N1 Service Provisioning System 4.1 インストールガイド

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

Part No: 817–5509–10
2004 年 2 月

原典: *N1 Service Provisioning System 4.1 Installation Guide*

Part No: 817-4820-10

Revision A

040325@7940

目次

表目次

例目次

はじめに

『N1 Service Provisioning System 4.1 インストールガイド』では、Solaris™ Operating System (OS)、Red Hat Linux、IBM AIX、または Windows 2000 環境での N1™ Service Provisioning System 4.1 のインストールとアップグレードの方法について説明します。

対象読者

このマニュアルは、 N1 Service Provisioning System 4.1 のインストールと構成を担当するシステム管理者を対象としています。

内容の紹介

『N1 Service Provisioning System 4.1 インストールガイド』は、次の各章で構成されます。

- 第 1 章では、ソフトウェアのインストールと構成に必要な作業の概要を示します。ソフトウェアとサポート対象プロトコルの概要も示します。
- 第 2 章では、ソフトウェアをインストールし、使用するためのシステム要件を示します。
- 第 3 章では、ソフトウェアのインストールに必要な情報を記入する、便利なワークシートを提供します。
- 第 4 章では、Solaris OS、Red Hat Linux、IBM AIX システムへのソフトウェアのインストール手順を示します。
- 第 5 章では、Windows システムへのソフトウェアのインストール手順を示します。

- 第 6 章では、SSH を使って通信する場合のソフトウェアの構成手順を示します。
- 第 7 章では、SSL を使って通信する場合のソフトウェアの構成手順を示します。
- 第 8 章では、JVM™ の構成方法について説明します。[1] セキュリティポリシー
- 第 9 章では、ソフトウェアのアップグレード手順を示します。
- 第 10 章では、ソフトウェアのアンインストール手順を示します。
- 第 11 章では、ソフトウェアのバックアップと復元の手順を示します。
- 付録 A では、ソフトウェアのインストールと構成の関連資料を提供します。
- 付録 B では、インストール時ならびに構成時に発生する障害の追跡手順を示します。

関連情報

N1 Service Provisioning System 4.1 のインストール時ならびに使用時には、次のマニュアルを参照してください。

- 『*N1 Service Provisioning System 4.1* ご使用にあたって』
- 『*N1 Service Provisioning System 4.1* ユーザーガイド』
- 『*N1 Service Provisioning System 4.1* リファレンスガイド』

Sun のオンラインマニュアル

`docs.sun.com` では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。URL は、http://docs.sun.com です。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

1 「Java 仮想マシン」という語と「JVM」という語は、Java™ プラットフォームの仮想マシンを意味します。

表 **P–1** 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% **su** password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm *filename* と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% **grep '^#define ** **XV_VERSION_STRING'**

コード例は次のように表示されます。

- C シェル

  machine_name% **command y|n** [*filename*]

- C シェルのスーパーユーザー

  machine_name# **command y|n** [*filename*]

- Bourne シェルおよび Korn シェル

  $ **command y|n** [*filename*]

- Bourne シェルおよび Korn シェルのスーパーユーザー

  # **command y|n** [*filename*]

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: `Shift` キーを押します)。ただし、キーボードによっては `Enter` キーが `Return` キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、`Ctrl-D` は `Control` キーを押したまま `D` キーを押すことを意味します。

一般規則

- このマニュアルでは、「x86」という用語は、Intel 32 ビット系列のマイクロプロセッサチップ、および AMD が提供する互換マイクロプロセッサチップを意味します。

第 1 章

N1 Service Provisioning System 4.1 の概要

この章では、N1 Service Provisioning System 4.1 のインストールと構成に必要な作業の概要を示します。 N1 Service Provisioning System 4.1 に付属しているアプリケーションの概要と、セキュリティの強化目的で使用できるネットワークプロトコルの種類も紹介します。

この章の内容は次のとおりです。

N1 Service Provisioning System 4.1 のインストールの概要

以下の作業マップには、N1 Service Provisioning System 4.1 を正常にインストールし、構成するために必要な作業を示します。

表 **1–1** 作業マップ: N1 Service Provisioning System 4.1 のインストール

作業	説明	参照先
システム要件を検討する	システムがインストールの最小要件を満たしているかどうかを判断します。	第 2 章
インストール情報を収集する	インストールの実行前に、製品をインストールするために必要な情報を収集します。	第 3 章

表 1–1 作業マップ: N1 Service Provisioning System 4.1 のインストール　(続き)

作業	説明	参照先
(任意) ユーザーアカウントを作成する	N1 Service Provisioning System 4.1 が使用する、特殊なオペレーティングシステムユーザーアカウントを作成できます。	使用するオペレーティングシステムのマニュアル
(任意) CLI クライアントマシンに Jython をインストールする	CLI クライアントの実行マシンに、Jython をインストールすることができます。 CLI クライアントは、Jython なしでも実行できます。 Jython は、`http://www.jython.org` からダウンロードできます。	Jython の Web サイト
アプリケーションをインストールする	N1 Service Provisioning System 4.1 アプリケーションを個別にインストールしたい場合は、製品メディア付属の適切なインストールスクリプトを使用します。	第 4 章 第 5 章
(任意) SSH を構成する	インターネット上の Master Server にアクセスする必要がある場合は、N1 Service Provisioning System 4.1 と Master Server 間の通信に SSH を使用するように構成して、Master Server のセキュリティを強化します。	第 6 章
(任意) SSL を構成する	アプリケーションが SSL 通信を使用するように構成して、アプリケーション間の通信セキュリティを最大限に強化します。 SSL のサポートは、ユーザーの所属組織が発行する自己署名付き電子証明書に基づいています。	第 7 章
(任意) JVM セキュリティポリシーを構成する	アプリケーション間のセキュリティ保護目的の SSL 通信を行わない場合は、アプリケーションがローカルホストからの接続のみを受け付けるように、JVM セキュリティポリシーを構成することができます。 この設定では、最小限のセキュリティしか提供されません。	第 8 章

N1 Service Provisioning System 4.1 アプリケーションの概要

N1 Service Provisioning System 4.1 は、次の専用アプリケーションを備えた分散ソフトウェアプラットフォームです。

- 19 ページの「Master Server」 – コンポーネントとプランを格納する中央サーバー。アプリケーション配備の管理用インタフェースを提供する
- 19 ページの「Local Distributor」 – データセンターやファイアウォールを介したネットワーク通信を最適化するため、Master Serverのプロキシとして機能するオプションサーバー
- 19 ページの「Remote Agent」 – 個々のホスト上で処理を実行する小さな管理アプリケーション。複数存在する場合もある。 N1 Service Provisioning System 4.1 の管理下の全ホストが必要とする

- 20 ページの「コマンド行インタフェースクライアント」 – Master Server 上で実行されるコマンドを受け付ける小さなオプションアプリケーション

Master Server

Master Server は、Solaris OS、Red Hat Linux、Microsoft Windows 2000 Server、Microsoft Windows 2000 Advanced Server の各システムで実行可能です。 Master Server は、次の処理を行う中央サーバーです。

- プロビジョニングソフトウェアに登録されているすべてのホストを識別するデータベースを管理する
- コンポーネントとプランをリポジトリに格納する
- リポジトリに格納されたオブジェクトのバージョン管理を実行する
- IT 管理者を認証し、承認されたユーザーだけが特定の処理を実行できるようにする
- 依存性の追跡、配備などの処理を実行する専用エンジンを格納する
- ユーザーに HTML インタフェースと CLI インタフェースを提供する

Local Distributor

Local Distributor は、Remote Agent の配信と管理を最適化するプロキシです。 データセンターは、Local Distributor を使って、次の処理を行います。

- 配備時のネットワークトラフィックを最小化する。 Local Distributor は、Master Server から送信されたコンポーネントのコピー 1 部を複製し、複数のシステムにインストールする
- ファイアウォールの再構成を最小化する。 ファイアウォールが Master Server と複数のシステムとの間に置かれている場合、管理者は、このファイアウォールを、配備に関わるすべてのシステムに対してではなく、Local Distributor を実行するシステムに対してだけオープンすることができる
- 大規模な配備の実行時に、Master Server への負荷を最小化する

Remote Agent

Remote Agent は、N1 Service Provisioning System 4.1 の管理下の全システム上で実行されるアプリケーションです。 Remote Agent は、Master Server からリクエストされた処理を実行します。 Remote Agent をサポートするプラットフォームには、Solaris OS、Red Hat Linux、IBM AIX、Microsoft Windows 2000 があります。 Remote Agent で実行できる処理は次のとおりです。

- サーバーのハードウェアおよびソフトウェアの構成を Master Server に報告する

- サービスを開始 / 停止する
- ディレクトリの内容とプロパティを管理する
- ソフトウェアをインストール / アンインストールする
- コンポーネントモデルに指定されたオペレーティングシステムコマンドとネイティブスクリプトを実行する

コマンド行インタフェースクライアント

コマンド行インタフェース (CLI) クライアントは、Master Server に通信パスを提供します。Master Server は、この通信パスを利用して、ローカルシステムやリモートシステムからコマンドを実行します。 CLI クライアントは、次の環境でのコマンドの実行を可能にします。

- Windows コマンド行
- bash などの UNIX シェル

これらのコマンドを実行するため、CLI クライアントは、Master Sever との TCP/IP 接続、または SSL や SSH によるセキュア接続を確立します。

CLI クライアントは、次の 2 つのモードで動作します。

- コマンドを 1 つずつ送信するシングルコマンドモード
- コマンドプロンプトの表示、コマンド履歴の管理、Jython スクリプティングの許可を実行する対話型モード

対話型モードのとき、CLI クライアントは Jython プログラミング言語を使用します。Jython は、動的でオブジェクト指向型の高水準言語である Python の Java 実装です。

注 – CLI クライアントを対話型モードで実行するすべてのシステムに、Jython をインストールしてください。 Jython の詳細情報の確認とダウンロードは、`http://www.jython.org` で行います。

ネットワークプロトコル

N1 Service Provisioning System 4.1 は、ソフトウェアアプリケーション間の通信用として、さまざまなネットワークプロトコルをサポートしています。 次の各ネットワーク通信について、使用するプロトコルを選択できます。

- Master Server と Local Distributor または Remote Agent 間の通信
- 特定の Local Distributor と Remote Agent 間の通信
- Master Server と CLI クライアント間の通信

N1 Service Provisioning System 4.1 がサポートするプロトコルは次のとおりです。

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

ネットワークセキュリティは、特定のネットワークトポロジのニーズを満たすようにカスタマイズできます。 たとえば、個々のデータセンター間の通信はセキュリティ保護されているが、リモートデータセンターとのネットワーク接続が公開インターネット経由で確立されているとしましょう。 インターネット経由のすべての通信をセキュリティ保護するためには、このリモートデータセンターのファイアウォールの内側にインストールされている Local Distributor と Master Server を接続する際、SSL を使用する設定にします。 ローカルネットワーク経由のすべての通信がセキュリティ保護されるので、Local Distributor は、Raw TCP/IP を使って Remote Agent に接続することができます。 さまざまなプロトコルの構成方法については、第 6 章と第 7 章を参照してください。

Raw TCP/IP

Raw TCP/IP は、その他の暗号化や認証を必要としない標準 TCP/IP です。 この TCP/IP には、追加設定や追加構成が不要であるという利点があります。 データセンターネットワークがファイアウォールの内側にあり、侵入から保護されている場合、N1 Service Provisioning System 4.1 アプリケーション間の通信手段として利用できます。

Secure Shell

Secure Shell (SSH) は、リモートコンピュータに安全にアクセスするための UNIX コマンド群であり、プロトコルでもあります。 電子証明書とパスワードの暗号化を利用して、接続の両端で認証を行うことにより、ネットワーク上のクライアントとサーバーの通信を保護します。 また、RSA 公開鍵暗号化を使って、接続と認証の管理を行います。 SSH は、telnet やその他のシェルベースの通信方式よりも安全性の面で優れています。

SSH を使って通信するように、N1 Service Provisioning System 4.1 アプリケーションを構成できます。 N1 Service Provisioning System 4.1 は、OpenBSD Project によって開発された無料バージョンの SSH である OpenSSH をサポートします。 OpenSSH の詳細については、`http://www.openssh.com` を参照してください。 その他のバージョンの SSH をサポートするソフトウェア構成も可能です。

Secure Sockets Layer

Secure Sockets Layer (SSL) は、IP ネットワーク経由の通信を保護するプロトコルです。 TCP/IP ソケットテクノロジを利用して、クライアントとサーバーのメッセージ交換を行います。交換されるメッセージは、RSA が開発した公開鍵と秘密鍵による暗号化システムで保護されます。 SSL は、Netscape Navigator™、Microsoft の Web ブラウザをはじめとする大多数の Web サーバー製品でサポートされます。

ソフトウェアメッセージの傍受や改ざんを防ぐため、SSL を使ってネットワーク通信を行うように N1 Service Provisioning System 4.1 アプリケーションを構成することができます。 ネットワークセキュリティをさらに強化するため、SSL を使って、通信前にアプリケーション同士が認証を行うように構成することもできます。

第 2 章

N1 Service Provisioning System 4.1 のシステム要件

この章では、N1 Service Provisioning System 4.1 をインストールし、使用するためのシステム要件を紹介します。 この章の内容は次のとおりです。

一般的なシステム要件

この節では、N1 Service Provisioning System 4.1 をインストールし、使用するための要件を紹介します。

サポート対象のオペレーティングシステム

N1 Service Provisioning System 4.1 Master Server は、次のオペレーティングシステムを実行するシステムにインストールできます。

- Solaris 8 または Solaris 9 システム
- Red Hat Linux 7.2、7.3、8.0 および Red Hat Advanced Server 2.1
- Microsoft Windows 2000 Server および Microsoft Windows 2000 Advanced Server

N1 Service Provisioning System 4.1 Remote Agent、Local Distributor、および CLI クライアントは、次のオペレーティングシステムを実行するシステムにインストールできます。

- Solaris 2.6、Solaris 7、Solaris 8、または Solaris 9 システム
- Red Hat Linux 7.2、7.3、8.0 および Red Hat Advanced Server 2.1
- IBM AIX 4.3.3、5.1、5.2
- Microsoft Windows 2000 Server および Microsoft Windows 2000 Advanced Server

サポート対象の Web ブラウザ

次の表に、N1 Service Provisioning System 4.1 Web インタフェースに必要な Web ブラウザを一覧表示します。

表 **2–1** HTML ユーザーインタフェースに必要な Web ブラウザ

プラットフォーム	ブラウザ
Solaris	Netscape Navigator 6.2.2、Netscape Navigator 7.0
Red Hat	Netscape Navigator 6、Netscape Navigator 7.1
Windows	Internet Explorer 5.5 および 6、Netscape Navigator 6、Netscape Navigator 7.1

必要なオペレーティングシステムパッチ

次の表に、サポート対象の各オペレーティングシステムに必要なパッチを一覧表示します。

表 **2–2** サポート対象のオペレーティングシステムに必要なパッチ

OS のバージョン	必要なパッチ
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

表 2–2 サポート対象のオペレーティングシステムに必要なパッチ　(続き)

OS のバージョン	必要なパッチ
Solaris 7	106980-16
	106541-16
	107544-03
	106950-13
	106327-08
	106300-09
Solaris 8	なし
Solaris 9	なし
IBM AIX 4.3.3.0	AIX 4330-09 メンテナンスレベル (APAR IY22024)
IBM AIX 5.1.0.0	AIX 5100-01 メンテナンスレベル (APAR IY21957)
	APAR IY19375
Red Hat Linux 7.2、7.3、または 8.0	なし
Red Hat Linux Advanced Server 2.1	なし
Windows 200 Server または Windows 2000 Advanced Server	Service Pack 3

SSH の要件

Solaris OS、Red Hat Linux、または IBM AIX システム上の接続を SSH で保護したい場合は、SSH を使用する各マシンに SSH プロトコルバージョン 2 をインストールする必要があります。

Jython の要件

CLI クライアントで Jython を使用したい場合は、Jython バージョン 2.0 以上をインストールする必要があります。 Jython の詳細については、`http://www.jython.org` を参照してください。

ロケールの要件

N1 Service Provisioning System 4.1 は国際化対応です。各国語の環境にインストールし、実行することができます。 ただし、このためには、次の要件を満たしている必要があります。

- すべてのアプリケーションを同一ロケールまたは同等のロケールで実行する。 Remote Agent、Local Distributor、および CLI クライアントは、Master Server と同一のロケールで実行する必要がある
- ファイル名、ディレクトリ名、その他の入力には ASCII 文字だけを使用する

アプリケーション要件

この節では、個々の N1 Service Provisioning System 4.1 アプリケーションをインストールし、使用するための要件を紹介します。

- 26 ページの「Solaris OS 上のアプリケーションのシステム要件」
- 27 ページの「Red Hat Linux 上のアプリケーションのシステム要件」
- 28 ページの「IBM AIX 上のアプリケーションのシステム要件」
- 29 ページの「Windows 2000 上のアプリケーションのシステム要件」

Solaris OS 上のアプリケーションのシステム要件

Solaris Master Server

Solaris Master Server には、Solaris 8 または Solaris 9 オペレーティングシステムが必要です。 システムのハードウェア要件は次のとおりです。

- 450 MHz のシングルまたはマルチ CPU (SPARC® ハードウェア専用)
- 1G バイト以上の RAM
- 2G バイトの HD 空き領域。 必要なリポジトリ容量は、配備したアプリケーションのサイズによって決定される

次の表に、Master Server を実行する Solaris システムに必要な `/etc/system` の設定を一覧表示します。

注 – Solaris 9 オペレーティングシステムを使用する場合、`shmsys:shminfo_shmmin` と `shmsys:shminfo_shmseg` の値を変更することはできません。 デフォルトの値をそのまま使用してください。

表 **2–3** Solaris `/etc/system` 設定

変数	最小値	推奨値
`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] 10 進値 536870912 (512M バイト)。ただし、Solaris 8 オペレーティングシステムでは、この数値を 16 進値で指定する必要がある

Solaris Local Distributor、Remote Agent、および CLI クライアント

Solaris Local Distributor には、Solaris 6、Solaris 7、Solaris 8、または Solaris 9 オペレーティングシステムが必要です。 システムのハードウェア要件は次のとおりです。

- 400 MHz のシングルまたはマルチ CPU (SPARC ハードウェア専用)
- 256M バイト以上の RAM
- 1G バイトの HD 空き領域。 必要なキャッシュ容量は、配備したアプリケーションのサイズによって決定される

Red Hat Linux 上のアプリケーションのシステム要件

Red Hat Linux Master Server のインストールの開始時に、ユーザーのパスに `bc` コマンドが置かれていなければなりません。 `bc` コマンドがない場合、インストールは終了し、`bc` コマンドのインストールを要求するメッセージが表示されます。 `bc-1.06-5.rpm` か、これ以降のバージョンのパッケージをインストールしてください。

