Sun N1 Service Provisioning System 5.1 インストールガイド

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

Part No: 819–4218–10
2005 年 9 月

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

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

本製品に含まれる HG-MinchoL、HG-MinchoL-Sun、HG-PMinchoL-Sun、HG-GothicB、HG-GothicB-Sun、および HG-PGothicB-Sun は、株式会社リコーがリョービイマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。 HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェースマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、docs.sun.com、AnswerBook、AnswerBook2 は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標、登録商標もしくは、サービスマークです。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャーに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。©Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. ©Copyright OMRON SOFTWARE Co.,Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

「ATOK Server/ATOK12」に含まれる郵便番号辞書 (7 桁/5 桁) は日本郵政公社が公開したデータを元に制作された物です (一部データの加工を行なっています)。

「ATOK Server/ATOK12」に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド』に添付のものを使用しています。

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザーおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザーインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。 米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは、OPEN LOOK のグラフィカル・ユーザーインタフェースを実装するか、またはその他の方法で米国 Sun Microsystems 社との書面によるライセンス契約を遵守する、米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われないものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: Sun N1 Service Provisioning System 5.1 Installation Guide

Part No: 819-1655–10

Revision A

Adobe PostScript

051017@13215

目次

はじめに　**13**

1　N1 Service Provisioning System 5.1 の概要　**17**
N1 Service Provisioning System 5.1 のインストール – 手順の概要　17
N1 Service Provisioning System 5.1 アプリケーションの概要　19
マスターサーバー　19
ローカルディストリビュータ　19
リモートエージェント　20
CLI Client　20
ネットワークプロトコル　21
raw (TCP/IP)　21
Secure Shell　22
Secure Sockets Layer　22
プラグインの導入　22
プラグインの入手　23

2　N1 Service Provisioning System 5.1 のシステム要件　**25**
オペレーティングシステム要件　25
オペレーティングシステムの構成　26
必要なオペレーティングシステムパッチ　28
ハードウェア要件　31
マスターサーバーのハードウェア要件　31
ローカルディストリビュータ、リモートエージェント、および CLI Client のハードウェア要件　31
一般システム要件　32
サポートされる Web ブラウザ　32

SSH の要件　33
Jython の要件　33
ロケールの要件　33

3　インストールの前に収集すべき情報　**35**
一般的な構成に関する決定事項　35
Java Runtime Environment　35
アプリケーションのユーザー所有権　36
ホスト名と IP アドレス　37
Jython　38
セキュリティー構成に関する決定事項　38
ネットワークプロトコル – raw、SSH、SSL　38
HTTP または HTTPS　39
全アプリケーションのワークシート　40
マスターサーバーのワークシート　41
ローカルディストリビュータのワークシート　42
リモートエージェントのワークシート　42
CLI Client のワークシート　43

4　**Linux** および **UNIX** システムへの **N1 Service Provisioning System 5.1** のインストール　**45**
N1 Service Provisioning System 5.1 のインストール　45
▼ Linux および UNIX システムに N1 Service Provisioning System 5.1 をインストールする　45
Linux および UNIX へのリモートエージェントへの非対話方式によるインストール　48
▼ Linux および UNIX にリモートエージェントを非対話方式でインストールする　48
Linux および UNIX システムへのリモートエージェントのリモートインストール　51
▼ Linux および UNIX システムにリモートエージェントをリモートインストールする　51
Linux および UNIX システムでのアプリケーションの起動　54

5　**Windows** システムへの **N1 Service Provisioning System 5.1** のインストール　**57**
マスターサーバーのインストール　57
▼ Windows に N1 Service Provisioning System 5.1 マスターサーバーをインストールする　57

▼ データベースを最適化するために定期タスクを作成する　59
リモートエージェント、ローカルディストリビュータ、CLI Client のインストール　60
▼ Windows にリモートエージェント、ローカルディストリビュータ、CLI Client をインストールする　60
Windows へのリモートエージェントの非対話方式インストール　62
▼ Windows にリモートエージェントを非対話方式でインストールする　62
Windows へのリモートエージェントのリモートインストール　63
▼ Windows にリモートエージェントをリモートインストールする　64
リモートエージェントの変数値　66
Windows システムでのアプリケーションの起動　67

6　HTTPS を使用するための N1 Service Provisioning System 5.1 の構成　69
HTTPS 接続のキーストアファイルとキーストアパスワードの作成　70
▼ SSL 証明書の生成方法　70
▼ SSL 証明書の署名を取得する　71
インストール後の HTTPS の構成　72
▼ キーストアファイルをコピーする　72
▼ 符号化されたキーストアパスワードを作成および構成する　73
インストール中に HTTP を選択したあとで行う HTTPS の構成　74
▼ マスターサーバーブラウザインタフェースからマスターサーバーの Web インタフェースまでの HTTPS 接続を使用可能にする　74
▼ SSL を使用してマスターサーバーブラウザインタフェースに接続するようにユーザーに要求する　75
HTTP への復帰　76
▼ HTTP に戻る　76

7　Secure Shell を使用するための N1 Service Provisioning System 5.1 の構成　77
SSH の概要と使用条件　78
`ssh-agent` または空のパスワード鍵　78
SSH の要件　79
その他の SSH セキュリティー　80
SSH の構成 – 作業概要　81
鍵の準備　82
▼ 鍵ペアを作成する　82
▼ `ssh-agent` の鍵を設定する　83
▼ 空のパスワードファイルに鍵を設定する (1 組の鍵ペアを使用する場合)　83
▼ 空のパスワードファイルに鍵を設定する (複数の鍵ペアを使用する場合)　84

マスターサーバーでの接続の設定とテスト　85
▼ マスターサーバーで ssh-agent を起動する　85
▼ マスターサーバーで接続の設定とテストを行う　86
アプリケーションに合わせて SSH を構成する　87
▼ ローカルディストリビュータとリモートエージェントで使用するように SSH を構成する　88
▼ CLI Client が ssh-agent を使用して SSH 接続を行うように構成する　89
▼ CLI Client が空のパスワードで SSH 接続を行うように構成する　90
SSH の上級パラメータとコマンドのリファレンス　92
上級パラメータのリファレンス　92
OpenSSH 2.0 コマンドリファレンス　93

8　SSL を使用するための N1 Service Provisioning System 5.1 の構成　95
N1 Service Provisioning System 5.1 での SSL サポートの概要　95
暗号群: 暗号化と認証の概要　95
認証キーストア　96
SSL でのパスワードの使用　98
N1 Service Provisioning System 5.1 における SSL の制限　98
SSL の構成 – 作業概要　99
▼ キーストアを作成する　100
▼ config.properties ファイルを編集して SSL を構成する　102
構成例　104
SSL 暗号群　109
暗号群　109
IBM AIX の暗号群　109

9　Java 仮想マシンのセキュリティーポリシーの構成　111
JVM セキュリティーポリシーの構成　111
▼ マスターサーバーの JVM ポリシーを構成する　112
▼ リモートエージェントの JVM ポリシーを構成する　112
▼ ローカルディストリビュータの JVM ポリシーを構成する　113
Postgres セキュリティー　113

10　N1 Service Provisioning System 5.1 へのアップグレード　115
アップグレードの概要　115
アップグレードの要件　115
アップグレード – 作業概要　116

マスターサーバーのアップグレード　117
▼ マスターサーバーのデータを移行する　117
マスターサーバーのデータ移行の詳細　119
リモートエージェントとローカルディストリビュータのアップグレード　121
▼ リモートエージェントとローカルディストリビュータをアップグレードする　121

11　N1 Service Provisioning System 5.1 のアンインストール　123
Linux および UNIX システムでのアプリケーションのアンインストール　123
▼ Solaris OS 上のマスターサーバーまたは CLI Client をアンインストールする　123
▼ Linux および UNIX 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　124
▼ データベースの自動最適化を無効にする　125
Windows システム上のアプリケーションのアンインストール　126

A　インストールと構成のリファレンス　127
Linux および UNIX システム上の N1 Service Provisioning System 5.1 用の参照データ　127
Linux および UNIX システム上の N1 Service Provisioning System 5.1 のディレクトリ構造　128
Linux および UNIX システムでのデータベースの最適化　130
Linux および UNIX システム向けのリモートエージェントパラメータサンプル　131
Windows 上の N1 Service Provisioning System 5.1 用の参照データ　133
Windows 上の N1 Service Provisioning System 5.1 のディレクトリ構造　134
Cygwin　136
Windows インストールスクリプトによって行われる処理　136

B　トラブルシューティング　139
Linux および UNIX システムのインストール時の問題　139
IBM AIX にインストールする際のパッケージ破損エラー (6279820)　139
JRE を IBM AIX にインストールする際の警告　140
Microsoft Windows にインストールする際の問題　140
Windows にインストールする際のエラー　140
SSH 接続　141
マスターサーバーが中間のローカルディストリビュータを経由してローカルディストリビュータに接続できない　141

SSH を使ってアプリケーションに接続できない　141
▼ SSH 接続の問題に対処する　141

用語集　**145**

索引　**153**

表目次

表 2–1 N1 Service Provisioning System 5.1 でサポートされているオペレーティングシステム 25
表 2–2 Solaris /etc/system 設定 27
表 2–3 Linux マスターサーバーシステムの設定 27
表 2–4 サポートされるオペレーティングシステムに必要なパッチ 28
表 2–5 マスターサーバーのハードウェア要件 31
表 2–6 ローカルディストリビュータ、リモートエージェント、および CLI Client の ハードウェア要件 31
表 2–7 ブラウザインタフェースの Web ブラウザ要件 32
表 3–1 全アプリケーションで要求される情報 40
表 3–2 マスターサーバーで要求される情報 41
表 3–3 ローカルディストリビュータで要求される情報 42
表 3–4 リモートエージェントで要求される情報 43
表 3–5 CLI Client で要求される情報 43
表 4–1 Linux および UNIX システムアプリケーションの起動コマンド 55
表 5–1 リモートエージェントの変数値 66
表 5–2 Windows マスターサーバー、ローカルディストリビュータ、およびリモートエージェント のために起動するサービスの名前 68
表 5–3 Windows CLI Client の起動コマンド 68
表 7–1 OpenSSH 2.0 コマンド 93
表 10–1 移行の概要 119
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 128
表 A–2 マスターサーバー用にインストールされるディレクトリ 128
表 A–3 ローカルディストリビュータ用にインストールされるディレクトリ 129
表 A–4 リモートエージェント用にインストールされるディレクトリ 129
表 A–5 CLI Client 用にインストールされるディレクトリ 130
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 134

表 A–7 マスターサーバー用にインストールされるディレクトリ 134
表 A–8 ローカルディストリビュータ用にインストールされるディレクトリ 135
表 A–9 リモートエージェント用にインストールされるディレクトリ 135
表 A–10 CLI Client 用にインストールされるディレクトリ 136

例目次

例 5–1 Windows へのリモートエージェントの非対話方式インストール 63
例 5–2 Windows へのリモートエージェントのリモートインストール 66
例 7–1 cprefix の例 92
例 7–2 sshpath の例 93
例 7–3 sshargs の例 93
例 8–1 `crkeys` のコマンド構文 101
例 8–2 マスターサーバー、ローカルディストリビュータ、リモートエージェント間で認証を使用せずに SSL を構成する方法 104
例 8–3 SSL サーバー認証を構成する方法 105
例 8–4 SSL サーバーおよびクライアント認証を構成する方法 106
例 8–5 CLI Client とマスターサーバー間の SSL 認証を構成する方法 108

はじめに

『Sun N1 Service Provisioning System 5.1 インストールガイド』では、Sun N1™ Service Provisioning System 5.1 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。

注 – このマニュアルでは、“x86” という用語は Intel 32 ビットファミリのマイクロプロセッサと、これと互換性のある AMD 製の 64 ビットおよび 32 ビットマイクロプロセッサを意味します。

注 – Solaris、IBM AIX、および HP-UX オペレーティングシステムは、UNIX™ プラットフォームをベースにしています。これらのオペレーティングシステムは、「UNIX システム」と総称されます。Red Hat Linux および SUSE Linux は、「Linux システム」と総称されます。

対象読者

このマニュアルは、N1 Service Provisioning System 5.1 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。

このマニュアルの構成

『N1 Service Provisioning System 5.1 インストールガイド』は、次の章から構成されます。

- 第 1 章では、この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。また、このソフトウェアとサポートされているネットワークプロトコルの概要も示します。
- 第 2 章では、このソフトウェアをインストールして使用する上でのシステム要件について説明します。
- 第 3 章には、このソフトウェアのインストールに必要な情報を収集するのに便利なワークシートを挙げます。
- 第 4 章では、Linux および UNIX サーバーにこのソフトウェアをインストールする手順について説明します。
- 第 5 章では、Windows にこのソフトウェアをインストールする手順について説明します。
- 第 6 章では、HTTPS を使用してマスターサーバーの Web インタフェースに接続するために ブラウザインタフェース を構成する手順について説明します。
- 第 7 章では、SSH を使用して通信を行うようにこのソフトウェアを構成する作業について説明します。
- 第 8 章では、SSL を使用して通信を行うようにこのソフトウェアを構成する作業について説明します。
- 第 9 章では、JVM™[1] セキュリティーポリシーの構成方法について説明します。
- 第 10 章では、このソフトウェアをアップグレードする手順について説明します。
- 第 11 章では、このソフトウェアをアンインストールする手順について説明します。
- 付録 A では、このソフトウェアのインストールと構成に関連する参考資料を挙げます。
- 付録 B では、インストールと構成に関連した問題の障害追跡の手順について説明します。

関連マニュアル

N1 Service Provisioning System 5.1 をインストールして使用する場合は、必要に応じて次のマニュアルを参照してください。

- 『Sun N1 Service Provisioning System 5.1 リリースノート』

1 「Java 仮想マシン」と「JVM」は、Java™ プラットフォームの仮想マシンを意味します。

- 『Sun N1 Service Provisioning System 5.1 システム管理者ガイド』
- 『Sun N1 Service Provisioning System 5.1 コマンド行インタフェース(CLI) リファレンスマニュアル』

マニュアル、サポート、およびトレーニング

Sun のサービス	**URL**	内容
マニュアル	`http://jp.sun.com/documentation/`	PDF 文書および HTML 文書をダウンロードできます。
サポートおよびトレーニング	`http://jp.sun.com/supportraining/`	技術サポート、パッチのダウンロード、および Sun のトレーニングコース情報を提供します。

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。

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。

表 **P–1** 表記上の規則　(続き)

字体または記号	意味	例
『 』	参照する書名を示します。	『コードマネージャー・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。

第 1 章

N1 Service Provisioning System 5.1 の概要

この章では、N1 Service Provisioning System 5.1 のインストールと構成に必要な作業の概要について説明します。また、N1 Service Provisioning System 5.1 に含まれるアプリケーションの概要と、セキュリティー強化に使用できるネットワークプロトコルの種類についても説明します。

この章の内容は、次のとおりです。

- 17 ページの「N1 Service Provisioning System 5.1 のインストール – 手順の概要」
- 19 ページの「N1 Service Provisioning System 5.1 アプリケーションの概要」
- 21 ページの「ネットワークプロトコル」
- 22 ページの「プラグインの導入」

N1 Service Provisioning System 5.1 のインストール – 手順の概要

次に、N1 Service Provisioning System 5.1 を確実にインストールし構成するために必要な作業の概要を示します。

1. 使用しているサーバーがインストールの最小要件を満たしているか確認します。

   第 2 章を参照してください。

2. 構成についての必要事項を決定して、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 章を参照してください。

3. (省略可能) N1 Service Provisioning System 5.1 で使用する特殊なオペレーティングシステムグループとユーザーアカウントを作成できます。

   新しいユーザーと新しいグループを作成した場合は、その新しいユーザーをそのグループに含めてください。ユーザーアカウントの作成の詳細は、使用しているオペレーティングシステムのマニュアルを参照してください。

4. (省略可能) CLI Client マシンに Jython をインストールします。

   Jython は、CLI Client を実行する任意のマシンにインストールできます。Jython は CLI Client の実行に必須ではありません。Jython は、`http://www.jython.org` から入手できます。

   CLI CLient を Jython と一緒に使用する方法については、『Sun N1 Service Provisioning System 5.1 コマンド行インタフェース(CLI) リファレンスマニュアル』の第 1 章「コマンド行インタフェースの使用」を参照してください。

5. 製品メディアに入っている該当するインストールスクリプトを使用し、N1 Service Provisioning System 5.1 の各アプリケーションを個々にインストールします。

   インストールの手順については、第 4 章または第 5 章を参照してください。

6. (省略可能) インターネット上でマスターサーバーにアクセスする予定がある場合は、そのサーバーとの通信に SSH を使用するように N1 Service Provisioning System 5.1 を構成することによってマスターサーバーのセキュリティーを高めることができます。

   第 7 章を参照してください。

7. (省略可能) アプリケーション間の通信に最大のセキュリティーを提供する場合は、通信に SSL を使用するようにアプリケーションを構成してください。

   第 8 章を参照してください。

8. (省略可能) アプリケーション間の通信にセキュリティーを提供する手段として SSL を使用しない場合は、アプリケーションが `localhost` からの接続だけを認めるように JVM セキュリティーポリシーを構成できます。この設定では、最小レベルのセキュリティーしか提供されません。

   第 9 章を参照してください。

9. (省略可能) アプリケーションを起動します。

   インストールが正常に完了すると、アプリケーションを起動するようにメッセージが表示されます。この時点でアプリケーションを起動しない場合は、54 ページの「Linux および UNIX システムでのアプリケーションの起動」または67 ページの「Windows システムでのアプリケーションの起動」に示されている説明にしたがってアプリケーションを起動してください。

