following hardware requirements:

- 1 GHz single or multiple CPU, Intel x86-compatible hardware only.

- At least 1 GByte RAM.
- 2 GBytes HD free space. The repository space requirement is determined by the size of your deployed applications.

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

TABLE 2–4 Red Hat System Settings

System Parameter	Minimum Value	Recommended Value
`shmall` in `/proc/sys/kernel/shmall`	536870912 (512Mb)	2147483647 (2048Mb)
`shmmax` in `/proc/sys/kernel/shmmax`	536870912 (512Mb)	2147483647 (2048Mb)

Red Hat Linux Local Distributor, Remote Agent, and CLI Client

The Red Hat Linux Local Distributor, Remote Agent, and CLI Client require one of the following versions of Red Hat Linux:

- Red Hat Linux 7.2, 7.3, or 8.0
- Red Hat Advanced Server 2.1

The system must meet the following hardware requirements:

- 1 GHz single or multiple CPU, Intel x86-compatible hardware only.
- At least 1 GByte RAM.
- 1 GByte HD free space. The cache space requirement is determined by the size of your deployed applications.

System Requirements for Applications on IBM AIX

The IBM AIX Local Distributor, Remote Agent, and CLI Client require the AIX 4.3.3, 5.1.0, or 5.2.0 operating system. The system must meet the following hardware requirements:

- 400 MHz single or multiple CPU, pSeries hardware only.
- At least 256 MBytes RAM.
- 1 GByte HD free space. The cache space requirement is determined by the size of your deployed applications.

System Requirements for Applications on Windows 2000

When running the Windows Master Server, Remote Agent, Local Distributor, or CLI Client, you must have enough space in the home directory for creating intermediate files. The space requirement is approximately equivalent to the size of the files in the that are installed to run the application.

Windows 2000 Master Server

The WindowsMaster Server requires one of the following versions of Windows:

- Windows 2000 Server
- Windows 2000 Advanced Server

The system must meet the following hardware requirements:

- 1 GHz single or multiple CPU, Intel x86-compatible hardware only.
- At least 1 GByte RAM.
- 2 GBytes HD free space The repository space requirement is determined by the size of your deployed applications.

Windows 2000 Local Distributor, Remote Agent, and CLI Client

The Windows Local Distributor, Remote Agent, and CLI Client require one of the following versions of Windows:

- Windows 2000 Server
- Windows 2000 Advanced Server

The system must meet the following hardware requirements:

- 1 GHz single or multiple CPU, Intel x86-compatible hardware only.
- At least 1 GByte RAM.
- 1 GByte HD free space The repository space requirement is determined by the size of your deployed applications.

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 N1 Service Provisioning System 4.1. This chapter covers the following topics:

Configuration Decisions

The installation program prompts you for configuration information for the N1 Service Provisioning System 4.1. Use the sections below to make configuration decisions before you begin the installation.

The Java Runtime Environment

When installing on Solaris OS, Red Hat Linux, or IBM AIX systems, 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.

If you are installing on a Red Hat Linux system, the installation script searches your machine for an instance of the JRE in the default location. If the JRE is not installed in the default location, you must install the JRE. 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 Solaris OS or IBM AIX systems, if 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 N1 Service Provisioning System 4.1, the installation script sets the `JRE_HOME` variable to the JRE that you specified. The installation script also creates a symbolic link, *N1SPS4.1-home*`/common/jre`, which points to the JRE directory. By creating a symbolic link, the N1 Service Provisioning System 4.1 applications use the JRE without changing its location, which other applications might depend upon.

Note – You should install the bundled JRE only once per 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 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.

Network Protocol

The installation program prompts you to choose a network protocol for communication among the software applications. Before you install the software, decide which encryption method to use, TCP/IP, SSH, or SSL. If you select SSL, you must also specify which cipher suite to use, encryption with no authentication or encryption with authentication.

For more information about network protocols, see “Network Protocols” on page 20.

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.

Worksheet for All Applications

The installation scripts for each of the N1 Service Provisioning System 4.1 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 N1 Service Provisioning System 4.1 applications.

TABLE 3–1 Worksheet for All Applications

Information Needed	Your Answer
In what base directory do you want the software installed?	
If the JRE is already installed on the machine, what is the path to the JRE? For example `/usr/local/jre` or the value of your `JAVA_HOME` environment variable.	
Which user do you want to own the application that you are installing?	
On Solaris OS, Red Hat Linux, and IBM AIX systems, which group do 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 Worksheet for the Master Server

Information Needed	Your Answer
What is the hostname or IP address for the Master Server machine?	

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

Information Needed	Your Answer
What is the IP port number the CLI Client should use to connect to the Master Server?	
What is the hostname or IP address of the SMTP mail server for the software to use to send notification mail messages?	
What do you want the subject line of email notifications from the software to be?	
What is the name of the user account (username) from which email notifications are sent?	
What is the name of the user account that the software should use when executing native commands?	
What is the port number on which the Postgres database will listen?	
What is the port number on which the Web Interface will be available?	
Do 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. An entry will be made in your `crontab` file to optimize the database every day.	Y/N HH:MM

Worksheet for Local Distributors

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

TABLE 3–3 Worksheet for Local Distributors

Information Needed	Your Answer
What is the IP address or hostname for the Local Distributor machine?	
What is the port number on which this Local Distributor will listen?	

Worksheet for Remote Agents

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

TABLE 3–4 Worksheet for Remote Agents

Information Needed	Your Answer
What is the IP address or hostname on which the Remote Agent will run?	
What is the port number on which this Remote Agent will listen?	

Worksheet for CLI Clients

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

TABLE 3–5 Worksheet for CLI Clients

Information Needed	Your Answer
What is the IP address or hostname of the Master Server for the command line user interface?	
What is the IP port number of the Master Server?	
If Jython is already installed on this machine, what is the path to Jython? For example `/usr/local/jython`.	

CHAPTER 4

Installing the N1 Service Provisioning System 4.1 on Solaris OS, Red Hat Linux, and IBM AIX Systems

This chapter describes the steps to install the N1 Service Provisioning System 4.1 on Solaris OS, Red Hat Linux, and IBM AIX systems. This chapter discusses the following topics:

Installing the N1 Service Provisioning System 4.1

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 N1 Service Provisioning System 4.1 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 N1 Service Provisioning System 4.1 on Solaris OS, Red Hat Linux, and IBM AIX Systems

Before You Begin Review the task map in Table 1–1. Complete any necessary tasks prior to installing the applications.

Steps

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.

2. **Insert the CD.**

   - If you are installing the software on a Solaris OS system, insert the N1 Service Provisioning System 4.1: Solaris CD.
   - If you are installing the software on IBM AIX or Red Hat Linux, insert the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD.

3. **If you are installing the software on the Solaris OS, determine whether you want to install the software as packages or as files.**

   You can choose to install the Master Server and the CLI Client applications as packages or as files. The installation program you select depends on whether you want to install the applications as packages or as files. If you do not want to run the installation as the root user, install the software as files.