Red Hat Linux Master Server

Red Hat Linux Master Server には、次のどれかのバージョンの Red Hat Linux が必要です。

- Red Hat Linux 7.2、7.3、または 8.0
- Red Hat Advanced Server 2.1

システムのハードウェア要件は次のとおりです。

- 1 GHz のシングルまたはマルチ CPU (Intel x86 対応ハードウェア専用)
- 1G バイト以上の RAM
- 2G バイトの HD 空き領域。 必要なリポジトリ容量は、配備したアプリケーションのサイズによって決定される

Red Hat Linux Master Server のインストーラは、次のシステムパラメータをチェックし、最小値が満たされていない場合、エラーを発行して終了します。

表 **2–4** Red Hat システム設定

システムパラメータ	最小値	推奨値
`/proc/sys/kernel/shmall` の `shmall`	536870912 (512M バイト)	2147483647 (2048M バイト)
`/proc/sys/kernel/shmmax` の `shmmax`	536870912 (512M バイト)	2147483647 (2048M バイト)

Red Hat Linux Local Distributor、Remote Agent、および CLI クライアント

Red Hat Linux Local Distributor、Remote Agent、および CLI クライアントには、次のいずれかのバージョンの Red Hat Linux が必要です。

- Red Hat Linux 7.2、7.3、または 8.0
- Red Hat Advanced Server 2.1

システムのハードウェア要件は次のとおりです。

- 1 GHz のシングルまたはマルチ CPU (Intel x86 対応ハードウェア専用)
- 1G バイト以上の RAM
- 1G バイトの HD 空き領域。 必要なキャッシュ容量は、配備したアプリケーションのサイズによって決定される

IBM AIX 上のアプリケーションのシステム要件

IBM AIX Local Distributor、Remote Agent、および CLI クライアントには、AIX 4.3.3、5.1.0、または 5.2.0 のオペレーティングシステムが必要です。 システムのハードウェア要件は次のとおりです。

- 400 MHz のシングルまたはマルチ CPU (pSeries のハードウェア専用)
- 256M バイト以上の RAM
- 1G バイトの HD 空き領域。 必要なキャッシュ容量は、配備したアプリケーションのサイズによって決定される

Windows 2000 上のアプリケーションのシステム要件

Windows Master Server、Remote Agent、Local Distributor、または CLI クライアントの実行時には、ホームディレクトリに、中間ファイルを作成するための空き領域が必要です。 必要な容量は、アプリケーションを実行するためにインストールされているファイルのサイズとほぼ同量になります。

Windows 2000 Master Server

WindowsMaster Server には、次のいずれかのバージョンの Windows が必要です。

- Windows 2000 Server
- Windows 2000 Advanced Server

システムのハードウェア要件は次のとおりです。

- 1 GHz のシングルまたはマルチ CPU (Intel x86 対応ハードウェア専用)
- 1G バイト以上の RAM
- 2G バイトの HD 空き領域。必要なリポジトリ容量は、配備したアプリケーションのサイズによって決定される

Windows 2000 Local Distributor、Remote Agent、および CLI クライアント

Windows Local Distributor、Remote Agent、および CLI クライアントには、次のいずれかのバージョンの Windows が必要です。

- Windows 2000 Server
- Windows 2000 Advanced Server

システムのハードウェア要件は次のとおりです。

- 1 GHz のシングルまたはマルチ CPU (Intel x86 対応ハードウェア専用)
- 1G バイト以上の RAM
- 1G バイトの HD 空き領域。必要なリポジトリ容量は、配備したアプリケーションのサイズによって決定される

第 3 章

インストール情報の収集

この章では、意思決定と、N1 Service Provisioning System 4.1 のインストールに必要な全情報の収集に役立つ情報とワークシートを提供します。 この章の内容は次のとおりです。

構成上の決定事項

インストールプログラムは、N1 Service Provisioning System 4.1 の構成情報の入力を求めるプロンプトを表示します。 インストールを開始する前に、以下の各節の情報を利用して、必要事項を決定してください。

Java Runtime Environment:

Solaris OS、Red Hat Linux、または IBM AIX システムでインストールを実行している場合、JRE をインストールするか、JRE への有効なパスを入力するように求めるプロンプトが表示されます。 Windows システムでインストールを実行している場合、JRE はインストールプログラムによって自動的にインストールされます。プロンプトは表示されません。

Red Hat Linux システムでインストールを実行している場合、インストールスクリプトにより、マシン上のデフォルトの場所に JRE のインスタンスがあるかどうかがチェックされます。 JRE がデフォルトの場所にない場合は、インストールする必要があります。 JRE がデフォルトの場所にある場合、JRE を再インストールするかどうかを選択できます。

Solaris OS またはIBM AIX システムでインストールを実行しているとき、JRE をインストールしない設定を選択すると、インストールスクリプトは、有効なJRE のパスの入力を要求するプロンプトを表示します。 続いて、この JRE がサポートされているかどうかが検証されます。 この JRE がサポートされていない場合 (サポートされている JRE よりバージョン番号が大きい場合)、サポートされていない JRE で処理を続行するかどうかを確認する警告メッセージが表示されます。 N1 Service Provisioning System 4.1 でサポートされているバージョンの JRE を指定した場合、インストールスクリプトにより、`JRE_HOME` 変数に指定の JRE が設定されます。 続いて、JRE ディレクトリをポイントするシンボリックリンク *N1SPS4.1-home*`/common/jre` が作成されます。シンボリックリンクにより、N1 Service Provisioning System 4.1 アプリケーションは、他のアプリケーションが依存している可能性のある場所 を変更しないでJRE を使用できます。

注 – バンドル版の JRE は、各マシンに 1 回だけインストールします。 たとえば、Master Server、Local Distributor、CLI クライアントを同一マシンにインストールする場合、JRE は Master Server と同時にインストールします。Local Distributor や CLI クライアントと同時にはインストールしません。

ユーザーのアプリケーション所有権

インストールプログラムは、インストールするアプリケーションを所有するユーザーとグループの選択を促すプロンプトを表示します。 SSH 通信を行うようにアプリケーションを構成したい場合は、Master Server、Local Distributor、Remote Agent を同じユーザー ID でインストールします。

スーパーユーザー (root) は、Master Server の所有者にはなれません。 Master Server は、Master Server を所有するユーザーの ID でインストールします。スーパーユーザー (root) としてアプリケーションをインストールし、プロンプトが表示されたら root を所有するユーザーを指定するという方法も選択できます。

Remote Agent に、この Remote Agent の実行マシンの root 権限を許可したい場合は、スーパーユーザー (root) としてインストールプログラムを実行する必要があります。 スーパーユーザー (root) 以外のユーザーを Remote Agent の所有者に指定した場合も、Remote Agent にその実行マシンの root 権限を許可したい場合は、スーパーユーザーとしてインストールプログラムを開始します。

ネットワークプロトコル

インストールプログラムは、ソフトウェアアプリケーション間の通信に使用するネットワークプロトコルの選択を促すプロンプトを表示します。 ソフトウェアをインストールする前に、使用する暗号化方式、TCP/IP、SSH、または SSL を決定します。 SSL を選択した場合は、使用する暗号群、認証なしの暗号化、または認証ありの暗号化も指定します。

ネットワークプロトコルの詳細については、20 ページの「ネットワークプロトコル」を参照してください。

Jython

CLI クライアントをインストールする際、インストールプログラムは、マシンに Jython がインストールされているかどうかを指定するプロンプトを表示します。 CLI クライアントは、Jython プログラミング言語を使って、対話型モードで実行できます。 なお、CLI クライアントは、Jython なしでも実行できます。 Jython と CLI クライアントの詳細については、20 ページの「コマンド行インタフェースクライアント」を参照してください。

全アプリケーション用ワークシート

個々の N1 Service Provisioning System 4.1 アプリケーションのインストールスクリプトは、共通の準備作業を実行し、ディレクトリとファイルに関する共通の質問に答えると実行されます。 次のワークシートを使って、個々の N1 Service Provisioning System 4.1 アプリケーションのインストールに必要な情報を収集します。

表 3–1 全アプリケーション用ワークシート

必要な情報	回答
ソフトウェアをインストールするベースディレクトリ	
JRE がインストール済みの場合、JRE のパス (たとえば `/usr/local/jre` または `JAVA_HOME` 環境変数の値)	
インストールするアプリケーションを所有するユーザー	
Solaris OS、Red Hat Linux、および IBM AIX システムで、インストールするアプリケーションを所有するグループ	

Master Server 用ワークシート

Master Server のインストールに必要な情報を収集する際に、次のワークシートを使用します。

表 **3–2** Master Server 用ワークシート

必要な情報	回答
Master Server マシンのホスト名または IP アドレス	
CLI クライアントが Master Server との接続に使用する IP ポート番号	
ソフトウェアが電子メール通知の送信に使用する SMTP メールサーバーのホスト名または IP アドレス	
ソフトウェアから送信される電子メール通知の件名	
電子メール通知の差出人のユーザーアカウント (ユーザー名)	
ネイティブコマンドの実行時にソフトウェアが使用するユーザーアカウント	
Postgres データベースが待機するポート番号	
Web インタフェースを使用できるポート番号	
Postgres データベースを自動的に最適化するか? 最適化する場合は、Master Server データベースを最適化する時刻 日次ベースでデータベースを最適化する場合、`crontab` ファイル内にエントリが作成される	Y/N HH:MM

Local Distributor 用ワークシート

次のワークシートを使って、Local Distributor のインストールに必要な情報を収集します。

表 **3–3** Local Distributor 用ワークシート

必要な情報	回答
Local Distributor マシンの IP アドレスまたはホスト名	
この Local Distributor が待機するポート番号	

Remote Agent 用ワークシート

次のワークシートを使って、Remote Agent のインストールに必要な情報を収集します。

表 **3–4** Remote Agent 用ワークシート

必要な情報	回答
Remote Agent の実行マシンの IP アドレスまたはホスト名	
Remote Agent が待機するポート番号	

CLI クライアント用ワークシート

次のワークシートを使って、CLI クライアントに必要な情報を収集します。

表 **3–5** CLI クライアント用ワークシート

必要な情報	回答
コマンド行ユーザーインタフェースの Master Server の IP アドレスまたはホスト名	
Master Server の IP ポート番号	
このマシンに Jython がインストールされている場合、Jython のパス (例: `/usr/local/jython`)	

第 4 章

Solaris OS、Red Hat Linux、IBM AIX システムへのインストール

この章では、Solaris OS、Red Hat Linux、IBM AIX システムへの N1 Service Provisioning System 4.1 のインストール手順について説明します。 この章の内容は次のとおりです。

N1 Service Provisioning System 4.1 のインストール

個々のアプリケーションのインストールには、製品メディア上の適切なインストールスクリプトを使用します。 N1 Service Provisioning System 4.1 アプリケーションのインストールスクリプトを実行する前に、共通の準備作業を行います。また、ディレクトリ、ファイル、Java™ 実行時環境 (JRE) のインストールに関する共通の質問に答える必要があります。 その後、インストールする各アプリケーションの構成情報を指定します。

▼ Solaris OS、Red Hat Linux、IBM AIX システムへ N1 Service Provisioning System 4.1 をインストールする

はじめに　表 1–1 の作業マップを確認し、 アプリケーションをインストールするための準備作業を完了してください。

手順 **1.** アプリケーションの所有者になるユーザーとしてログインします。

スーパーユーザー (root) としてログインし、ソフトウェアをインストールすることができます。 インストールプログラムは、必要に応じて、ソフトウェアを所有するユーザーの情報を要求します。

2. **CD** を挿入します。

- Solaris OS システムへのインストールの場合、N1 Service Provisioning System 4.1: Solaris CD を挿入する
- IBM AIX または Red Hat Linux へのインストールの場合、N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD を挿入する

3. **Solaris OS** へのインストールの場合、ソフトウェアをパッケージとしてインストールするか、ファイルとしてインストールするかを指定します。

Master Server と CLI クライアントアプリケーションについても、パッケージとしてインストールするか、ファイルとしてインストールするかを選択できます。 パッケージとしてインストールする場合とファイルとしてインストールする場合では、選択するインストールプログラムが異なります。 スーパーユーザー以外のユーザーとしてインストールを実行する場合は、ソフトウェアをファイルとしてインストールします。

4. ソフトウェア **CD** の挿入後、インストールスクリプトが格納されているディレクトリに移動します。

```
% cd /script-directory
```

script-directory には、次のいずれかの値が入ります。

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

5. インストールするアプリケーションのインストールスクリプトを実行します。

```
% cr_app_opsystem_4.1.sh
```

app には、次のいずれかの値が入ります。

- ms – Master Server をインストール
- ra – Remote Agent をインストール
- ld – Local Distributor をインストール
- cli – CLI クライアントをインストール

opsystem には、次のいずれかの値が入ります。

- solaris – インストール先は Solaris OS
- aix – インストール先は IBM AIX
- linux – インストール先は Red Hat Linux

注 – Solaris Master Server をパッケージとしてインストールするインストールスクリプトは、`cr_ms_solaris_pkg_4.1.sh` です。 Solaris CLI クライアントをパッケージとしてインストールするインストールスクリプトは、`cr_cli_solaris_pkg_4.1.sh` です。 これらのスクリプトを使って、パッケージ版の Solaris Master Server と CLI クライアントをインストールします。

6. ログファイルの場所を記録します。
   インストールプログラムにより、ログファイルが作成され、その格納場所が表示されます。 あとでログファイルの内容を確認できるように、ファイルの場所を記録しておきます。
7. インストールプログラムのプロンプトに答えて、構成情報を指定します。
   インストールが完了すると、アプリケーションの再起動を促すメッセージが表示されます。
8. インストールスクリプトを **CD** から実行している場合は、**n** と入力します。

Solaris OS、Red Hat Linux、IBM AIX システムへの Remote Agent の非対話型インストール

構成情報を指定するパラメータファイルを使用すれば、Remote Agent を非対話的にインストールすることができます。 この場合、パラメータファイル内の構成情報が参照されます。したがって、インストール中に構成情報を要求するメッセージは表示されません。

▼ Solaris OS、Red Hat Linux、IBM AIX システムへ非対話方式で Remote Agent をインストールする

はじめに　Remote Agent のインストール前に、Master Server をインストールする必要があります。 Master Server をインストールするマシンが Remote Agent のインストールマシンと一致している必要はありません。

手順　1. **Remote Agent** をインストールするマシンに、**Remote Agent** を所有するユーザーとしてログインします。
   スーパーユーザー (root) としてログインし、ソフトウェアをインストールすることができます。 インストールプログラムは、必要に応じて、ソフトウェアを所有す

るユーザーの情報を要求します。

2. **CD** を挿入します。
   - Solaris OS システムへのインストールの場合、N1 Service Provisioning System 4.1: Solaris CD を挿入する
   - IBM AIX または Red Hat Linux へのインストールの場合、N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD を挿入する
3. ソフトウェア **CD** の挿入後、インストールスクリプトが格納されているディレクトリに移動します。

   ```
   % cd /script-directory
   ```

   script-directory には、次のいずれかの値が入ります。
   - solaris – N1 Service Provisioning System 4.1: Solaris CD 上
   - aix – N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD 上
   - linux – N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD 上
4. **Remote Agent** のインストールマシンにインストールスクリプトをコピーします。

   ```
   % cp cr_ra_opsystem_41.sh RA-machine/
   ```

   RA-machine には、Remote Agent をインストールするマシン上のディレクトリを指定します。*opsystem* には、次のいずれかの値が入ります。
   - solaris – インストール先は Solaris OS
   - aix – インストール先は IBM AIX
   - linux – インストール先は Red Hat Linux
5. インストールスクリプトのコピー先ディレクトリにパラメータファイルをコピーします。

   Master Server のインストール時に、Master Server 上の *N1SPS4.1-MasterServer-home*`/server/bin` ディレクトリに、サンプルパラメータファイルがインストールされます。 このファイル内のデフォルト値を使用するか、ファイルを編集してカスタム値を追加します。 サンプルパラメータファイル `cr_ra_41_remote_params.sh` の内容については、110 ページの「Solaris OS、Red Hat Linux、IBM AIX の Remote Agent パラメータファイル (サンプル)」」を参照してください。

   新しいパラメータファイルを作成することもできます。 パラメータファイルは実行可能ファイルでなければなりません。

   N1SPS4.1-MasterServer-home は、Master Server のインストールディレクトリです。
6. インストールスクリプトを実行します。

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

opsystem には、次のいずれかの値が入ります。

- solaris – インストール先は Solaris OS
- aix – インストール先は IBM AIX
- linux – インストール先は Red Hat Linux

parameters-file には、インストールプログラムが構成情報のソースとして使用するパラメータファイルの名前を指定します。 パラメータファイルは実行可能ファイルでなければなりません。

Solaris OS、Red Hat Linux、IBM AIX システムへの Remote Agent のリモートインストール

Remote Agent は、ネットワーク上の別のマシンからリモートインストールすることもできます。 この場合、Master Server のインストール時に、Remote Agent のリモートインストールに必要なスクリプトが `/server/bin` ディレクトリにインストールされます。 インストールは非対話方式で行われます。また、Remote Agent のインストールと構成を管理する環境変数を使用します。 環境変数は、インストールスクリプトのデフォルト値をそのまま受け入れるか、パラメータファイル内またはコマンド行に設定します。

▼ Solaris OS、Red Hat Linux、IBM AIX に Remote Agent をリモートインストールする

はじめに　ターゲットマシンの必要条件は以下のとおりです。

- UNIX の `sshd` ユーティリティを実行し、ソースマシンとの直接 IP 接続が確立されていること
- リモートインストールスクリプトが呼び出す、UNIX の `hostname` コマンド。 Remote Agent は、`hostname` コマンドが返すホスト名の IP アドレス上で待機する

Master Server マシンの実行時に、パス内に UNIX の `ssh` ユーティリティと `scp` ユーティリティがインストールされている必要があります。

リモートインストールプログラムは、環境変数を使って、Remote Agent のインストールと構成を管理します。 環境変数は、インストールスクリプトのデフォルト値をそのまま受け入れるか、パラメータファイル内またはコマンド行に設定します。 以下に、必要な環境変数とそのデフォルト値を示します。

- `CR_RA_INSTALLER_USER=root` – インストールプログラムは、スーパーユーザーをアプリケーションの所有者として Remote Agent をインストールする

- CR_RA_INSTALLER_WORKDIR=/tmp – インストールスクリプトのコピーはターゲットマシンの /tmp ディレクトリに保存される
- CR_RA_INSTALLER_LEAVEFILES=no – インストールプログラムは、インストールプログラムの完了時に、作業ディレクトリにコピーされているすべてのファイルを削除する
- CR_RA_INSTALLER_HOSTS=host1,host3.enterprise.com,10.10.0.207 – コマンド行または環境変数を使ってホスト名を指定しなかった場合、インストールスクリプトはエラーなしで終了する