10. 初期設定を行います。

    初期設定の手順については、『Sun N1 Service Provisioning System 5.1 システム管理者ガイド』の「Sun N1 Service Provisioning System の構成 – 手順の概要」を参照してください。

N1 Service Provisioning System 5.1 アプリケーションの概要

N1 Service Provisioning System 5.1 は、分散型のソフトウェアプラットフォームです。このプロビジョニングシステムには、次に示す特殊用途のアプリケーションが含まれます。ユーザーは、これらのアプリケーションをネットワーク内のサーバーにインストールして使用します。これらのアプリケーションは、ユーザーがネットワーク内のサーバーにソフトウェアを配備できるよう連係して動作します。

- 19 ページの「マスターサーバー」 – コンポーネントとプランを保持するセントラルサーバーであり、アプリケーション配備を管理するインタフェースを提供します。
- 19 ページの「ローカルディストリビュータ」 – マスターサーバーのプロキシとして機能するオプションサーバーであり、データセンター間のネットワーク通信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します。
- 20 ページの「リモートエージェント」 – ホスト上で処理を行う管理アプリケーション。N1 Service Provisioning System 5.1 を使用して制御する各サーバーに、リモートエージェントアプリケーションが存在しなければなりません。
- 20 ページの「CLI Client」 – マスターサーバー上で実行されるコマンドを受け付けるオプションのアプリケーション。

マスターサーバー

マスターサーバーは、Linux、UNIX、および Windows ベースのサーバーで稼働します。マスターサーバーは、次のような処理を行うセントラルサーバーです。

- この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
- リポジトリ内にコンポーネントとプランを格納する
- リポジトリ内に格納されているオブジェクトのバージョン制御を行う
- プロビジョニングシステムユーザーの認証を行い、承認されたユーザーだけが一定の処理を行えるようにする
- 依存性の追跡および配備などのタスクを実行する専用エンジンを含める
- ユーザーにブラウザインタフェースとコマンド行インタフェースの両方を提供する

ローカルディストリビュータ

ローカルディストリビュータは、リモートエージェントの配布と管理を最適化するプロキシです。データセンターでは、ローカルディストリビュータを使用して次の処理を行えます。

- 配備時のネットワークトラフィックを最小限に抑える。マスターサーバーは、コンポーネントのコピー一式をローカルディストリビュータに送信します。ローカルディストリビュータは、ほかのサーバーでのインストールのためにこのコンポーネントを複製します。
- ファイアウォールの再構成を最小限に抑える。ファイアウォールがマスターサーバーと複数のサーバー間に位置付けられている場合、管理者がファイアウォールをオープンできるのは (配備に関連するすべてのサーバーに対してではなく) ローカルディストリビュータ に対してだけです。
- 大規模配備の際にマスターサーバーにかかる負荷を最小限に抑える

リモートエージェント

リモートエージェントは、N1 Service Provisioning System 5.1 によって管理されているすべてのサーバーで動作するアプリケーションです。リモートエージェントは、マスターサーバーによって要求されるタスクを実行します。リモートエージェントは次の作業に利用できます。

- サーバーのハードウェア構成とソフトウェア構成をマスターサーバーに報告する
- サービスの開始と停止を行う
- ディレクトリのコンテンツとプロパティを管理する
- ソフトウェアのインストールとアンインストールを行う
- コンポーネントとプランで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

CLI Client

CLI (Command Line Interface: コマンド行インターフェース) Client は、ローカルサーバーとリモートサーバーからのコマンド実行を可能にする、マスターサーバーへの通信パスを提供します。CLI Client は、次の環境でコマンドの実行を許可します。

- Windows コマンド行
- UNIX シェル (bash など)

これらのコマンドを実行する場合、CLI Client は TCP/IP を使用するか、あるいは SSL または SSH のセキュリティー保護された接続を介してマスターサーバーへの接続を確立します。

CLI Client は、次の 2 つのモードで動作します。

- 単一コマンドモード: コマンドを 1 つずつ実行します。
- 対話モード: コマンド入力のプロンプトが表示され、コマンドの履歴が保持されます。Jython によるスクリプト記述も行えます。

対話モードで処理する場合、CLI Client は Jython プログラミング言語を使用します。Jython は、オブジェクト指向の高度な動的言語 Python の Java 実装です。

注 – 対話モードで CLI Client を実行する予定のあるすべてのサーバーに Jython をインストールしてください。Jython の詳細の確認とダウンロードは、`http://www.jython.org` で行えます。

ネットワークプロトコル

N1 Service Provisioning System 5.1 は、ソフトウェアアプリケーション間の通信に使用されるさまざまなネットワークプロトコルをサポートします。ユーザーは、次に示すネットワーク通信に適合するプロトコルを選択できます。

- マスターサーバーとローカルディストリビュータまたはリモートエージェント間の通信
- 特定のローカルディストリビュータとリモートエージェント間の通信
- マスターサーバーと CLI Client 間の通信

N1 Service Provisioning System 5.1 は、次のプロトコルをサポートします。

- raw (TCP/IP)
- Secure Shell
- Secure Sockets Layer

ネットワークセキュリティーは、特定のネットワークトポロジのニーズに合わせて変更できます。たとえば、各データセンター間にはセキュリティー保護された通信を使用しているが、リモートデータセンターとのネットワーク接続は公衆網であるインターネットを使用しているとします。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たローカルディストリビュータと通信する場合は、SSL を使用するようにマスターサーバーを構成できます。この結果、インターネットを介したリモートデータセンターとの通信は、安全なものとなります。ローカルネットワーク上でセキュリティー保護された通信が行われるため、ローカルディストリビュータは raw TCP/IP を使用してリモートエージェントに接続できます。各種プロトコルの内容と設定については、第 7 章および第 8 章を参照してください。

raw (TCP/IP)

raw (TCP/IP) は、暗号化や認証が別に行われることのない標準の TCP/IP です。raw 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で保護されている場合は、N1 Service Provisioning System 5.1 アプリケーション間の通信に raw を使用すると便利です。

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 / プロトコルです。SSH は、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 公開鍵暗号を使用して接続と認証を管理します。SSH は、シェルベースの通信方式 (Telnet など) よりも安全です。

N1 Service Provisioning System 5.1 アプリケーションは、SSH を使用して通信するように構成できます。N1 Service Provisioning System 5.1 は、OpenBSD Project によって開発された SSH のフリーバージョン、OpenSSH をサポートします。OpenSSH の詳細は、`http://www.openssh.com` を参照してください。このソフトウェアは、ほかの SSH バージョンをサポートするようにも構成できます。

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ためのプロトコルです。SSL は、TCP/IP ソケット技術を使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 が開発した公開 / 非公開鍵ペア暗号化システムで保護されます。SSL は、ほとんどの Web サーバー製品でサポートされるほか、Netscape Navigator™ ブラウザと Microsoft Web ブラウザでもサポートされます。

傍受や改ざんからソフトウェアメッセージを保護するため、SSL を使用してネットワーク通信を行うように N1 Service Provisioning System 5.1 アプリケーションを構成できます。また、通信前に SSL を使用して相互に認証を行うようにアプリケーションを構成すれば、さらにネットワークセキュリティーが高まります。

プラグインの導入

一般的な利用においては、「プラグインアプリケーション」は簡単にインストールできて Web ブラウザの一部として使用できるプログラムを意味します。プラグインアプリケーションはブラウザによって自動的に認識され、表示されるメインの HTML ファイルにその機能が統合されます。Web ブラウザプラグインアプリケーションは、一般にサウンドや動画ビデオの再生などを行います。

N1 Service Provisioning System 環境では、プラグインの概念が一般的な利用の場合とわずかに異なります。N1 Service Provisioning System 製品のプラグインは、特定のプラットフォーム、アプリケーション、または環境を対象とした製品のプロビジョニング機能を拡張するパッケージソリューションです。たとえば、Oracle 8i のような特定のアプリケーション用にプラグインソリューションを作成することも、Solaris Zone のような OS 機能用にプラグインソリューションを作成することもできます。

プラグインには、新しいカスタムアプリケーションのサポートに必要な関連データがすべて含まれます。プラグインのコンテンツについては、プラグイン記述子ファイルで説明されます。このファイルは、プラグインパッケージ構造内の標準の場所に置かれます。

プラグインの入手

N1 Service Provisioning System で使用できるように、いくつかのプラグインが用意されています。プラグインは、Sun N1 Service Provisioning System 5.1: Supplement CD および Sun Download Center からダウンロードするイメージから使用可能です。

これらのプラグインは、Java アーカイブファイル (.jar ファイル) にパッケージ化されています。特定のプラグインを N1 Service Provisioning System 製品に認識させるには、そのプラグインをインポートする必要があります。プラグインのインポート方法については、インポートするプラグインに関連するユーザーガイドを参照してください。このガイドは、`http://docs.sun.com/db/coll/1329.1` の Plug-In User's Guide ドキュメントコレクションに含まれています。

第 2 章

N1 Service Provisioning System 5.1 のシステム要件

この章では、N1 Service Provisioning System 5.1 をインストールして使用する上でのシステム要件を示します。この章の内容は、次のとおりです。

- 25 ページの「オペレーティングシステム要件」
- 31 ページの「 ハードウェア要件」
- 32 ページの「一般システム要件」

オペレーティングシステム要件

N1 Service Provisioning System 5.1 のマスターサーバー、リモートエージェント、ローカルディストリビュータ、および CLI Client は、次のオペレーティングシステムを実行しているサーバーにインストールできます。

表 **2–1** N1 Service Provisioning System 5.1 でサポートされているオペレーティングシステム

N1 Service Provisioning System のアプリケーション	サポートされているオペレーティングシステム
マスターサーバー	■ SPARC ® ベースサーバーで稼働している Solaris 8 OS ■ SPARC または x86 ベースサーバーで稼働している Solaris 9 および Solaris 10 OS ■ Red Hat Linux Advanced Server 2.1 と Red Hat Linux Advanced Server 3.0 ■ Microsoft Windows 2000 Server と Microsoft Windows 2000 Advanced Server

表 **2–1** N1 Service Provisioning System 5.1 でサポートされているオペレーティングシステム　(続き)

N1 Service Provisioning System のアプリケーション	サポートされているオペレーティングシステム
ローカルディストリビュータ、リモートエージェント、および CLI Client	■ SPARC ベースサーバーで稼働している Solaris 7 および Solaris 8 OS ■ SPARC または x86 ベースサーバーで稼働している Solaris 9 および Solaris 10 OS ■ Red Hat Linux Advanced Server 2.1 と Red Hat Linux Advanced Server 3.0 ■ IBM AIX 5.1、5.2、および 5.3 ■ SUSE Linux Enterprise Server 8 および SUSE Linux Enterprise Server 9 ■ PA-RISC ベースサーバー上の HP-UX 11i V1 ■ Microsoft Windows 2000 Server と Microsoft Windows 2000 Advanced Server ■ Microsoft Windows Server 2003: Standard、Enterprise、または Web Edition ■ Microsoft Windows Standard 2003 x64 Edition または Microsoft Windows Enterprise x64 Edition

オペレーティングシステムの構成

プロビジョニングシステムをインストールおよび実行するには、これ以降の節の説明どおりにシステムを構成する必要があります。

Solaris システムの構成要件

マスターサーバーを実行している Solaris システムでは、次の `/etc/system` 設定が必要です。

注 – Solaris 9 または Solaris 10 OS を使用している場合は、`shmsys:shminfo_shmmin` と `shmsys:shminfo_shmseg` の値を変更できません。Solaris 10 OS を使用している場合は、`shmsys:shminfo_shmmax`、`shmsys:shminfo_shmmni` 、`semsys:seminfo_semmns`、および `semsys:seminfo_semvmx` の値を変更できません。

これらの設定のデフォルト値のまま使用できます。

表 **2–2** Solaris `/etc/system` 設定

変数	最小値
`shmsys:shminfo_shmmax`	0x20000000[1]
`shmsys:shminfo_shmmin`	1
`shmsys:shminfo_shmmni`	2
`shmsys:shminfo_shmseg`	1
`semsys:seminfo_semmni`	32
`semsys:seminfo_semmns`	512
`semsys:seminfo_semmsl`	17
`semsys:seminfo_semvmx`	537

[1] 10 進の 536870912 (512MB)。ただし、この数字は Solaris 8 オペレーティングシステムでは 16 進で示す必要があります。

`/etc/system` 設定を変更する手順は、『Solaris Tunable Parameters Reference Manual』を参照してください。

Linux システムの構成要件

`bc` コマンドは、N1 Service Provisioning System をインストールするときにユーザーのパスにある必要があります。`bc` コマンドが存在しないと、インストールが終了し、`bc` 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` パッケージまたは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。

SUSE Linux 8 サーバー上の NFS マウントしたディレクトリに N1 Service Provisioning System をインストールし、接続のセキュリティーを確保するために SSH または SSL を使用する場合は、`nolocks` で NFS クライアントを構成するか NFS サーバーを構成してロックを許可してください。NFS クライアントまたは NFS サーバーのどちらも正しく設定されていない場合は、N1 Service Provisioning System アプリケーションは起動しません。

Linux マスターサーバーのインストールプログラムは、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はエラーを表示して終了します。

表 **2–3** Linux マスターサーバーシステムの設定

システムのパラメータ	最小値
`/proc/sys/kernel/shmall` の `shmall`	536870912 (512MB)
`/proc/sys/kernel/shmmax` の `shmmax`	536870912 (512MB)

必要なオペレーティングシステムパッチ

次に、サポートされる各オペレーティングシステムに必要なパッチを示します。

表 **2–4** サポートされる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 7	106980-16
	106541-16
	107544-03
	106950-13
	106327-08
	106300-09
Solaris 8、SPARC ベースサーバー	111310-01
	109147-28
	111308-04
	112438-03
	108434-15
	108435-15
	111111-04
	112396-02
	110386-03
	111023-03
	111317-05
	113648-03
	115827-01
	116602-01
	108987-13
	108528-29
	108989-02
	108993-33
	109326-14
	110615-10
Solaris 9、SPARC ベースサーバー	114356-06

表 **2–4** サポートされる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 9、x86 ベースサーバー	114357–06
Solaris 10、SPARC ベースサーバー	なし
Solaris 10、x86 ベースサーバー	なし
IBM AIX 5.1	AIX 5.1–5.1.4.0 メンテナンスレベル: APAR IY44478
IBM AIX 5.2	AIX 5.2–5.2.1.0 メンテナンスレベル: APAR IY44479
IBM AIX 5.3	AIX 5.3.1.0 メンテナンスレベル: APAR IY58143
Red Hat Linux Advanced Server 2.1	なし
Red Hat Linux Advanced Server 3.0	なし
SUSE Linux Enterprise Server 8	なし
SUSE Linux Enterprise Server 9	なし

表 **2–4** サポートされる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
PA-RISC ベースシステム上の HP-UX 11i V1	PHNE_23502
	PHKL_24253
	PHKL_24254
	PHKL_24255
	PHKL_24256
	PHKL_24257
	PHKL_24751
	PHNE_25084
	PHCO_25226
	PHKL_25227
	PHKL_25367
	PHCO_25452
	PHKL_25468
	PHKL_25614
	PHKL_25728
	PHKL_25729
	PHKL_25840
	PHKL_25842
	PHKL_25871
	PHKL_27091
	PHKL_27092
	PHKL_28489
	PHNE_29887
	PHCO_29960
	PHSS_30049
Windows 2000 Server または Windows 2000 Advanced Server	Service Pack 3
Windows Server 2003	なし
Windows Server 2003 x64	なし

ハードウェア要件

マスターサーバーのハードウェア要件

サポートされているオペレーティングシステムにマスターサーバーをインストールするためのハードウェア要件を次の表に示します。

表 **2–5** マスターサーバーのハードウェア要件

	Solaris	Red Hat Linux	Windows
ハードウェア	SPARC または x86 ベース[1]	x86 ベース	x86 ベース
CPU	450 MHz シングルまたはマルチ CPU	1 GHz シングルまたはマルチ CPU	1 GHz シングルまたはマルチ CPU
RAM	1 G バイト以上の RAM	1 G バイト以上の RAM	1 G バイト以上の RAM
ハードディスクの空き容量	2 G バイト	2 G バイト	2 G バイト

[1] Solaris 8 OS の場合、SPARC のみ

ローカルディストリビュータ、リモートエージェント、および CLI Client のハードウェア要件

サポートされているオペレーティングシステムにローカルディストリビュータ、リモートエージェント、および CLI Client をインストールするためのハードウェア要件を次の表に示します。

表 **2–6** ローカルディストリビュータ、リモートエージェント、および CLI Client の ハードウェア要件

	SolarisSolaris	Red Hat Linux	IBM AIX	Suse Linux	HP-UX	Windows
ハードウェア	SPARC または x86 ベース[1]	x86 ベース	pSeries	x86 ベース	PA-RISC ベース	x86 ベース

[1] Solaris 7 および Solaris 8 OS の場合は、SPARC のみ

表 **2–6** ローカルディストリビュータ、リモートエージェント、および CLI Client の ハードウェア要件　*(続き)*

	SolarisSolaris	Red Hat Linux	IBM AIX	Suse Linux	HP-UX	Windows
CPU	400 MHz シングルまたはマルチ CPU	1 GHz シングルまたはマルチ CPU	400 MHz シングルまたはマルチ CPU	400 MHz シングルまたはマルチ CPU	400 MHz シングルまたはマルチ CPU	1 GHz シングルまたはマルチ CPU
RAM	256 M バイト 以上の RAM	1 G バイト 以上の RAM	256 M バイト 以上の RAM	1 G バイト 以上の RAM	256M バイト 以上の RAM	1 G バイト 以上の RAM
ハードディスクの空き容量	1G バイト	1G バイト	1G バイト	1G バイト	1G バイト	HD 上に 1G バイトの空き領域