4. **Change to the directory on the software CD where the installation script is located.**

   `% cd /`*script-directory*

   script-directory is one of the following values:

   - solaris – on the N1 Service Provisioning System 4.1: Solaris CD
   - aix – on the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD
   - linux – on the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD

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

   `% cr_`*app*`_`*opsystem*`_4.1.sh`

   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 – installs the application on the Solaris OS
   - aix – installs the application on IBM AIX
   - linux – installs the application on Red Hat Linux

Note – The installation script that installs the Solaris Master Server as packages is named `cr_ms_solaris_pkg_4.1.sh`. The installation script that installs the Solaris CLI Client as packages is named `cr_cli_solaris_pkg_4.1.sh`. Begin these scripts to install the package version of the Solaris Master Server and CLI Client.

6. **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 that file. Note the location of the file so that you can view it later.

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

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

8. **If you are running the installation script from the CD, if the script prompts you to start the application, answer `n`.**

Non-Interactive Installation of a Remote Agent on Solaris OS, Red Hat Linux, and IBM AIX 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 provided in the parameters file.

▼ How to Non-Interactively Install a Remote Agent on Solaris OS, Red Hat Linux, and IBM AIX 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.

Steps

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. **Insert the CD.**

   - If you are installing the Remote Agent on a Solaris OS system, insert the N1 Service Provisioning System 4.1: Solaris CD.
   - If you are installing the Remote Agent on IBM AIX or Red Hat Linux, insert the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD.

3. **Change to the directory on the software CD where the installation script is located.**

   `% cd /`*script-directory*

   script-directory is one of the following values:

   - solaris – on the N1 Service Provisioning System 4.1: Solaris CD
   - aix – on the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD
   - linux – on the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD

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

   `% cp cr_ra_`*opsystem*`_41.sh` *RA-machine*`/`

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

   - solaris – installs the application on the Solaris OS
   - aix – installs the application on IBM AIX
   - linux – installs the application on Red Hat Linux

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 *N1SPS4.1-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_41_remote_params.sh` sample parameters file are in "Sample Remote Agent Parameters File for Solaris OS, Red Hat Linux, and IBM AIX" on page 110.

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

   N1SPS4.1-MasterServer-home is the directory where you installed the Master Server.

6. **Start the installation script.**

```
% cr_ra_opsystem_41.sh -paramfile parameters-file.sh
```

opsystem is one of the following values:

- solaris – installs the application on the Solaris OS
- aix – installs the application on IBM AIX
- linux – installs the application on Red Hat Linux

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.

Remote Installation of Remote Agents on Solaris OS, Red Hat Linux, and IBM AIX 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 `/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 Solaris OS, Red Hat Linux, and IBM AIX

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 hostname 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, at the command line, or use the default values provided by the installation script. Following are the required environment variables and their default values.

- `CR_RA_INSTALLER_USER=root` – The installation program installs the Remote Agent with the root user as the owner of the application.
- `CR_RA_INSTALLER_WORKDIR=/tmp` – A copy of the installation script is saved in the `/tmp` directory on the target machine.
- `CR_RA_INSTALLER_LEAVEFILES=no` – The installation program deletes any files copied to the working directory when the installation program completes.
- `CR_RA_INSTALLER_HOSTS=host1,host3.enterprise.com,10.10.0.207` – If you do not supply hostnames on the command line or as an environment variable, the installation script exits with an error.

Steps

1. **On the Master Server machine, insert the CD.**
   - If you are installing the Remote Agent on a Solaris OS system, insert the N1 Service Provisioning System 4.1: Solaris CD.
   - If you are installing the Remote Agent on IBM AIX or Red Hat Linux, insert the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD.
2. **Change to the directory on the software CD where the installation script is located.**

   `% cd /`*script-directory*

   script-directory is one of the following values:
   - solaris – on the N1 Service Provisioning System 4.1: Solaris CD
   - aix – on the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD
   - linux – on the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD
3. **Copy the installation script to the Master Server.**

   `% cp cr_ra_`*opsystem*`_4.1.sh` *N1SPS4.1-MasterServer-home*`/server/bin`

   N1SPS4.1-MasterServer-home is the directory where you installed the Master Server and *opsystem* is one of the following values:
   - solaris – installs the application on the Solaris OS
   - aix – installs the application on IBM AIX
   - linux – installs the application on Red Hat Linux
4. **Change directories to where the scripts are located.**

   `% cd` *N1SPS4.1-MasterServer-home*`/server/bin`

   N1SPS4.1-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 N1 Service Provisioning System 4.1. When you install the Master Server, a parameters file is installed. The file is named *N1SPS4.1-MasterServer-home*`/server/bin/cr_ra_41_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 are in "Sample Remote Agent Parameters File for Solaris OS, Red Hat Linux, and IBM AIX" on page 110. The parameters file must be an executable file.
- Set the environment variables.

```
% export CR_RA_INSTALLER_USER=username
% export CR_RA_INSTALLER_WORKDIR=/working_directory
% export CR_RA_INSTALLER_LEAVEFILES=yes_or_no
% export CR_RA_INSTALLER_HOSTS=hostnames.enterprise.com,10.10.0.207
```

6. **Start the remote installation.**

```
% cr_ra_opsystem_4.1_remote.sh path-to-file/parameters-file.sh -f
cr_ra_opsystem_4.1.sh hostnames
```

- *opsystem* is one of the following values:
  - solaris – installs the application on the Solaris OS
  - aix – installs the application on IBM AIX
  - linux – installs the application on Red Hat Linux
- `cr_ra_`*opsystem*`_4.1.sh` is the installation script that you copied from the N1 Service Provisioning System 4.1 CD.
- *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 hostnames of the machines on which to perform the installation. Separate the hostnames by a space. If you specified the hostnames in the `CR_RA_INSTALLER_HOSTS` parameter, either in the parameters file or as an environment variable, you do not need to specify the hostnames on the command line. Any hostnames that you specify on the command line take precedence over those specified in the `CR_RA_INSTALLER_HOSTS` parameter.

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 that 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.

CHAPTER 5

Installing the N1 Service Provisioning System 4.1 on Windows Systems

This chapter describes the steps to install the N1 Service Provisioning System 4.1 on systems running Windows. 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 N1 Service Provisioning System 4.1 applications begin by performing the same set of preparatory tasks and asking the same questions about directories and files. Each script then asks specific configuration questions about the application that it will install.

This chapter discusses the following topics:

Installing the Master Server

▼ How to Install the N1 Service Provisioning System 4.1 Master Server on Windows

Before You Begin Review the task map in Table 1–1. Complete any necessary tasks prior to installing the Master Server.

Steps

1. **Insert the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD.**

2. **Use the Windows File Manager or a DOS window to access the `windows` directory on the CD.**

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

   - If you are using the File Manager, double-click the `cr_ms_win32_4.1.msi` file.
   - If you are in a DOS window, type the name of the installation file at the prompt.

     ```
     E:\N1SPS4.1_2\windows> cr_ms_win32_4.1.msi
     ```

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 it 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 N1 Service Provisioning System 4.1.

7. **Log in to the system.**

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

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

Note – The installer opens DOS windows and executes commands. Some of the commands might take several minutes to run. Do not close the DOS 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 Web interface address 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 47.