手順

1. **Master Server** マシンに **CD** を挿入します。
   - Solaris OS システムへのインストールの場合、N1 Service Provisioning System 4.1: Solaris CD を挿入する
   - IBM AIX または Red Hat Linux へのインストールの場合、N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD を挿入する
2. ソフトウェア **CD** の挿入後、インストールスクリプトが格納されているディレクトリに移動します。

   % cd /*script-directory*

   script-directory には、次のいずれかの値が入ります。
   - solaris – N1 Service Provisioning System 4.1: Solaris CD 上
   - aix – N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD 上
   - linux – N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD 上
3. インストールスクリプトを **Master Server** にコピーします。

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

   N1SPS4.1-MasterServer-home は Master Server のインストールディレクトリです。*opsystem* には、次のいずれかの値が入ります。
   - solaris – インストール先は Solaris OS
   - aix – インストール先は IBM AIX
   - linux – インストール先は Red Hat Linux
4. スクリプトが格納されているディレクトリに移動します。

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

   N1SPS4.1-MasterServer-home は、Master Server のインストールディレクトリです。
5. インストールスクリプトが使用する構成情報の指定方法を決定します。

- 新しいパラメータファイルを作成するか、N1 Service Provisioning System 4.1 によってインストールされたサンプルパラメータファイルを編集する。 パラメータファイルは、Master Server のインストール時にインストールされる。 ファイル名は *N1SPS4.1-MasterServer-home*/server/bin/cr_ra_41_remote_params.sh。 このファイル内のデフォルト値を使用するか、ファイルを編集してカスタム値を追加する。 新しいパラメータファイルを作成することもできる。 サンプルパラメータファイルの内容については、「110 ページの「Solaris OS、Red Hat Linux、IBM AIX の Remote Agent パラメータファイル (サンプル)」」を参照。 パラメータファイルは実行可能ファイルでなければならない
- 環境変数を設定する

```
% 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. リモートインストールを実行します。

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

- *opsystem* には、次のいずれかの値が入る
  - solaris – インストール先は Solaris OS
  - aix – インストール先は IBM AIX
  - linux – インストール先は Red Hat Linux
- cr_ra_*opsystem*_4.1.sh には、N1 Service Provisioning System 4.1 CD からコピーするインストールファイルを指定する
- *path-to-file/parameters-file* には、パラメータファイルのパスと名前を指定する。 インストールプログラムは、この名前を利用して構成情報を取得する。 環境変数を設定する場合や、インストールスクリプトのデフォルト値を使用する場合、パラメータファイルを指定する必要はない
- *hostnames* には、インストールを実行するマシンのホスト名 (複数可) を指定する。 個々のホスト名は空白文字で区切る。 パラメータファイルまたは環境変数を使って、CR_RA_INSTALLER_HOSTS パラメータでホスト名を指定した場合、コマンド行にホスト名を指定する必要はない。 コマンド行にホスト名を指定した場合、CR_RA_INSTALLER_HOSTS パラメータで指定された値は、コマンド行に指定された値で上書きされる

7. ログファイルの場所を記録します。

インストールプログラムにより、ログファイルが作成され、その格納場所が表示されます。 あとでログファイルの内容を確認できるように、ファイルの場所を記録しておきます。

8. プロンプトに続いて、リモートマシンのパスワードを入力します。

インストールスクリプトにより、リモートマシン上にログファイルが生成されます。

第 5 章

N1 Service Provisioning System 4.1 を Windows システムへインストールする

この章では、Windows を実行しているシステムに N1 Service Provisioning System 4.1 をインストールする手順について説明します。 個々のアプリケーションのインストールには、製品メディア上の適切なインストールスクリプトを使用します。 個々の N1 Service Provisioning System 4.1 アプリケーションのインストールスクリプトは、共通の準備作業を実行し、ディレクトリとファイルに関する共通の質問に答えると実行されます。 その後、インストールする各アプリケーションの構成情報を指定します。

この章の内容は次のとおりです。

Master Server のインストール

▼ Windows システムに N1 Service Provisioning System 4.1 Master Server をインストールする

はじめに　表 1–1 の作業マップを確認し、 Master Server をインストールするための準備作業を完了してください。

手順　**1. N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD** を挿入します。

2. **Windows** ファイルマネージャか **DOS** ウィンドウを使って、**CD** 上の **`windows`** ディレクトリにアクセスします。

3. **Master Server** のインストールスクリプトを実行します。

   - ファイルマネージャを使用する場合は、`cr_ms_win32_4.1.msi` ファイルをダブルクリックする
   - DOS ウィンドウを使用する場合は、プロンプトに続いてインストールファイルの名前を入力する

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

4. インストールプログラムのプロンプトに答えて、構成情報を指定します。
   一連の情報を指定すると、「Ready to Install」画面が表示されます。

5. 「**Install**」をクリックして、インストールを開始します。
   インストールプログラムにより、プログラムファイルがインストールされます。続いて、マシンの再起動を促すメッセージが表示されます。

6. マシンを再起動すれば、インストールは完了です。
   マシンを再起動するまで、N1 Service Provisioning System 4.1 のインストールは完了しません。

7. システムにログインします。
   インストールプログラムにより、「ようこそ」画面が表示されます。

8. 「**Next**」をクリックして、インストールを行います。

注 – インストーラは、DOS ウィンドウを開き、コマンドを実行します。 実行するのに数分かかるコマンドもあります。 操作が完了するまでは、DOS ウィンドウを開いたままにしておいてください。 操作は、数分間で自動的に完了します。

---

9. 「**Finish**」をクリックして、インストールプログラムを終了します。
   Master Server がインストールされました。 Web ブラウザを起動し、インストール時に指定した Web インタフェースアドレスを使って、Master Server にアクセスします。

10. **(**任意**)** タスクのスケジュールにより、データベースを最適化することができます。
    データベースのパフォーマンスを最適化したい場合は、`vacuumdb` ユーティリティを毎日実行するようなタスクを作成します。 タスクの作成方法については、47 ページの「タスクをスケジュールしてデータベースを最適化する」を参照してください。

▼ タスクをスケジュールしてデータベースを最適化する

手順

1. **Windows 2000** のタスクフォルダを開きます。
   タスクフォルダを開くには、「スタート」メニューをクリックし、「すべてのプログラム」→「アクセサリ」→「システム ツール」→「タスク」を選択します。
2. 新しいタスクを作成するには、フォルダ内を右クリックし、「新規」→「タスク」を選択します。
3. タスクに名前を付けます。
4. タスクをダブルクリックして、編集します。
5. 「**Run**」フィールドに、次のコマンドを入力します。途中で改行はしません。

   `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 は、Master Server のインストールディレクトリです。
6. 「スケジュール」タブで、日単位で **1** 回実行するようにタスクを構成します。

Remote Agent、Local Distributor、CLI クライアントのインストール

▼ Windows システムへ Remote Agent、Local Distributor、CLI クライアントをインストールする

はじめに　表 1–1 の作業マップを確認し、 Master Server をインストールするための準備作業を完了してください。

手順

1. **N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD** を挿入します。
2. **Windows** ファイルマネージャか **DOS** ウィンドウを使って、**CD** 上の **`windows`** ディレクトリにアクセスします。
3. インストールしたいアプリケーションのインストールスクリプトを起動します。

- ファイルマネージャを使用する場合は、cr_*app*_win32_4.1.msi ファイルをクリックする
- DOS ウィンドウを使用する場合は、プロンプトに続いてインストールファイルの名前を入力する

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

app には、次のいずれかの値が入ります。

- ra – Remote Agent をインストール
- ld – Local Distributor をインストール
- cli – CLI クライアントをインストール

4. インストールプログラムのプロンプトに答えて、構成情報を指定します。
   一連の情報を指定すると、「Ready to Install」画面が表示されます。
5. 「**Install**」をクリックして、インストールを開始します。
   インストールプログラムにより、プログラムファイルがインストールされます。
6. 「**Finish**」をクリックして、インストールプログラムを終了します。

Windows システムへの Remote Agent の非対話型インストール

Remote Agent をインストールする際、コマンド行の変数で構成内容を指定することができます。 Remote Agent の非対話型インストールには、Windows インストーラサービスの一部としてインストールされる msiexec コマンドを使用します。

▼ Windows システムへ Remote Agent を非対話方式でインストールする

手順

1. **Remote Agent** をインストールするマシンで、**DOS** ウィンドウを開きます。
2. **N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD** を挿入します。
3. ソフトウェア **CD** の挿入後、**Windows** のインストールスクリプトが格納されているディレクトリに移動します。

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

   path-to-CD には、ソフトウェア CD のパスを指定します。

4. **Remote Agent** のインストールマシンにインストールスクリプトをコピーします。

   ```
   % cp cr_ra_win32_4.1.sh RA-machine/
   ```

   RA-machine には、Remote Agent をインストールするマシン上のディレクトリを指定します。

5. インストールを開始します。

   ```
   C:RA-machine\>  msiexec /i cr_ra_win32_4.1.msi /qn
   VARIABLE=value  VARIABLE=value
   ```

   指定できる変数の数に制限はありません。 ディレクトリ名など、変数値に空白文字が含まれる場合は、その部分を引用符で囲む必要があります。 非対話型インストールプログラムで有効な変数および値については、表 5–1 を参照してください。 変数や値を指定しない場合、Remote Agent はデフォルトの構成値でインストールされます。

例 5–1 Windows システムへの Remote Agent の非対話型インストール

Windows システムに Remote Agent を非対話方式でインストールしたい場合、次の例のように入力します。

```
C:\> msiexec /i cr_ra_win32_4.1.msi/ qn
INSTALLDIR=C:\Program Files\N1 Service Provisioning System\"R
A_PARENT_CONNECTION=false
```

Windows への Remote Agent のリモートインストール

Remote Agent を非対話モードでリモートインストールする場合は、Remote Agent のインストールスクリプトを活用できます。 Windows スクリプティングホストが使用する `.wsh` スクリプトを使って、インストールを行います。 このスクリプトファイルには、次の処理を行う VB スクリプトが含まれています。

- リモートシステムの WMI DCOM インタフェースに接続する
- WMI を使って、ターゲットシステム上に一時的な Windows ファイル共有を作成する
- ローカルの `cr_ra_win32_4.1.msi` をターゲットの共有へコピーする
- WMI をリモートで使用して、ターゲットマシン上でサイレント MSI を実行する

▼ Windows へ Remote Agent をリモートインストールする

手順

1. **Master Server** マシンで、**DOS** ウィンドウを開きます。
2. **N1 Service Provisioning System 4.1: IBM AIX, Redhat Linux, Windows 2000 Server/Advanced Server CD** を挿入します。
3. ソフトウェア **CD** の挿入後、**Windows** のインストールスクリプトが格納されているディレクトリに移動します。

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

   path-to-CD には、ソフトウェア CD のパスを指定します。
4. インストールスクリプトを **Master Server** にコピーします。

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

   MS-machine には、Master Server マシン上のディレクトリを指定します。
5. インストールを開始します。

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

   Hostname には、Remote Agent のインストールマシンのホスト名を指定します。

   コマンド行の *parameters* の位置に何も値を指定しない場合、Remote Agent は次に示すデフォルトの構成値でインストールされます。

   Remote Agent の非対話型インストールプログラムは、次の表のパラメータを受け付けます。

パラメータ	説明	デフォルト
`-user`	ターゲットマシンの WMI に接続するユーザー	なし
`-password`	ターゲットマシンの WMI に接続するときのパスワード	なし
変数	`cscript WinInstaller.wsf` コマンドの Windows 変数。表 5–1 を参照。 すべての変数、すべての値は、途中で改行しないで入力し、文字列全体を引用符で囲む	なし
`-msiLocation`	インストールする `.msi/.input` ファイルのパス	現在の作業ディレクトリ

パラメータ	説明	デフォルト
-shareLocation	ターゲットマシン上の既存のディレクトリ。一時的な Windows ファイル共有は、このディレクトリに作成される。 少なくとも、インストールスクリプトを格納できるディレクトリ容量が必要	C:\WINNT\Temp

終了コード 0 は正常なインストールの完了、終了コード 1 はインストールの失敗を表します。

例 5–2 Windows への Remote Agent のリモートインストール

Windows へ Remote Agent をリモートインストールする場合は、次の例のように入力します。