一般システム要件

この節では、N1 Service Provisioning System 5.1 をインストールして使用するための一般的なシステム要件を示します。

サポートされる Web ブラウザ

次に、N1 Service Provisioning System 5.1 ブラウザインタフェースの Web ブラウザ要件を示します。

表 **2–7** ブラウザインタフェースの Web ブラウザ要件

プラットフォーム	ブラウザ
Solaris、Red Hat、SUSE、HP-UX	Netscape Navigator™ 7.1、Mozilla™ 1.4
Windows	Netscape Navigator 7.1、Mozilla 1.4、Internet Explorer 5.5、Internet Explorer 6.0
AIX	Mozilla 1.4

注 – 一部の Web プロキシサーバーは、ポップアップウィンドウをブロックするように構成されます。N1 Service Provisioning System 5.1 を正しく動作させるためには、ポップアップウィンドウを表示させる必要があります。ポップアップウィンドウをブロックする Web プロキシサーバーを実行したり、ポップアップウィンドウをブロックするようにブラウザを設定したりしないでください。

正しく動作するように、N1 Service Provisioning System 5.1 は cookie を使用する機能に依存しています。cookie を使用できるようにブラウザを設定してください。

SSH の要件

Linux および UNIX システム上で SSH を使用してセキュリティー保護された接続を行う場合は、SSH を使用する各サーバーに 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 Client で Jython を使用する場合は、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 5.1 は、各国語環境でインストールして使用できるように国際化されています。N1 Service Provisioning System 5.1 は、ASCII 以外の文字も受け付けます。ソフトウェアで ASCII 以外の文字をサポートする必要がある場合は、次の要件を満たす必要があります。

- すべてのアプリケーションを同じロケールまたは同等のロケールで実行する。リモートエージェント、ローカルディストリビュータ、および CLI Client は、マスターサーバーが稼働しているロケールと互換性があるロケールを実行する必要があります。
- Internet Explorer 5.5 または 6.0、あるいは Netscape 7.0 を使用する必要がある
- Web ブラウザの Character Interface が UTF-8 (Unicode または Universal Alphabet とも呼ばれる) を使用するように設定する
- `config.properties` ファイルなどの構成ファイルでは、ASCII 以外のすべての文字は、Unicode でコード化する必要があります。構成ファイルは任意のエンコードで作成できます。続いて、Java™ Development Kit (JDK) パッケージで提供されている `native2ascii` コマンドを使用してファイルを Unicode で符号化された ASCII 文字に変換します。

第 3 章

インストールの前に収集すべき情報

この章では、N1 Service Provisioning System 5.1 をインストールする上で必要な意思決定と情報収集に役立つ情報とワークシートを挙げています。この章の内容は次のとおりです。

- 35 ページの「一般的な構成に関する決定事項」
- 38 ページの「セキュリティー構成に関する決定事項」
- 40 ページの「全アプリケーションのワークシート」
- 41 ページの「マスターサーバーのワークシート」
- 42 ページの「ローカルディストリビュータのワークシート」
- 42 ページの「リモートエージェントのワークシート」
- 43 ページの「CLI Client のワークシート」

一般的な構成に関する決定事項

インストールプログラムを実行すると、N1 Service Provisioning System 5.1 の構成情報の入力を求めるメッセージが表示されます。インストールを始める前に、以下の各節を利用して構成についての必要事項を決定してください。

Java Runtime Environment

Linux および UNIX サーバーにインストールする場合は、JRE をインストールするか、あるいは有効な JRE のパスを指定するようにメッセージが表示されます。Windows システムにインストールする場合は、メッセージは表示されずに自動的に JRE がインストールされます。

注 – Solaris 10 サーバーでは、すでに `/usr/j2se` にインストールされている JRE v1.4.2 を使用するようメッセージが表示されます。新しいバージョンをインストールせずに、このバージョンの JRE を使用することも選択できます。

Red Hat Linux または SUSE Linux サーバーにインストールする場合は、インストールスクリプトによってマシンが検索され、デフォルトの場所に JRE インスタンスがないか確認されます。

- デフォルトの場所に JRE がインストールされていない場合は、JRE をインストールする必要があります。
- デフォルトの場所に JRE が見つかった場合は、JRE をインストールし直すかどうかを選択できます。

Solaris OS、IBM AIX、または HP-UX サーバーにインストールする場合で、JRE をインストールしないことを選択したときは、有効な JRE のパスの指定を求めるメッセージが表示されます。インストールスクリプトは、その JRE がサポートされているバージョンかどうかを確認します。

- その JRE がサポートされているバージョンよりも高いバージョンの場合は、その JRE がサポートされていないことを知らせる警告と、継続するかどうかを訪ねるメッセージが表示されます。
- N1 Service Provisioning System 5.1 でサポートされている JRE バージョンを指定した場合、インストールスクリプトは `JRE_HOME` 変数をその JRE に設定します。インストールスクリプトは、JRE ディレクトリを指すシンボリックリンク *N1SPS5.1-home*`/common/jre` も作成します。*N1SPS5.1-home* は、N1 Service Provisioning System 5.1 のホームディレクトリです。シンボリックリンクを作成することによって N1 Service Provisioning System 5.1 アプリケーションは、ほかのアプリケーションが依存している可能性のある JRE の場所を変更することなく JRE を使用します。

注 – バンドル版の JRE は、各マシンに 1 度だけインストールします。たとえば、マスターサーバー、ローカルディストリビュータ、CLI Client を同じマシンにインストールする場合、マスターサーバーと共に JRE をインストールし、ローカルディストリビュータと CLI Client のインストールでは JRE をインストールしないようにします。

アプリケーションのユーザー所有権

インストールでは、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を選択するようにメッセージが表示されます。アプリケーションが SSH を使用して通信を行うように構成する場合は、同じユーザーでマスターサーバー、ローカルディストリビュータ、リモートエージェントをインストールしてください。

スーパーユーザー (root) は、マスターサーバーの所有者にはなれません。マスターサーバーは、マスターサーバーを所有するユーザーでインストールするか、あるいはスーパーユーザー (root) でこのアプリケーションをインストールします。どのユーザーがマスターサーバーを所有するかを尋ねるメッセージが表示される時点で、所有するユーザーを指定することもできます。

注 – マスターサーバーまたは CLI Client を Solaris サーバーにインストールする場合は、root でログインする必要があります。

リモートエージェントがその稼働マシン上で root 権限を持つようにする場合は、インストールプログラムをスーパーユーザー (root) で実行する必要があります。リモートエージェントの所有者にスーパーユーザー (root) 以外のユーザーを指定できますが、その稼働マシン上でリモートエージェントに root 権限を持たせるときは、インストールプログラムをスーパーユーザー (root) で起動してください。

ホスト名と IP アドレス

N1 Service Provisioning System 5.1 アプリケーションは、すべてのサーバーが静的な IP アドレスを持つことを要求します。これは、サーバー上にインストールされている N1 Service Provisioning System 5.1 アプリケーションがその IP アドレスを使用してネットワーク要求を待機するためです。インストールでは、ホスト名または IP アドレスを指定するように求められます。サーバー上のホスト名からそのサーバーの IP アドレスが解決されない場合は、そのサーバーを N1 Service Provisioning System 内で接続するように構成することはできません。

インストールでホスト名を指定する場合、そのホスト名はサーバーの実際の IP アドレスを解決するものでなければなりません。サーバーによっては、ホスト名が IP アドレスを解決しないように構成されるものや、ホスト名がループバックアドレス 127.0.0.1 を解決するように構成されるものがあります。このように構成されたサーバーのホスト名が N1 Service Provisioning System アプリケーションに設定されると、そのアプリケーションは起動に失敗する可能性があります。また、ほかの N1 Service Provisioning System アプリケーションからこのサーバーへの接続も失敗する可能性があります。

N1 Service Provisioning System アプリケーションをインストールする場合は、サーバーのホスト名ではなく IP アドレスを指定してください。ホスト名を指定する場合は、そのホスト名からサーバーの実際の IP アドレスが解決されるか確認してください。

注 – 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求めるメッセージが表示されます。マスターサーバーと CLI Client 間の通信に SSH を使用することを選択すると、マスターサーバーの IP アドレスは、127.0.0.1 に設定されます。

Jython

CLI Client のインストールでは、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ようにメッセージが表示されます。CLI Client は、対話モードで動作する場合に Jython プログラミング言語を使用します。しかし、Jython は CLI Client を使用する上での必須要素ではありません。Jython と CLI Client の詳細は、20 ページの「CLI Client」を参照してください。

セキュリティー構成に関する決定事項

ネットワークプロトコル – raw、SSH、SSL

インストールでは、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を選択するようにメッセージが表示されます。マスターサーバーの場合、raw (TCP/IP) または SSL を選択できます。ローカルディストリビュータ、リモートエージェント、および CLI Client の場合、raw (TCP/IP)、SSH、または SSL を選択できます。

raw (TCP/IP) は、通信プロトコルとしては安全性に劣ります。プロビジョニングシステムでこの接続プロトコルを使用すると、N1 Service Provisioning System 5.1 アプリケーションがインストールされたサーバーにネットワークアクセスできる人なら誰でもプロビジョニングシステムに接続してコマンドを発行できる状態になります。raw を選択する場合は、N1 Service Provisioning System 5.1 アプリケーションがインストールされているサーバーからの接続だけを受け入れるようにセキュリティーポリシーファイルを構成することによってプロビジョニングシステムを保護できます。詳細は、第 9 章を参照してください。

SSL は raw よりも安全性が高いプロトコルです。SSL を選択する場合は、どの暗号群を使用するか、認証を行うかどうか (認証を伴う暗号化にするか認証を伴わない暗号化にするか) も指定する必要があります。認証を伴わない暗号化は、プロビジョニングシステムアプリケーションがインストールされたサーバーにネットワークアクセスできる人であれば誰でもプロビジョニングシステムに接続してコマンドを実行できるという点で raw の使用に似ています。SSL を使用する場合は、認証を伴う暗号化モー

ドを選択するのがもっとも安全です。プロビジョニングシステムは、N1 Service Provisioning System 5.1 アプリケーションがインストールされているサーバーからの接続だけを受け入れるようにセキュリティーポリシーファイルを構成することによって、安全性をさらに高めることができます。詳細は、第 9 章を参照してください。SSL については、第 8 章を参照してください。

注 – AIX サーバー上のローカルディストリビュータで SSL を使用する場合、SSL 暗号群は認証による暗号化に設定されます。認証を伴わない暗号化は、AIX サーバーで稼働しているローカルディストリビュータでは利用できません。

SSH はもっとも安全性の高いネットワークプロトコルであり、サポートされるのは Linux および UNIX ベースのプラットフォーム上だけです。N1 Service Provisioning System 5.1 で SSH を使用するには、サーバーに SSH ソフトウェアをインストールする必要があります。詳細は、第 7 章を参照してください。

注 – マスターサーバーと CLI Client 間の通信を行うためのネットワークプロトコルとして SSH を使用することを選択した場合、マスターサーバーの IP アドレスは 127.0.0.1 に設定されます。マスターサーバーの通信プロトコルは raw に設定されます。SSH を使用してマスターサーバーに接続するには、CLI Client を構成する必要があります。

HTTP または HTTPS

ブラウザインタフェースがマスターサーバーへの接続に HTTP (Hypertext Transmission Protocol) または HTTPS (Hypertext Transmission Protocol, Secure) を使用することを選択できます。HTTP を選択する場合は、ネットワーク上の誰でもブラウザインタフェースとマスターサーバー間で転送されるデータを傍受できます。また、ユーザーがマスターサーバーとして動作して、ユーザーパスワードなどのセキュリティー保護されているデータをブラウザインタフェースから取得しようとすることも考えられます。

HTTPS は、HTTP より安全にデータを転送します。HTTPS では、キーストアファイルとキーストアパスワードが必要です。プロビジョニングシステムが使用するキーストアファイルとキーストアパスワードを作成する必要があります。

インストールプログラムにより、HTTP または HTTPS を選択するようメッセージが表示されます。HTTPS を選択すると、キーストアファイルとキーストアパスワードを入力するようにメッセージが表示されます。キーストアファイルとキーストアパスワードは、2 とおりの方法で指定できます。

- インストールの前にキーストアファイルとキーストアパスワードを作成する方法: プロビジョニングシステムをインストールする前にキーストアファイルとキーストアパスワードを作成する場合は、インストール中にキーストアファイルとキーストアパスワードの情報を指定できます。プロビジョニングシステムをインストールす

る前にキーストアファイルとキーストアパスワードを作成する方法については、70 ページの「HTTPS 接続のキーストアファイルとキーストアパスワードの作成」を参照してください。

注 – キーストアパスワードが有効かどうかは確認されません。無効なパスワードを入力した場合、マスターサーバーを起動できないことがあります。

- インストールしたあとにキーストアファイルとキーストアパスワードを作成する方法: インストールする前にキーストアファイルとキーストアパスワードを作成しない場合は、インストール中にキーストアファイルとキーストアパスワードの情報を指定できません。この結果、インストールスクリプトによって、空のキーストアファイルが *N1SPS5.1-home*/server/tomcat ディレクトリに作成されます。空のキーストアファイルとキーストアパスワード情報は、マスターサーバーを起動する前に手動で交換する必要があります。手順については、72 ページの「インストール後の HTTPS の構成」を参照してください。

インストール中に HTTP を選択すると、 HTTPS を使用するためにマスターサーバーを手動で構成できます。手順については、74 ページの「インストール中に HTTP を選択したあとで行う HTTPS の構成」を参照してください。

全アプリケーションのワークシート

各 N1 Service Provisioning System 5.1 アプリケーションのインストールスクリプトは、初めに同じ予備作業を行い、ディレクトリとファイルについての同じ質問を表示します。次のワークシートを使って、各 N1 Service Provisioning System 5.1 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てください。

表 **3–1** 全アプリケーションで要求される情報

説明	値
ソフトウェアのインストール先となるベースディレクトリ。 例: /opt/SUNWn1sps	_____
JRE がすでにマシンにインストールされている場合は、JRE のパス。 例: /usr/local/jre、 JAVA_HOME 環境変数の値など	_____
インストールするアプリケーションの所有者となるユーザー。	_____

表 **3–1** 全アプリケーションで要求される情報　(続き)

説明	値
Linux および UNIX システムでは、インストールするアプリケーションを保持するグループ。	_____

マスターサーバーのワークシート

マスターサーバーのインストールに必要な情報を収集する際には、次のワークシートを使用してください。

表 **3–2** マスターサーバーで要求される情報

説明	値
マスターサーバーマシンの IP アドレスまたはホスト名。	_____
マスターサーバーと CLI Client 間の通信プロトコルとして SSH を使用することを選択した場合、マスターサーバーの IP アドレスは自動的に 127.0.0.1 に設定されます。	
マスターサーバーへの接続に CLI Client が使用すべき IP ポート番号。 例: 1130	_____
通知メールメッセージを送信するためにソフトウェアが使用する SMTP メールサーバーの IP アドレスまたはホスト名。	_____
ソフトウェアから送信される電子メール通知のタイトル行。 例: N1 Service Provisioning System 通知	_____
電子メール通知の送信元となるユーザーアカウントの名前 (ユーザー名)。インストールプログラムは、入力されるユーザーアカウント名の有効性を検証しません。	_____
ネイティブコマンドの実行時にソフトウェアが使用すべきユーザーアカウントの名前。インストールプログラムは、入力されるユーザーアカウント名の有効性を検証しません。	_____
Postgres データベースが待機するポート番号。 例: 5432	_____
インストールが完了したあとで admin ユーザーがマスターサーバー ブラウザインタフェースにアクセスするために使用するパスワード。	_____

表 **3–2** マスターサーバーで要求される情報　(続き)

説明	値
ブラウザインタフェースを使用できるポート番号。 例: HTTP の場合、8080。HTTPS の場合、8443	_____
Postgres データベースの最適化を自動的に行うかどうか。 自動的に行う場合は、マスターサーバーデータベースを最適化する時間を HH:MM の形式で指定してください。 例: 23:00 毎日データベースの最適化が行われるよう、`crontab` ファイルにエントリが作成されます。インストールの前に `crontab` ファイルが存在するか確認してください。存在しない場合は、作成してください。	_____

ローカルディストリビュータのワークシート

ローカルディストリビュータのインストールに必要な情報を収集する際には、次のワークシートを使用してください。

表 **3–3** ローカルディストリビュータで要求される情報

説明	値
ローカルディストリビュータマシンの IP アドレスまたはホスト名。	_____
このローカルディストリビュータが待機するポート番号。 親アプリケーションとローカルディストリビュータ間の通信プロトコルとして SSH を使用することを選択した場合、ポート番号は自動的に 70001 に設定されます。 例: 1132	_____

リモートエージェントのワークシート

リモートエージェントのインストールに必要な情報を収集する際には、次のワークシートを使用してください。

表 **3–4** リモートエージェントで要求される情報

説明	値
リモートエージェントが実行される IP アドレスまたはホスト名。	_____
このリモートエージェントが待機するポート番号。 親アプリケーションとリモートエージェント間の通信プロトコルとして SSH を使用することを選択した場合、ポート番号は自動的に 70000 に設定されます。 例: 1131	_____

CLI Client のワークシート

CLI Client のインストールに必要な情報を収集する際には、次のワークシートを使用してください。

表 **3–5** CLI Client で要求される情報

説明	値
コマンド行ユーザーインタフェースに使用される、マスターサーバーの IP アドレスまたはホスト名	_____
マスターサーバーと CLI Client 間の通信プロトコルとして SSH を使用することを選択した場合、マスターサーバーの IP アドレスは 127.0.0.1 に設定されます。	
マスターサーバーの IP ポート番号。 マスターサーバーと CLI Client 間の通信プロトコルとして SSH を使用することを選択した場合、ポート番号は自動的に 80001 に設定されます。 例: 1130	_____
マスターサーバーと CLI Client 間の通信プロトコルとして SSH を使用することを選択した場合、マスターサーバーのインストールディレクトリを入力する必要があります。 例: `/opt/SUNWn1sps/N1_Service_Provisioning_System_5.1/server`	_____
Jython がこのマシンにすでにインストールされている場合は、Jython のパス。 デフォルト値: `/usr/local/jython`	_____