▼ How to Create a Scheduled Task to Optimize the Database

Steps

1. **Open the Windows 2000 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/4.1`***`/server/bin/roxdbcmd vacuumdb -h localhost -a -z"`**
   c/Program\ Files/N1\ Service\ Provisioning\ System/4.1 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 task map in Table 1–1. Complete any necessary tasks prior to installing the Master Server.

Steps

1. **Insert the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD.**
2. **Use the Windows File Manager or a DOS window to access the `windows` directory on the CD.**
3. **Start the installation script for the application you want to install.**

- If you are using the File Manager, double-click the `cr_app_win32_4.1.msi` file.
- If you are in a DOS window, type the name of the installation file at the prompt.

```
E:\N1SPS4.1_2\windows> cr_app_win32_4.1.msi
```

app is one of the following values:

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

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 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

Steps

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

2. **Insert the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD.**

3. **Change to the directory on the software CD where the Windows installation script is located.**

```
C:\> cd path-to-CD/windows
```

path-to-CD is the path to the software CD.

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

   `% cp cr_ra_win32_4.1.sh` *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_win32_4.1.msi /qn`**
   VARIABLE`=`*value* *VARIABLE*`=`*value*

   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. If you do not include any variables or values, the installation program installs the Remote Agent using the default values.

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_win32_4.1.msi/ qn
INSTALLDIR="C:\Program Files\N1 Service Provisioning System\"R
A_PARENT_CONNECTION=false
```

Remote Installation of Remote Agents on Windows

The Remote Agent installation script 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 Sindows file share on the target system
- Copies `cr_ra_win32_4.1.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

Steps

1. **On the Master Server machine, open a DOS window.**
2. **Insert the N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD.**
3. **Change to the directory on the software CD where the Windows installation script is located.**

   C:\> **cd** *path-to-CD*/**windows**

   path-to-CD is the path to the software CD.
4. **Copy the installation script to the Master Server.**

   % **cp cr_ra_win32_4.1.sh** *MS-machine*/

   MS-machine is a directory on the Master Server machine.
5. **Start the installation.**

   C:*MS-machine*> **cscript WinInstaller.wsf**
   parameters Hostname

   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 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.	none
`-msiLocation`	Paths to the `.msi/.input` files to install.	Current working directory

Parameter	Description	Default
`-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 installation script.	`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
"INSTALLDIR='C:\Program Files\N1 Service Provisioning System'" 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.	2313	Any valid port number.
`RA_PARENT_CONNECTION`	Specify that the parent application connect to this Remote Agent using unencrypted (raw) or SSL connections.	`false`	`true` specifies to use SSL. `false` specifies to use raw.

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

Variable Name	Description	Default	Values
RA_SSL_CIPHER	If you selected SSL, specify the type of SSL cipher suite to use.	1	0 specifies to use encryption with authentication. 1 specifies to use encryption without 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.
RA_SERVICE_AUTOSTART	Specify whether to start the Remote Agent automatically on system 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.

CHAPTER 6

Configuring the N1 Service Provisioning System 4.1 to Use Secure Shell

This chapter contains instructions for configuring the N1 Service Provisioning System 4.1 to communicate using Secure Shell (SSH).

The N1 Service Provisioning System 4.1 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.

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 67.

This chapter discusses the following topics:

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 systems.

Unlike the other connection types, when an SSH connection is set up between two N1 Service Provisioning System 4.1 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 an Local Distributor to connect to an Remote Agent using SSH, do not manually start the Remote Agent. The Local Distributor automatically stars the Remote Agent when necessary. The Remote Agent continues to run for as long as it is being used. The Local Distributor will automatically shutdown the Remote Agent when it has not been used for a configurable period of time.

Empty Password Keys or `ssh-agent`

You can configure SSH to use empty password keys or to use the `ssh-agent`. 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.

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 N1 Service Provisioning System 4.1, 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.

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 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. Agent forwarding allows Local Distributors to authenticate to Local Distributors and Remote Agents that are downstream. This approach provides more security.

SSH Requirements

The N1 Service Provisioning System 4.1 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 N1 Service Provisioning System 4.1.

- The ssh-keygen command must generate a public-private keypair 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
  ```

- On host3 after hopping from the current host to host1 to host2 to 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.

Task Map for Configuring SSH

The following table describes the tasks necessary to configure the N1 Service Provisioning System 4.1 to use SSH.

TABLE 6–1 Task Map: Configuring SSH

Task	Description	For Instructions
Decide security level.	Determine whether you want to use empty password keys or the `ssh-agent`.	"Empty Password Keys or `ssh-agent`" on page 54
Generate keys.	Generate the keys on the applications that initiate SSH connections.	"How to Generate Key Pairs" on page 57
Set up the keys.	Copy the generated keys to the Local Distributors and the Remote Agents. Choose the appropriate task based on whether you are using empty password keys or the `ssh-agent`.	"How to Set Up Keys for Empty Password Files When Using One Key Pair" on page 57 "How to Set Up Keys for Empty Password Files When Using Multiple Key Pairs" on page 58 "How to Set Up Keys for the `ssh-agent`" on page 59
Set up and test the connectivity on the Master Server.	Set up the SSH connectivity and test the connectivity before you start the Master Server.	"Setting Up and Testing the Connectivity on the Master Server" on page 59
Configure the Local Distributors and Remote Agents to use SSH.	Configure the Local Distributors and Remote Agents to use SSH.	"How to Configure SSH for Local Distributors and Remote Agents" on page 62
(Optional) Configure the CLI Clients to use SSH.	If you have any CLI Clients, configure them to use SSH.	"How to Configure SSH for the CLI Client With the `ssh-agent`" on page 63

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.

If you are using `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. Or, 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.

▼ How to Generate Key Pairs

Steps

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

   ```
   % ssh-keygen -t rsa
   ```

   The system prompts you for a password.

2. **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.

   The system prompts you to save the keys.

3. **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.

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

Steps

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 and *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 and *User-home-downstream* is the home directory on the Local Distributor or the Remote Agent machine to which the machine 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

Before You Begin

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

Steps

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 and *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.**

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

Steps

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

   `% cp /`*User-hone*`/.ssh/id_rsa` *path_to_file/*

   User-home is the home directory of the currently logged in user on the Master Server machine and *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 and *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. **Add the following line to the `config.properties` files on the Master Server and the Local Distributors to enable `ssh-agent` forwarding.**

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

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 N1 Service Provisioning System 4.1. 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.

Steps

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 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 60. 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 one 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.

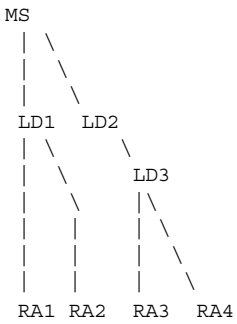
Steps

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



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, would 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.**

   Using the same command that you used in Step 1, test the connection paths again to be sure that the system does not prompt you for any information.