```
C:\> cscript WinInstaller.wsf -sharelocation C:\installs -options
"INSTALLDIR='C:\Program Files\N1 Service Provisioning System'" targetHost
```

Remote Agent 変数の値

Remote Agent の非対話型インストールやリモートインストールのプログラムは、次の変数を受け付けます。

表 5–1 Remote Agent 変数の値

変数名	説明	デフォルト	値
INSTALLDIR	Remote Agent のインストールディレクトリを指定する	C:\Program Files\N1 Service Provisioning System	任意の有効なディレクトリ
REMOTE_AGENT_HOSTNAME	Remote Agent のインストールマシンのホスト名または IP アドレスを指定する	Windows のコンピュータ名	任意の有効なホスト名または IP アドレス
RA_PORT_NUMBER	この Remote Agent 用の IP ポート番号を指定する	2313	任意の有効なポート番号
RA_PARENT_CONNECTION	暗号化されていない (raw) 接続、または SSL 接続を使ってこの Remote Agent に接続する親アプリケーションを指定する	false	true の場合 SSL、false の場合 raw 接続を使用する

表 **5–1** Remote Agent 変数の値　(続き)

変数名	説明	デフォルト	値
RA_SSL_CIPHER	SSL を選択した場合、使用する SSL 暗号タイプを指定する	1	0 の場合は認証付きの暗号化、 1 の場合は認証なしの暗号化を使用する
RA_SERVICE_USERNAME RA_SERVICE_PASSWORD	Remote Agent を実行するユーザーのユーザーアカウントを指定する	システムユーザー	ローカルのユーザー名を使用する場合は、接頭辞 .\ が必要。 これらの変数を定義した場合、 RA_SERVICE_CONTROL を other に設定する必要がある
RA_SERVICE_AUTOSTART	システムの再起動時に Remote Agent を自動的に起動するかどうかを指定する。インストール時に Remote Agent を起動するかどうかも、この変数で決定される	1	1 の場合は自動的に起動する。 0 の場合は自動的には起動しない

第 6 章

Secure Shell を使用するための N1 Service Provisioning System 4.1 の構成

この章では、Secure Shell (SSH) 通信を行うように N1 Service Provisioning System 4.1 を構成する方法について説明します。

N1 Service Provisioning System 4.1 は OpenSSH 2.0 をサポートします。 OpenSSH 2.0 は、OpenBSD Project によって開発された SSH の無料版です。 詳細については、`http://www.openssh.com` を参照してください。 その他のバージョンの SSH をサポートする構成も可能です。

注 – この章で紹介するコマンドとインタフェースは OpenSSH 2.0 用です。その他のバージョンの SSH を使用する場合は、その SSH の付属文書で、対応するコマンドやオプションを確認してください。OpenSSH 2.0 のコマンドとオプションの詳細については、66 ページの「OpenSSH 2.0 コマンドリファレンス」」を参照してください。

この章の内容は次のとおりです。

SSH の概要と使用条件

SSH は、リモートコンピュータに安全にアクセスするための UNIX ベースのコマンド群兼プロトコルです。 電子証明書とパスワードの暗号化を利用して、接続の両端で認証を行うことにより、ネットワーク上のクライアントとサーバーの通信を保護します。 また、RSA 公開鍵暗号化を使って、接続と認証の管理を行います。 telnet やその他のシェルベースの通信方式より安全性が高く、Web サーバーやリモートシステムの管理に使用されます。

2 つの N1 Service Provisioning System 4.1 アプリケーション間にSSH 接続を設定した場合、ダウンストリームのアプリケーションは手動で起動する必要はありません。 アップストリームのアプリケーションによって必要に応じて自動的に起動されます。 その後、使用されている間は継続して動作しますが、一定の時間使用されていない場合、自動的に停止します。この一連の動作は、SSH 独特のものであり、その他の接続タイプでは見られません。

SSH 接続を使用する場合は、ダウンストリームのアプリケーションを手動で起動しないでください。 たとえば、Local Distributor が SSH を使って Remote Agent と接続するように設定した場合、Remote Agent を手動では起動しないでください。 Local Distributer は、必要に応じて Remote Agent を自動的に起動します。Remote Agent は使用されている間は継続して作動します。Local Distributor は、一定の時間使用されていない場合、Remote Agent を自動的に停止します。この時間は設定可能です。

空のパスワードキーと ssh-agent

SSH の構成時に、空のパスワードキーを使用するか、`ssh-agent` を使用するかを選択できます。 空のパスワードキーを使用する構成の場合、生成された SSH 非公開鍵は空のパスワード付きで格納されます。 よって、この非公開鍵には、パスワードなしでアクセスできます。 公開鍵を信頼するその他のマシンと SSH を使って通信するとき、パスワードプロンプトは表示されません。 `ssh-agent` を使用する場合、生成された非公開鍵はセキュリティ保護されたパスワード付きで安全なメディアに格納されます。 別のマシンと通信するときは、`ssh-agent` を起動し、安全なメディアから非公開鍵をアップロードして、パスワードを入力します。 非公開鍵はファイルシステムには格納されず、`ssh-agent` プロセスのメモリーに格納されます。

空のパスワードを使用する場合、非公開鍵はパスワードなしでマシンのファイルシステムに格納されます。 この非公開鍵は、SSH 通信を開始するすべてのマシン上に存在しなければなりません。 N1 Service Provisioning System 4.1 の場合、SSH を使ってダウンストリームのアプリケーションに接続するすべての Master Server と Local Distributor に、非公開鍵を配布する必要があります。 この方法では、高いセキュリティ効果は期待できません。

`ssh-agent` を使用する場合、非公開鍵は Master Server 上でしか時実行されない `ssh-agent` によって格納されます。 ネットワーク上のその他のマシンには、公開鍵が配布されます。 認証を必要とする SSH アプリケーションは、`ssh-agent` を使って

認証を行います。 中間 SSH 接続を、Master Server 上で認証のために実行される ssh-agent へ Local Distributer からのプロキシ設定が可能になるように設定するときには、エージェントの転送機能を有効にする必要があります。 Local Distributor は、エージェントの転送機能を使って、ダウンストリームの Local Distributor や Remote Agent に認証情報を渡します。 この方法のほうが、先ほどの方法よりセキュリティの面で優れています。

SSH の要件

N1 Service Provisioning System 4.1 は、次の SSH 機能を要求します。

- ssh によるリモートコマンド呼び出し
- 公開 - 非公開鍵認証
- BatchMode yes (ユーザーの介入なしで ssh コマンドを呼び出す機能) のサポート

ssh-agent を使用する場合、次の SSH 機能が必要になります。

- ssh-agent のサポート
- SSH の ssh-agent 転送機能のサポート。 OpenSSH での -A オプションの使用

SSH 接続を行うマシンでは、次のような便利な機能を利用できます。ただし、これらの機能は必須ではありません。

- リモートコマンド呼び出し実行時の tty の強制割り当て。 OpenSSH での -t オプションの使用
- ssh エージェントの強制終了。 OpenSSH での ssh-agent コマンドの -k オプションの使用
- 高度なセキュリティを実現する RSA 鍵の生成。 OpenSSH での -t rsa の使用

次のチェックリストを使って、SSH の実装が N1 Service Provisioning System 4.1 の要件を満たしているかどうかを確認します。

- ssh-keygen コマンドは、SSH 呼び出しの認証に使用できる公開鍵と非公開鍵のペアを生成する必要がある
- 追加情報なしでホストキーの交換やパスワードの取得が可能な指定のホスト上で、認証用非公開鍵がパスワードなしで (または空のパスワード付きで) 生成された場合、次の ssh コマンドを実行できなければならない

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

- パスワード付きで生成された非公開鍵を使ってアップロードされたホストで、ssh-agent を実行して、host1、host2 を経て host3 にホップしたとする。この host3 上で、追加情報なしでホストキーの交換やパスワードの取得が可能な場合、次の ssh コマンドを実行できなければならない

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

- ssh コマンドを使って、標準入力ストリーム、標準出力ストリーム、標準エラーストリームを、リモートマシンで実行されるコマンドに正常にパイプできなければならない
- ssh-add コマンドを使って、パスワード付きの非公開鍵を認証用として ssh-agent にアップロードできなければならない

SSH の構成 (作業マップ)

次の表に、N1 Service Provisioning System 4.1 が SSH を使用するように構成するために必要なタスクを示します。

表 **6–1** 作業マップ: SSH の構成

作業	説明	参照先
セキュリティレベルの決定	空のパスワードを使用するか ssh-agent を使用するかを判断する	54 ページの「空のパスワードキーと ssh-agent」
鍵の生成	SSH 接続を開始するアプリケーション上で鍵を生成する	57 ページの「鍵のペアを生成する」
鍵の設定	生成された鍵を Local Distributor と Remote Agent にコピーする。 空のパスワード鍵を使用するか ssh-agent を使用するかに応じて、適切な作業を選択する	57 ページの「1 組の鍵ペアを使用するとき、空のパスワードファイルに鍵を設定する」 58 ページの「複数の鍵ペアを使用するとき、空のパスワードファイルに鍵を設定する」 59 ページの「ssh-agent に鍵を設定する」
Master Server 上の接続の設定とテスト	Master Server を起動する前に、SSH 接続を設定し、テストする	59 ページの「Master Server 上の接続の設定とテスト」
Local Distributor と Remote Agent で SSH を使用する構成	SSH を使用するように、Local Distributor と Remote Agent を構成する	62 ページの「SSH を使用するように Local Distributor と Remote Agent を構成する」
(任意) CLI クライアントで SSH を使用する構成	CLI クライアントを使用する場合、SSH を使用するように構成する	63 ページの「ssh-agent を使って、SSH を使用するように CLI クライアントを構成する」

鍵の準備

Master Server とLocal Distributor やRemote Agent との通信の認証に使用する公開- 非公開鍵のペアを生成します

ssh-agent を使用する場合、必要な鍵のペアは 1 組だけです。 空のパスワードを使用する場合は、2 台のマシン間をつなぐ SSH 接続ごとに 1 組ずつ鍵のペアを生成します。 または、1 組の鍵のペアをすべての接続で使用することもできます。 生成するペアごとに、次の処理を行います。

▼ 鍵のペアを生成する

手順 **1. Master Server (**空のパスワードを使用し、接続ごとに鍵のペアを生成する場合はアップストリームのマシン**)** 上で、鍵を生成します。

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

パスワードプロンプトが表示されます。

2. パスワードが必要かどうかを確認します。

- 空のパスワード鍵を使用する場合は不要。 Return キーを押して継続
- ssh-agent を使用する場合は鍵のパスワードを入力

鍵を保存するかどうかを確認するプロンプトが表示されます。

3. Return キーを押して、デフォルトの場所に鍵を保存します。

非公開鍵の保存場所は、/*User-home*/.ssh/id_rsa です。 公開鍵の保存場所は、/*HOME*/.ssh/id_rsa.pub です。

User-home には、現在 Master Server マシンにログインしているユーザーのホームディレクトリが入ります。

▼ 1 組の鍵ペアを使用するとき、空のパスワードファイルに鍵を設定する

手順 **1.** 非公開鍵を、**Master Server** からアップストリームの各マシンにコピーします。 この鍵をホームディレクトリに保存します。

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

User-home には現在 Master Server マシンにログインしているユーザーのホームディレクトリ、*User-home-upstream* にはアップストリームのマシンのホームディレクトリを指定します。 アップストリームのマシンは、ダウンストリームのマシンとの SSH 接続を開始するマシンです。

Local Distributor ごとに固有の非公開鍵を持たせるか、すべての Local Distributor に同じ非公開鍵を共有させるかを選択できます。

2. 公開鍵をダウンストリームの各マシンにコピーします。 この鍵を **/.ssh/authorized_keys2** ファイルに保存します。

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

   User-home には、Master Server のホームディレクトリを指定します。一方、*User-home-downstream* には、前の手順で設定したマシンの接続先となる Local Distributor または Remote Agent マシンのホームディレクトリを指定します。SSH を使って接続を確立するすべての Local Distributor および Remote Agent に公開鍵をコピーします。

3. **.ssh/** ディレクトリとその親ディレクトリ **(**存在する場合**)** が **world writable** でないことを確認します。

4. その他のユーザーまたはグループが、非公開鍵ファイル **.ssh/id_rsa** へのアクセスを許可されていないことを確認します。

5. **.ssh/authorized_keys2** ファイルのアクセス権ビットを **600** に変更します。

▼ 複数の鍵ペアを使用するとき、空のパスワードファイルに鍵を設定する

はじめに　SSH 接続ごと (ネットワーク上の鍵ペアごと) に、次の処理を行います。

手順　1. 公開鍵を、アップストリームのマシンからダウンストリームの各マシンにコピーします。 この鍵を *User-home***/.ssh/authorized_keys2** ファイルに保存します。

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

   User-home-upstream には、アップストリームのマシンのホームディレクトリを指定します。*Use-home-downstream* には、アップストリームのマシンの接続先となる Local Distributor または Remote Agent マシンのホームディレクトリを指定します。

2. **.ssh/** ディレクトリとその親ディレクトリ **(**存在する場合**)** が **world writable** でないことを確認します。

3. その他のユーザーまたはグループが、非公開鍵ファイル **.ssh/id_rsa** へのアクセスを許可されていないことを確認します。

4. **.ssh/authorized_keys2** ファイルのアクセス権ビットを **600** に変更します。

▼ ssh-agent に鍵を設定する

手順 1. **Master Server** 上の非公開鍵ファイル **~/.ssh/id_rsa** を安全なメディアにコピーします。

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

User-home には現在 Master Server マシンにログインしているユーザーのホームディレクトリ、*path_to_file/* には非公開鍵ファイルの保存先となる安全なメディアのパスを指定します。

2. 非公開鍵ファイルをローカルファイルシステムから削除します。

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

3. 公開鍵を、**SSH** を使用するように設定する個々の **Local Distributor** と **Remote Agent** にコピーします。 この鍵を **~/.ssh/authorized_keys2** ファイルに保存します。

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

User-home には、Master Server マシンのホームディレクトリを指定します。*User-home-APP* には、現在 Local Distributor または Remote Agent マシンにログインしているユーザーのホームディレクトリを指定します。

4. **.ssh/** ディレクトリとその親ディレクトリ **(**存在する場合**)** が **world writable** でないことを確認します。

5. **.ssh/authorized_keys2** ファイルのアクセス権ビットを **600** に変更します。

6. 次の行を、**Master Server** と **Local Distributor** の **config.properties** ファイルに追加します。これで、**ssh-agent** の転送機能が有効になります。

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

Master Server 上の接続の設定とテスト

この節では、SSH の初期設定とテストについて説明します。この初期設定とテストが完了してから、SSH を使用するように N1 Service Provisioning System 4.1 を構成します。 ssh-agent を使用する場合は、設定とテストを開始する前に、ssh-agent を実行する必要があります。

▼ Master Server 上で ssh-agent を起動する

ssh-agent を使用する場合は、Master Server を起動する前に、次の処理を行います。ssh-agent を使用しない場合、この処理は不要です。

手順 **1. `ssh-agent` を起動します。**

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

ssh-agent により、2 つの環境変数 SSH_AUTH_SOCK と SSH_AGENT_PID が設定されます。ssh と ssh-add は、これらの環境変数を使って ssh-agent に接続します。

2. 生成した非公開鍵をアップロードします。

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

path-to-file/ には、非公開鍵ファイルを保存した安全なメディアのパスを指定します。

パスワードプロンプトが表示されます。

3. 鍵の生成時に作成したパスワードを入力します。

参考 ssh-agent の停止

eval `ssh-agent -k` コマンドで、ssh-agent を停止できます。

このコマンドは、SSH_AGENT_PID 変数を使って、ssh-agent プロセスに停止信号を送ります。 また、ssh-agent の起動時に設定した環境変数の設定を解除します。

▼ Master Server 上で接続を設定し、テストする

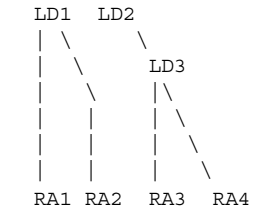
はじめに ssh-agent を使用する場合は、「59 ページの「Master Server 上で ssh-agent を起動する」」に説明されている手順に従って、ssh-agent を起動します。 設定はセッションを認識するため、ssh-agent を起動したセッションと同じセッションで、すべてのSSH コマンド(ssh、ssh-add、cr_server start) を実行する必要があります。 このセッションが終了している場合は、実行中の ssh-agent プログラムを強制終了して、新規に ssh-agent プログラムを実行しなければなりません。 また、非公開鍵をアップロードする必要があります。

手順 **1. SSH 接続パスをテストします。**

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

-A オプションは、ssh-agent を使用する場合だけ使用してください。 *target-host-IP* には、このマシンの接続先マシンの IP アドレスを指定します。

たとえば、Master Server (MS)、Local Distributor (LD1、LD2、LD3)、Remote Agent (RA1、RA2、RA3、RA4) が設定されているネットワークについて考えてみましょう。



このネットワークでは、Master Server 上で次のコマンドを実行し、ネットワーク上のLocal Distributor とRemote Agent のIP アドレスを LD1、LD2、RA1、RA2、RA3、RA4 と置き換えることで、SSH 接続パスをテストします。

```
% 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
```

これらのコマンドは、Master Server が SSH を使ってダウンロードストリームのマシンに接続するとき使用するパスをたどります。 各コマンドにより、SSH で、引数として指定されたマシンと通信するために必要なホストキーの交換が可能になります。

ホストキーの交換を許可するかどうかを確認する SSH プロンプトが表示されます。

2. すべてのプロンプトに「**yes**」で答えます。
3. すべてのコマンドの出力で、環境変数が正しく設定されていることを確認します。

   PATH 変数には、/bin、/usr/bin のほか、ユーザーの環境を構成するすべてのディレクトリを指定します。
4. **SSH** 接続パスを再度テストします。

   手順 1 と同じコマンドを実行して、接続パスを再度テストし、情報の入力を求めるシステムプロンプトが表示されないことを確認します。

参考 設定とテストのくり返し

この処理は、鍵に変更を加えるたびに、くり返し行う必要があります。 システムの設定によっては、マシンのリブートを行うたびに、この処理を行わなければならない場合もあります。

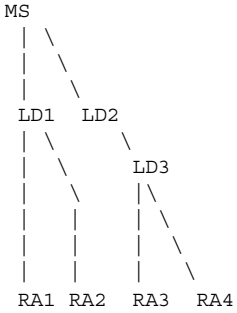
アプリケーションの構成

Master Server 上での SSH の設定とテストが完了したら、Master Server が SSH を使って接続する N1 Service Provisioning System 4.1 のその他のマシンを構成します。

▼ SSH を使用するように Local Distributor と Remote Agent を構成する

Master Server から Remote Agent へと N1 Service Provisioning System 4.1 ネットワークをたどり、中間の Local Distributor を検出順に構成することにより、SSH 構成を完了しておく必要があります。 これは、本質的には、ツリーネットワークを先行順にたどる (preorder traversal) 処理になります。

たとえば、Master Server (MS)、Local Distributor (LD1、LD2、LD3)、Remote Agent (RA1、RA2、RA3、RA4) が設定されているネットワークについて考えてみましょう。



LD1、RA1、RA2、LD2、LD3、RA3、RA4 の順でネットワークを構成します。 この順で、あるマシンの構成を完了してから次のマシンに進みます。

手順

1. **Web** インタフェースを使って、構成するマシンの「**Host Details**」ページを表示します。う
2. そのマシンでどのアプリケーションを構成するかによって、**Local Distributor** と **Remote Agent** のどちらかのセクションに接続の詳細情報を追加します。
3. 接続タイプとして **ssh** を指定します。
4. 「**Advanced Parameters**」フィールドに次のテキストを追加します。

   cprefix=/*N1SPS4.1-Home***/***application*

   N1SPS4.1–Home には、アプリケーションのホームディレクトリを指定します。 *application* には、Remote Agent を構成している場合はエージェント、Local Distributor を構成している場合は ID を指定します。

   たとえば、N1 Service Provisioning System 4.1 のインストールディレクトリが `/opt/SUNWn1sps/N1_Service_Provisioning_System_4.1/` で、Remote Agent を構成している場合は、次のテキストを追加します。

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

5. ホストの詳細情報を保存します。

6. このマシン上で **Remote Agent** または **Local Distributor** のインスタンスが実行されていないことを確認します。

7. このアプリケーションインスタンスの「**Host Details**」ページを開き、「**Test Connection**」をクリックします。

8. ここまでの処理を、ネットワーク内のすべてのマシンについてくり返します。

▼ ssh-agent を使って、SSH を使用するように CLI クライアントを構成する

`ssh-agent` を使って、SSH 接続を行うように CLI クライアントを構成するには、次の手順に従ってください。

手順

1. **Master Server** と、**CLI** クライアントのインストール先のマシン上で、新しいオペレーティングシステムユーザーアカウントを作成します。

   Master Server、Local Distributor、Remote Agent のインストール時に指定したアカウント以外となります。

2. 前の手順で作成した新規ユーザーのアカウントで、**Master Server** にログインします。

3. 「**57** ページの「鍵のペアを生成する」」の手順に従って、新規ユーザーの公開鍵と非公開鍵を生成します。

   Master Server、Local Distributor、Remote Agent の通信用として生成した鍵を再利用することはできません。

4. **Master Server** 上で、非公開鍵ファイルを安全なメディアにコピーします。

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

   User-home には、現在 Master Server マシンにログインしているユーザーのホームディレクトリが入ります。 *path-to-file/* には、非公開鍵ファイルを保存する安全なメディアのパスを指定します。

5. 非公開鍵ファイルをローカルファイルシステムから削除します。

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

6. **Master Server** 上で、ユーザーの **/.ssh/authorized_keys2** ファイルの末尾に公開鍵を連結します。

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

   User-home には、Master Server マシンのホームディレクトリを指定します。

7. 先ほど作成した新規ユーザーのアカウントで、**CLI** クライアントにログインします。

8. **ssh-agent** を起動します。

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

User-home には、CLI クライアントマシンに現在ログインしているユーザーのホームディレクトリを指定します。

9. **.profile** または **.cshrc** ファイルに次の行を追加します。

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

User-home には、CLI クライアントマシンのホームディレクトリを指定します。

10. **Master Server** からいったんログアウトし、再度ログインします。

11. 生成した非公開鍵をアップロードします。

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

path-to-file/ には、非公開鍵ファイルを保存する安全なメディアのパスを指定します。

CLI クライアントは、Master Server と接続するときの認証に、SSH と `ssh-agent` を使用するようになります。

12. ローカルホストからの接続だけを許可するように、**Master Server** を構成します。具体的な手順については、**85** ページの「**JVM** セキュリティポリシーの構成」を参照してください。

参考 `ssh-agent` の停止

注 – `ssh-agent` を停止したい場合は、CLI クライアント上で次のコマンドを実行します。

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

User-home には、CLI クライアントマシンに現在ログインしているユーザーのホームディレクトリを指定します。

▼ 空のパスワードで CLI クライアントが SSH 接続を行うように構成する

空のパスワードで CLI クライアントが SSH 接続を行うように構成するには、次の手順に従ってください。

手順 1. **Master Server** と、**CLI** クライアントをインストールするマシン上で、新しいオペレーティングシステムユーザーアカウントを作成します。

Master Server、Local Distributor、Remote Agent のインストール時に指定したアカウント以外となります。

2. 前の手順で作成した新規ユーザーのアカウントで、**CLI** クライアントマシンにログインします。

3. **57** ページの「鍵のペアを生成する」の手順に従って、新規ユーザーの公開鍵と非公開鍵を生成します。

   Master Server、Local Distributor、Remote Agent の通信用として生成した鍵を再利用することはできません。

4. **CLI** クライアント上の公開鍵ファイルを、**Master Server** マシン上の新規ユーザーの **`authorized_keys2`** ファイルにコピーします。

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

   User-home-CLI には CLI クライアントマシンのホームディレクトリ、*User-home-MS* には Master Server マシンのホームディレクトリを指定します。

5. **Master Server** 上で、ユーザーの **`/.ssh/authorized_keys2`** ファイルの末尾に公開鍵を連結します。

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

   User-home には、現在 Master Server マシンにログインしているユーザーのホームディレクトリが入ります。

6. 先ほど作成した新規ユーザーのアカウントで、**CLI** クライアントにログインします。

7. **SSH** 接続をテストします。

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

   IP-Address-MS には、Master Server マシンの IP アドレスを指定します。

   鍵の交換を促すメッセージが表示される場合があります。

8. 鍵の交換を促すメッセージが表示された場合は、「**yes**」で答えます。

9. **`PATH`** 変数が正しく設定されていることを確認します。

   `PATH` 変数には、`/bin`、`/usr/bin` のほか、ユーザーの環境を構成するディレクトリをすべて指定する必要があります。

10. ローカルホストからの接続だけを許可するように、**Master Server** を構成します。具体的な手順については、**85** ページの「**JVM** セキュリティポリシーの構成」を参照してください。

jexec ラッパー

SSH を介して Remote Agent を起動した場合、Remote Agent は jexec ラッパーを使って Java 仮想マシンを起動します。 jexec ラッパーはスーパーユーザー (root) が所有するネイティブの実行可能ファイルであり、ビットセット `setuid` を保持していま

す。 さらに、Remote Agent をインストールしたユーザーと同じ groupid を保持しているため、グループに実行権が許可されます。 このファイルは、Remote Agent をインストールしたユーザーが所有する `protect` ディレクトリに格納されます。 Remote Agent を所有するユーザーだけに、このファイルの実行権が許可されます。 その他のユーザーは、jexec ラッパーを実行できません。

`jexec` と `protect` のファイルアクセス権が誤って変更されることがないように、常に注意を払ってください。

次の変更を加えることにより、jexec のセキュリティを強化することができます。

- JVM 実行ファイル (通常、シェルスクリプト) の所有者としてスーパーユーザー (root) またはアプリケーションを所有するユーザーを指定し、それ以外のユーザーやグループに書き込み権を許可しない。 N1 Service Provisioning System 4.1 で JRE をインストールする場合は、*N1SPS4.1-home*`/common/jre` 内の全ファイルは必ずアプリケーションの所有者によって所有されるようにし、それ以外のユーザーやグループに書き込みアクセスを許可しない
- アプリケーションを所有するユーザーのユーザー ID は、SSH を使ってログインするとき以外は許可しない。 SSH を使ってログインする場合は、公開鍵認証だけを許可する。 アプリケーションを所有するユーザー以外のユーザーやグループには、*/N1SPS4.1-home/*`.ssh` ディレクトリへのあらゆるアクセスを禁止する
- 公開鍵認証だけを許可するように、SSH サーバーを構成できる。この場合は、パスワード認証を無効にするため、`etc/sshd_config` ファイルに次の行が含まれていなければならない