第 4 章

Linux および UNIX システムへの N1 Service Provisioning System 5.1 のインストール

この章では、Linux および UNIX システムに N1 Service Provisioning System 5.1 をインストールする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。

- 45 ページの「N1 Service Provisioning System 5.1 のインストール」
- 48 ページの「Linux および UNIX へのリモートエージェントへの非対話方式によるインストール」
- 51 ページの「Linux および UNIX システムへのリモートエージェントのリモートインストール」
- 54 ページの「Linux および UNIX システムでのアプリケーションの起動」

N1 Service Provisioning System 5.1 のインストール

製品メディアから該当するインストールスクリプトを選択し、各アプリケーションを個別にインストールします。各 N1 Service Provisioning System 5.1 アプリケーションのインストールスクリプトは、初めに同じ予備作業を実行し、ディレクトリ、ファイル、および Java™ Runtime Environment (JRE) についての同じ質問を表示します。続いて、インストールするアプリケーションに固有の、構成に関する質問を表示します。

▼ Linux および UNIX システムに N1 Service Provisioning System 5.1 をインストールする

始める前に　17 ページの「N1 Service Provisioning System 5.1 のインストール – 手順の概要」のインストール手順の概要に目を通します。アプリケーションのインストール前に必要な作業があれば、それらを済ませます。

手順 **1.** そのアプリケーションを所有するユーザーでログインします。

root でログインすれば、ルートユーザーでソフトウェアをインストールできます。必要に応じインストールプログラムは、どのユーザーがソフトウェアを所有すべきか指定するようにメッセージを表示します。

注 – マスターサーバーまたは CLI Client を Solaris サーバーにインストールする場合は、root でログインする必要があります。

---

2. インストールスクリプトにアクセスします。

- CD からインストールしている場合は、該当する CD を挿入します。
  - Solaris OS、SPARC サーバーにソフトウェアをインストールする場合は、Sun N1 Service Provisioning System 5.1: Solaris, SPARC CD を挿入します。
  - Solaris OS、x86 サーバーにソフトウェアをインストールする場合は、Sun N1 Service Provisioning System 5.1: Solaris, x86 CD を挿入します。
  - Red Hat Linux または SUSE Linux にソフトウェアをインストールする場合は、Sun N1 Service Provisioning System 5.1: Red Hat Linux, SUSE Linux CD を挿入します。
  - HP-UX にリモートエージェントまたはローカルディストリビュータをインストールする場合は、Sun N1 Service Provisioning System 5.1: HP-UX (CD 1 of 2) CD を挿入します。
  - HP-UX に CLI をインストールする場合は、Sun N1 Service Provisioning System 5.1: HP-UX (CD 2 of 2), IBM-AIX CD を挿入します。
  - IBM AIX にソフトウェアをインストールする場合は、Sun N1 Service Provisioning System 5.1: HP-UX (CD 2 of 2), IBM-AIX CD を挿入します。
- ダウンロードしたイメージからインストールする場合は、そのイメージが保存されているディレクトリに移動します。

3. スクリプトが置かれている場所 **(**ソフトウェア **CD** 上のディレクトリまたはダウンロードしたイメージ内のディレクトリ**)** に移動します。

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

script-directory には、次に示す値の 1 つを指定します。

- `solaris_sparc`
- `solaris_x86`
- `aix`
- `linux`
- `hpux_risc`

4. 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起動します。

```
# cr_app_opsystem_5.1.sh [-allowForwardVersion]
```

app には、次に示す値の 1 つを指定します。

- ms – マスターサーバーをインストールする
- ra – リモートエージェントをインストールする
- ld – ローカルディストリビュータをインストールする
- cli – CLI Client をインストールする

opsystem には、次に示す値の 1 つを指定します。

- solaris_sparc – Solaris OS を使用している SPARC ベースのハードウェアにアプリケーションをインストールする。マスターサーバーまたは CLI Client をインストールするには、`solaris_sparc_pkg` を使用します。
- solaris_x86 – Solaris OS を使用している x86 ベースのハードウェアにアプリケーションをインストールする。マスターサーバーまたは CLI Client をインストールするには、`solaris_x86_pk` を使用します。
- aix – IBM AIX にアプリケーションをインストールする
- linux – Red Hat Linux および SUSE Linux Enterprise Server にアプリケーションをインストールする
- hpux_risc – HP-UX を実行している PA-RISC ベースのシステムにアプリケーションをインストールする

`-allowForwardVersion` オプションを指定すると、N1 Service Provisioning System 5.1 がそのオペレーティングシステムでサポートしている最高バージョンを超える OS バージョンに N1 Service Provisioning System 5.1 アプリケーションをインストールできます。`-allowForwardVersion` オプションを使用すると、インストールプログラムはアプリケーションがインストールされるオペレーティングシステムがサポートされているかどうかを確認しません。サポートされていないオペレーティングシステムでの N1 Service Provisioning System 5.1 の使用については、Sun サービスは標準サポートを提供しません。



注意 – サポートされていないオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールすると、不確定な動作や予期しない動作が発生することがあります。サポートされていないオペレーティングシステムに対する N1 Service Provisioning System 5.1 のインストールは、テスト目的以外では行わないでください。運用環境では、サポートされていなオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールして使用することは避けてください。

5. インストール中に構成についての質問が表示される場合は、その質問に答えてください。

   インストールが完了すると、アプリケーションを起動するか尋ねるメッセージが表示されます。

   インストールプログラムは、`/tmp/N1SPSInstaller.log` ファイルにイベントログを保存します。

Linux および UNIX へのリモートエージェントへの非対話方式によるインストール

リモートエージェントは、構成設定を示すパラメータファイルを指定することによって非対話型形式でインストールできます。インストールプログラムに対してパラメータファイルを指定すると、インストール中に構成についての選択を求めるメッセージは表示されません。インストールプログラムは、パラメータファイル内に指定された構成情報を使用します。

▼ Linux および UNIX にリモートエージェントを非対話方式でインストールする

始める前に　リモートエージェントをインストールするには、あらかじめマスターサーバーをインストールする必要があります。マスターサーバーは、リモートエージェントをインストールするマシンにインストールする必要はありません。

手順

1. リモートエージェントをインストールするマシンで、リモートエージェントを所有するユーザーでログインします。

   root でログインすれば、ルートユーザーでソフトウェアをインストールできます。必要に応じインストールプログラムは、どのユーザーがソフトウェアを所有すべきか指定するようにメッセージを表示します。

2. インストールスクリプトにアクセスします。
   - CD からインストールしている場合は、該当する CD を挿入します。
     - Solaris OS、SPARC サーバーにリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: Solaris, SPARC CD を挿入します。
     - Solaris OS、x86 サーバーにリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: Solaris, x86 CD を挿入します。
     - Red Hat Linux または SUSE Linux にリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: Red Hat Linux, SUSE Linux CD を挿入します。
     - HP-UX にリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: HP-UX (CD 1 of 2) CD を挿入します。
     - IBM AIX にリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: HP-UX (CD 2 of 2), IBM-AIX CD を挿入します。

- ダウンロードしたイメージからインストールする場合は、そのイメージが保存されているディレクトリに移動します。

3. スクリプトが置かれている場所 **(**ソフトウェア **CD** 上のディレクトリまたはダウンロードしたイメージ内のディレクトリ**)** に移動します。

   % cd /*script-directory*

   script-directory には、次に示す値の 1 つを指定します。

   - `solaris_sparc`
   - `solaris_x86`
   - `aix`
   - `linux`
   - `hpux_risc`

4. リモートエージェントをインストールするマシンに、インストールスクリプトをコピーします。

   % cp cr_ra_*opsystem***_5.1.sh** *RA-machine***/**

   RA-machine は、リモートエージェントをインストールするマシン上のディレクトリです。*opsystem* には、次に示す値の 1 つを指定します。

   - solaris_sparc – Solaris OS を使用している SPARC ベースのハードウェアに リモートエージェントをインストールする
   - solaris_x86 – Solaris OS を使用している x86 ベースのハードウェアにリモートエージェントをインストールする
   - aix – IBM AIX にリモートエージェントをインストールする
   - linux – Red Hat Linux および SUSE Linux Enterprise Server にリモートエージェントをインストールする
   - hpux_risc – HP-UX を実行している PA-RISC ベースのシステムにアプリケーションをインストールする

5. インストールスクリプトと同じディレクトリにパラメータファイルをコピーします。

   マスターサーバーをインストールする際に、*N1SPS5.1-MasterServer-home*`/server/bin` ディレクトリのマスターサーバー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。このファイル内に指定されているデフォルト値を使用することも、あるいはファイルを編集してカスタム値を追加することもできます。`cr_ra_remote_params.sh` サンプルパラメータファイルの内容と設定可能な変数の説明は、131 ページの「Linux および UNIX システム向けのリモートエージェントパラメータサンプル」を参照してください。

   使用するパラメータファイルを新たに作成することもできます。パラメータファイルは、実行可能なファイルでなければなりません。

   N1SPS5.1-MasterServer-home は、マスターサーバーをインストールしたディレクトリです。



注意 – CR_RA_SUID と CR_RA_CTYPE 変数の値は、サンプルパラメータファイルには設定されていません。これらの変数の値は設定する必要があります。設定されていない場合、インストールスクリプトが中止されます。raw や認証なしの SSL 暗号化などのセキュリティー保護されていない接続タイプを使用することを選択した場合は、CR_RA_CTYPE_CONFIRM 変数の値を true に設定する必要があります。

6. インストールスクリプトを起動します。

```
% cr_ra_opsystem_5.1.sh -paramfile parameters-file.sh [-allowForwardVersion]
```

opsystem には、次に示す値の 1 つを指定します。

- solaris_sparc – Solaris OS を使用している SPARC ベースのハードウェアにリモートエージェントをインストールする
- solaris_x86 – Solaris OS を使用している x86 ベースのハードウェアにリモートエージェントをインストールする
- aix – IBM AIX にリモートエージェントをインストールする
- linux – Red Hat Linux および SUSE Linux Enterprise Server にリモートエージェントをインストールする
- hpux_risc – HP-UX を実行している PA-RISC ベースのシステムにアプリケーションをインストールする

parameters-file には、インストールプログラムに構成情報を取得させ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、実行可能なファイルでなければなりません。

-allowForwardVersion オプションを指定すると、N1 Service Provisioning System 5.1 がそのオペレーティングシステムでサポートしている最高バージョンを超える OS バージョンに N1 Service Provisioning System 5.1 リモートエージェントをインストールできます。-allowForwardVersion オプションを使用すると、インストールプログラムはリモートエージェントがインストールされるオペレーティングシステムがサポートされているかどうかを確認しません。サポートされていないオペレーティングシステムでの N1 Service Provisioning System 5.1 の使用については、Sun サービスは標準サポートを提供しません。



注意 – サポートされていないオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールすると、不確定な動作や予期しない動作が発生することがあります。サポートされていないオペレーティングシステムに対する N1 Service Provisioning System 5.1 のインストールは、テスト目的以外では行わないでください。運用環境では、サポートされていなオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールして使用することは避けてください。

Linux および UNIX システムへのリモートエージェントのリモートインストール

ネットワーク上のほかのマシンから、リモートエージェントをインストールできます。リモートエージェントをリモートインストールするために必要なスクリプトは、マスターサーバーのインストール時に *N1SPS5.1-MasterServer-home* /server/bin ディレクトリにインストールされます。このインストールでは、環境変数を使用して非対話方式でリモートエージェントのインストールと構成が行われます。この環境変数はパラメータファイル内またはコマンド行で設定することも、インストールスクリプトに入っているデフォルト値を使用することもできます。

▼ Linux および UNIX システムにリモートエージェントをリモートインストールする

リモートエージェントのインストールスクリプトは、対象マシンのオペレーティングシステムに固有のものです。これらのステップは、対象マシン上で使用されている各オペレーティングシステムに合わせて行う必要があります。

始める前に　対象マシンは、次の要件を満たす必要があります。

- UNIX ユーティリティ sshd が実行されており、ソースマシンに対して直接 IP 接続を確立できる
- hostname UNIX コマンドをサポートしている必要がある。リモートマシンのインストールスクリプトがこのコマンドを呼び出します。リモートエージェントは、hostname コマンドが返すホスト名の IP アドレスで待機するように構成する必要があります。

実行時には、UNIX ユーティリティー ssh と scp がマスターサーバーマシンにインストールされ、パス内に存在し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、環境変数を使用してリモートエージェントのインストールと構成を行います。環境変数は、パラメータファイルまたはコマンド行で設定できます。次の環境変数の値を宣言する必要があります。

- CR_RA_CTYPE=raw – リモートエージェントは、暗号化を行わずに マスターサーバーまたはローカルディストリビュータに接続する。有効な値は、raw、ssh、および ssl です。
- CR_RA_SUID – setuid root 権限でリモートエージェントをインストールします。値 yes を指定するには、インストールスクリプトをスーパーユーザー (root) で実行する必要があります。有効な値は y と n です。
- CR_RA_INSTALLER_HOSTS – コマンド行で、または環境変数としてホスト名を指定しないと、インストールスクリプトはエラーを生成して終了します。有効な値の例を次に示します。

```
CR_RA_INSTALLER_HOSTS=host1,host3.enterprise.com,10.10.0.207.
```

手順 **1.** マスターサーバーマシンで、インストールスクリプトにアクセスします。

- CD からインストールしている場合は、該当する CD を挿入します。
  - Solaris OS、SPARC サーバーにリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: Solaris, SPARC CD を挿入します。
  - Solaris OS、x86 サーバーにリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: Solaris, x86 CD を挿入します。
  - Red Hat Linux または SUSE Linux にリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: Red Hat Linux, SUSE Linux CD を挿入します。
  - HP-UX にリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: HP-UX (CD 1 of 2) CD を挿入します。
  - IBM AIX にリモートエージェントをインストールする場合は、Sun N1 Service Provisioning System 5.1: HP-UX (CD 2 of 2), IBM-AIX CD を挿入します。
- ダウンロードしたイメージからインストールする場合は、そのイメージが保存されているディレクトリに移動します。

2. スクリプトが置かれている場所 **(**ソフトウェア **CD** 上のディレクトリまたはダウンロードしたイメージ内のディレクトリ**)** に移動します。

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

script-directory には、次に示す値の 1 つを指定します。

- `solaris_sparc`
- `solaris_x86`
- `aix`
- `linux`
- `hpux_risc`

3. インストールスクリプトをマスターサーバーにコピーします。

```
% cp cr_ra_opsystem_5.1.sh N1SPS5.1-MasterServer-home/server/bin
```

N1SPS5.1-MasterServer-home は、マスターサーバーをインストールしたディレクトリです。*opsystem* には、次に示す値のどれかを指定します。

- solaris_sparc – Solaris OS を使用している SPARC ベースのハードウェアにリモートエージェントをインストールする
- solaris_x86 – Solaris OS を使用している x86 ベースのハードウェアにリモートエージェントをインストールする
- aix – IBM AIX にリモートエージェントをインストールする

- linux – Red Hat Linux および SUSE Linux Enterprise Server にリモートエージェントをインストールする
- hpux_risc – HP-UX を実行している PA-RISC ベースのシステムにアプリケーションをインストールする

4. スクリプトが置かれているディレクトリへ移動します。

```
% cd N1SPS5.1-MasterServer-home/server/bin
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

5. インストールスクリプトに構成情報をどのように提供するかを決定します。

- 新しいパラメータファイルを作成するか、あるいは N1 Service Provisioning System 5.1 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。マスターサーバーをインストールする際に、パラメータファイルがインストールされます。このファイルには、*N1SPS5.1-MasterServer-home*/server/bin/cr_ra_remote_params.sh という名前が付けられます。このファイル内に指定されているデフォルト値を使用することも、あるいはファイルを編集してカスタム値を追加することもできます。使用するパラメータファイルを新たに作成することもできます。サンプルパラメータファイルの内容と設定可能な変数の説明は、131 ページの「Linux および UNIX システム向けのリモートエージェントパラメータサンプル」を参照してください。パラメータファイルは、実行可能なファイルでなければなりません。



注意 – CR_RA_SUID と CR_RA_CTYPE 変数の値は、サンプルパラメータファイルには設定されていません。これらの変数の値は設定する必要があります。設定されていない場合、インストールスクリプトが中止されます。raw や認証なしの SSL 暗号化などのセキュリティー保護されていない接続タイプを使用することを選択した場合は、CR_RA_CTYPE_CONFIRM 変数の値を true に設定する必要があります。

- 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, no
% export CR_RA_CTYPE=raw, ssh, ssl
% export CR_RA_SUID=y, n
% export CR_RA_INSTALLER_HOSTS=hostnames.enterprise.com,10.10.0.207
```

6. リモートインストールを開始します。

```
% cr_ra_remote.sh -paramfile path-to-file/parameters-file.sh -f
cr_ra_opsystem_5.1.sh hostnames
```

- cr_ra_*opsystem*_5.1.sh は、N1 Service Provisioning System 5.1 CD またはダウンロードしたイメージからコピーしたインストールスクリプトを指定します。