More Information

Repeating Set Up and Testing

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

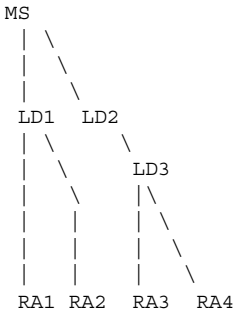
Configuring SSH For the Applications

After you set up and test SSH on the Master Server, configure the other machines in the N1 Service Provisioning System 4.1 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 N1 Service Provisioning System 4.1 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 may 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 one.

Steps

1. **Use the Web 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=/`*N1SPS4.1-Home*`/`*application*

N1SPS4.1–Home is the home directory of the application and *application* is agent if you are configuring a Remote Agent or ld if you are configuring a Local Distributor.

For example, if the N1 Service Provisioning System 4.1 is installed in `/opt/SUNWn1sps/N1_Service_Provisioning_System_4.1/` and you are configuring a Remote Agent, the text you add to the Advanced Parameters field is:

```
cprefix=/opt/SUNWn1sps/N1_Service_Provisioning_System_4.1/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`.

Steps

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 57.**

   Do not reuse the keys 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` or the `.cshrc` 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 85.**

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.

Steps

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 57.**

   Do not reuse the keys 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 and *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 85.**

The jexec Wrapper

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 has the `setuid` bit set. This file has the same groupid as the user you used to install the Remote Agent and it gives execute permission to the group. Additionally, the file is stored in a directory called `protect` that is owned by the user you used to install the Remote Agent. The file only gives execute permission 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 N1 Service Provisioning System 4.1, ensure that all the files in *N1SPS4.1-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 */N1SPS4.1-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 containing RhostsRSAAuthentication, as 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 */N1SPS4.1-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.

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 available in that version of SSH that are equivalent to the following commands.

TABLE 6–2 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 7

Configuring the N1 Service Provisioning System 4.1 for SSL

This chapter contains instructions for configuring the N1 Service Provisioning System 4.1 to communicate using Secure Socket Layer (SSL). This chapter discusses the following topics:

Overview of SSL Support in the N1 Service Provisioning System 4.1

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.

N1 Service Provisioning System 4.1 applications can be configured to use SSL for their network communications, preventing messages from being read or tampered. 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 will take 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 in `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 authentication with encryption, the cipher suite is set to `SSL_RSA_WITH_3DES_EDE_CBC_SHA`.

For lists of SSL cipher suites that do and do not require server authentication, see “SSL Cipher Suites” on page 84. You can configure client authentication only for cipher suites requiring server authentication.

Authentication Key Stores

The N1 Service Provisioning System 4.1 supports self-signed certificates. Two types of key stores exist:

- Private Key Store – The private key store contains the public-private key pairs that the application uses to authenticate itself when connecting to other applications.

- Trust Key Store – The trust key store contains the public key, in self-signed certificates, of other applications that the key store 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 key stores 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 key store and the application acting as the SSL client requires a public, or trusted, key store. The public key stores 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 key stores. The password for both of the key stores 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 key store, and application A must have application B's public key in its trust key store. 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 key store.

If application B is configured to require client authentication, application A must have a public-private key pair in its private key store, and application B must have application A's public key in its trust key store. 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 key store.

Using Passwords With SSL

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

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

The `crkeys` script validates that you specified the same password for both the key stores. 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.

To use the `crkeys` script to prompt for and verify the key store password when starting applications, use the `-vpass` option. The `crkeys` script prompts the user for the key store password if any of the key stores exist, and verifies the password against the key store. If the verification succeeds, it prints the password on the standard output so that it can be fed into the application.

Limitations of SSL on the N1 Service Provisioning System 4.1

The SSL implementation on the N1 Service Provisioning System 4.1 has the following limitations:

- Only self-signed certificates are supported. The trust key store contains self-signed certificates only. You cannot use CA-signed certificates.
- Both the trust and the private key stores must be configured with the same password. Also, within the private key store, 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.
- Passwords echo to the terminal. To overcome this limitation on POSIX platforms, you may have the startup script disable terminal echo and then prompt the user for a password.
- 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 key store passwords. If the key stores have been created, the key stores must be provided in the CLI Client properties file.

  The N1 Service Provisioning System 4.1 uses single trust key store for both incoming and outgoing connections. Hence, 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.

  Client authentication is not supported for CLI Client, therefore, the CLI Client only has a trust store. The benefit of supplying a password is that you can verify that the trust store has not been tampered with. You can specify the password in the properties file, but prompting the user for the password each time the CLI Client is run is more secure.
- For SSH connections, the remote application, the Local Distributor or Remote Agent, is automatically started. The system does not prompt you for the key store passwords to start these applications. If the applications are initialized with key stores, the passwords to their key stores 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.

Task Map for Configuring SSL

The following table describes the tasks necessary to configure the N1 Service Provisioning System 4.1 to use SSL.

TABLE 7–1 Task Map: Configuring SSL

Task	Description	For Instructions
Decide security level.	Determine the SSL connectivity that you want to use.	"Overview of SSL Support in the N1 Service Provisioning System 4.1" on page 69
(Optional) Enable SSL in Tomcat.	You can enable the Web Interface to use HTTPS.	"Enabling SSL in Tomcat" on page 73
Create key stores.	Use the `crkeys` command to create key stores.	"Creating Key Stores" on page 75
Configure SSL.	Edit the `config.properties` file to configure SSL.	"Configuring SSL" on page 77

Enabling SSL in Tomcat

By default, the N1 Service Provisioning System 4.1 Web Interface does not use SSL. Requests are performed over HTTP rather than HTTPS. You can enable HTTPS with an SSL Certificate. SSL Certificates are issued by Certifying Authorities (CA). Certificates are usually specific to individual machines.

An SSL Certificate is enclosed within the following delimiters:

```
-----BEGIN CERTIFICATE-----
```

and

```
-----END CERTIFICATE-----
```

▼ How to Generate SSL Certificates for Tomcat

Steps **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 N1 Service Provisioning System 4.1, the JRE is installed in the *N1SPS4.1-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 where you want to store the generated keys. `/etc/keystore` is commonly used.

   Set *password* to whatever password you choose.

3. **Follow the prompts to complete.**

▼ How to Enable SSL in Tomcat

Steps

1. **Import the SSL Certificate**

   `% keytool -import -alias tomcat -keystore` *keystore-location***`/ -trustcacerts`**

   keystore-location is the path to and the name of the file in which you saved your certificate text. The output of this command shows the name of the file in which the imported certificate is stored. This file is usually saved in the home directory of the user who ran the command.

2. **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="org.apache.catalina.net.SSLServerSocketFactory"
           clientAuth="false" protocol="TLS"/>
</Connector>
```

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

```
<Factory className="org.apache.catalina.net.SSLServerSocketFactory"
           clientAuth="false" protocol="TLS"
       keystoreFile=path-to-tomcat-keystore-file/ keystorePass="password"/>
```

   path-tomcat-keystore-file is the path to the Tomcat `keystore` file and *password* is the password that you used to create the original keypass.

Requiring Users to Connect to the Web Interface Using SSL

After you have configured the N1 Service Provisioning System 4.1 to use SSL, you can configure it further so that users must use SSL to connect to the server.