  ```
  PasswordAuthentication no
  ```

- `etc/sshd_config` ファイルに、RhostsRSAAuthentication という語を含む行が含まれていてはならない。この認証は、デフォルトでは許可されていない。RSAAuthentication という語を含む行がある場合、その値を `yes` (デフォルト) に設定する必要がある
- Remote Agent のセキュリティをさらに強化したい場合は、*/N1SPS4.1-home/*`.ssh/authorized_keys2` ファイルを編集して、Master Server の公開鍵が含まれる行の前に次のテキストを追加する

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

その他の詳細情報については、`sshd(1M)` のマニュアルページを参照してください。

OpenSSH 2.0 コマンドリファレンス

この節では、この章で扱う OpenSSH 2.0 コマンドと各種オプションについて説明します。 別のバージョンの SSH を使用している場合は、そのバージョンで、次のコマンドと同等のコマンドならびにオプションが使用できるかどうかを確認してください。

表 **6–2** OpenSSH 2.0 コマンド

ツール	説明
`ssh`	アプリケーションにその他のアプリケーションのリモート呼び出しを許可する。 SSH 通信を行う構成にした場合、ソフトウェアは `ssh` コマンドを使ってリモートアプリケーション (Remote Agent または Local Distributor) を呼び出す。リモートアプリケーションとの通信には、SSH の標準入出力ストリームを使用する
`ssh-agent`	パスワード付きの非公開鍵を使用したい場合に使用する。 アプリケーションの SSH 呼び出しが `ssh-agent` 通信で認証を行えるように、`ssh-agent` を使って鍵をアップロードする
`ssh-add`	`ssh-agent` に非公開鍵をアップロードする
`ssh-keygen`	SSH 接続を保護するため、公開鍵 - 非公開鍵のペアを生成する

`ssh` コマンドには、次のオプションがあります。

`-A` 認証エージェント転送を有効にする

`-o 'BatchMode yes'` パスフレーズクエリーを無効にする

`-t` コマンドが発行されている場合も `tty` を割り当てる

`ssh-keygen` コマンドには、次のオプションがあります。

`-t rsa` 生成する鍵のタイプを RSA にする

`ssh-agent` コマンドには、次のオプションがあります。

`-k` 環境変数 `SSH_AGENT_PID` 内の `pid` セットを使ってエージェントを強制終了する。 その他の実装では、別の環境変数を使用する可能性がある

第 7 章

SSL を使用する構成

この章では、Secure Socket Layer (SSL) 通信を行うように N1 Service Provisioning System 4.1 を構成する方法について説明します。 この章の内容は次のとおりです。

N1 Service Provisioning System 4.1 における SSL のサポートの概要

SSL は、IP ネットワーク経由の通信を保護するためのプロトコルです。 SSL では、TCP/IP ソケットテクノロジを利用してクライアントとサーバー間のメッセージ交換を行います。交換されるメッセージは、RSA の開発による公開鍵 - 非公開鍵暗号化システムで保護されます。 SSL は、Netscape Navigator、Microsoft の Web ブラウザをはじめとする大多数の Web サーバー製品でサポートされます。

第三者によるメッセージの読み取りや改ざんを防ぐため、SSL を使ってネットワーク通信を行うように N1 Service Provisioning System 4.1 アプリケーションを構成できます。 オプションとして、通信前に認証を行うように構成すれば、さらに高いネットワークセキュリティ効果を期待できます。

暗号群: 暗号化と認証の概要

SSL プロトコルは、さまざまな暗号アルゴリズム、暗号方式をサポートします。こうした暗号は、サーバーとクライアント間の認証、証明書の送信、セッションキーの確立などに使用されます。 認証が行われるかどうかは、SSL が接続に使用する暗号群によって決まります。

暗号群の選択は慎重に行なってください。 どのアプリケーションも、ノードが要求する最小限のセキュリティを提供する暗号群だけを有効にする必要があります。 SSL は、クライアントとサーバーの両方でサポートされる最も安全な暗号群を使用します。 低セキュリティの暗号群を有効化した場合、サン以外のクライアントは、暗号群のネゴシエーション時に、最小限のセキュリティしか提供しない暗号群を選択することがあります。この場合、サーバーは、安全性の低い暗号群を使用しなければならなくなります。

SSL は、次のモードで動作します。

- 暗号化のみ、認証なし – 接続は暗号化されるが、接続するアプリケーションの認証は行われない
- サーバーの認証 – クライアントは接続先のサーバーを認証する
- サーバーとクライアントの認証 – クライアントとサーバーの両方が双方を認証する

インストール時、アプリケーション間の通信を保護する手段として SSL を選択すると、使用する暗号群を指定するプロンプトが表示されます。 暗号群の値は、`config.properties` ファイルの `net.ssl.cipher.suites` に格納されます。 選択内容によって、暗号群に次の値を設定できます。

- 「暗号化のみ、認証なし」を選択した場合、暗号群の値は `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA`
- 「認証と暗号化」を選択した場合、暗号群の値は `SSL_RSA_WITH_3DES_EDE_CBC_SHA`

サーバーの認証を必要とする、または必要としない SSL 暗号群の一覧は、「84 ページの「SSL 暗号群」」を参照してください。 サーバーの認証を必要とする暗号群を対象に、クライアントの認証を構成することもできます。

認証キーストア

N1 Service Provisioning System 4.1 は、自己署名付き証明書をサポートします。 次の 2 種類のキーストアがあります。

- プライベートキーストア – アプリケーションが別のアプリケーションに接続するとき自身の認証用として使用する公開鍵 - 非公開鍵のペアが格納される
- トラストキーストア – このキーストアが信頼し、アプリケーションへの接続を許可する、その他のアプリケーションの自己署名付き証明書内の公開鍵が格納される

クライアントサーバー認証で SSL を有効にした場合、各アプリケーションに、SSL が使用する 2 つのキーストアを構成する必要があります。2 つのキーストアのうち 1 つは、ほかのアプリケーションに対する自身の認証用、もう 1 つはほかのアプリケーションの認証用です。

サーバー認証だけで SSL を有効にした場合、SSL サーバーとして機能するアプリケーションにはプライベートキーストア、SSL クライアントとして機能するアプリケーションにはトラスト (パブリック) キーストアが必要になります。 パブリックキーストアは、Java Secure Sockets Extension (JSSE) v1.0.3 によって提供される独自の JKS フォーマットです。

両方のキーストアに、同一のパスワードを指定する必要があります。

たとえば、SSL を使って、SSL クライアントであるアプリケーション A と、SSL サーバーであるアプリケーション B を接続する場合について考えてみましょう。 どちらのアプリケーションも、サーバー認証を要求する暗号群を使用するように構成されています。 アプリケーション B のプライベートキーストアには公開鍵 - 非公開鍵のペア、アプリケーション A のトラストキーストアにはアプリケーション B の公開鍵が格納されていなければなりません。 アプリケーション A からアプリケーション B への接続を試みると、アプリケーション B はアプリケーション A に公開鍵を送信します。アプリケーション A は、この鍵と、トラストキーストア内の鍵を照合します。

アプリケーション B がクライアント認証を要求する場合、アプリケーション A のプライベートキーストアには公開鍵 - 非公開鍵のペア、アプリケーション B のトラストキーストアにはアプリケーション A の公開鍵が格納されていなければなりません。 アプリケーション A によって認証されたあと、アプリケーション B はアプリケーション A の公開鍵とトラストキーストア内の公開鍵を照合できます。

SSL でのパスワードの使用

トラストキーストアの操作にパスワードを設定した場合、このパスワードを使って、キーストアの整合性のチェックが行われます。 このパスワードでは、トラストキーストアの更新を防ぐことはできますが、トラストキーストア内へのアクセスを禁止することはできません。 トラストキーストアの内容に変更を加えたい場合は、パスワードを入力する必要があります。

プライベートキーストアの操作にパスワードを設定した場合、このパスワードを使って、キーストアの整合性のチェック、キーストアの内容の更新の禁止、非公開鍵のアクセスの暗号化と保護が行われます。

`crkeys` スクリプトは、両方のキーストアに同一のパスワードが指定されているかどうかの確認に使用されます。 証明書をインポートして、初めてトラストストアを作成するとき、このトラストストアには、`crkeys` スクリプトにより、プライベートストアと同じパスワードが設定されます (プライベートストアにパスワードが設定されている場合)。 同様に、初めてプライベートストアを作成するときも、`crkeys` により、プライベートストアにトラストストアと同じパスワードが設定されます (トラストストアにパスワードが設定されている場合)。

crkeys スクリプトを使って、アプリケーションの起動時にキーストアのパスワードプロンプトを表示し、パスワードを検証する場合は、-vpass オプションを指定します。 キーストアが存在する場合、crkeys スクリプトは、 キーストアのパスワードを求めるプロンプトを表示し、入力されたパスワードを照合します。 照合の結果、正しいパスワードであることが確認されると、標準出力にパスワードが表示されます。このパスワードは、その後、アプリケーションに渡されます。

N1 Service Provisioning System 4.1 上の SSL の制限事項

N1 Service Provisioning System 4.1 上の SSL 実装には、次の制限があります。

- 自己署名付き証明書以外はサポートされない。 トラストキーストアには自己署名付き証明書しか格納されない。 CA 署名付き証明書は使用できない
- トラストキーストアとプライベートキーストアに同一のパスワードを設定する必要がある。 また、プライベートキーストア内の各キーにストアと同一のパスワードを設定する必要がある。 この制限は、キーを作成するのに使用された crkeys スクリプトによって実行される
- パスワードは端末にエコーされる。 POSIX プラットフォーム上でこの制限を克服するには、端末エコーを無効にし、パスワードプロンプトを表示する起動スクリプトを作成する
- CLI クライアントアプリケーションのクライアント認証を有効にしても、この設定は、セキュリティの制限上サポートされない。 CLI クライアントアプリケーションは、キーストアパスワードの入力を求めるプロンプトを表示しない。 作成されたキーストアは、CLI クライアントのプロパティファイル内になければならない。

  N1 Service Provisioning System 4.1 は、接続する側と接続される側で同一のトラストキーストアを使用する。 よって、たとえば Master Server が Remote Agent に接続し、この Remote Agent の公開鍵を信頼する場合は、たとえ Remote Agent に欠陥が生じても、CLI クライアントがクライアント認証を使用する設定になっているならば、この Remote Agent の鍵を使って、Master Server に対して CLI クライアントの認証を行うことができる。

  CLI クライアントのクライアント認証はサポートされていない。よって、CLI クライアントはトラストストアしか持たない。 パスワードの使用には、トラストストアが改ざんされていないことを確認できるという利点がある。 パスワードはプロパティファイル内に指定できるが、CLI クライアントの実行のたびにユーザーにパスワードの入力を求めるほうが、セキュリティ効果が高い
- SSH 接続の場合、リモートアプリケーション、Local Distributor、Remote Agent は自動的に起動する。 これらのアプリケーションを起動するキーストアパスワードの入力プロンプトは表示されない。 ただし、アプリケーションの初期化時にキーストアを使用した場合、プロパティファイルにキーストアパスワードを指定する必要がある

- SSH を使って Master Server に接続するようにCLI クライアントを構成した場合、CLI クライアントは、Master Server にソケットを使って接続する SshProxy アプリケーションを利用して Master Server に接続する。 SshProxy は SSL を介して Master Server に接続できるが、この構成はサポートされていない

SSL の構成 (作業マップ)

次の表に、SSL を使用するように N1 Service Provisioning System 4.1 を構成するために必要な作業を示します。

表 **7–1** 作業マップ: SSL の構成

作業	説明	参照先
セキュリティレベルの決定	使用する SSL 接続を決定する	69 ページの「N1 Service Provisioning System 4.1 における SSL のサポートの概要」
(任意) Tomcat での SSL の有効化	Web インタフェースでの HTTPS の使用を有効にすることができる	73 ページの「Tomcat での SSL の有効化」
キーストアの作成	`crkeys` コマンドを使ってキーストアを作成する	75 ページの「キーストアの作成」
SSL の構成	`config.properties` ファイルを編集して SSL を構成する	78 ページの「SSL の構成」

Tomcat での SSL の有効化

N1 Service Provisioning System 4.1 Web インタフェースは、デフォルトの設定では、SSL を使用しません。 要求は、HTTPS ではなく HTTP 経由で送信されます。 HTTPS を有効化したい場合は、認証局 (CA) から発行された SSL 証明書を使用します。 証明書は、通常、マシンごとに固有です。

SSL 証明書は、次の区切り文字で囲まれた形式になっています。

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

および

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

▼ Tomcat の SSL 証明書を生成する

手順 **1.** **JRE** のインストールディレクトリに移動します。

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

JAVA-HOME には、JRE のインストールディレクトリを指定します。 JRE を N1 Service Provisioning System 4.1 と同時にインストールした場合、*N1SPS4.1-home*/common/JRE/bin がインストールディレクトリになります。

2. 証明書を生成します。

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

/keystore-location には、生成した鍵の格納先を指定します。 通常は、/etc/keystore を使用します。

password には、任意のパスワードを指定します。

3. すべてのプロンプトに答えて、情報を入力してください。

▼ Tomcat で SSL を有効にする

手順 **1.** **SSL** 証明書をインポートします。

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

keystore-location には、証明書テキストを保存するファイルのパスと名前を指定します。 このコマンドは、インポートした証明書の格納先ファイルの名前を出力します。 このファイルは通常、コマンドを実行したユーザーのホームディレクトリに保存されます。

2. **server.xml** ファイル内の次の行の **<!--** と **-->** を削除し、コメント状態を解除します。

```
<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. **Factory** 要素を次のように編集します。

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

path-tomcat-keystore-file には、Tomcat の keystore ファイルのパスを指定します。 *password* には、元のキーパスを作成したとき使用したパスワードを指定します。

SSL による Web インタフェースへの接続

SSL を使用するように N1 Service Provisioning System 4.1 を構成したあと、SSL を使ってサーバーに接続するようにユーザーに要求する構成にすることができます。

▼ SSL を使って接続するようにユーザーに要求する

手順

- 現在の **web.xml** ファイルを **Tomcat** の **/webapp/WEB-INF/web.xml.secure** ファイルで置き換えます。

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

N1SPS4.1-home にはアプリケーションのホームディレクトリを指定します。

▼ 元の構成に戻す

手順

- 元の構成に戻すには、**web.xml** ファイルを **/webapp/WEB-INF/web.xml.default** ファイルで置き換えます。

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

N1SPS4.1-home にはアプリケーションのホームディレクトリを指定します。

キーストアの作成

N1 Service Provisioning System 4.1 は、JRE 付属のキーツールユーティリティを使用します。 ユーザーがキーストアを作成できるように、キーツールユーティリティはシェルスクリプト `crkeys` にラップされています。 このスクリプトには、'keytool' ユーティリティに正しいパラメータが指定されているかどうかを確認する働きがあります。

キーストアを作成すると、自己署名付き証明書の X.509 識別名が次のように設定されます。

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

▼ キーストアを作成する

手順 ● キーを生成します。

```
% crkeys -options
```

使用する SSL 接続の種類に基づいてキーストアを作成したい場合、次のオプションを使用します。

`-alias` *application_hostname*	証明書または鍵のペアの別名を指定する。 アプリケーションのホスト名を別名として使用する。 キーストア内に重複する別名があってはならない
`-cpass`	キーストアとキーストア内のすべてのキーのパスワードを変更する
`-delete`	エンティティを指定して、鍵のペアまたは証明書をキーストアから削除する
`-export`	エンティティを指定して、自己署名付き証明書を指定のファイルにエクスポートする
`-file` *cert_file*	証明書をどのファイルからインポートするか、どのファイルへエクスポートするかを指定する
`-generate`	指定された別名に、新しい鍵のペアを生成する
`-help`	すべてのオプションを一覧表示する
`-import`	このノードへの接続を許可されたエンティティの自己署名付き証明書をインポートする。 証明書をインポートする際は、別名として、この証明書に記載されたノードのホスト名を使用する
`-keyalg` *keyalg*	鍵生成アルゴリズム。 デフォルトは 'RSA'。 'RSA' と 'DSA' のいずれかを指定できる
`-keysize` *keysize*	キーサイズ。 デフォルトは 102。DSA 鍵の場合は 512 から 1024、RSA 鍵の場合は 512 から 2048 の範囲内の 64 の倍数
`-list`	キーストアに格納されているすべてのエンティティを一覧表示する
`-new` *newpassword*	キーストアとキーストア内のすべてのキーの新しいパスワードを指定する
`-password` *password*	キーストアのパスワードを指定する。 パスワードを指定しない場合、ユーザーにパスワードの入力を求めるプロンプトが表示される
`-private`	操作の対象として、プライベートキーストアを指定する

`-validity` *days_valid*	自己署名付き証明書の有効期間を日数で指定する
`-trust`	操作の対象としてトラストキーストアを指定する

例 7–1 crkeys コマンドの例

次に、crkeys コマンドの使用例を示します。

公開鍵 - 非公開鍵のペアを生成するには:

```
crkeys -private –generate|-delete
    -alias application_hostname [-keyalg keyalg]
[-keysize keysize] [-validity days_valid]
[–password password]
```