- *path-to-file/parameters-file* には、構成情報を取得するためにインストールプログラムが使用するパラメータファイルのパスと名前を指定します。環境変数を設定する場合と、インストールスクリプトにデフォルト値を使用させる場合は、パラメータファイルを指定する必要はありません。
- *hostnames* は、インストール先となるマシンのホスト名です。ホスト名はスペースで区切ってください。パラメータファイル内で、または環境変数として `CR_RA_INSTALLER_HOSTS` パラメータにホスト名を指定した場合は、コマンド行でホスト名を指定する必要はありません。コマンド行でホスト名を指定すると、それらのホストにインストールされ、`CR_RA_INSTALLER_HOSTS` パラメータに指定したホストにはインストールされません。

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は、ログファイルが作成されていることと、そのログファイルの場所を表示します。あとで確認できるように、このファイルの場所をメモに記録しておいてください。

8. リモートマシンのパスワードを求めるメッセージが表示された場合は、そのパスワードを指定してください。

   インストールスクリプトは、リモートマシン上にログファイルを生成します。

Linux および UNIX システムでのアプリケーションの起動

次の表に、Linux および UNIX システムで N1 Service Provisioning System 5.1 アプリケーションを起動するコマンドを示します。*N1SPS5.1-home* は、アプリケーションのホームディレクトリです。



注意 – N1 Service Provisioning System 5.1 アプリケーション (マスターサーバーなど) の起動に Bourne シェルは使用しないでください。Bourne シェルで `cr_server start` コマンドを使用してマスターサーバープロセスを起動し、マスターサーバーを起動したのと同じシェルで、後続のコマンドに対して `^C` コマンドを実行すると、データベースとマスターサーバーのプロセスが停止します。

N1SPS5.1-home`/server/bin/roxdb.out` ファイルで、次のメッセージが最近のエントリとして表示されます。

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

表 **4–1** Linux および UNIX システムアプリケーションの起動コマンド

アプリケーション	コマンドのパス	起動用のコマンド
マスターサーバー	*N1SPS5.1-home*/server/bin/	cr_server start
ローカルディストリビュータ	*N1SPS5.1-home*/ld/bin/	cr_ld start
リモートエージェント	*N1SPS5.1-home*/agent/bin/	cr_ra start
CLI Client	*N1SPS5.1-home*/cli/bin/	cr_cli *CLI-command*
Jython バージョンの CLI Client	*N1SPS5.1-home*/cli/bin/	cr_clij *CLI-command*

第 5 章

Windows システムへの N1 Service Provisioning System 5.1 のインストール

この章では、Windows を使用しているサーバーに N1 Service Provisioning System 5.1 をインストールする手順を説明します。製品メディアから該当する Microsoft Installer (MSI) パッケージを選択し、各アプリケーションを個別にインストールします。各 N1 Service Provisioning System 5.1 アプリケーションの MSI パッケージは、初めに同じ予備作業を行い、ディレクトリとファイルについての同じ質問を表示します。続いて、インストールするアプリケーションに固有の、構成に関する質問を表示します。

この章の内容は、次のとおりです。

- 57 ページの「マスターサーバーのインストール」
- 60 ページの「リモートエージェント、ローカルディストリビュータ、CLI Client のインストール」
- 62 ページの「Windows へのリモートエージェントの非対話方式インストール」
- 63 ページの「Windows へのリモートエージェントのリモートインストール」
- 66 ページの「リモートエージェントの変数値」
- 67 ページの「Windows システムでのアプリケーションの起動」

マスターサーバーのインストール

▼ Windows に N1 Service Provisioning System 5.1 マスターサーバーをインストールする

始める前に　17 ページの「N1 Service Provisioning System 5.1 のインストール – 手順の概要」のインストール手順の概要に目を通します。マスターサーバーのインストール前に必要な作業があれば、それらを済ませます。

マスターサーバーをほかのマシンからインストールする場合は、Virtual Network Computing (VNC) ソフトウェアを使用してそのサーバーにアクセスしてください。Terminal Client Services を使用してサーバーにアクセスすると、インストールは失敗します。

MSI パッケージが保存されているフォルダに書き込みが行えるか (書き込み許可があるか) 確認してください。

手順 **1. MSI** パッケージにアクセスします。

- CD からインストールする場合は、Sun N1 Service Provisioning System 5.1: Windows CD を挿入します。
- ダウンロードしたイメージからインストールする場合は、そのイメージが保存されているディレクトリに移動します。

2. Windows の「ファイル マネージャ」または「コマンド プロンプト」を使用し、**CD** 上の **`windows`** ディレクトリ、またはダウンロードしたイメージを保存したディレクトリにアクセスします。

3. 「ファイル マネージャ」のインストールを開始します。

- 「ファイル マネージャ」を使用している場合は、 `cr_ms_win32_5.1.msi` ファイルをダブルクリックします。
- 「コマンド プロンプト」を使用している場合は、プロンプトにインストールファイルの名前を入力します。

```
E:\N1GSPS5.1\windows> cr_ms_win32_5.1.msi [ALLOWFORWARDVERSION=true]
```

`ALLOWFORWARDVERSION=true` オプションを指定すると、N1 Service Provisioning System 5.1 がそのオペレーティングシステムでサポートしている最高バージョンを超える OS バージョンに N1 Service Provisioning System 5.1 マスターサーバーをインストールできます。`ALLOWFORWARDVERSION=true` オプションを使用すると、インストールプログラムはマスターサーバーがインストールされるオペレーティングシステムがサポートされているかどうかを確認しません。サポートされていないオペレーティングシステムでの N1 Service Provisioning System 5.1 の使用については、Sun サービスは標準サポートを提供しません。



注意 – サポートされていないオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールすると、不確定な動作や予期しない動作が発生することがあります。サポートされていないオペレーティングシステムに対する N1 Service Provisioning System 5.1 のインストールは、テスト目的以外では行わないでください。運用環境では、サポートされていなオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールして使用することは避けてください。

4. インストール中に構成についての質問が表示される場合は、その質問に答えてください。

   構成に関する質問がいくつか表示され、続いて「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ってプログラムファイルのインストールが行われます。インストールが終了すると、マシンを再起動するようにメッセージが表示されます。

6. マシンを再起動してインストールを終了します。

   N1 Service Provisioning System 5.1 のインストールを完全に終了するためには、マシンを再起動する必要があります。

7. サーバーにログインします。

   ログインが終わると、「Welcome」画面が表示されます。

8. 「**Next**」をクリックし、インストールを終了します。

注 – 「コマンド プロンプト」ウィンドウが表示され、インストールプログラムによってコマンドがいくつか実行されます。コマンドのいくつかは、実行に数分かかる可能性があります。「コマンド プロンプト」ウィンドウを閉じたり、処理をキャンセルしたりしないでください。数分後、処理は自動的に終了します。

---

9. インストールプログラムを終了するには「**Finish**」をクリックします。

   以上の操作で、マスターサーバーがインストールされます。マスターサーバーにアクセスするには、Web ブラウザと、インストールで指定したブラウザインタフェースアドレスを使用します。

10. **(**省略可能**)** データベースを最適化する定期タスクを作成します。

    データベースのパフォーマンスを最適化するには、`vacuumdb` ユーティリティを毎日実行する定期タスク (スケジュール化されたタスク) を作成します。定期タスクを作成するには、59 ページの「データベースを最適化するために定期タスクを作成する」の説明に従ってください。

▼ データベースを最適化するために定期タスクを作成する

手順 1. **Windows** の「**Scheduled Tasks**」フォルダを開きます。

   「タスク」フォルダは、「スタート」メニューをクリックして、「プログラム」、「アクセサリ」、「システム ツール」、「タスク」の順にクリックして開くことができます。

2. 新しいタスクを作成するには、フォルダ内でマウスの右ボタンをクリックし、「新規」、「タスク」の順に選択します。

3. タスクに名前を付けます。

4. タスクを編集するには、そのタスクをダブルクリックします。

5. 「実行するファイル名」フィールドに、次のコマンドを **1** 行で入力します。

   `bash -c "/cygdrive/`*`c/Program\ Files/N1\ Service\ Provisioning\ System/5.1`***`/server/bin/roxdbcmd vacuumdb -h localhost -a -z"`**

   c/Program\ Files/N1\ Service\ Provisioning\ System/5.1 は、マスターサーバーをインストールしたディレクトリです。

6. 「スケジュール」タブで、日に **1** 度実行するタスクを設定します。

リモートエージェント、ローカルディストリビュータ、CLI Client のインストール

▼ Windows にリモートエージェント、ローカルディストリビュータ、CLI Client をインストールする

始める前に　17 ページの「N1 Service Provisioning System 5.1 のインストール – 手順の概要」のインストール手順の概要に目を通します。マスターサーバーのインストール前に必要な作業があれば、それらを済ませます。

MSI パッケージが保存されているフォルダに書き込みが行えるか (書き込み許可があるか) 確認してください。

手順　1. インストール **MSI** パッケージにアクセスします。

- CD からインストールする場合は、Sun N1 Service Provisioning System 5.1: Windows CD を挿入します。
- ダウンロードしたイメージからインストールする場合は、そのイメージが保存されているディレクトリに移動します。

2. **Windows** の「ファイル マネージャ」または「コマンド プロンプト」を使用し、**CD** 上の **`windows`** ディレクトリ、またはダウンロードしたイメージを保存したディレクトリにアクセスします。

3. インストールするアプリケーションのインストール操作を開始します。

   - 「ファイル マネージャ」を使用している場合は、 `cr_app_win32_5.1.msi` ファイルをダブルクリックします。
   - 「コマンド プロンプト」を使用している場合は、プロンプトにインストールファイルの名前を入力します。

     E:\N1GSPS5.1\windows> **cr_app_win32_5.1.msi [ALLOWFORWARDVERSION=true]**

     app には、次に示す値の 1 つを指定します。

     - ra – リモートエージェントをインストールする
     - ld – ローカルディストリビュータをインストールする
     - cli – CLI Client をインストールする

     `ALLOWFORWARDVERSION=true` オプションを指定すると、N1 Service Provisioning System 5.1 がそのオペレーティングシステムでサポートしている最高バージョンを超える OS バージョンに N1 Service Provisioning System 5.1 アプリケーションをインストールできます。`ALLOWFORWARDVERSION=true` オプションを使用すると、インストールプログラムはアプリケーションがインストールされるオペレーティングシステムがサポートされているかどうかを確認しません。サポートされていないオペレーティングシステムでの N1 Service Provisioning System 5.1 の使用については、Sun サービスは標準サポートを提供しません。



注意 – サポートされていないオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールすると、不確定な動作や予期しない動作が発生することがあります。サポートされていないオペレーティングシステムに対する N1 Service Provisioning System 5.1 のインストールは、テスト目的以外では行わないでください。運用環境では、サポートされていないオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールして使用することは避けてください。

---

4. インストール中に構成についての質問が表示される場合は、その質問に答えてください。

   構成に関する質問がいくつか表示され、続いて「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ってプログラムファイルのインストールが行われます。

6. インストールプログラムを終了するには「**Finish**」をクリックします。

Windows へのリモートエージェントの非対話方式インストール

リモートエージェントは、構成を設定する変数をコマンド行で指定してインストールできます。この非対話方式インストールは、Windows Installer Service の一部でインストールされる `msiexec` コマンドを使用して行えます。

▼ Windows にリモートエージェントを非対話方式でインストールする

始める前に　MSI パッケージが保存されているフォルダに書き込みが行えるか (書き込み許可があるか) 確認してください。

手順

1. リモートエージェントをインストールするマシンで、「コマンド プロンプト」ウィンドウを開きます。
2. **Sun N1 Service Provisioning System 5.1: Windows CD** を挿入します。
3. ソフトウェア **CD** 上で、あるいはダウンロードしたイメージを保存したディレクトリから、**MSI** パッケージが置かれているディレクトリに移動します。
4. リモートエージェントをインストールするマシンに、インストール **MSI** パッケージをコピーします。

   ```
   % copy cr_ra_win32_5.1.msi RA-machine\
   ```

   RA-machine は、リモートエージェントをインストールするマシン上のディレクトリです。
5. インストールを開始します。

   ```
   C:RA-machine\>  msiexec /i cr_ra_win32_5.1.msi /qn
   RA_PARENT_CONNECTION=value CR_RA_CTYPE_CONFIRM=value VARIABLE=value [ALLOWFORWARDVERSION=true]
   ```

   変数は、必要なだけいくつでも含めることができます。スペースを含む変数値 (ディレクトリ名など) は、引用符で囲む必要があります。非対話方式のインストールプログラムで指定できる変数と値については、表 5–1 を参照してください。値を指定しない変数についてはデフォルト値を使用してリモートエージェントをインストールします。



注意 – RA_PARENT_CONNECTION 変数の値は設定する必要があります。この値が設定されていない場合、インストールスクリプトは中止されます。raw や認証なしの SSL 暗号化などのセキュリティー保護されていない接続タイプを使用することを選択した場合は、CR_RA_CTYPE_CONFIRM 変数の値を true に設定する必要があります。

ALLOWFORWARDVERSION=true オプションを指定すると、N1 Service Provisioning System 5.1 がそのオペレーティングシステムでサポートしている最高バージョンを超える OS バージョンに N1 Service Provisioning System 5.1 リモートエージェントをインストールできます。ALLOWFORWARDVERSION=true オプションを使用すると、インストールプログラムはリモートエージェントがインストールされるオペレーティングシステムがサポートされているかどうかを確認しません。サポートされていないオペレーティングシステムでの N1 Service Provisioning System 5.1 の使用については、Sun サービスは標準サポートを提供しません。



注意 – サポートされていないオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールすると、不確定な動作や予期しない動作が発生することがあります。サポートされていないオペレーティングシステムに対する N1 Service Provisioning System 5.1 のインストールは、テスト目的以外では行わないでください。運用環境では、サポートされていなオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールして使用することは避けてください。

例 5–1 Windows へのリモートエージェントの非対話方式インストール

次に、リモートエージェントを Windows に非対話方式でインストールするコマンドの例を示します。

```
C:\> msiexec /i cr_ra_win32_5.1.msi /qn
INSTALLDIR="C:\Program Files\N1 Service Provisioning System\"
RA_PARENT_CONNECTION=false CR_RA_CTYPE_CONFIRM=true
```

Windows へのリモートエージェントのリモートインストール

リモートエージェント MSI パッケージは、非対話モードでリモートインストールをスムーズに行うためのツールです。インストールは、Windows Scripting Host が使用する .wsh スクリプトを使用して行われます。このスクリプトファイルには、次の処理を行う VB スクリプトコードが含まれます。

- リモートシステムの WMI DCOM インタフェースに接続する
- WMI を使用し、ターゲットサーバー上に一時的な Windows ファイルシェアを作成する
- ローカルロケーションからターゲットサーバー上のファイルシェアに `cr_ra_win32_5.1.msi` をコピーする
- WMI をリモートマシンで使用し、ターゲットマシン上のサイレント MSI を実行する

▼ Windows にリモートエージェントをリモートインストールする

始める前に　MSI パッケージが保存されているフォルダに書き込みが行えるか (書き込み許可があるか) 確認してください。

手順　**1.** マスターサーバーマシンで、「コマンド プロンプト」ウィンドウを開きます。

2. **MSI** パッケージにアクセスします。

- CD からインストールする場合は、Sun N1 Service Provisioning System 5.1: Windows CD を挿入します。
- ダウンロードしたイメージからインストールする場合は、そのイメージが保存されているディレクトリに移動します。

3. ソフトウェア **CD** 上で、あるいはダウンロードしたイメージを保存したディレクトリから、**MSI** パッケージが置かれているディレクトリに移動します。

4. **MSI** パッケージをマスターサーバーにコピーします。

C:\> **copy cr_ra_win32_5.1.msi** *MS-machine*\

MS-machine は、マスターサーバーマシン上のディレクトリです。

5. マスターサーバーのホームディレクトリに移動します。

C:\> **cd** *N1SPS5.1-home*\server\bin\WinInstaller

N1SPS5.1-home は、マスターサーバーをインストールしたディレクトリです。

6. インストールを開始します。

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

`ALLOWFORWARDVERSION=true` オプションを指定すると、N1 Service Provisioning System 5.1 がそのオペレーティングシステムでサポートしている最高バージョンを超える OS バージョンに N1 Service Provisioning System 5.1 リモートエージェントをインストールできます。`ALLOWFORWARDVERSION=true` オプ

ションを使用すると、インストールプログラムはリモートエージェントがインストールされるオペレーティングシステムがサポートされているかどうかを確認しません。サポートされていないオペレーティングシステムでの N1 Service Provisioning System 5.1 の使用については、Sun サービスは標準サポートを提供しません。



注意 – サポートされていないオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールすると、不確定な動作や予期しない動作が発生することがあります。サポートされていないオペレーティングシステムに対する N1 Service Provisioning System 5.1 のインストールは、テスト目的以外では行わないでください。運用環境では、サポートされていなオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールして使用することは避けてください。

Hostname は、リモートエージェントのインストール先となるマシンのホスト名です。

コマンド行で次に示す *parameters* のどれにも値を指定しないと、リモートエージェントはデフォルト値でインストールされます。

リモートエージェントの非対話方式インストールプログラムは、次の表に示すパラメータを受け入れます。

パラメータ	説明	標準
`-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 に接続するためのパスワード	なし
変数	表 5–1 に示された `cscript WinInstaller.wsf` コマンドの Windows 変数。変数と値はすべて、引用符で囲まれた 1 つの文字列内に含める必要があります。 注意 – `RA_PARENT_CONNECTION` 変数の値は設定する必要があります。この値が設定されていない場合、インストールスクリプトは中止されます。`raw` や認証なしの SSL 暗号化などのセキュリティー保護されていない接続タイプを使用することを選択した場合は、`CR_RA_CTYPE_CONFIRM` 変数の値を `true` に設定する必要があります。	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	標準
-shareLocation	一時的な Windows ファイルシェアを作成する、ターゲットマシン上の既存のディレクトリ。ファイルシェアディレクトリは、MSI パッケージのサイズ以上の容量がなければなりません。	C:\WINNT\Temp