▼ How to Require Users to Connect Using SSL

Step ● **Replace the current `web.xml` file with the Tomcat `/webapp/WEB-INF/web.xml.secure` file.**

```
% cd /N1SPS4.1-home/webapp/WEB-INF
% cp web.xml.secure web.xml
```

N1SPS4.1-home is the home directory of the application.

▼ How to Revert to the Original Configuration

Step ● **To return to the original configuration, replace the `web.xml` file with the `/webapp/WEB-INF/web.xml.default` file.**

```
% cd /N1SPS4.1-home/webapp/WEB-INF
% cp web.xml.default web.xml
```

N1SPS4.1-home is the home directory of the application.

Creating Key Stores

The N1 Service Provisioning System 4.1 uses the keytool utility provided with the JRE. The keytool utility is wrapped in a shell script, `crkeys`, to enable you to create key stores. The script ensures that the correct parameters are supplied to the 'keytool' utility.

When you create a key store, 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`

▼ How to Create Key Stores

Step ● **Generate the keys.**

```
% crkeys -options
```

Use the following options to create key stores 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 hostname of the application as the alias. The alias names must be unique within a key store.
`-cpass`	Changes the password of the key store and all the keys within the key store.
`-delete`	Specifies that the key pair or certificate for the specified entity should be deleted from key store.
`-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 hostname 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 key store.
`-new` *newpassword*	Specifies the new password for the key store and all the keys in the key store.
`-password` *password*	Specifies the password for the key store. If a password is not specified, the user is prompted for one.
`-private`	Specifies the private key store as the target of the operation.
`-validity` *days_valid*	Number of days the self-signed certificate is valid.
`-trust`	Specifies the trust key store as the target of the operation.

Example 7–1 `crkeys` Command Examples

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

To generate a public-private key pair:

```
crkeys -private –generate|-delete
    –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 key store, both the trust and the private store, password:

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

To print instructions for using the `crkeys` command:

```
crkeys -help
```

Configuring 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 key store in the *N1SPS4.1-home*/*app*/`data/private.store` file.
- Find the trust key store in the *N1SPS4.1-home*/*app*/`data/trust.store` file.
- Supply empty passwords for each key store.

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 84.
- Specify that the application authenticate the SSL clients connecting to it.
- Specify the location and password of the private key and trust stores.

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

▼ How to Configure SSL

Step ● **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 84.
`net.ssl.client.auth`	`false`	Specifies whether the SSL server should authenticate clients connecting to it.
`net.ssl.trust.store.path`	*N1SPS4.1–home*`/data/trust.store`	The path to the trust key store. The key store that contains the public keys of the nodes that are allowed to connect to this node.

Parameter	Default Value	Description
`net.ssl.private.store.path`	*N1SPS4.1–home*`/data/private.store`	The path to the private key store. The key store that contains the public-private key pairs that this node uses to authenticate itself to other nodes.
`net.ssl.key.store.pass`		The key store password.

Sample Configuration Scenarios

▼ How to Configure SSL Without Authentication Between the Master Server, Local Director, and Remote Agent

Steps

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 Web 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.**

▼ 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.

Steps

1. **Generate a key pair for the Local Distributor and store it in the private store for the Local Distributor.**

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

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

   ```
   % ld/bin/crkeys -private -export -file ld.cert -alias ldhostname.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 store.**

   ```
   % server/bin/crkeys -trust -import -file ld.cert -alias ldhostname.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 the Master Server and the Local Distributor.**

10. **Start the Master Server and the Local Distributor.**

11. **Provide the key store 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.**

    ```
    % agent/bin/crkeys -private -generate -alias rahostname.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.cert -alias rahostname.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.cert -alias rahostname.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 key store password for the Local Distributor and Remote Agent.**
24. **Test the connection to the Remote Agent.**

▼ How to Configure SSL Server and Client Authentication

Steps

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. **Generate a key pair for the Local Distributor and store it in the private store for the Local Distributor.**

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

3. **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
   ```

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

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

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

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

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

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

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

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

   `% agent/bin/crkeys -private -generate -alias` *rahostname.cr.com* `-validity 365`
17. **Export the self-signed certificate from private store for the Remote Agent into a file.**

   `% agent/bin/crkeys -private -export -file ra.cert -alias` *rahostname.cr.com*
18. **Copy the self-signed certificate for the Remote Agent to the Local Distributor.**
19. **Import the self-signed certificate into the Local Distributor trust store.**

   `% ld/bin/crkeys -trust -import -file ra.cert -alias` *rahostname.cr.com*
20. **Copy the self-signed certificate for the Local Distributor, exported in Step 4, to the Remote Agent machine.**
21. **Import the self-signed certificate into the Remote Agent trust store.**

   `% agent/bin/crkeys -trust -import -file ld.cert -alias` *ldhostname.cr.com*
22. **Create a new host.**
23. **On the new host, add a Remote Agent with the connection type SSL.**
24. **Copy the `transport.config` file to the Remote Agent.**
25. **If already running, stop the Local Distributor and Remote Agent.**
26. **Start the Local Distributor and the Remote Agent.**
27. **Provide the key store password for the Local Distributor and Remote Agent.**
28. **Test the connection to the Remote Agent.**

▼ How to Configure SSL Authentication Between a CLI Client and Master Server

Steps

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.**

   ```
   % 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 key store 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

The following lists describe the supported SSL cipher suites.

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
```

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
```

The following suites require server authentication with no encryption.