鍵のペアの自己署名付き公開鍵をファイルにエクスポートするには:

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

前の例のようにしてエクスポートした自己署名付き公開鍵をトラストストアにインポートするには:

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

鍵または鍵のペアを削除するには:

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

すべての公開鍵を一覧表示するには:

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

SSL キーストア (トラストキーストアとプライベートキーストア) のパスワードを変更するには:

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

crkeys コマンドの使用方法を出力するには:

```
crkeys -help
```

SSL の構成

インストール時に、各アプリケーションは次のように構成されます。

- サーバー認証を要求する暗号群をサポートする
- クライアント認証は要求しない
- *N1SPS4.1-home*/*app*/data/private.store ファイル内のプライベートキーストアを検出する
- *N1SPS4.1-home*/*app*/data/trust.store ファイル内のトラストキーストアを検出する
- 各キーストアに空のパスワードを渡す

アプリケーションごとに、次のセキュリティチェックが行われるように SSL 構成を変更することができます。

- 各アプリケーションの暗号群を選択的に有効化する

  有効化する暗号群を明示的に指定できます。 指定しない場合、リファレンス実装はデフォルトで有効になっている暗号群を使用します。 リファレンス実装によって有効化されるデフォルトの暗号群は、サーバー認証を要求します。 サポートされる暗号群については、「84 ページの「SSL 暗号群」」を参照してください。
- アプリケーションが接続する SSL クライアントを認証するように指定する
- プライベートキーストアおよびトラストキーストアの場所とパスワードを指定する

注 – 認証を有効にするには、アプリケーションのインストール後にキーストアを初期化する必要があります。

▼ SSL を構成する

手順 ● **config.properties** ファイルを手動で編集し、**SSL** 構成を変更します。

次の表に、config.properties ファイル内の SSL 構成関連の設定を示します。使用する SSL 接続の種類に応じて、パラメータを変更してください。

パラメータ	デフォルト値	説明
net.ssl.cipher.suites	SSL_RSA_WITH_3DES_EDE_CBC_SHA	有効にする SSL 暗号群をコンマで区切って表示。 サポートされる SSL 暗号群の一覧は、「84 ページの「SSL 暗号群」」を参照
net.ssl.client.auth	false	SSL サーバーで、接続するクライアントを認証するかどうかを指定する
net.ssl.trust.store.path	*N1SPS4.1–home*/data/trust.store	トラストキーストアのパス。 トラストキーストアには、このノードへの接続を許可されたノードの公開鍵が格納されている
net.ssl.private.store.path	*N1SPS4.1–home*/data/private.store	プライベートキーストアのパス。 プライベートキーストアには、このノードがほかのノードに対して自身を認証するとき使用する公開鍵と非公開鍵のペアが格納されている
net.ssl.key.store.pass		キーストアのパスワード

構成シナリオ (サンプル)

▼ Master Server、Local Director、Remote Agent 間の認証なしで SSL を構成する

手順

1. **Master Server**、**Local Distributor**、**Remote Agent** をインストールし、接続タイプを選択するプロンプトが表示されたら、**SSL** を選択します。 暗号群を選択するプロンプトが表示されたら、認証なしの暗号化を選択します。
2. 各アプリケーションの **config.properties** ファイルに、次のプロパティを追加します。

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

   複数の暗号群または異なった暗号群を有効にすることができます。 複数の暗号群を有効にする場合は、パラメータとして、暗号群をコンマで区切ったリストを指定します。
3. **Web** インタフェースで、新しいホストを作成します。

4. 作成したホストで、接続タイプ **SSL** の **Local Distributor** を追加します。
5. **Local Distributor** との接続をテストします。
6. 新しいホストを作成します。
7. 作成したホストで、接続タイプ **SSL** の **Remote Agent** を追加します。
8. **Remote Agent** との接続をテストします。

▼ SSL サーバー認証を構成する

サーバー認証を要求する暗号群はデフォルトで有効になっています。したがって、暗号群を有効にするため、`config.properties` ファイルに変更を加える必要はありません。

手順

1. **Local Distributor** 用の鍵のペアを生成し、**Local Distributor** のプライベートストアに格納します。

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

2. **Local Distributor** 上のプライベートストア内の自己署名付き証明書をファイルにエクスポートします。

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

3. **Local Distributor** の自己署名付き証明書を **Master Server** にコピーします。
4. 自己署名付き証明書を **Master Server** のトラストストアにインポートします。

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

5. 新しいホストを作成します。
6. 新しいホストで、接続タイプ **SSL** の **Local Distributor** を追加します。
7. **Local Distributor** に対し、**CLI** `net.gencfg` コマンドを使って、手動で `transport.config` ファイルを生成します。
8. `transport.config` ファイルを **Local Distributor** にコピーします。
9. 実行中の **Master Server** や **Local Distributor** がある場合は、停止します。
10. **Master Server** と **Local Distributor** を起動します。
11. **Master Server** と **Local Distributor** のキーストアのパスワードを入力します。
12. **Local Distributor** との接続をテストします。
13. **Remote Agent** 用の鍵のペアを生成し、**Remote Agent** のプライベートストアに格納します。

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

14. **Remote Agent** 上のプライベートストア内の自己署名付き証明書をファイルにエクスポートします。

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

15. **Remote Agent** の自己署名付き証明書を **Local Distributor** にコピーします。
16. 自己署名付き証明書を **Local Distributor** のトラストストアにインポートします。

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

17. 新しいホストを作成します。
18. 新しいホストで、接続タイプ **SSL** の **Remote Agent** を追加します。
19. **CLI** `net.gencfg` コマンドを使って、手動で `transport.config` ファイルを生成します。
20. `transport.config` ファイルを **Remote Agent** にコピーします。
21. 実行中の **Local Distributor** や **Remote Agent** がある場合は、停止します。
22. **Local Distributor** と **Remote Agent** を起動します。
23. **Local Distributor** と **Remote Agent** のキーストアのパスワードを入力します。
24. **Remote Agent** との接続をテストします。

▼ SSL サーバーとクライアントの認証を構成する

手順

1. **Master Server**、**Local Distributor**、**Remote Agent** をインストールし、接続タイプを選択するプロンプトが表示されたら、**SSL** を選択します。 暗号群を選択するプロンプトが表示されたら、認証ありの暗号化を選択します。
2. **Local Distributor** 用の鍵のペアを生成し、**Local Distributor** のプライベートストアに格納します。

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

3. **Master Server** 用の鍵のペアを生成し、**Master Server** のプライベートストアに格納します。

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

4. **Local Distributor** のプライベートストア内の自己署名付き証明書をファイルにエクスポートします。

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

5. **Local Distributor** の自己署名付き証明書を **Master Server** にコピーします。
6. 自己署名付き証明書を **Master Server** のトラストストアにインポートします。

   % server/bin/crkeys -trust -import -file ld.cert -alias *ldhostname.cr.com*
7. **Master Server** のプライベートストア内の自己署名付き証明書をファイルにエクスポートします。

   % server/bin/crkeys -private -export -file ms.cert -alias *mshostname.cr.com*
8. **Master Server** の自己署名付き証明書を **Local Distributor** にコピーします。
9. 自己署名付き証明書を **Local Distributor** のトラストストアにインポートします。

   % ld/bin/crkeys -trust -import -file ms.cert -alias *mshostname.cr.com*
10. 新しいホストを作成します。
11. 新しいホストで、接続タイプ **SSL** の **Local Distributor** を追加します。
12. 実行中の **Master Server** や **Local Distributor** がある場合は、停止します。
13. **Master Server** と **Local Distributor** を起動します。
14. **Master Server** と **Local Distributor** のキーストアのパスワードを入力します。
15. **Local Distributor** との接続をテストします。
16. **Remote Agent** 用の鍵のペアを生成し、**Remote Agent** のプライベートストアに格納します。

   % agent/bin/crkeys -private -generate -alias *rahostname.cr.com* **-validity 365**
17. **Remote Agent** のプライベートストア内の自己署名付き証明書をファイルにエクスポートします。

   % agent/bin/crkeys -private -export -file ra.cert -alias *rahostname.cr.com*
18. **Remote Agent** の自己署名付き証明書を **Local Distributor** にコピーします。
19. 自己署名付き証明書を **Local Distributor** のトラストストアにインポートします。

   % ld/bin/crkeys -trust -import -file ra.cert -alias *rahostname.cr.com*
20. 手順 **4** でエクスポートした **Local Distributor** の自己署名付き証明書を **Remote Agent** マシンにコピーします。
21. 自己署名付き証明書を **Remote Agent** のトラストストアにインポートします。

   % agent/bin/crkeys -trust -import -file ld.cert -alias *ldhostname.cr.com*
22. 新しいホストを作成します。
23. 新しいホストで、接続タイプ **SSL** の **Remote Agent** を追加します。

24. **transport.config** ファイルを **Remote Agent** にコピーします。

25. 実行中の **Local Distributor** や **Remote Agent** がある場合は、停止します。

26. **Local Distributor** と **Remote Agent** を起動します。

27. **Local Distributor** と **Remote Agent** のキーストアのパスワードを入力します。

28. **Remote Agent** との接続をテストします。

▼ CLI クライアントと Master Server 間の SSL 認証を構成する

手順

1. **Master Server** と **CLI** クライアントをインストールし、接続タイプを選択するプロンプトが表示されたら、**SSL** を選択します。 暗号群を選択するプロンプトが表示されたら、認証ありの暗号化を選択します。

2. **Master Server** 用の鍵のペアを生成し、**Master Server** のプライベートストアに格納します。

   % server/bin/crkeys -private -generate -alias *mshostname.cr.com* **-validity 365**

3. **CLI** クライアント用の鍵のペアを生成し、**CLI** クライアントのプライベートストアに格納します。

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

4. **Master Server** のプライベートストア内の自己署名付き証明書をファイルにエクスポートします。

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

5. **Master Server** の自己署名付き証明書を **CLI** クライアントにコピーします。

6. 自己署名付き証明書を **CLI** クライアントのトラストストアにインポートします。

   % cli/bin/crkeys -trust -import -file ms.cert -alias *mshostname.cr.com*

7. **CLI** クライアント上のプライベートストア内の自己署名付き証明書をファイルにエクスポートします。

   % cli/bin/crkeys -private -export -file cli.cert -alias *clihostname.cr.com*

8. **CLI** クライアントの自己署名付き証明書を **Master Server** にコピーします。

9. 自己署名付き証明書を **Master Server** のトラストストアにインポートします。

   % server/bin/crkeys -trust -import -file cli.cert -alias *clihostname.cr.com*

10. **Master Server** が実行中の場合は、停止します。

11. **Master Server** を起動します。

12. **Master Server** のキーストアのパスワードを入力します。

13. **CLI** クライアントで、**`config.properties`** ファイルに次の行を追加します。

```
net.ssl.key.store.pass=trust-store-password
```

14. **CLI** クライアントコマンドを実行して、接続を検証します。

SSL 暗号群

ここでは、サポートされている SSL 暗号群を紹介します。

以下は、サーバー認証を要求する暗号群です。

```
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
```

以下は、サーバー認証を要求しない暗号群です。

```
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
```

以下は、暗号化なしのサーバー認証を要求する暗号群です。

```
SSL_RSA_WITH_NULL_MD5
SSL_RSA_WITH_NULL_SHA
```

第 8 章

Java 仮想マシンのセキュリティポリシーの構成

この章では、N1 Service Provisioning System 4.1 アプリケーションが特定の IP アドレスおよびポート範囲との接続しか確立しないようにするセキュリティポリシーの構成方法を示します。

JVM セキュリティポリシーの構成

Java 仮想マシン (JVM セキュリティポリシーファイルは、各アプリケーションの `lib/security/rox.policy` にあります。 このファイルによって、アプリケーションに割り当てられるアクセス権が指定されます。 ポリシーファイルをインストールした時点では、アプリケーションは、どのホストとの接続も確立できます。 SSH で CLI クライアントを使用する場合は、ポリシーファイルに変更を加えて、接続をローカルホストのみに制限します。

これらのアクセス権は、`lib/security/rox.policy` ファイル内の次の行で指定します。

```
permission java.net.SocketPermission "*", "connect,accept,listen";
```

アプリケーションのネットワークアクセスを制限したい場合は、この行を削除し、より制限の厳しいアクセス権を追加します。

以下は、SocketPermission のホストパラメータです。

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

hostname にはマシンのホスト名、*IPaddress* には IP アドレスを指定します。 *portrange* は次のとおりです。

```
portrange = portnumber | -portnumber | portnumber-[portnumber]
```

セキュリティポリシーファイルの構文の詳細については、`http://java.sun.com/j2se/1.3/docs/guide/security/PolicyFiles.html` の「Policy File Syntax」のリンクをクリックしてください。

▼ Master Server の JVM ポリシーを構成する

手順 **1.** アプリケーションにすべてのホストとの接続を許可する行を削除します。

2. 次の行を追加し、アプリケーションに選択的なアクセス権を付与します。

```
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* は、CLI クライアントが Master Server に接続するとき使用するポート。 1 行目で、Master Server は CLI クライアントにローカル接続または `ssh-proxy` 経由の接続のみを許可する
- *dbport* は、Postgres データベースサーバーのポート番号
- *domain* は、Web インタフェースへの接続を許可されたホストのドメイン。 *httpport* は、Web インタフェースのポート番号
- *ipAddress1:port1* と *ipAddress2:port2* は、Master Server に直接接続する Remote Agent または Local Distributor の IP アドレスおよびポート番号

▼ Remote Agent の JVM ポリシーを構成する

手順 **1.** アプリケーションにすべてのホストとの接続を許可する行を削除します。

2. 次の行を追加し、アプリケーションにアクセス権を付与します。

```
permission java.net.SocketPermission "ipAddress", "accept";
```

ipAddress は、この Remote Agent が接続する Local Distributor または Master Server の IP アドレス

参考 ホストに接続するためのアクセス権の追加

`urltest` など、ネットワークアクセスを必要とする手順を含むプランを実行したい場合は、この Remote Agent から特定のホストへの接続を許可するアクセス権を設定することができます。

▼ Local Distributor の JVM ポリシーを構成する

手順 **1.** アプリケーションにすべてのホストとの接続を許可する行を削除します。

2. 次の行を追加し、アプリケーションに選択的なアクセス権を付与します。

```
permission java.net.SocketPermission "ipAddress", "accept";
permission java.net.SocketPermission "ipAddress1:port1", "connect";
permission java.net.SocketPermission "ipAddress2:port2", "connect"; ...
```

- *ipAddress* は、この Local Distributor の親になっている Local Distributor または Master Server の IP アドレス
- *ipAddress1:port1* と *ipAddress2:port2* は、この Local Distributor の子になっている Remote Agent または Local Distributor の IP アドレスおよびポート番号

Postgres セキュリティ

Postgres データベースがその他のホストからの接続を受け付けない構成になっていることを確認します。 デフォルトの構成では、UNIX ソケットとローカルホストからの接続を受け付けることになっていますが、`server/postgres/data/pg_hba.conf` 構成ファイルを編集して、この設定を変更します。 また、インストール後に、alter user username とパスワード 'password' クエリーを使って、データベースパスワードを変更します。 Postgres 構成ファイル *N1SPS4.1-MasterServer-home*`/config/config.properties` を使ってこれらの変更を加える場合、値を `db.password` に変更します。

第 9 章

N1 Service Provisioning System 4.1 へのアップグレード

この章では、バージョン 4.0 の製品を N1 Service Provisioning System 4.1 へアップグレードする手順を示します。

注 – 4.0 以前のバージョンを使用している場合は、4.0 にアップグレードしてから、N1 Service Provisioning System 4.1 にアップグレードする必要があります。

この章の内容は次のとおりです。

アップグレードの概要

バージョン 4.0 の Master Server と同じシステムにバージョン 4.1 の Master Server をインストールして、Master Server をアップグレードします。 このインストール方法は「サイドバイサイド方式」として知られています。 次に、バージョン 4.0 の Master Server のデータをバージョン 4.1 の Master Server へ移行します。 Master Server のデータの移行処理が完了したら、指示に従って Remote Agent と Local Distributor をアップグレードします。 CLI クライアントは、アップグレードする必要はありません。バージョン 4.1 の CLI クライアントをインストールし、バージョン 4.0 をアンインストールするだけで済みます。

Solaris OS と Red Hat の Master Server のアップグレード

Master Server アプリケーションのアップグレード方法は、通常のソフトウェアのアップグレード方法とは異なっています。 具体的には、新しいバージョンの Master Server を以前のバージョンの Master Server と同じシステム上にインストールし、以前のバージョンの Master Server のデータを新しいバージョンの Master Server へ移行します。

▼ Solaris OS または Red Hat の Master Server のデータを移行する

バージョン 4.0 の Master Server のデータをバージョン 4.1 へ移行すると、バージョン 4.1 の Master Server のデータがすべて削除されます。 新旧の Master Server は、移行スクリプトの働きにより、移行が完了するまで停止状態になります。 移行中、Master Server を使用することはできません。

はじめに

「37 ページの「Solaris OS、Red Hat Linux、IBM AIX システムへ N1 Service Provisioning System 4.1 をインストールする」」の手順に従って、バージョン 4.0 の Master Server がインストールされているシステムに N1 Service Provisioning System 4.1 Master Server をインストールします。 移行を開始する前に、新旧バージョンの Master Server を同一のマシンにインストールスル必要があります。

移行データのバックアップを作成する方法については、「103 ページの「Master Server のバックアップを作成する」」を参照してください。

手順

1. 移行中にデータベースの最適化が行われないことを確認します。
   データの移行中に、データベースの最適化を行う `cron` ジョブがスケジュールされていないかどうかチェックします。
2. **Master Server** ディレクトリを所有するユーザーとしてログインします。
3. 移行スクリプトを開始します。

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

   N1SPS4.1-home には、アプリケーションのインストールディレクトリを指定します。 デフォルトのディレクトリは `/opt/SUNWn1sps/N1_Service_Provisioning_System_4.1` です。
4. 画面の指示に従って、移行を完了します。
   移行が完了すると、次のメッセージが表示されます。