正常なインストールが行われた場合の終了コードは 0、失敗時の終了コードは 1 です。

例 5–2 Windows へのリモートエージェントのリモートインストール

次に、Windows に リモートマシンからリモートエージェントをインストールするコマンド例を示します。

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

リモートエージェントの変数値

非対話方式で行われるリモートエージェントのリモートインストールプログラムは、次の変数を受け入れます。

表 5–1 リモートエージェントの変数値

変数名	説明	標準	値
INSTALLDIR	リモートエージェントのインストール先であるディレクトリを指定します。	C:\Program Files\N1 Service Provisioning System	任意の有効なディレクトリ
REMOTE_AGENT_HOSTNAME	リモートエージェントのインストール先であるマシンのホスト名または IP アドレスを指定します。	Windows 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	このリモートエージェントに使用する IP ポート番号を指定します。	1131	任意の有効なポート番号

表 **5–1** リモートエージェントの変数値　*(続き)*

変数名	説明	標準	値
RA_PARENT_CONNECTION	暗号化されない (raw) 接続または SSL 接続を使用して親アプリケーションがこのリモートエージェントに接続することを指定します。	なし	true は SSL を使用することを指定します。false は raw を使用することを指定します。
RA_SSL_CIPHER	SSL を選択した場合は、使用する SSL 暗号群の種類を選択します。	1	0 は、認証を伴う SSL 暗号化を使用することを指定します。1 は、認証を伴わない SSL 暗号化を使用することを指定します。
CR_RA_CTYPE_CONFIRM	使用する暗号化の種類を確認します。	なし	raw または認証なしの SSL 暗号化を選択した場合は、true です。認証付きの SSL 暗号化を選択した場合は、false です。
RA_SERVICE_USERNAME RA_SERVICE_PASSWORD	どのユーザーアカウントでリモートエージェントを実行するかを指定します。	システムユーザー	ローカルユーザー名は、先頭に .\ を付けます。 これらの変数を定義する場合は、RA_SERVICE_CONTROL を other に設定する必要があります。
RA_SERVICE_AUTOSTART	サーバーの再起動時にリモートエージェントを自動的に起動するかどうかを指定します。この変数は、インストール時にリモートエージェントを起動するかどうかも決定します。	1	1 は自動的に起動することを指定します。0 は自動的に起動しないことを指定します。

Windows システムでのアプリケーションの起動

Windows サーバーでは、マスターサーバー、ローカルディストリビュータ、およびリモートエージェントを Services Panel で起動します。CLI Client は、「コマンド プロンプト」ウィンドウから起動します。

マスターサーバー、ローカルディストリビュータ、またはリモートエージェントを起動するには、「スタート」メニューをクリックし、「プログラム」、「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、このアプリケーションの名前を見つけ、起動します。

表 **5–2** Windows マスターサーバー、ローカルディストリビュータ、およびリモートエージェント のために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
マスターサーバー	N1 Service Provisioning System 5.1 Server
	N1 Service Provisioning System 5.1 PostgresSQL Server
	N1 Service Provisioning System 5.1 IPC Daemon
	N1 Service Provisioning System 5.1 Database Preparer
ローカルディストリビュータ	N1 Service Provisioning System 5.1 Distributor
リモートエージェント	N1 Service Provisioning System 5.1 Agent

Windows サーバーで CLI Client を起動するには、「コマンド プロンプト」で次に示すコマンドの 1 つを入力します。*N1SPS5.1-home* は、アプリケーションのホームディレクトリです。

表 **5–3** Windows CLI Client の起動コマンド

アプリケーション	コマンドのパス	起動用のコマンド
CLI Client	*N1SPS5.1-home*\cli\bin\	cr_cli.cmd *CLI-command*
Jython バージョンの CLI Client	*N1SPS5.1-home*\cli\bin\	cr_clij.cmd *CLI-command*

第 6 章

HTTPS を使用するための N1 Service Provisioning System 5.1 の構成

N1 Service Provisioning System インストールプログラムによって、マスターサーバーへの接続に HTTP または HTTPS を使用するために ブラウザインタフェース を選択するようにメッセージが表示されます。HTTPS を選択すると、キーストアファイルとキーストアパスワードを入力するようにメッセージが表示されます。キーストアファイルとキーストアパスワードは、2 とおりの方法で指定できます。

- インストールの前にキーストアファイルとキーストアパスワードを作成する方法: プロビジョニングシステムをインストールする前にキーストアファイルとキーストアパスワードを作成する場合は、インストール中にキーストアファイルとキーストアパスワードの情報を指定できます。プロビジョニングシステムをインストールする前にキーストアファイルとキーストアパスワードを作成する方法については、70 ページの「HTTPS 接続のキーストアファイルとキーストアパスワードの作成」を参照してください。

  注 – キーストアパスワードが有効かどうかは確認されません。無効なパスワードを入力した場合、マスターサーバーを起動できないことがあります。

- インストール後にキーストアファイルとキーストアパスワードを作成する方法: インストールする前にキーストアファイルとキーストアパスワードを作成しない場合は、インストール中にキーストアファイルとキーストアパスワードの情報を指定できません。この結果、インストールスクリプトによって、空のキーストアファイルが *N1SPS5.1-home*`/server/tomcat` ディレクトリに作成されます。空のキーストアファイルとキーストアパスワード情報は、マスターサーバーを起動する前に手動で置き換える必要があります。手順については、72 ページの「インストール後の HTTPS の構成」を参照してください。

インストール中に HTTP を選択すると、 HTTPS を使用するようにマスターサーバーを手動で構成できます。手順については、74 ページの「インストール中に HTTP を選択したあとで行う HTTPS の構成」を参照してください。

HTTPS を使用するようにプロビジョニングシステムを構成した場合でも、プロビジョニングシステムを手動で再構成して HTTP を使用できます。手順については、76 ページの「HTTP への復帰」を参照してください。

HTTPS 接続のキーストアファイルとキーストアパスワードの作成

HTTPS では SSL 証明書が必要です。SSL 証明書を生成すると、ブラウザインタフェース がマスターサーバーに接続するときに使用されるキーストアファイルとキーストアパスワードが作成されます。

認証局に証明書に署名してもらうか、自己署名付き証明書を使用するかを選択できます。認証局によって署名された証明書は、ブラウザによって信頼されます。このため、ブラウザはユーザーがマスターサーバー上のブラウザインタフェースに接続する場合に警告を出しません。一般に、認証局は証明書の署名に費用を請求します。自己署名付き証明書は認証局による署名を待つ必要がないため、生成後すぐに使用できます。ただし、ブラウザは自己署名付き証明書を信用しないため、ユーザーがマスターサーバーに接続するたびに警告を発します。

▼ SSL 証明書の生成方法

ブラウザインタフェースに SSL を使用させるには、初めに SSL 証明書を生成する必要があります。キーストアファイルとキーストアパスワードは、SSL 証明書を生成中に作成します。

始める前に　キーストアファイルとキーストアパスワードを作成するには、`keytool` を使用します。`keytool` は、JRE で使用可能なセキュリティーツールです。`keytool` がインストールされていない場合は、HTTPS を使用するようにプロビジョニングシステムを構成する前に `keytool` をインストールする必要があります。JRE は、N1 Service Provisioning System によってインストールされます。プロビジョニングシステムをインストールしたあとに HTTPS を構成する場合、`keytool` はシステム上にインストールされています。

手順　**1. JRE** をインストールしたディレクトリに移動します。

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

JAVA-HOME には、JRE をインストールしたディレクトリを指定します。N1 Service Provisioning System 5.1 でインストールした場合、JRE は *N1SPS5.1-home*`/common/jre/bin` ディレクトリにインストールされます。

2. 証明書を生成します。

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

/keystore-location に生成されたキーを格納するキーストアファイルの場所とファイル名を設定します。

password にキーストアパスワードとして使用する何らかのパスワードを設定します。

3. メッセージに従って操作を進めます。

組織名には区切り文字を入れないようにしていください。区切り文字が入ると、Java Certificate ツールは要求の生成に失敗します。共通名 (Common Name: CN) は、ドメイン名、URI のコンポーネントを含む完全修飾のホスト名である必要があります。

▼ SSL 証明書の署名を取得する

認証局によって署名された証明書を使用する場合は、この手順に従って証明書を認証局に提出し、署名を依頼してください。

手順 1. 証明書要求を生成します。

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

/keystore-location には、生成したキーを格納した場所とファイル名を指定します。

2. この証明書要求を認証局へ送信します。

認証局の指示に従います。認証局から、証明書応答 (Certificate Reply) が返送されてきます。

3. 証明書応答をファイルに保存します。

4. 証明書応答を確認します。

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

certificate-reply-file には、認証局から受け取った証明書応答のファイル名を指定します。

5. 証明書応答ファイルを **keystore** ファイルにインポートします。

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

/keystore-location には、生成したキーを格納した場所とファイル名を指定します。*certificate-reply-file* には、認証局から受け取った証明書応答のファイル名を指定します。

6. インポートされた証明書応答を確認します。

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

/keystore-location には、生成したキーを格納した場所とファイル名を指定します。

インストール後の HTTPS の構成

N1 Service Provisioning System インストールプログラムによって、マスターサーバーの Web インタフェースに接続するために ブラウザインタフェース が使用する HTTPS または HTTP を選択するようメッセージが表示されます。インストール中に HTTPS を選択したもののキーストアファイルとキーストアパスワードをインストール中に指定しなかった場合は、キーストアファイルをプロビジョニングシステムのキーストアディレクトリに手動でコピーする必要があります。また、符号化されたキーストアパスワードを作成して、`server.xml` ファイル内で、そのパスワードを構成することも必要です。

▼ キーストアファイルをコピーする

キーストアファイルを *N1SPS5.1-MasterServer-home*`/server/tomcat/keystore` ファイルにコピーするには、次の操作を行います。

始める前に　SSL 証明書を生成し、必要に応じて認証局から署名を取得します。手順については、70 ページの「HTTPS 接続のキーストアファイルとキーストアパスワードの作成」を参照してください。

手順

1. キーストアファイルをマスターサーバーのホームディレクトリに移動させます。

```
%mv /keystore-location  N1SPS5.1-MasterServer-home/server/tomcat/keystore
```

/keystore-location には、生成したキーを格納した場所とファイル名を指定します。*N1SPS5.1-MasterServer-home* は、マスターサーバーをインストールしたディレクトリです。

2. キーストアファイルを移動したディレクトリに移動します。

```
% cd N1SPS5.1-MasterServer-home/server/tomcat/
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

3. キーストアファイルに所有権とアクセス権を設定します。

```
%chmod 600 keystore
```

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

MS_user には、マスターサーバーアプリケーションを所有するユーザーを指定します。*MS_group* には、マスターサーバーアプリケーションを所有するグループを指定します。

▼ 符号化されたキーストアパスワードを作成および構成する

SSL 証明書を生成するときには、キーストアファイルを作成し、キーストアパスワードを指定します。プロビジョニングシステムは、このキーストアパスワードを `server.xml` ファイルに格納する必要があります。セキュリティー向上のため、プロビジョニングシステムでは、パスワードの符号化バージョンを `server.xml` ファイルに格納する必要があります。

手順 **1.** **`crkeys`** コマンドを含むマスターサーバー上のディレクトリに移動します。

```
%cd N1SPS5.1-MasterServer-home/server/bin
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

2. キーストアファイルを作成したときに作成したパスワードの符号化バージョンを作成します。

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

password は、SSL 証明書の生成中に作成したパスワードです。

`crkeys` ツールは、符号化されたパスワードを画面に出力します。符号化されたパスワードを控えてください。

3. **Tomcat** 構成ファイルが置かれているディレクトリに移動します。

```
% cd /N1SPS5.1-MasterServer-home/server/tomcat/conf
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

4. **Factory** 要素を編集して、符号化されたパスワードを含めます。

```
<Factory className="com.raplix.rolloutexpress.ui.web.EncodedPasswordSSLFactory"
          clientAuth="false" protocol="TLS"
    keystoreFile="/opt/SUNWn1sps/N1_Service_Provisioning_System_5.1/server/tomcat/keystore"
 keystorePass="ADD_ENCODED_PASSWORD_HERE"/>
```

`ADD_ENCODED_PASSWORD_HERE` を符号化されたパスワードに変更します。

インストール中に HTTP を選択したあとで行う HTTPS の構成

N1 Service Provisioning System インストールプログラムによって、マスターサーバーに接続するために ブラウザインタフェース が使用する HTTPS または HTTP を選択するようメッセージが表示されます。インストール中に HTTP を選択した場合でも、N1 Service Provisioning System を手動で再構成して HTTPS を使用できます。

▼ マスターサーバーブラウザインタフェースからマスターサーバーの Web インタフェースまでの HTTPS 接続を使用可能にする

次の操作を行い、マスターサーバーへの接続に HTTP ではなく HTTPS を使用するようにブラウザインタフェースを構成します。

始める前に　SSL 証明書を生成し、必要に応じて認証局から署名を取得します。手順については、70 ページの「HTTPS 接続のキーストアファイルとキーストアパスワードの作成」を参照してください。

手順

1. マスターサーバーを停止します。

   ```
   % N1SPS5.1-MasterServer-home/server/bin/cr_server stop
   ```

   N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

2. キーストアファイルをマスターサーバーのホームディレクトリに移動させます。

   ```
   %mv /keystore-location N1SPS5.1-MasterServer-home/server/tomcat/keystore
   ```

   /keystore-location には、生成したキーを格納した場所とファイル名を指定します。*N1SPS5.1-MasterServer-home* は、マスターサーバーをインストールしたディレクトリです。

3. キーストアファイルを移したディレクトリに移動します。

   ```
   % cd N1SPS5.1-MasterServer-home/server/tomcat/
   ```

   N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

4. キーストアファイルに所有権とアクセス権を設定します。

   ```
   %chmod 600 keystore
   ```

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

MS_user には、マスターサーバーアプリケーションを所有するユーザーを指定します。*MS_group* には、マスターサーバーアプリケーションを所有するグループを指定します。*/keystore-location* には、生成したキーを格納したファイル名を指定します。

5. **Tomcat** 構成ファイルが置かれているディレクトリに移動します。

```
% cd /N1SPS5.1-MasterServer-home/server/tomcat/conf
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

6. **`server.xml`** ファイルで、次の行のコメントを解除します。**XML comments begin with `<!--` and end with `-->`.**

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

7. **Factory** 要素を次のように編集します。

```
<Factory className="com.raplix.rolloutexpress.ui.web.EncodedPasswordSSLFactory"
           clientAuth="false" protocol="TLS"
    keystoreFile="N1SPS5.1-MasterServer-home/server/tomcat/keystore"
 keystorePass="password"/>
```

N1SPS5.1-MasterServer-home は、マスターサーバーをインストールしたディレクトリです。*password* は、符号化したパスワードです。

▼ SSL を使用してマスターサーバーブラウザインタフェースに接続するようにユーザーに要求する

SSL を使用するようにマスターサーバーのブラウザインタフェースを構成したあとで、ユーザーが N1 Service Provisioning System マスターサーバー上のブラウザインタフェースに接続する際に SSL を使用するように構成できます。

手順

1. **Tomcat** の **`web.xml`** ファイルをセキュリティー保護された **`web.xml`** ファイルに置き換えます。

```
% cd /N1SPS5.1-MasterServer-home/server/webapp/WEB-INF
% cp web.xml.secure web.xml
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

2. マスターサーバーを再起動します。

```
% N1SPS5.1-MasterServer-home/server/bin/cr_server start
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

HTTP への復帰

HTTPS を使用するためにプロビジョニングシステムを構成した場合でも、プロビジョニングシステムを手動で再構成して HTTP を使用できます。

▼ HTTP に戻る

手順 **1.** マスターサーバーを停止します。

```
% N1SPS5.1-MasterServer-home/server/bin/cr_server stop
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

2. 元の構成に戻るには、セキュリティー保護された **web.xml** ファイルをデフォルトの **web.xml** ファイルに置き換えます。

```
% cd /N1SPS5.1-MasterServer-home/server/webapp/WEB-INF
% cp web.xml.default web.xml
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

3. マスターサーバーを再起動します。

```
% N1SPS5.1-MasterServer-home/server/bin/cr_server start
```

N1SPS5.1-MasterServer-home には、マスターサーバーをインストールしたディレクトリを指定します。

第 7 章

Secure Shell を使用するための N1 Service Provisioning System 5.1 の構成

この章では、Secure Shell (SSH) を使用して通信を行うように N1 Service Provisioning System 5.1 を構成する方法について説明しています。

N1 Service Provisioning System 5.1 は、OpenSSH 2.0 を明示的にサポートしています。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。このソフトウェアは、ほかの SSH バージョンをサポートするようにも構成できます。

ネットワーク内の各サーバーで同じ SSH 実装を使用すると、鍵の互換性が保たれ、サーバー通信が正しく行われます。ネットワーク上の各サーバーに異なる SSH 実装を使用することもできますが、それらの実装に互換性があることと、相互運用が可能であるかを検証する必要があります。

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。そのほかのバージョンのSSH を使用する場合は、そのバージョンの SSH の付属文書で、OpenSSH 2.0 で使用されているコマンドに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については、93 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。

- 78 ページの「SSH の概要と使用条件」
- 81 ページの「SSH の構成 – 作業概要」
- 82 ページの「鍵の準備」
- 85 ページの「マスターサーバーでの接続の設定とテスト」
- 87 ページの「アプリケーションに合わせて SSH を構成する」
- 92 ページの「SSH の上級パラメータとコマンドのリファレンス」