```
SSL_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA
```

CHAPTER 8

Configuring the Java Virtual Machine Security Policy

This chapter describes how to configure the security policy of the N1 Service Provisioning System 4.1 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 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:

`host =` *hostname*`|`*IPaddress* `:`*portrange*

hostname is the host name of the machine and `|`*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/1.3/docs/guide/security/PolicyFiles.html` and click on the Policy File Syntax link.

▼ How to Configure the JVM Policy for the Master Server

Steps

1. **Delete the line that allows the application to connect to or accept connections from all hosts.**
2. **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 Web Interface, and *httpport* is the port number the Web 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

Steps

1. **Delete the line that allows the application to connect to or accept connections from all hosts.**
2. **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

Steps

1. **Delete the line that allows the application to connect to or accept connections from all hosts.**
2. **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 *N1SPS4.1-MasterServer-home*`/config/config.properties` file, you must change the value of `db.password`.

CHAPTER 9

Upgrading to the N1 Service Provisioning System 4.1

This chapter contains instructions for upgrading from the 4.0 version of product to the N1 Service Provisioning System 4.1.

Note – 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 the N1 Service Provisioning System 4.1.

This chapter describes the following upgrade topics:

Upgrading Overview

You will first upgrade the Master Server by installing the 4.1 version of the Master Server on the same system as the 4.0 version of the Master Server. This process is known as a side-by-side installation. You will them migrate the data from the 4.0 version of the Master Server to the 4.1 version of the Master Server. After you complete the data migration for the Master Server, then follow the instructions to upgrade the Remote Agents and the Local Distributors. CLI Clients do not need to be upgraded, simply install the 4.1 version of the CLI Client and uninstall the 4.0 version.

Upgrading Solaris OS and Red Hat Master Servers

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 system 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 Data on a Solaris OS or a Red Hat Master Server

Migrating data from the 4.0 version of the Master Server to the 4.1 version of the Master Server deletes any data in the 4.1 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

Follow the instructions in "How to Install the N1 Service Provisioning System 4.1 on Solaris OS, Red Hat Linux, and IBM AIX Systems" on page 37 to install the N1 Service Provisioning System 4.1 Master Server on the system where the 4.0 version of the Master Server is installed. Both the 4.0 and the 4.1 versions of the Master Server must be installed on the same system before you begin the migration procedure.

To back up your data before you migrate, follow the instructions in "How to Back Up a Master Server" on page 103.

Steps

1. **Verify that the database will not optimize 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.

2. **Log in as the user that owns the Master Server directories.**

3. **Start the migration script.**

   # **/***N1SPS4.1-home***/server/bin/migrate/cr_4.0.2-4.1_migration.sh**

   N1SPS4.1-home is the directory where you installed the application. The default directory is `/opt/SUNWn1sps/N1_Service_Provisioning_System_4.1`.

4. **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, Web Interface, and the Master Server are not migrated. The N1 Service Provisioning System 4.1 Master Server uses the port numbers that you supplied during installation.

5. **Check the log file for any errors that may have occurred during migration.**
   The migration script displays the location of the log file.
6. **Back up the data that you migrated to the new Master Server by using the instructions in "How to Back Up a Master Server" on page 103.**
   You cannot restore data from the 4.0 Master Server to the 4.1 Master Server. Back up the data on the 4.1 Master Server so that you have a complete and accurate backup of the data to use if necessary.

Upgrading Windows Master Servers

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 system 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 Install A Side-by-Side Windows Master Server

You must follow the instructions below to install a 4.1 version of the Windows Master Server on the same machine as the 4.0 version of the Master Server.

Steps

1. **Stop the 4.0 Master Server by using the Service application in the Windows Administrative Tools to stop the IPC Daemon service.**
2. **Set the 4.0 Master Services to start manually, specifically the IPC Daemon and Server.**
3. **Install the 4.1 version of the Master Server by following the instructions in "How to Install the N1 Service Provisioning System 4.1 Master Server on Windows" on page 45.**
   Install the 4.1 version of the Master Server with ownership by the same user and group that owns the 4.0 version of the Master Server.

Note – Only one version of the Master Server can be running on a system. You can start and stop the Master Servers by using the Service application in the Windows Administrative Tools to stop or start the IPC Daemon service.

4. **Migrate the data from the 4.0 version of the Master Server to the 4.1 version of the Master Server.**

   To migrate the data, follow the instructions in "How to Migrate Data on a Windows Master Server" on page 92.

5. **(Optional) Uninstall the 4.0 version of the Master Server.**

   You can uninstall the 4.0 version of the Master Server if you do not want to use it by following the instructions in "Uninstalling Applications on Windows Systems" on page 99.

▼ How to Migrate Data on a Windows Master Server

Migrating data from the 4.0 version of the Master Server to the 4.1version of the Master Server deletes any data in the 4.1 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

Follow the instructions in "How to Install A Side-by-Side Windows Master Server" on page 91 to install the N1 Service Provisioning System 4.1 Master Server on the system where the previous version of the Master Server is installed. Both the 4.0 and the 4.1 versions of the Master Server must be installed on the same system before you begin the migration procedure.

Steps

1. **Open a command prompt window.**

2. **Change to the** *C:\Program Files\N1 Service Provisioning System\4.1***`server\bin\migrate`** **directory.**

   `cd` *C:\Program Files\N1 Service Provisioning System\4.1***`server\bin\migrate`**

   C:\Program Files\N1 Service Provisioning System\4.1 is the directory in which you installed the Master Server.

3. **Begin the migration by typing:**

   ```
   .\cr_migrate.cmd
   ```

4. **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, Web Interface, and the Master Server are not migrated. The N1 Service Provisioning System 4.1 Master Server uses the port numbers that you supplied during installation.

5. **Check the log file for any errors that might have occurred during migration.**
   The migration script displays the location of the log file.

Upgrading Remote Agents and Local Distributors

▼ How to Upgrade Remote Agents and Local Distributors

You upgrade Remote Agents and Local Distributors using the Web Interface to the Master Server. To complete the upgrade, you will need to click the Update Entire N1 SPS network button twice.

Before You Begin Migrate the Master Server before upgrading Remote Agents and Local Distributors.

Steps

1. **Log in to the Web Interface of the N1 Service Provisioning System 4.1 Master Server.**
2. **Click Hosts.**
3. **Click the masterserver.**
4. **Click the Update Entire N1 SPS network... button.**
   A window opens that 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 on 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 Remote Agents that you upgraded.**
   Before you can run a Plan on a Remote Agent that you upgraded, you must Prepare the Remote Agent. To Prepare Remote Agents, follow the instructions in the *N1 Service Provisioning System 4.1 User's Guide*.

Migrating Master Server Data

Migration Overview

The following table details the types of data that is migrated on the Master Server.

TABLE 9–1 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 node	Yes	Properties listed in the file are migrated using the details found in "Migration Details for the Properties File" on page 94.
Resource migration	Yes	Copy the resources directory.

Migration Details for the Properties File

The `centerrun.properties` file is migrated to the `config.properties` file. During the migration, the value of each property in the 4.0 file is compared to the value of the property in the 4.1 `config.properties` file. If the value is the same, the

property is ignored. If the value is different, then the 4.0 value is copied to the 4.1 `config.properties` file. If a property exists in the 4.0 file and is absent in the 4.1 file, then the 4.0 value is added to the 4.1 file. Values for the following properties are not migrated to the 4.1 `config.properties` file:

- `webserver.TomcatHome`
- `rsrc.localrepo`
- `db.port`
- `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 4.0 properties file, you will need to manually change the value in the 4.1 `config.properties` file.

CHAPTER 10

Uninstalling the N1 Service Provisioning System 4.1

This chapter describes procedures for uninstalling the N1 Service Provisioning System 4.1 in the following sections:

Uninstalling Applications on Solaris OS, Red Hat, and IBM AIX Systems

The procedure to uninstall the N1 Service Provisioning System 4.1 depends upon the method that you used to install the software.

- If you installed the application as packages on the Solaris OS, use the instructions in "How to Uninstall Package-Based Applications on a Solaris OS System" on page 97.
- If you installed the application as files Solaris OS or if you have a Red Hat or IBM AIX system, use the instructions in "How to Uninstall File-Based Applications on Solaris OS, Red Hat, and IBM AIX Systems" on page 98.

▼ How to Uninstall Package-Based Applications on a Solaris OS System

Only the Master Server and the CLI Client are available for installation as packages. The uninstall script removes only 4.1 versions of the Master Server or CLI Client.

Note – The uninstall script is only installed if you installed the Master Server or the CLI Client as package. If the script is not in the directory, then uninstall by using the instructions in "How to Uninstall File-Based Applications on Solaris OS, Red Hat, and IBM AIX Systems" on page 98.

Steps

1. **On the system that you want to uninstall the application, verify that you are not in the directory of the application that you want to uninstall.**

2. **Begin the uninstallation.**

   # **/***N1SPS4.1-home***/***app_directory***/bin/cr_uninstall_***app***.sh**

   N1SPS4.1-home is the directory where you installed the application. The default directory is `/opt/SUNWn1sps/N1_Service_Provisioning_System_4.1`. *app_directory* is one of the following values:

   - server – uninstalls a Master Server
   - cli – uninstalls a 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 SUNWsp app
       Successfully removed SUNWspsc1
       Successfully removed SUNWspsj1
   ```

   app is ms when uninstalling a Master Server and cl when uninstalling a CLI Client.