```
Master Server migration completed successfully.
```

注 – Postgres データベース、Web インタフェース、Master Server のリスナーポート番号は、移行対象から除外されます。 N1 Service Provisioning System 4.1 Master Server は、インストール時に指定されたポート番号を使用します。

---

5. 移行中にエラーが発生していないか、ログファイルをチェックします。
   移行スクリプトにより、ログファイルの場所が表示されます。
6. 「**103** ページの「**Master Server** のバックアップを作成する」」の指示に従って、新しい **Master Server** へ移行したデータのバックアップを作成します。
   バージョン 4.0 の Master Server のデータをバージョン 4.1 へ復元することはできません。 バージョン 4.1 の Master Server のデータのバックアップを作成します。これは、必要に応じて使用できる、完全かつ厳密なデータバックアップです。

Windows Master Server のアップグレード

Master Server アプリケーションのアップグレード方法は、通常のソフトウェアのアップグレード方法とは異なっています。 具体的には、新しいバージョンの Master Server を以前のバージョンの Master Server と同じシステム上にインストールし、以前のバージョンの Master Server のデータを新しいバージョンの Master Server へ移行します。

▼ サイドバイサイド方式で Windows Master Server をインストールする

以下の手順に従って、バージョン 4.1 の Windows Master Server をバージョン 4.0 の Master Server と同じマシン上にインストールします。

手順
1. **IPC** デーモンサービスを停止するため、**Windows** 管理ツールの「サービス」アプリケーションを使って、バージョン **4.0** の **Master Server** を停止します。
2. バージョン **4.0** の **Master Service**、特に **IPC** デーモンとサーバーを手動で起動するように設定します。
3. 「**45** ページの「**Windows** システムに **N1 Service Provisioning System 4.1 Master Server** をインストールする」」の指示に従って、バージョン **4.1** の **Master Server** をインストールします。

バージョン 4.0 の Master Server を所有するユーザーおよびグループの権限で、バージョン 4.1 の Master Server をインストールします。

注 – 同じシステム上で同時に複数のバージョンの Master Server を実行することはできません。 IPC デーモンサービスを停止または開始するため、Windows 管理ツールの「サービス」アプリケーションを使って、Master Server を開始したり停止したりできます。

---

4. バージョン **4.0** の **Master Server** のデータをバージョン **4.1** の **Master Server** へ移行します。

   データの移行方法については、「92 ページの「Windows Master Server 上のデータを移行する」」を参照してください。

5. **(**任意**)** バージョン **4.0** の **Master Server** をアンインストールします。

   バージョン 4.0 の Master Server をもう使用しない場合は、「100 ページの「Windows システム上のアプリケーションのアンインストール」」の手順に従ってアンインストールすることができます。

▼ Windows Master Server 上のデータを移行する

バージョン 4.0 の Master Server のデータをバージョン 4.1 へ移行すると、バージョン 4.1 の Master Server のデータがすべて削除されます。 新旧の Master Server は、移行スクリプトの働きにより、移行が完了するまで停止状態になります。 移行中、Master Server を使用することはできません。

はじめに 「91 ページの「サイドバイサイド方式で Windows Master Server をインストールする」」の手順に従って、以前のバージョンの Master Server がインストールされているシステムに N1 Service Provisioning System 4.1 Master Server をインストールします。 移行を開始する前に、新旧バージョンの Master Server を同一のマシンにインストールスル必要があります。

手順 1. コマンドプロンプトウィンドウを開きます。

2. *C:\Program Files\N1 Service Provisioning System\4.1***server\bin\migrate** ディレクトリに移動します。

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

   C:\Program Files\N1 Service Provisioning System\4.1 は、Master Server のインストールディレクトリです。

3. 次のように入力して、移行を開始します。

   ```
   .\cr_migrate.cmd
   ```

4. 画面の指示に従って、移行を完了します。
   移行が完了すると、次のメッセージが表示されます。

```
Master Server migration completed successfully.
```

注 – Postgres データベース、Web インタフェース、Master Server のリスナーポート番号は、移行対象から除外されます。 N1 Service Provisioning System 4.1 Master Server は、インストール時に指定されたポート番号を使用します。

---

5. 移行中にエラーが発生していないか、ログファイルをチェックします。
   移行スクリプトにより、ログファイルの場所が表示されます。

Remote Agent と Local Distributor のアップグレード

▼ Remote Agent と Local Distributor をアップグレードする

Remote Agent と Local Distributor のアップグレードには、Master Server の Web インタフェースを使用します。 アップグレードを完了するためには、「Update Entire N1 SPS network」ボタンを 2 回クリックする必要があります。

はじめに　Remote Agent と Local Distributor のアップグレードを開始する前に、Master Server を移行します。

手順

1. **N1 Service Provisioning System 4.1 Master Server** の **Web** インタフェースにログインします。
2. 「**Hosts**」をクリックします。
3. マスターサーバーをクリックします。
4. 「**Update Entire N1 SPS network...**」ボタンをクリックします。
   ウィンドウが開き、アップグレード対象のホストが一覧表示されます。 アップグレードの進捗状況も表示されます。 処理が完了すると、次のメッセージが表示されます。

```
Host Update not yet complete.
```

5. 「**Close**」ボタンをクリックします。

6. アップグレードの第 **2** 段階を完了するには、「**Update Entire N1 SPS network...**」ボタンを再度クリックします。

   ウィンドウが開き、アップグレード対象のホストが一覧表示されます。 アップグレードの進捗状況も表示されます。 処理が完了すると、各ホストのステータスが「Updated」になります。

7. 「**Close**」ボタンをクリックします。

   アップグレードは完了しました。

8. アップグレードした **Remote Agent** の準備を行います。

   アップグレードした Remote Agent でプランを実行するには、Remote Agent の準備が必要です。 具体的な手順については、『『*N1 Service Provisioning System 4.1* ユーザーガイド』』を参照してください。

Master Server のデータの移行

移行の概要

次の表に、Master Server 上の移行データの種類を示します。

表 **9–1** 移行の概要

Master Server 上のデータ	移行データか**?**	移行方法
PostgresSQL データ	はい	SQL スクリプト
既存のコマンドに変更を加えるための CLI クライアントスクリプト	いいえ	
CLI クライアントから直列化されたオブジェクトの移行	いいえ	
各ノードの `config.properties` ファイルへの変更内容の移行	はい	ファイルに記載されているプロパティの移行は、「95 ページの「プロパティファイルの移行の詳細情報」」に従って行われる
リソースの移行	はい	リソースディレクトリをコピーする

プロパティファイルの移行の詳細情報

centerrun.properties ファイルは config.properties ファイルへ移行されます。 移行中、バージョン 4.0 のファイル内の各プロパティ値がバージョン 4.1 の config.properties ファイル内のプロパティ値と比較されます。 値が同じであれば、そのプロパティは移行対象から除外されます。 値が異なっている場合、バージョン 4.0 の値がバージョン 4.1 の config.properties ファイルにコピーされます。 バージョン 4.0 のファイルには存在するのにバージョン 4.1 のファイルには存在しないプロパティがある場合、バージョン 4.0 の値がバージョン 4.1 のファイルに追加されます。 次のプロパティの値は、バージョン 4.1 の config.properties ファイルへ移行されません。

- 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

バージョン 4.0 のプロパティファイルでこれらのプロパティの値を変更した場合、バージョン 4.1 の config.properties ファイルの値を手動で変更する必要があります。

第 10 章

N1 Service Provisioning System 4.1 のアンインストール

この章では、N1 Service Provisioning System 4.1 のアンインストール方法について説明します。次の節があります。

Solaris OS、Red Hat、IBM AIX システム上のアプリケーションのアンインストール

N1 Service Provisioning System 4.1 のアンインストール方法は、ソフトウェアのインストール方法によって異なります。

- アプリケーションをパッケージとして Solaris OS 上にインストールしている場合、「98 ページの「Solaris OS システム上のパッケージベースのアプリケーションをアンインストールする」」を参照
- アプリケーションをファイルとして Solaris OS 上にインストールしているか、Red Hat または IBM AIX システムを使用している場合、「99 ページの「Solaris OS、Red Hat、IBM AIX システム上のファイルベースのアプリケーションをアンインストールする」」を参照

▼ Solaris OS システム上のパッケージベースのアプリケーションをアンインストールする

パッケージとしてインストールできるのは、Master Server とCLI クライアントだけです。 アンインストールスクリプトで削除できるのは、バージョン 4.1 の Master Server と CLI クライアントだけです。

注 – アンインストールスクリプトは、Master Server または CLI クライアントをパッケージとしてインストールした場合にかぎりインストールされます。 ディレクトリ内にスクリプトが存在しない場合は、「99 ページの「Solaris OS、Red Hat、IBM AIX システム上のファイルベースのアプリケーションをアンインストールする」」の手順に従ってアンインストールを行います。

手順 **1.** アンインストールしたいアプリケーションがあるシステムに移動し、アンインストール対象のアプリケーションが格納されているディレクトリ以外のディレクトリに移動します。

2. アンインストールを開始します。

```
# /N1SPS4.1-home/app_directory/bin/cr_uninstall_app.sh
```

N1SPS4.1-home には、アプリケーションのインストールディレクトリを指定します。 デフォルトのディレクトリは `/opt/SUNWn1sps/N1_Service_Provisioning_System_4.1` です。
app_directory には、次のいずれかの値を指定します。

- server – Master Server をアンインストールする
- cli – CLI クライアントをアンインストールする

app には、次のいずれかの値を指定します。

- ms – Master Server をアンインストールする
- cli – CLI クライアントをアンインストールする

アンインストールが完了すると、次のメッセージが表示されます。

```
Successfully removed SUNWsp app
    Successfully removed SUNWspsc1
    Successfully removed SUNWspsj1
```

app には、Master Server のアンインストール時には「ms」、CLI クライアントのアンインストール時には「cl」が入ります。

注 – このシステムに別のアプリケーションがインストールされている場合、SUNWspsc パッケージと SUNWspsj1 パッケージは削除されません。 たとえば、同一システム上に Master Server と CLI クライアントがインストールされている場合、Master Server だけをアンインストールしても、CLI クライアントをアンインストールするまで、SUNWspsc パッケージと SUNWspsj1 パッケージは削除されません。

▼ Solaris OS、Red Hat、IBM AIX システム上のファイルベースのアプリケーションをアンインストールする

手順

1. アンインストールしたいアプリケーションがあるシステムに移動し、アンインストール対象のアプリケーションが格納されているディレクトリ以外のディレクトリに移動します。
2. アンインストールするアプリケーションを停止します。
3. **Remote Agent** をアンインストールする場合は、**/protect** ディレクトリ内のファイルのアクセス権を変更します。

   ```
   % chmod -R 755 /N1SPS4.1-home/agent/bin/protect
   ```

   N1SPS4.1-home には、Remote Agent のインストールディレクトリを指定します。
4. アンインストールするアプリケーションが格納されているディレクトリを削除します。

   ```
   # rm -r /N1SPS4.1-home/app-directory
   ```

   N1SPS4.1-home には、アプリケーションのインストールディレクトリを指定します。 デフォルトのディレクトリは /opt/SUNWn1sps/ です。 *app-directory* には、次のいずれかの値を指定します。

   - server – Master Server をアンインストールする
   - agent – Remote Agent をアンインストールする
   - cli – CLI クライアントをアンインストールする
   - ld – Local Distributor をアンインストールする
5. マシンからすべてのアプリケーションをアンインストールする場合は、*N1SPS4.1–home* ディレクトリの下にアプリケーションディレクトリがないのを確認して、**common/** ディレクトリを削除します。

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

   アンインストールが完了しました。

Windows システム上のアプリケーションのアンインストール

Windows システム上のアプリケーションをアンインストールする場合は、Windows コントロールパネルの「プログラムの追加と削除」を使用します。 アンインストールを実行するとき、Microsoft 管理コンソールの「サービス」スナップイン (「サービス」コンソール) が開いていないことを確認してください。 開いていると、Master Server、Remote Agent、Local Distributor を正常にアンインストールできないことがあります。

第 11 章

N1 Service Provisioning System 4.1 の管理

この章では、N1 Service Provisioning System 4.1 のバックアップと復元の方法を説明します。 この章の内容は次のとおりです。

N1 Service Provisioning System 4.1 アプリケーションの起動

Solaris OS、Red Hat Linux、IBM AIX システムでのアプリケーションの起動

次の表に、Solaris OS、Red Hat Linux、IBM AIX システム上で N1 Service Provisioning System 4.1 アプリケーションを起動するコマンドを一覧表示します。*N1SPS4.1-home* には、アプリケーションのホームディレクトリを指定します。

表 **11–1** Solaris OS、Red Hat Linux、IBM AIX アプリケーションの起動コマンド

アプリケーション	コマンドパス	起動コマンド
Master Server	*N1SPS4.1-home*/server/bin/	cr_server start

表 **11–1** Solaris OS、Red Hat Linux、IBM AIX アプリケーションの起動コマンド　(続き)

アプリケーション	コマンドパス	起動コマンド
Local Distributor	*N1SPS4.1-home*/ld/bin/	cr_ld start
Remote Agent	*N1SPS4.1-home*/agent/bin/	cr_ra start
CLI クライアント	*N1SPS4.1-home*/cli/bin/	cr_cli start
Jython 版の CLI クライアント	*N1SPS4.1-home*/cli/bin/	cr_clij start

Windows システムでのアプリケーションの起動

Windows システムでは、Master Server、Local Distributor、Remote agent は「サービス」パネル、CLI クライアントは DOS ウィンドウから起動します。

Master Server、Local Distributor、Remote Agent のいずれかを起動する場合は、「スタート」メニューの「すべてのプログラム」をクリックし、「管理ツール」、「サービス」の順にクリックします。 「サービス」パネルで、アプリケーション名を選択し、起動します。

表 **11–2** Windows Master Server、Local Distributor、Remote Agent 用として起動するサービスの名前

アプリケーション	起動するサービスの名前
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

Windows システム上で CLI クライアントを起動する場合、DOS プロンプトに次のいずれかのコマンドを入力します。 *N1SPS4.1-home* にはアプリケーションのホームディレクトリを指定します。

表 **11–3** Windows CLI クライアントの起動コマンド

アプリケーション	コマンドパス	起動コマンド
CLI クライアント	*N1SPS4.1-home*/cli/bin/	cr_cli.cmd start
Jython 版の CLI クライアント	*N1SPS4.1-home*/cli/bin/	cr_clij.cmd start

Master Server のバックアップと復元

ソフトウェアには、Master Server を完全にバックアップし、復元するユーティリティが付属しています。 これらのユーティリティは、*N1SPS4.1-home*/server/bin ディレクトリに格納されています。

バックアップまたは復元対象として、Resource Manager と Postgres データベース、またはいずれか一方を選択できます。 デフォルトでは、Resource Manager ディレクトリと Postgres データベースの内容がバックアップまたは復元対象となります。 いずれかのコンポーネントのバックアップまたは復元を省略したい場合は、適切なコマンド行引数を指定します。

▼ Master Server のバックアップを作成する

はじめに　Master Server のバックアップを作成する前に、Master Server を停止する必要があります。 比較などのタスクと同様に、実行中のプランやプリフライトも停止します。

注意 – バックアップスクリプトの実行時には、出力ディレクトリの指定も忘れないでください。 出力ディレクトリを指定しないと、バックアップファイルは *N1SPS4.1-home*/server/bin ディレクトリに格納されます。 Master Server をアンインストールし、再度インストールする場合、このディレクトリ内のバックアップファイルは削除されます。 したがって、Master Server を復元することができなくなります。

手順

1. **Master Server** を停止します。
2. **Master Server** 上で、スーパーユーザー **(root)** か、アプリケーションを所有するユーザーになります。
3. バックアップスクリプトが格納されているディレクトリに移動します。

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

   N1SPS4.1-home にはアプリケーションのホームディレクトリを指定します。
4. 次のように入力して、バックアップを開始します。

   % ./cr_backup.sh *options*

   cr_backup.sh コマンドには、次のオプションを指定できます。

-b	Master Server のベースディレクトリ
-q	Quiet (対話なし) モード。 情報メッセージは出力されない
-nors	リソースストアのバックアップを省略する
-nodb	Resource Manager だけをバックアップする

`-o` *directory*　バックアップファイルの保存先ディレクトリを指定する。 バックアップスクリプトは、ユーザーが指定したディレクトリに対する書き込み権を持っているかどうかを検証し、書き込み権がない場合はエラーを生成する。

ディレクトリを指定しないと、ファイルは *N1SPS4.1-home*`/server/bin` ディレクトリに保存される

`-z`　UNIX の圧縮方式でバックアップファイルを圧縮する

`-l` *logfile*　デフォルトの `logfile` ファイルではなく、指定の *logfile* ファイルにログを出力する

`-gz`　PATH に `gzip` が指定されている場合、バックアップファイルを gunzip で圧縮する

`-shutdown`　ユーザーに確認しないで Master Server を停止する

`-u`　この情報を出力する

`-h`　この情報を出力する

まだ停止していない Master Server プロセスがある場合、処理を継続すると、検索、プラン、比較など、実行中のすべてのタスクを取り消して Master Server プロセスを停止するという警告メッセージが表示されます。

5. バックアップを継続する場合、**`y`** を入力します。
   バックアップの進捗状況と、バックアップ tar ファイルの場所が表示されます。
   Master Server が再起動します。

▼ Master Server を復元する

はじめに　復元を行うためには、データを含まない Master Server のインストールが必要です。

手順

1. **Master Server** を停止します。
2. **Master Server** 上で、スーパーユーザー **(root)** か、アプリケーションを所有するユーザーになります。
3. バックアップスクリプトが格納されているディレクトリに移動します。

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

   N1SPS4.1-home にはアプリケーションのホームディレクトリを指定します。
4. 次のように入力して、復元を開始します。

   `% ./cr_restore.sh` *options*

   `cr_restore.sh` コマンドには、次のオプションを指定できます。

   `-b`　Master Server のベースディレクトリ。

	-b オプションを指定してバックアップファイルの復元先ディレクトリを指定しないと、ファイルは現在のディレクトリ *N1SPS4.1–home*/server/bin に復元される。 ユーザーがこのディレクトリに対する書き込み権を持っていない場合、エラーが生成される
-q	Quiet (対話なし) モード。 情報メッセージは出力されない
-nors	Resource Store の復元を省略する
-nodb	データベースの復元を省略する
-f *backupfile*	*backupfile* ファイルの内容を復元する
-l *logfile*	デフォルトの logfile ファイルではなく、指定の *logfile* ファイルにログを出力する
-t *temp_directory*	*temp_directory* ディレクトリに一時ファイルを保存する
-overwrite yes	復元時に既存のデータを上書きする
-u	この情報を出力する
-h	この情報を出力する

スクリプトは、バックアップファイルにエラーがないことを検証します。 さらに、まだ停止していない Master Server プロセスがある場合、処理を継続すると、検索、プラン、比較など、実行中のすべてのタスクを取り消して Master Server プロセスを停止するという警告メッセージを表示します。

5. 継続する場合は、**y** を入力します。
   現在データベース内にあるデータをバックアップファイルのデータで上書きするという警告メッセージが表示されます。
6. 継続する場合は、**y** を入力します。
   復元処理が継続されます。 続いて、Master Server が起動します。

Remote Agent のバックアップと復元

Remote Agent を手動でバックアップしたい場合は、エージェントを停止し、*N1SPS4.1-home*/data ディレクトリの内容を安全な場所へコピーします。 Remote Agent を復元したい場合は、エージェントを停止し、保存したディレクトリの内容をコピーします。

N1 Service Provisioning System 4.1 のバージョンとビルド番号の確認

Solaris OS、Red Hat Linux、IBM AIX システムで、インストール済みのアプリケーションのバージョンやビルド番号を確認したい場合は、アプリケーションの起動コマンドに `-version` または `-build` オプションを指定します。

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

- *N1SPS4.1-app* はアプリケーションのホームディレクトリ
- *app* はバージョンまたはビルド番号を確認したいアプリケーション
- *option* は `-version` または `-build`

Windows システム上で、インストール済みのアプリケーションのバージョンやビルド番号を確認したい場合は、DOS プロンプトから `ShowBuild -version` コマンドを実行します。

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

付録 A

インストールおよび構成リファレンス

この付録には、N1 Service Provisioning System 4.1 のインストールの詳細情報を記載します。次の節があります。

Solaris OS、Red Hat Linux、IBM AIX 上の N1 Service Provisioning System 4.1 リファレンスデータ

この節では、Solaris OS、Red Hat Linux、IBM AIX に N1 Service Provisioning System 4.1 をインストールする際の詳細情報を提供します。 次の各小節があります。

Solaris OS、Red Hat Linux、IBM AIX 上の N1 Service Provisioning System 4.1 のディレクトリ構造

N1 Service Provisioning System 4.1 のインストールの際、ソフトウェアのホームディレクトリを選択するプロンプトが表示されます。 デフォルトのホームディレクトリは /opt/SUNWn1sps です。 ホームディレクトリ内には、インストールプログラムにより、次のディレクトリツリーが作成されます。

- N1_Service_Provisioning_System_4.1。Master Server と CLI クライアント用のソフトウェア格納ディレクトリ
- N1_Service_Provisioning_System。Local Distributor と Remote Agent 用のソフトウェア格納ディレクトリ

N1 Service Provisioning System 4.1 ソフトウェアは、インストールスクリプトにより、ソフトウェアのホームディレクトリの下のデフォルトのインストールディレクトリにインストールされます。 以下の表に注記のないすべてのディレクトリのアクセスビットは、755 (rwxr-xr-x) に設定されています。 実行可能ファイルとスクリプトのアクセスビットも 755 です。それ以外のほとんどのファイルのアクセスビットは 644 (rw-r--r--) です。

次の表に、すべての N1 Service Provisioning System 4.1 アプリケーション、Master Server、Local Distributor、Remote Agent、CLI クライアントに共通のインストールディレクトリを示します。

表 **A–1** すべてのアプリケーションに共通のディレクトリ

ディレクトリ	内容
/common	すべてのサブアプリケーションの共通ファイル
/common/jre	プラットフォーム固有の JRE のバンドル版コピー
/common/lib	一部またはすべてのサブアプリケーションに共通のライブラリファイル

次の表に、Master Server 用インストールディレクトリを示します。

表 **A–2** Master Server 用ディレクトリ

ディレクトリ	内容
/server/config	Master Server 構成ファイル
/server/data	Master Server データファイル
/server/bin	Master Server 実行可能ファイル
/server/lib	Master Server 固有のライブラリファイル

表 **A–2** Master Server 用ディレクトリ　(続き)

ディレクトリ	内容
/server/postgres	Postgres のバンドル版コピー
/server/tomcat	Apache Tomcat のバンドル版コピー
/server/webapp	HTML ユーザーインタフェース Web アプリケーション
/server/setup	Master Server の初期化に使用するファイル
/server/config/proxy/config	コマンド行ユーザーインタフェース SSH プロキシプロパティファイル
/server/data/tmp	Master Server 一時ディレクトリ (アクセスビット 777)

次の表に、Local Distributor 用インストールディレクトリを示します。

表 **A–3** Local Distributor 用ディレクトリ

ディレクトリ	内容
/ld/config	Local Distributor 構成ファイル
/ld/bin	Local Distributor 実行可能ファイル
/ld/lib	Local Distributor ライブラリファイル
/ld/data	Local Distributor 固有のデータ
/ld/data/tmp	Local Distributor 一時ディレクトリ (アクセスビット 777)

次の表に、Remote Agent 用インストールディレクトリを示します。

表 **A–4** Remote Agent 用ディレクトリ

ディレクトリ	内容
/agent/config	Remote Agent 構成ファイル
/agent/bin	Remote Agent 実行可能ファイル
/agent/bin/protect	Jexec ディレクトリ (アクセスビット 100、--x------)
/agent/bin/protect/jexec	Jexec。Jexec は、エージェントがアクセスビット 4110 の root 権限を必要とするとき使用される
/agent/lib	Remote Agent ライブラリファイル
/agent/data	Remote Agent 固有のデータ

表 **A–4** Remote Agent 用ディレクトリ　(続き)

ディレクトリ	内容
/agent/work	exeNative の実行用デフォルトディレクトリ
/agent/data/tmp	Remote Agent 一時ディレクトリ (アクセスビット 777)

次の表に、CLI クライアント用インストールディレクトリを示します。

表 **A–5** CLI クライアント用ディレクトリ

ディレクトリ	内容
/cli/config	CLI 構成ファイル
/cli/bin	CLI 実行可能ファイル
/cli/lib	CLI ライブラリファイル
/cli/data	CLI 固有のデータ
/cli/data/tmp	CLI 一時ディレクトリ (アクセスビット 777)

Solaris OS、Red Hat Linux、IBM AIX 上のデータベースの最適化

インストールプログラムは、日次ベースでデータベースの最適化を行うかどうかを確認するプロンプトを表示します。 日次ベースでデータベースの最適化を行う設定を選択した場合、cronjob ファイルに次のコマンドが追加されます。 コマンドは、日次ベースでデータベースの最適化を開始する時点で追加することもできます。

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

N1SPS4.1-home には、Master Server のホームディレクトリを指定します。

Solaris OS、Red Hat Linux、IBM AIX の Remote Agent パラメータファイル (サンプル)

Master Server のインストール時に、Master Server の /server/bin ディレクトリには、各種スクリプトと一緒にサンプルのパラメータファイルがインストールされます。 以下に、サンプルのパラメータファイルの内容を示します。