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/プロトコルです。SSH は、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 公開鍵暗号を使用して接続と認証を管理します。SSH は、Telnet やほかのシェルベース通信方式よりも安全性が高く、Web サーバーなどのリモートサーバーの管理に使用されます。

ほかの接続タイプと異なり、2 つの N1 Service Provisioning System 5.1 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がありません。ダウンストリームのアプリケーションは、必要な場合にアップストリームのアプリケーションによって自動的に起動されます。ダウンストリームのアプリケーションは、使用されている間は継続して動作しますが、一定の時間 (この時間は設定可能) 使用されないと自動的に停止します。

SSH 接続では、ダウンストリームのアプリケーションの手動起動は行わないでください。たとえば、SSH を使用してリモートエージェントに接続するようにローカルディストリビュータを設定した場合、リモートエージェントを手動では起動しないでください。ローカルディストリビュータは、必要時にリモートエージェントを自動的に起動します。リモートエージェントは、使用されている間は継続して作動します。リモートエージェントが一定時間使用されなかった場合、ローカルディストリビュータは自動的にリモートエージェントを停止します。

ssh-agent または空のパスワード鍵

SSH は構成時に、ssh-agent を使用するか、空のパスワード鍵を使用するかを選択できます。空のパスワード鍵を使用すると、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。この結果、この非公開鍵はパスワードなしでアクセスできます。公開鍵を信頼するほかのマシンとの通信に SSH を使用する場合、パスワードの入力は求められません。ssh-agent を使用する場合は、生成される非公開鍵はセキュリティー保護されたパスワードと共に格納され、安全なメディアに格納されます。ほかのマシンと通信を行うときは、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、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内に格納されます。

ssh-agent を使用する場合、非公開鍵はマスターサーバー上でしか実行されない ssh-agent によって格納されます。ネットワーク上のほかのマシンには公開鍵が配布されます。認証が必要な SSH アプリケーションは、ssh-agent と通信して認証を行います。認証のためにマスターサーバーで実行されている ssh-agent に対してローカルディストリビュータからプロキシ設定が行えるように中間的な SSH 接続するには、ssh-agent の転送機能を有効にする必要があります。ローカルディストリビュータは、ssh-agent の転送機能を使用してダウンストリームのローカルディストリビュータとリモートエージェントに認証情報を渡します。この方法は、前述の方法よりも安全です。また、N1 Service Provisioning System 5.1 で ssh-agent を使用するように構成した方が、空のパスワードを設定するよりも簡単です。

空のパスワードを使用すると、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上に格納されます。非公開鍵は、SSH 通信を開始するすべてのマシン上に置く必要があります。N1 Service Provisioning System 5.1 の場合、SSH を使用してダウンストリームのアプリケーションに接続しているマスターサーバーとローカルディストリビュータはすべて、非公開鍵を持っていなければなりません。この方法は安全性に劣ります。

SSH の要件

N1 Service Provisioning System 5.1 には、次の SSH 機能が必要です。

- `ssh` によるリモートコマンド呼び出し
- 公開 / 非公開鍵ペアによる認証
- `BatchMode yes` インタラクション (オペレータの介入なしで `ssh` コマンドを呼び出す機能) のサポート

`ssh-agent` を使用する場合は、次の SSH 機能が必要です。

- `ssh-agent` のサポート
- SSH における `ssh-agent` 転送機能のサポート。Open SSH での `-A` オプションの使用

SSH 接続を行うようにマシンを構成する場合は、次の機能が役に立ちます。ただし、これらは必須機能ではありません。

- リモートコマンドを呼び出す時の `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ーを実現する RSA 鍵の生成。OpenSSH で `- t rsa` を使用します。

次のチェックリストを使用し、SSH の実装が N1 Service Provisioning System 5.1 の要件を満たしているか確認してください。

- `ssh-keygen` コマンドは、SSH 呼び出しの認証に使用できる 公開 / 非公開鍵ペアを生成する必要があります。
- 追加情報なしでホスト鍵の交換やパスワードの取得が可能な指定のホストで、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る必要があります。

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

- パスワード付きで生成された非公開鍵を使用してアップロードされた現在のホストで `ssh-agent` を実行し、現在のホストから host1、host2、host3 へとホップしたとします。このあと、追加情報なしでホスト鍵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る必要があります。

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

- ssh コマンドは、その標準の入出力とエラーストリームを、リモートマシンで実行されているコマンドに対して正しくパイプできる必要があります。
- ssh-add コマンドは、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る必要があります。

その他の SSH セキュリティー

SSH を使用してリモートエージェントを起動する場合、リモートエージェントは jexec ラッパーを使用して Java 仮想マシンを起動します。この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、setuid ビットセットを持っています。このファイルはリモートエージェントのインストールに使用したユーザーと同じグループ ID を持ち、このグループに実行許可を与えます。このファイルは、リモートエージェントのインストールに使用したユーザーアカウントが所有する protect というディレクトリに保存されます。このファイルの実行許可が与えられるのは、リモートエージェントを所有するユーザーだけです。ほかのユーザーは jexec ラッパーを実行できません。

どのような場合でも 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対策を施す必要があります。

jexec のセキュリティーは、次の変更を加えることによって強化できます。

- JVM 実行ファイル (通常、シェルスクリプト) はアプリケーションを所有するスーパーユーザー (root) またはユーザーによって所有されるようにし、ほかのユーザーやグループには書き込み許可を与えないないようにします。N1 Service Provisioning System 5.1 で JRE をインストールする場合は、*N1SPS5.1-home* /common/jre 内の全ファイルがアプリケーションを所有するユーザーによって所有されるようにし、ほかのどんなユーザーやグループにも書き込みアクセス権を与えないようにします。
- 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 を使用したログインにだけ許可します。SSH を使用してログインする場合は、公開鍵認証だけを許可します。ほかのどんなユーザーまたはグループにも、*/N1SPS5.1-home* / .ssh ディレクトリに対するいかなるアクセス権も付与しません。
- etc/sshd_config ファイルにパスワード認証を無効にする次の行を含めることにより、公開鍵認証だけを許可するように SSH サーバーを構成します。

  ```
  PasswordAuthentication no
  ```

- etc/sshd_config ファイルに、RhostsRSAAuthentication を含む行が存在しないようにします (デフォルトではこの認証が許可されていないため)。さらに、RSAAuthentication という語が存在する場合は、この値を yes (デフォルト) に設定する必要があります。
- リモートエージェントのセキュリティーをさらに強化する場合は、*/N1SPS5.1-home/* .ssh/authorized_keys2 ファイルを編集し、マスターサーバーの公開鍵が含まれる行の前に次のテキストを追加します。

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

詳細については、sshd(1M) のマニュアルページを参照してください。

SSH の構成 – 作業概要

以下に、SSH を使用するように N1 Service Provisioning System 5.1 を構成するために必要な作業の概要を示します。

1. 空のパスワード付きの鍵を使用するか、ssh-agent を使用するかを決定します。

   SSH セキュリティーレベルの選択についての詳細は、78 ページの「ssh-agent または空のパスワード鍵」を参照してください。

2. SSH 接続を開始するアプリケーション上で鍵を生成します。

   82 ページの「鍵ペアを作成する」を参照してください。

3. 生成した鍵をローカルディストリビュータとリモートエージェントにコピーします。

   空のパスワードが付いた鍵を使用するか、ssh-agent を使用するかにもとづいて、次の一覧から適切な作業を選択してください。

   - 83 ページの「ssh-agent の鍵を設定する」
   - 83 ページの「空のパスワードファイルに鍵を設定する (1 組の鍵ペアを使用する場合)」
   - 84 ページの「空のパスワードファイルに鍵を設定する (複数の鍵ペアを使用する場合)」

4. SSH 接続を設定します。続いて、接続テストを行います 。このテストはマスターサーバーを起動する前に行なってください。

   85 ページの「マスターサーバーでの接続の設定とテスト」を参照してください。

5. SSH を使用するようにローカルディストリビュータとリモートエージェントを設定します。

   88 ページの「ローカルディストリビュータとリモートエージェントで使用するように SSH を構成する」を参照してください。

6. (省略可能) CLI Client が存在する場合は、SSH を使用するようにそれらのクライアントを設定します。

   89 ページの「CLI Client が ssh-agent を使用して SSH 接続を行うように構成する」を参照してください。

鍵の準備

マスターサーバーとローカルディストリビュータ間、マスターサーバーと リモートエージェント間の通信の認証に使用する 公開 / 非公開鍵ペアを生成します。続いて、生成された鍵をローカルディストリビュータとリモートエージェントにコピーします。`ssh-agent` を使用するか空のパスワード付きの鍵を使用するかにもとづいて、行う作業を選択してください。

注 – 次に、デフォルトの長さで鍵を作成する方法を示します。セキュリティーを最大限に高めるには、できるだけ長い鍵を作成してください。

▼ 鍵ペアを作成する

`ssh-agent` を使用する場合は、鍵ペアを 1 つ生成すればすみます。空のパスワードを使用する場合は、2 台のマシン間に確立される SSH 接続ごとに鍵ペアを 1 つ生成できます。あるいは、すべての接続に使用するように鍵ペアを 1 つだけ生成することも可能です。この作業を、生成する鍵ペアごとに行います。

始める前に　N1 Service Provisioning System 5.1 に使用するユーザー ID とグループ ID はネットワーク内の全サーバーで同じになるようにしてください。

手順

1. マスターサーバー **(**空のパスワードを使用し、接続ごとに鍵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
   ```

   鍵を保存するように指示するメッセージが表示されます。

2. **Return** を押して、デフォルトの場所に鍵を保存します。

   非公開鍵が */User-home*`/.ssh/id_rsa` に保存されます。公開鍵は、*/HOME*`/.ssh/id_rsa.pub` に保存されます。

   User-home には、現在ログインしているユーザーのマスターサーバーマシン上のホームディレクトリを指定します。

   パスワードを指定するためのプロンプトが表示されます。

3. パスワードを指定する必要があるかどうかを決定します。
   - 空のパスワードの鍵を使用する場合は、パスワードを指定しないでください。Return キーを押して継続します。
   - `ssh-agent` を使用する場合は、鍵のパスワードを指定してください。

▼ ssh-agent の鍵を設定する

ssh-agent を使用する場合は、この作業を行なってローカルディストリビュータとリモートエージェントに鍵をコピーしてください。

手順 **1.** マスターサーバーで、非公開鍵ファイル **~/.ssh/id_rsa** をセキュリティー保護された媒体にコピーします。

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

User-home には、現在ログインしているユーザーのマスターサーバーマシン上のホームディレクトリを指定します。*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。

2. ローカルファイルシステムから非公開鍵ファイルを削除します。

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

3. **SSH** を使用するように設定する各ローカルディストリビュータとリモートエージェントに公開鍵をコピーします。**~/.ssh/authorized_keys2** ファイルに鍵を保存します。

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

User-home には、マスターサーバーマシン上のホームディレクトリを指定します。*User-home-APP* には、現在ログインしているユーザーのローカルディストリビュータまたはリモートエージェントマシン上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 (**誰でも書き込める状態**)** 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を **600** に変更します。

6. **ssh-agent** の転送を使用可能にするため、マスターサーバーとローカルディストリビュータの **config.properties** ファイルにある次の行を編集します。

現在の構成は次のとおりです。

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

この行を編集して -A オプションを含めます。

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

▼ 空のパスワードファイルに鍵を設定する (1 組の鍵ペアを使用する場合)

空のパスワードファイルを使用する場合で、1 組の鍵ペアだけを生成した場合は、この作業を行なってローカルディストリビュータとリモートエージェントに鍵をコピーしてください。

手順

1. マスターサーバーからアップストリームである各マシンに非公開鍵をコピーします。この鍵をホームディレクトリに保存してください。

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

   User-home には、現在ログインしているユーザーのマスターサーバーマシン上のホームディレクトリを指定します。*User-home-upstream* には、アップストリームである マシン上のホームディレクトリを指定します。アップストリームマシンとは、ダウンストリームであるマシンとの SSH 接続を開始するマシンです。

   各ローカルディストリビュータに固有の非公開鍵を設定することも、あるいはすべてのローカルディストリビュータに同じ非公開鍵を設定することもできます。

2. 各ダウンストリームマシンに公開鍵をコピーします。**/.ssh/authorized_keys2** ファイルに鍵を保存してください。

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

   User-home には、マスターサーバーマシン上のホームディレクトリを指定します。*User-home-downstream* には、先の手順で設定したマシンの接続先であるローカルディストリビュータマシンまたはリモートエージェントマシン上のホームディレクトリを指定します。SSH を使用して接続するすべてのローカルディストリビュータとリモートエージェント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 (**誰でも書き込める状態**)** でないことを確認します。

4. 非公開鍵ファイル **.ssh/id_rsa** がほかのユーザーまたはグループによってアクセスできないことを確認します。

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

▼ 空のパスワードファイルに鍵を設定する (複数の鍵ペアを使用する場合)

空のパスワードファイルを使用する場合で、SSH 接続ごとに鍵ペアを生成した場合は、この作業を行なってローカルディストリビュータとリモートエージェントに鍵をコピーしてください。

始める前に　この作業は、ネットワーク上で確立されるSSH 接続ごと (つまり鍵ペアごと) に行なってください。

手順

1. アップストリームマシンから、各ダウンストリームマシンに公開鍵をコピーします。*User-home* **/.ssh/authorized_keys2** ファイルに鍵を保存します。

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

User-home-upstream には、アップストリームマシン上のホームディレクトリを指定します。*Use-home-downstream* には、アップストリームマシンの接続先であるローカルディストリビュータまたはリモートエージェントマシン上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 (**誰でも書き込める状態**)** でないことを確認します。
3. 非公開鍵ファイル **.ssh/id_rsa** がほかのユーザーまたはグループによってアクセスできないことを確認します。
4. **.ssh/authorized_keys2** ファイルのアクセス権を **600** に変更します。

マスターサーバーでの接続の設定とテスト

この節では、N1 Service Provisioning System 5.1 で SSH を使用する前に実施すべき SSH の初期設定とテストについて説明します。 `ssh-agent` を使用する場合は、設定とテストの作業を開始する前に `ssh-agent` を起動する必要があります。

▼ マスターサーバーで `ssh-agent` を起動する

この作業は、`ssh-agent` を使用する場合にのみ行います。この作業は、マスターサーバーを起動する前に行なってください。

注 **–** 次の設定作業で使用する `ssh`、`ssh-add`、および `cr_server start` の SSH 設定コマンドは、すべて `ssh-agent` の起動に使用したセッションで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、稼働している `ssh-agent` プログラムを停止し、新しい `ssh-agent` プログラムを起動しなければなりません。

手順 1. **ssh-agent** を起動します。

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

`ssh-agent` が起動し、このプログラムによって 2 つの環境変数が設定されます。`SSH_AUTH_SOCK` と `SSH_AGENT_PID` は、`ssh-agent` に接続するために `ssh` と `ssh-add` によって使用されます。

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。

パスワードを指定するようにメッセージが表示されます。

3. 鍵を生成した際に作成したパスワードを指定します。

参考 ssh-agent の停止

ssh-agent は、コマンド eval `ssh-agent -k` を実行して停止できます。

このコマンドは、SSH_AGENT_PID 変数を使用して ssh-agent プロセスに信号を送り、このプロセスを停止します。また、ssh-agent の起動時に設定された環境変数の解除も行います。

▼ マスターサーバーで接続の設定とテストを行う

始める前に ssh-agent を使用する場合は、85 ページの「マスターサーバーで ssh-agent を起動する」 の操作説明に従って、必ず ssh-agent を起動してください。



注意 – 設定作業はセッションに依存します。このため、SSH コマンド、ssh、ssh-add、および cr_server start はすべて ssh-agent の起動時に使用したセッションで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、稼働している ssh-agent プログラムを停止し、新しい ssh-agent プログラムを起動しなければなりません。また、非公開鍵もアップロードする必要もあります。

手順 1. **SSH** の接続経路をテストします。

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

-A オプションは、 ssh-agent を使用する場合にのみ使用してください。*target-host-IP* には、このマシンの接続先となるマシンの IP アドレスを指定します。

例として、マスターサーバー (MS)、ローカルディストリビュータ (LD1、LD2、LD3)、リモートエージェント (RA1、RA2、RA3、RA4) から構成された次のようなネットワークを考えてみましょう。

```
MS
 | \
 |  \
 |   \
LD1  LD2
 | \    \
 |  \    LD3
 |   \   |\
```



このネットワーク例では、マスターサーバー上で次のコマンドを実行し、LD1、LD2、RA1、RA2、RA3、RA4 をネットワーク上のローカルディストリビュータとリモートエージェントの IP アドレスに置き換えることにより、SSH 接続経路をテストします。

```
% 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、SSH を使用してダウンストリームマシンに接続する場合にマスターサーバーが使用する経路をたどります。各コマンドを実行することで、SSH は引数として指定されたマシンとの通信に必要なホスト鍵を交換できます。

SSH は、ホスト鍵の交換を許可するかどうかを尋ねるメッセージを表示します。

2. 各メッセージに「**yes**」と答えます。
3. すべてのコマンドの出力を検証し、環境変数が正しく設定されているか確認します。

   PATH 変数には、ユーザー環境の一部であるディレクトリ (/bin、/usr/bin など) が設定されていなければなりません。
4. **SSH** の接続経路をもう一度テストします。

   手順 1 と同じコマンドを使用して接続経路をテストし直し、情報の入力を求めるプロンプトが表示されないことを確認します。

参考 設定とテストの繰り返し

鍵のどれかを変更する場合は、必要に応じてこの作業を再度行なってください。サーバーの設定によっては、マシンのどれかをリブートするたびにこの作業を行わなければならないこともあります。