Note – The `SUNWspsc` and `SUNWspsj1` packages are not removed if another application is installed on this system. For example, if you have a Master Server and a CLI Client both installed on the same system, when you uninstall only the Master Server, the `SUNWspsc` and `SUNWspsj1` packages remain on the system until you uninstall the CLI Client.

▼ How to Uninstall File-Based Applications on Solaris OS, Red Hat, and IBM AIX Systems

Steps

1. **On the system 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 /N1SPS4.1-home/agent/bin/protect
   ```

   N1SPS4.1-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 /N1SPS4.1-home/app-directory
   ```

   N1SPS4.1-home is the directory where you installed the application. The default directory is `/opt/SUNWn1sps/`. *app-directory* is one of the following values:

   - 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 *N1SPS4.1–home* directory contains no more application directories, delete the `common/` directory.**

   ```
   # rm -r N1SPS4.1-home/common
   ```

   The uninstallation is complete.

Uninstalling Applications on Windows Systems

To uninstall applications on Windows systems, 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.

CHAPTER 11

Administering the N1 Service Provisioning System 4.1

This chapter provides instructions for backing up and restoring the N1 Service Provisioning System 4.1. This chapter covers the following topics:

Starting the N1 Service Provisioning System 4.1 Applications

Starting Applications on Solaris OS, Red Hat Linux, and IBM AIX Systems

The following table lists the commands to start the N1 Service Provisioning System 4.1 applications on Solaris OS, Red Hat Linux, and IBM AIX systems. *N1SPS4.1-home* is the home directory of the application.

TABLE 11–1 Start Commands for Solaris OS, Red Hat Linux, and IBM AIX Applications

Application	Path to Command	Command to Start
Master Server	*N1SPS4.1-home*`/server/bin/`	`cr_server start`

TABLE 11–1 Start Commands for Solaris OS, Red Hat Linux, and IBM AIX Applications *(Continued)*

Application	Path to Command	Command to Start
Local Distributor	*N1SPS4.1-home*/ld/bin/	cr_ld start
Remote Agent	*N1SPS4.1-home*/agent/bin/	cr_ra start
CLI Client	*N1SPS4.1-home*/cli/bin/	cr_cli start
Jython version of CLI Client	*N1SPS4.1-home*/cli/bin/	cr_clij start

Starting Applications on Windows Systems

On Windows systems, you start the Master Server, Local Distributor, and Remote agent in the Services Panel. You start the CLI Client from a DOS 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 11–2 Names of Services to Start for the Windows Master Server, Local Distributor, and Remote Agent

Application	Name of Service to Start
Master Server	N1 Service Provisioning System 4.1 Server
	N1 Service Provisioning System 4.1 PostgresSQL Server
	N1 Service Provisioning System 4.1 IPC Daemon
	N1 Service Provisioning System 4.1 Database Preparer
Local Distributor	N1 Service Provisioning System 4.1 Distributor
Remote Agent	N1 Service Provisioning System 4.1 Agent

To start the CLI Client on a Windows system, type one of the following commands at a DOS prompt. *N1SPS4.1-home* is the home directory of the application.

TABLE 11–3 Start Commands for the Windows CLI Client

Application	Path to Command	Command to Start
CLI Client	*N1SPS4.1-home*/cli/bin/	cr_cli.cmd start
Jython version of CLI Client	*N1SPS4.1-home*/cli/bin/	cr_clij.cmd start

Backing Up and Restoring the Master Server

The software includes utilities for completely backing up and restoring a Master Server. These utilities are found in the *N1SPS4.1-home*/`server/bin` directory.

You can choose to back up or restore either or both the Resource Manager and the Postgres database. By default, the Resource Manager directory and the Postgres database contents are backed up or restored. You can skip one or the other of these components by using the appropriate command line arguments.

▼ How to Back Up a Master Server

Before You Begin

Before you back up the Master Server, you must stop the Master Server. Any Plans or Preflights must be stopped, as well as any other tasks, such as comparisons.

Caution – Be sure to specify an output directory when you run the backup script. If you do not specify an output directory, the resultant backup file is saved in the *N1SPS4.1-home*/`server/bin` directory. If you are uninstalling and then reinstalling the Master Server and the backup file is in this directory, the backup file is deleted. Consequently, you are not able to restore the Master Server.

Steps

1. **Stop the Master Server.**
2. **On the Master Server, change users so that you are root or the owner of the application.**
3. **Change to the directory where the backup script is located.**

   `% cd` *N1SPS4.1-home*`/server/bin`

   N1SPS4.1-home is the home directory of the application.
4. **Start the backup by typing:**

   `% ./cr_backup.sh` *options*

   The following options are available for use with the `cr_backup.sh` command.

`-b`	Master Server base directory.
`-q`	Quiet. No informative messages are printed.
`-nors`	Skip the Resource Store back up.
`-nodb`	Back up the Resource Manager only.

`-o` *directory*	Save the backup file in this directory. The backup script verifies that you have write permissions for the directory you specified. If you do not have write permissions for this directory, the script generates an error. If you do not specify a directory, the files are saved in the *N1SPS4.1-home*`/server/bin` directory.
`-z`	Compress the resulting backup file using UNIX compression.
`-1` *logfile*	Write log output to the *logfile* file instead of the default `logfile` file.
`-gz`	GZip the resulting backup file if `gzip` is in PATH.
`-shutdown`	Shut down the Master Server without prompting the user.
`-u`	Print this information.
`-h`	Print this information.

The script warns that if you proceed Master Server process will stop, if it is not already stopped, thereby canceling any tasks, such as searches, plans, and comparisons, that are in progress

5. **Type `y` to proceed with the backup.**

   The script shows the progress of the backup and displays the location of the backup tar file.

   The script restarts the Master Server.

▼ How to Restore a Master Server

Before You Begin

Before performing a restore, you must have an installation of the Master Server that contains no data.

Steps

1. **Stop the Master Server.**
2. **On the Master Server, change users so that you are root or the owner of the application.**
3. **Change to the directory where the backup script is located.**

   `% cd` *N1SPS4.1-home*`/server/bin`

   N1SPS4.1-home is the home directory of the application.

4. **Start the restoration by typing:**

   `% ./cr_restore.sh` *options*

   The following options are available for use with the `cr_restore.sh` command.