```
# 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
```

Windows 上の N1 Service Provisioning System 4.1 リファレンスデータ

この節では、Windows システムに N1 Service Provisioning System 4.1 をインストールする際の詳細情報を提供します。次の各小節があります。

Windows 上の N1 Service Provisioning System 4.1 のディレクトリ構造

N1 Service Provisioning System 4.1 のインストールの際、ソフトウェアのホームディレクトリを選択するプロンプトが表示されます。 デフォルトのディレクトリは次のいずれかです。

- `C:\Program Files\N1 Service Provisioning System 4.1`。Master Server と CLI クライアント用のソフトウェア格納ディレクトリ
- `C:\Program Files\N1 Service Provisioning System`。Local Distributor と Remote Agent 用のソフトウェア格納ディレクトリ

N1 Service Provisioning System 4.1 ソフトウェアは、インストールスクリプトにより、ソフトウェアのホームディレクトリの下のデフォルトのインストールディレクトリにインストールされます。 次の表に、すべての N1 Service Provisioning System 4.1 アプリケーション、Master Server、Local Distributor、Remote Agent、CLI クライアントに共通のインストールディレクトリを示します。

表 **A–6** すべてのアプリケーションに共通のディレクトリ

ディレクトリ	内容
`\common`	すべてのサブアプリケーションの共通ファイル
`\common\jre`	Windows 用 JRE のバンドル版コピー
`\common\lib`	一部またはすべてのサブアプリケーションに共通のライブラリファイル

次の表に、Master Server 用インストールディレクトリを示します。

表 **A–7** Master Server 用ディレクトリ

ディレクトリ	内容
`\server\config`	Master Server 構成ファイル
`\server\data`	Master Server データファイル
`\server\bin`	Master Server 実行可能ファイル
`\server\lib`	Master Server 固有のライブラリファイル
`\server\postgres`	Postgres のバンドル版コピー
`\server\cygwin`	Red Hat cygwin のバンドル版サブセット
`\server\tomcat`	Apache Tomcat のバンドル版コピー
`\server\webapp`	HTML ユーザーインタフェース Web アプリケーション
`\server\setup`	Master Server の初期化に使用するファイル

表 **A–7** Master Server 用ディレクトリ　(続き)

ディレクトリ	内容
\server\data\tmp	Master Server 一時ディレクトリ (アクセスビット 777)

次の表に、Local Distributor 用インストールディレクトリを示します。

表 **A–8** Local Distributor 用ディレクトリ

ディレクトリ	内容
\ld\config	Local Distributor 構成ファイル
\ld\bin	Local Distributor 実行可能ファイル
\ld\lib	Local Distributor ライブラリファイル
\ld\data	Local Distributor 固有のデータ
\ld\data\tmp	Local Distributor 一時ディレクトリ

次の表に、Remote Agent 用インストールディレクトリを示します。

表 **A–9** Remote Agent 用ディレクトリ

ディレクトリ	内容
\agent\config	Remote Agent 構成ファイル
\agent\bin	Remote Agent 実行可能ファイル
\agent\lib	Remote Agent ライブラリファイル
\agent\data	Remote Agent 固有のデータ
\agent\work	exeNative の実行用デフォルトディレクトリ
\agent\data\tmp	Remote Agent 一時ディレクトリ

次の表に、CLI クライアント用インストールディレクトリを示します。

表 **A–10** CLI クライアント用ディレクトリ

ディレクトリ	内容
\cli\config	CLI 構成ファイル
\cli\bin	CLI 実行可能ファイル
\cli\lib	CLI ライブラリファイル
\cli\data	CLI 固有のデータ

表 A–10 CLI クライアント用ディレクトリ　(続き)

ディレクトリ	内容
\cli\data\tmp	CLI 一時ディレクトリ (アクセスビット 777)

Cygwin

Solaris OS、Red Hat Linux、IBM AIX システム上のアプリケーションの相互運用性を向上させるため、Windows 版ソフトウェアには Red Hat cygwin UNIX 環境のサブセットが付属しています。 次に示す cygwin の解説は、Cygwin の公式 Web サイト (http://www.cygwin.com) からの引用です。

> Cygwin は、Red Hat が開発した Windows 用 UNIX 環境です。 この環境は、DLL (cygwin1.dll) と、UNIX から移植されたツール群の 2 つの部分で構成されます。 cygwin1.dll は、事実上、UNIX API 機能を提供する UNIX エミュレーションレイヤとして機能します。ツール群は、UNIX/Linux 的な見た目と使い心地を実現します。 Cygwin DLL は、ベータ版、release candidate″、Windows CE を除く、Windows 95 以降のすべての ix86 版 Windows に対応しています。

Windows インストールスクリプトの機能

Windows Master Server インストールスクリプトには、次の機能があります。

- すべてのインストール内容を指定のディレクトリにコピーする
- cygwin の適切なマウントポイントにレジストリエントリを設定する
- cygipc サービスを登録する
- cygipc サービスに依存するサービスとして、postmaster サービスを登録する
- postmaster サービスに依存するサービスとして Master Server サービスを登録する
- 「スタート」メニューのショートカットを作成する
- 通信プロトコルとして SSL を選択した場合、SSL 用の構成ファイルを生成するスクリプトを実行する

Windows Local Distributor インストールスクリプトには、次の機能があります。

- インストール内容を指定のディレクトリにコピーする
- 通信プロトコルとして SSL を選択した場合、SSL 用の構成ファイルを生成するスクリプトを実行する
- Local Distributor サービスを登録する
- 「スタート」メニューのショートカットを作成する
- インストールスクリプトから Local Distributor を起動する設定にしている場合、Local Distributor を起動する

Windows Remote Agent インストールスクリプトには、次の機能があります。

- インストール内容を指定のディレクトリにコピーする
- 通信プロトコルとして SSL を選択した場合、SSL 用の構成ファイルを生成するスクリプトを実行する
- Remote Agent サービスを登録する
- 「スタート」メニューのショートカットを作成する

Windows CLI クライアントのインストールスクリプトには、次の機能があります。

- インストール内容を指定のディレクトリにコピーする
- 通信プロトコルとして SSL を選択した場合、SSL 用の構成ファイルを生成するスクリプトを実行する
- 「スタート」メニューのショートカットを作成する

付録 B

トラブルシューティング

この付録では、N1 Service Provisioning System 4.1 のインストールと構成のトラブルシューティング情報を提供します。

Solaris OS、Red Hat Linux、IBM AIX のインストール時の問題

JRE を IBM AIX にインストールする際の警告

AIX マシンの共通ディレクトリにすでに JRE インスタンスが存在することが検出された場合、次の警告が表示されます。

```
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]
```

JRE を AIX マシンにインストールすると、JRE のネイティブライブラリがメモリーにキャッシュされます。 キャッシュされたライブラリは、ディスクにロックされます。これらのロックされたライブラリの上に新しく JRE をインストールしようとすると、エラーが発生します。

新しいバージョンの JRE をインストールしてはなりません。 JRE のインストールを促すプロンプトが表示されたら、no を選択し、すでにインストールされている JRE のパスを入力します。

Solaris OS システム上でインストールを実行したあと、N1 Service Provisioning System 4.1 CD を取り出せない

ソフトウェア CD から Solaris OS システムに Master Server や Remote Agent をインストールし、アプリケーションの起動を促すプロンプトに対して **yes** と答えた場合、ソフトウェア CD を取り出せなくなります。 次のエラーメッセージが表示されます。

```
Device busy
```

CD を取り出すには、アプリケーションを停止する必要があります。

▼ Solaris OS システム上のアプリケーションを停止する

手順 **1.** アプリケーションの起動スクリプトが格納されているディレクトリに移動します。

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

N1sps4.1–home にはアプリケーションのホームディレクトリ、*app* にはアプリケーションを指定します。 *app* には、Master Server のサーバーと Remote Agent のエージェントを指定します。

2. **stop** オプションを指定してアプリケーションスクリプトを起動します。

```
% cr_app stop
```

app には、停止するアプリケーションを指定します。

3. 次のように入力して、**CD** を取り出します。

```
% eject cdrom
```

実行時の問題

Master Server とデータベースサービスの停止

Bourne シェルで、`cr_server start` コマンドを実行して Master Server プロセスを起動したあと、Master Server を起動したのと同じシェルで実行される後続のコマンドに対して `^C` コマンドが発行された場合、データベースと Master Server のプロセスが停止します。

N1SPS4.1-home`/server/bin/roxdb.out` ファイルに、最新のエントリとして次のメッセージが書き込まれます。

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

Master Server や、その他の N1 Service Provisioning System 4.1 アプリケーションの起動に、Bourne シェルは使用しないでください。

SSH 接続

Master Server が中間 Local Distributor 経由で Local Distributor に接続できない

「Host Details」ページで Master Server の接続先マシンまたはアップストリームのマシンの構成を更新したあと、Master Server がそのマシンに接続できず、TTL 期限切れエラーが表示された場合は、Master Server と接続先マシン間の一部または全部の中間 Local Distributor に対して、`transport.config` ファイルを手動で生成しなければならなくなることがあります。 問題が発生したマシンから Master Server まで移動しながら、問題が発生したマシンのアップストリームの個々の Local Distributor との接続をテストします。 接続に成功した Local Distributor のうち、問題が発生したマシンに最も近いものに対して、`transport.config` ファイルと、ダウンストリームの Local Distributor を再生成します。 CLI クライアントの `net.gencfg` コマンドを使って、`transport.config` ファイルを生成します。

SSH を使ってアプリケーションに接続できない

SSH を使用するように N1 Service Provisioning System 4.1 を構成したあと、マシンに接続できなくなった場合は、次の手順に従って対処してください。

▼ SSH 接続の問題に対処する

はじめに　ssh-agent を使用している場合は、ssh-agent を起動したセッションから、この作業を行います。

手順　**1.** アップストリームのマシンから、ダウンストリームのマシンへの接続をテストします。

- アップストリームのマシンより 1 つダウンストリームのマシンをテストする場合、次のコマンドを使用する

  ```
  % ssh target-IPaddress ls -l
  ```

  target-IPaddress には、テストするダウンストリームのマシンのうち最も遠くにあるものの IP アドレスを指定する

- ssh-agent コマンドを実行するマシンより 2 つ以上ダウンストリームのマシンをテストする場合は、次のコマンドを使用する

  ```
  % 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 には、テストするマシンのアップストリームの Local Distributor マシンの IP アドレスを順に指定します。 たとえば、1 はテストするマシンに最も近いマシン、n は Master Server の直前のマシンです。 *target-IPaddress* には、テストするダウンストリームのマシンのうち最も遠くにあるものの IP アドレスを指定します。

  target-IPadress-parentmachine には、接続をテストするアップストリームのマシンとダウンストリームのマシンの中間にあるマシンの IP アドレスを指定します。

  プロンプトが表示されたら、必要な情報を入力します。 テストを再試行します。

  プロンプトが表示されない場合は、次の手順に進みます。

2. アップストリームのマシンの **logger_config.xml** ファイルの **<root>** セクションの前に、次の行を挿入します。これで、**priority="debug"** でログを記録できるようになります。

```
<category name="SSH.STDERR">
<priority value="debug" />
</category>
<category name="com.raplix.rolloutexpress.net.transport.SshClientConnectionHandler">
<priority value="debug" />
</category>
```

アップストリームのマシンがログファイルの更新を読み取り終わるまで待ちます。

3. 手順 **1** のコマンドを使って、再度接続をテストします。

コマンド行と SSH.STDERR ログファイルに出力された内容を確認します。 ログファイルに出力された問題を修正し、再度テストを行います。

ダウンストリームのアプリケーションの起動に使用した SSH コマンドに対してアップストリームのマシンに出力されたアプリケーションログと、SSH コマンドの `stderr` 出力を確認します。 ログメッセージを元に問題を修正し、再度テストを行います。

ログファイルに問題がない場合、アップストリームのマシンはダウンストリームのマシンに正常に接続しています。問題は、アプリケーションが正常に起動していない点にあります。 次の手順に進みます。

4. **`ROX`** ログファイルで、ダウンストリームのマシン上でアプリケーションを起動する際にエラーが発生していないか確認します。
   - Red Hat Linux および IBM AIX マシンでは `/tmp/ROX`*appnumbers*`.log` ファイルを確認する
   - Solaris OS マシンでは、`/var/tmp/ROX`*appnumbers*`.log` ファイルを確認する

   app は、テストするダウンロードストリームのマシン上のアプリケーションです。Remote Agent には Agent、Local Distributor には Dist、CLi クライアントには Proxy を使用します。 *numbers* はランダムに生成された数値であり、ファイル名の一部になります。
5. ログファイル内で見つかったエラーを修正します。