`-b`	Master Server base directory If you do not use the `-b` option to specify a directory to in which to restore the backup files, the files are put in the current directory, *N1SPS4.1–home*`/server/bin`. If you do not have write permissions for this directory, the software generates an error.
`-q`	Quiet. No informative messages are printed
`-nors`	Skip the Resource Store restore
`-nodb`	Skip the database restore
`-f` *backupfile*	Restore the contents of the *backupfile* file
`-l` *logfile*	Write log output to the *logfile* file instead of the default `logfile`
`-t` *temp_directory*	Use the *temp_directory* directory to save temporary files
`-overwrite yes`	Overwrite the existing data during restore
`-u`	Print this information
`-h`	Print this information

The script verifies that the backup files do not contain any errors. The script warns that if you proceed the Master Server process will stop, if it is not already stopped, thereby canceling any tasks, such as searches, plans, and comparisons, that are in progress.

5. **Type `y` to proceed.**

   The script warns that any data currently in the databases will be overwritten with the data in the backup files.

6. **Type `y` to proceed.**

   The restoration proceeds. The script starts the Master Server:

Backing Up and Restoring Remote Agents

To manually back up the Remote Agent, stop the agent and copy the contents of the *N1SPS4.1-home*`/data` directory to a safe location. To restore the Remote Agent, stop the agent, and copy the contents of the directory that you saved.

Determining the Version and Build of the N1 Service Provisioning System 4.1

On a Solaris OS, Red Hat Linux, or IBM AIX system, to determine the version or build of any application that you installed, provide the `-version` or `-build` option to the command to start the application.

```
% N1SPS4.1-app/server/bin/cr_app -option
```

- *N1SPS4.1-app* is the home directory of the application.
- *app* is the application for which you want to find the version or build information.
- *option* is either `-version` or `-build`.

On a Windows system, to determine the version or build of any application you installed, at a DOS prompt use the `ShowBuild` command with the `-version` option.

```
C:\> N1SPS4.1-app/server/bin/ShowBuild -version
```

APPENDIX **A**

Installation and Configuration Reference

This appendix contains details about the installation of the N1 Service Provisioning System 4.1 in the following sections:

Reference Data for the N1 Service Provisioning System 4.1 on Solaris OS, Red Hat Linux, and IBM AIX

This section contains details about the installation of the N1 Service Provisioning System 4.1 on Solaris OS, Red Hat Linux, and IBM AIX. The topics include the following sections:

Directory Structure of the N1 Service Provisioning System 4.1 on Solaris OS, Red Hat Linux, and IBM AIX

When installing the N1 Service Provisioning System 4.1, you are prompted to select a home directory for the software. The default directory is `/opt/SUNWn1sps`. The installation program creates the following directory tree within the home directory:

- `N1_Service_Provisioning_System_4.1` 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 N1 Service Provisioning System 4.1 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 N1 Service Provisioning System 4.1 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 subapplications
`/common/jre`	Bundled copy of platform-specific JRE
`/common/lib`	Library files common for some or all subapplications

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/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

TABLE A–2 Directories Installed for the Master Server *(Continued)*

Directory	Contents
/server/webapp	HTML User 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

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

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

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
`/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

Database Optimization on Solaris OS, Red Hat Linux, and IBM AIX

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 * * * N1SPS4.1-home/server/bin/roxdbcmd vacuumdb -d rox > /dev/null 2> /dev/null
```

N1SPS4.1-home is the home directory of the Master Server.

Sample Remote Agent Parameters File for Solaris OS, Red Hat Linux, and IBM AIX

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. 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_41_remote_params.sh,v 1.2 2003/11/21 22:50:20 tchang 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 2313.
CR_RA_PORT=2313

# CR_RA_CTYPE - Parent connection type.  How the parent connects to
# this RA.  One of 'raw' (unencrypted), 'ssh', or 'ssl'.  Default is
# raw.
#
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_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 privledges.  Defaults to 'y'.  Valid values are 'y'
# or 'n'.  This only works when the remote installer is run as the root user.
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
```

Reference Data for the N1 Service Provisioning System 4.1 on Windows

This section contains details about the installation of the N1 Service Provisioning System 4.1 on Windows in the following sections:

Directory Structure of the N1 Service Provisioning System 4.1 on Windows

When installing the N1 Service Provisioning System 4.1, 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 4.1` 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 N1 Service Provisioning System 4.1 software into a default destination directories that are subdirectories of the home directory for the software. The following table lists the directories that are installed for every N1 Service Provisioning System 4.1 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 subapplications
`\common\jre`	Bundled copy of the JRE for Windows
`\common\lib`	Library files common for some or all subapplications

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`	HTML User 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

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

TABLE A–8 Directories Installed for the Local Distributor *(Continued)*

Directory	Contents
\ld\lib	Local Distributor library files
\ld\data	Local Distributor-specific data
\ld\data\tmp	Local Distributor temporary directory

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

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

Cygwin

To facilitate interoperability with applications running on Solaris OS, Red Hat Linux, and IBM AIX 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 found 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, runs scripts to generate the configuration files 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, runs scripts to generate the configuration files needed for SSL.
- Registers the Local Distributor service.
- Creates a Start menu shortcut.
- If you requested that the installation script start the Local Distributor, 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 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, runs scripts to generate the configuration files needed for SSL.
- Creates a Start menu shortcut.

APPENDIX **B**

Troubleshooting

This appendix provides troubleshooting information for installation and configuration of the N1 Service Provisioning System 4.1.

Issues During Installation on Solaris OS, Red Hat Linux, and IBM AIX

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 these locked libraries will create errors.

Do not install a new version of the JRE. When prompted to install the JRE, choose no, and then provide a path to the JRE that is already installed on the machine.

Cannot Eject N1 Service Provisioning System 4.1 CD After Installation on a Solaris OS System

If you install the Master Server or Remote agent on a Solaris OS system from the software CD, and answer **yes** when prompted to start the application, you cannot eject the software CD. The following error displays:

```
Device busy
```

To eject the CD, stop the application.

▼ How to Stop an Application on a Solaris OS System

Steps

1. **Change to the directory where the script to start the application is located.**

   % cd *N1SPS4.1-home*/*app*/**bin**

   N1sps4.1–home is the home directory of the application and *app* is the application. For *app*, use server for the Master Server and agent for the Remote Agent.

2. **Invoke the application script with the `stop` option.**

   % cr_*app* **stop**

   app is the application to stop.

3. **Eject the CD by typing:**

   % eject cdrom

Runtime Issue

Master Server and Database Services Stop

After starting the Master Server process using the `cr_server start` command in a bourne shell, if a `^C` command is issued to any subsequent command in the same shell that started the Master Server, the database and Master Server processes stop.

In the *N1SPS4.1-home*/server/bin/roxdb.out file the following messages appear as the most recent entries.

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

Do not use the bourne shell to start the Master Server or other N1 Service Provisioning System 4.1 applications.

SSH Connectivity

Master Server Unable to Connect to Local Distributor Through an Intermediate Local Distributor

If the Master Server is unable to connect to a 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 may 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 N1 Service Provisioning System 4.1 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 one you used to start the ssh-agent.

Steps

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 most 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 closest to the machine being tested and `n` is the machine right before the Master Server. *target-IPaddress* is the IP address of the machine that is the furthest most 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 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 Red Hat Linux and IBM AIX machines, examine the `/tmp/ROX`*appnumbers*`.log` file.
   - On Solaris OS machines, examine the `/var/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.**