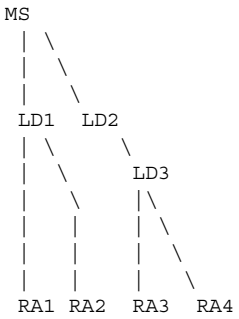
アプリケーションに合わせて SSH を構成する

マスターサーバーで SSH の設定とテストを完了したあと、マスターサーバー が SSH を使用して接続できるように N1 Service Provisioning System 5.1 内のその他のマシンを構成します。

▼ ローカルディストリビュータとリモートエージェントで使用するように SSH を構成する

マスターサーバーからリモートエージェントの方向へ向かって N1 Service Provisioning System 5.1 ネットワーク内をたどり、中間的なローカルディストリビュータ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。基本的に、これはツリーネットワークを先行順にトラバースする処理です。

例として、マスターサーバー (MS)、ローカルディストリビュータ (LD1、LD2、LD3)、リモートエージェント (RA1、RA2、RA3、RA4) から構成された次のようなネットワークを考えてみましょう。



LD1、RA1、RA2、LD2、LD3、RA3、RA4 の順でネットワークを構成します。この順序を厳密に守り、1 つの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.** マスターサーバー ブラウザインタフェースを使用し、構成するマシンの「**Host Details**」ページを表示します。

2. そのマシンにどのアプリケーションを構成するかに応じて、ローカルディストリビュータセクションまたはリモートエージェントセクションに接続の詳細情報を追加します。

3. 接続タイプを **ssh** と指定します。

4. 「**Advanced Parameters**」フィールドに次のテキストを追加します。

`cprefix=/`*`N1SPS5.1-Home`***`/`***`application`***

N1SPS5.1–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、agent (リモートエージェントを構成している場合) または ld (ローカルディストリビュータを構成している場合) と指定します。

たとえば、N1 Service Provisioning System 5.1 が `/opt/SUNWn1sps/N1_Service_Provisioning_System_5.1/` にインストールされる場合で、リモートエージェントを構成しているときは、「Advanced Parameters」フィールドに次のテキストを追加します。

```
cprefix=/opt/SUNWn1sps/N1_Service_Provisioning_System_5.1/agent
```

5. ホストの詳細情報を保存します。
6. このマシンでリモートエージェントインスタンスまたはローカルディストリビュータ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**」ページで「**Test Connection**」をクリックします。
8. この作業を、ネットワーク内のマシンごとに繰り返します。

▼ CLI Client が `ssh-agent` を使用して SSH 接続を行うように構成する

この作業は、CLI Client が `ssh-agent` を使用して SSH 接続を行うように構成する場合に行なってください。

手順

1. マスターサーバーと、**CLI Client** のインストール先であるマシンで、新しい **OS** ユーザーアカウントを作成します。

   このアカウントは、マスターサーバー、ローカルディストリビュータまたはリモートエージェントのインストール時に指定したアカウントとは別のものでなければなりません。
2. 先のステップで作成した新しいユーザーでマスターサーバー にログインします。
3. 82 ページの「鍵ペアを作成する」の手順に従って、作成した新規ユーザー用の公開鍵と非公開鍵を生成します。

   マスターサーバー、ローカルディストリビュータ、リモートエージェント間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. マスターサーバーで、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、現在ログインしているユーザーのマスターサーバーマシン上のホームディレクトリを指定します。*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. ローカルファイルシステムから非公開鍵ファイルを削除します。

   ```
   % rm /User-home/.ssh/id_rsa
   ```
6. マスターサーバーで、そのユーザーの **`/.ssh/authorized_keys2`** ファイル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、マスターサーバーマシン上のホームディレクトリを指定します。
7. 先に作成した新規ユーザーのアカウントで、**CLI Client** マシンにログインします。

8. **`ssh-agent`** を起動します。

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

   User-home には、現在ログインしているユーザーの CLI Client マシン上のホームディレクトリを指定します。

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

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

   User-home には、CLI Client マシン上のホームディレクトリを指定します。

10. マスターサーバーからログアウトし、ログインし直します。

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した安全なメディアのパスを指定します。

    以上の操作で、CLI Client はマスターサーバーに接続する時の認証に 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受け入れるように、マスターサーバーを構成します。手順については、111 ページの「**JVM** セキュリティーポリシーの構成」を参照してください。

参考 `ssh-agent` の停止

注 – `ssh-agent` を停止する場合は、CLI Client 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、現在ログインしているユーザーの CLI Client マシン上のホームディレクトリを指定します。

▼ CLI Client が空のパスワードで SSH 接続を行うように構成する

この作業は、CLI Client が空のパスワードを使用して SSH 接続を行うように構成する場合に行なってください。

手順

1. マスターサーバーと、**CLI Client** のインストール先であるマシンで、新しい **OS** ユーザーアカウントを作成します。

   このアカウントは、マスターサーバー、ローカルディストリビュータ、またはリモートエージェントのインストール時に指定したアカウントとは別のものでなければなりません。

2. 先のステップで作成した新しいユーザーで、**CLI Client** マシンにログインします。

3. 82 ページの「鍵ペアを作成する」の手順に従って、作成した新規ユーザー用の公開鍵と非公開鍵を生成します。

   マスターサーバー 、ローカルディストリビュータ、リモートエージェント間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 Client** 上で、マスターサーバーマシン上にある新しいユーザーの **`authorized_keys2`** ファイルに公開鍵ファイルをコピーします。

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

   User-home-CLI には、CLI Client マシン上のホームディレクトリを指定します。*User-home-MS* には、マスターサーバーマシン上のホームディレクトリを指定します。

5. マスターサーバーで公開鍵をそのユーザーの **`/.ssh/authorized_keys2`** ファイルに連結します。

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

   User-home には、現在ログインしているユーザーの マスターサーバーマシン上のホームディレクトリを指定します。

6. 先に作成した新規ユーザーのアカウントで、**CLI Client** マシンにログインします。

7. **SSH** 接続をテストします。

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

   IP-Address-MS には、マスターサーバーマシンの IP アドレスを指定します。

   ここで、鍵の交換をするかどうかを尋ねるメッセージが表示されることがあります。

8. このメッセージが表示された場合は、「**yes**」と答えます。

9. **`PATH`** 変数が正しく設定されているか確認します。

   `PATH` 変数には、 `/bin`, `/usr/bin`、およびユーザー環境の一部であるそのほかのあらゆるディレクトリが含まれている必要があります。

10. ローカルホストからの接続だけを受け入れるように、マスターサーバーを構成します。手順については、**111** ページの「**JVM** セキュリティーポリシーの構成」を参照してください。

SSH の上級パラメータとコマンドのリファレンス

上級パラメータのリファレンス

「Host Details」ページの「Advanced Parameters」フィールドに、SSH 構成情報をさらに指定できます。次に、上級パラメータとして受け入れられるものを示します。複数のパラメータを使用する場合は、それらをコンマで区切って入力します。
「Advanced Parameters」フィールドには、不要なスペースは入れないようにしてください。

cprefix

cprefix パラメータは、すべてのローカルディストリビュータおよびリモートエージェントに必要です。「Advanced Parameters」フィールド内のパラメータの構文は次のとおりです。

```
cprefix=/N1SPS5.1-Home/application
```

N1SPS5.1–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、agent (リモートエージェントを構成している場合) または ld (ローカルディストリビュータを構成している場合) と指定します。

例 **7–1** cprefix の例

N1 Service Provisioning System 5.1 が `/opt/SUNWn1sps/` にインストールされていて、リモートエージェントの構成を行う場合には、「Advanced Parameters」フィールドに次のテキストを追加します。

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

sshpath

ターゲットサーバー上の PATH に SSH 実行可能ファイルのパスをまだ追加しておらず、追加を望まない場合は、このパラメータを使用して SSH 実行可能ファイルのパスを指定できます。「Advanced Parameters」フィールド内のパラメータの構文は次のとおりです。

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

path-to-SSH には、SSH 実行可能ファイルがインストールされているディレクトリを指定します。

例 **7–2** sshpath の例

SSH 実行可能ファイルが /usr/local/bin/ssh ディレクトリにインストールされている場合は、「Advanced Parameters」フィールドには次のテキストを追加します。

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

sshargs

特定のホストで ssh コマンドを実行するときにその ssh コマンドにコマンド行引数を追加する場合は、そのホストの「Advanced Parameters」フィールドに引数を指定できます。「Advanced Parameters」フィールド内のパラメータの構文は次のとおりです。

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

option には、ssh コマンドに提供するコマンド行オプションを指定します。複数のオプションを追加する場合は、それらを | で区切ってください。

例 **7–3** sshargs の例

ssh コマンドによって認証エージェントの転送機能を有効にする場合は、「Advanced Parameters」フィールドに -A オプションを指定します。

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

OpenSSH 2.0 コマンドリファレンス

この節では、この章の操作説明で挙げられている OpenSSH 2.0 のコマンドとオプションについて説明します。別バージョンの SSH を使用している場合は、次に示すものと同等のコマンドとオプションをそのバージョンで確認してください。SSH の構成操作を行う際には、それらの同等コマンドを使用してください。

表 **7–1** OpenSSH 2.0 コマンド

ツール	説明
ssh	アプリケーションがほかのアプリケーションをリモートマシンから呼び出すことを許可します。SSH 通信を行うように構成されたソフトウェアは、ssh コマンドを使用してリモートアプリケーション (リモートエージェントまたはローカルディストリビュータ) を呼び出し、SSH の標準入出力ストリームを使用してそれらのリモートアプリケーションとの通信を行います。
ssh-agent	パスワード付きの非公開鍵を使用する場合に使用します。アプリケーションの SSH 呼び出しが ssh-agent による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します。

表 **7–1** OpenSSH 2.0 コマンド　(続き)

ツール	説明
ssh-add	ssh-agent に非公開鍵をアップロードします。
ssh-keygen	SSH 接続を保護するため、公開 / 非公開鍵のペアを生成します。

ssh コマンドでは、次のオプションを使用できます。

-A　認証エージェントの転送機能を有効にします。

-o 'BatchMode yes'　パスフレーズクエリーを無効にします。

-t　コマンドが発行されている場合も tty を割り当てます。

ssh-keygen コマンドでは、次のオプションを使用できます。

-t rsa　生成する鍵のタイプとして RSA を指定します。

ssh-agent コマンドでは、次のオプションを使用できます。

-k　環境変数 SSH_AGENT_PID 内の pid セットを使用してエージェントを強制終了します。ほかの実装では、異なる環境変数を使用する可能性があります。

第 8 章

SSL を使用するための N1 Service Provisioning System 5.1 の構成

この章では、Socket Layer (SSL) を使用して通信を行うように N1 Service Provisioning System 5.1 を構成する方法について説明しています。この章の内容は、次のとおりです。

- 95 ページの「N1 Service Provisioning System 5.1 での SSL サポートの概要」
- 99 ページの「SSL の構成 – 作業概要」
- 104 ページの「構成例」
- 109 ページの「SSL 暗号群」

N1 Service Provisioning System 5.1 での SSL サポートの概要

SSL は、IP ネットワーク上の通信を保護するためのプロトコルです。SSL は、TCP/IP ソケット技術を使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、RSA が開発した公開 / 非公開鍵ペア暗号化システムを使用してそのメッセージを保護します。SSL は、ほとんどの Web サーバー製品でサポートされるほか、Netscape Navigator ブラウザと Microsoft Web ブラウザでもサポートされます。

メッセージの傍受や改ざんを防止するため、ネットワーク通信に SSL を使用するように N1 Service Provisioning System 5.1 アプリケーションを構成できます。また、通信前に SSL を使用して認証を行うようにアプリケーションを構成すれば、ネットワークセキュリティーがさらに高まります。

暗号群: 暗号化と認証の概要

サーバーとクライアント間の相互認証、証明書の送信、セッション鍵の確立などのために、SSL プロトコルでは各種の暗号アルゴリズム (暗号方式) がサポートされます。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。

暗号群は慎重に選択する必要があります。各アプリケーションの設定では、ノードが要求する最小限のセキュリティーを提供する暗号群だけを有効にしてください。SSL は、クライアントとサーバーの双方がサポートする暗号群の中からもっともセキュリティーレベルの高い暗号群を使用します。低セキュリティーの暗号群を有効にすると、Sun 以外のクライアントは暗号群のネゴシエーションの際にセキュリティーレベルがもっとも低い暗号群しかサポートしなくなり、サーバーはセキュリティーレベルが比較的低い暗号群の使用を余儀なくされる可能性があります。

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ません。
- サーバーの認証 – クライアントは接続先であるサーバーの認証を行います。
- 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双方が互いに認証し合います。

インストール時に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、使用する暗号群の選択を求めるメッセージが表示されます。暗号群の値は、`net.ssl.cipher.suites` の値として `config.properties` ファイルに格納されます。暗号群の値は、選択に応じて次のように設定できます。

- 「暗号化のみ (認証なし)」を選択した場合、暗号群は `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA` に設定されます。
- 「暗号化 (認証あり)」を選択した場合、暗号群は `SSL_RSA_WITH_3DES_EDE_CBC_SHA` に設定されます。

AIX サーバー上のローカルディストリビュータで SSL を使用する場合、SSL 暗号群は認証による暗号化に設定されます。認証を伴わない暗号化は、AIX サーバーで稼働しているローカルディストリビュータでは利用できません。

サーバー認証を必要とする SSL 暗号群とサーバー認証を必要としない SSL 暗号群の一覧は、109 ページの「SSL 暗号群」を参照してください。クライアント認証は、サーバー認証を必要とする暗号群だけを対象に構成することもできます。

注 – N1 Service Provisioning System 5.1 アプリケーションの SSL 構成は、認証を伴わない暗号化と認証を伴う暗号化のどちらも行えます。認証を伴う暗号化では、クライアントとサーバーの認証が行われます。前述のような構成が可能ですが、認証を伴う暗号化がもっとも安全です。

認証キーストア

N1 Service Provisioning System 5.1 は、自己署名付き証明書と認証局によって署名される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ほかのアプリケーションに接続する際にそれ自体の認証に使用する公開 / 非公開鍵ペアが格納されます。

- トラストキーストア – キーストアによって信頼されアプリケーションへの接続が許可されるその他のアプリケーションの公開鍵が、自己署名付き証明書の形で格納されます。

「クライアントとサーバーの相互認証」で SSL を有効にした場合は、有効にした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。その 1 つは、SSL がほかのアプリケーションに対しそれ自体を認証するためのキーストアで、もう 1 つはほかのアプリケーションを認証するためのキーストアです。

「サーバーのみの認証」で SSL を有効にした場合は、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公開 (トラステッド) キーストアが必要になります。公開キーストアは、Java Secure Sockets Extension (JSSE) v1.0.3 によって提供されている独自の JKS フォーマットになります。

これらのキーストアは、どちらもパスワードを指定する必要があります。このパスワードは、両キーストアで同一のものでなければなりません。

例として、アプリケーション A (SSL クライアント) とアプリケーション B (SSL サーバー) を SSL を使用して相互接続する場合を考えてみましょう。両方とも、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 は、そのプライベートキーストアに公開 / 非公開鍵ペアが格納されていなければなりません。一方アプリケーション A は、そのトラストキーストアにアプリケーション 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 がアプリケーション 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、トラストキーストア内にこの公開鍵が入っているかを確認することによってこの鍵を検証できます。

クライアント認証を要求するようにアプリケーション B が構成されている場合、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 / 非公開鍵ペアが格納されていなければなりません。同時に、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る必要があります。アプリケーション A がアプリケーション B の認証を行なったあと、アプリケーション B はそのトラストキーストアに鍵が入っているかを確認することによってアプリケーション A の公開鍵を検証できます。

`crkeys` コマンドを使用してキーストアを生成する場合は、`-mode upstream|downstream` オプションを使用して、認証を実行しているマシンとの関係上での認証されているマシンの場所を指定します。SSL 接続の試行時には、接続の妥当性検査を行うサーバーが、証明書を送信したサーバーの相対的な位置に一致するモードが受信した証明書に含まれていることを確認します。たとえば、ローカルディストリビュータがマスターサーバーから SSL 接続要求を受信するとします。ローカルディストリビュータは、マスターサーバー証明書に `downstream` 注釈が含まれていないことを確認します。

注 – サーバーは、証明書に upstream 注釈が含まれているかは確認しません。サーバーが確認するのは、証明書に downstream 注釈が含まれていないことだけです。この結果、元々 upstream 注釈なしの SSL を使用するように構成されていたサーバーはいずれも、引き続き SSL を接続に使用します。

CLI クライアントからアップストリームになれるサーバーはないため、CLI クライアントは、upstream 注釈のある証明書を転送するサーバーとの接続の妥当性を証明できません。リモートエージェントからダウンストリームになれるサーバーはないため、リモートエージェントは、downstream 注釈のある証明書を転送するサーバーとの接続の妥当性を証明できません。

SSL でのパスワードの使用

トラストキーストアの処理にパスワードを指定した場合、このパスワードはキーストアの整合性の検査にだけ使用されます。このパスワードはトラストキーストアのコンテンツに対するアクセスは防止しませんが、トラストキーストアの更新は防止します。ユーザーは、パスワードを指定しなければキーストアの内容を変更することはできません。

プライベートキーストアの処理にパスワードを指定した場合、このパスワードは、キーストアの整合性の検査、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に使用されます。

crkeys コマンドは、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に、プライベートストア内にパスワードが存在すると、これと同じパスワードが crkeys スクリプトによってトラストストアに設定されます。 同様に、初めてプライベートストアを作成するときに、トラストストア内にパスワードが存在すると、これと同じパスワードが crkeys スクリプトによってプライベートストアに設定されます。

crkeys コマンドを使用すると、符号化されたキーストアパスワードを作成できます。符号化されたパスワードは、キーストアパスワードを保存するどのプロパティファイルでも使用できます。安全性は、プロパティファイルに符号化されたパスワードを保存するほうが、プロパティファイルにプレーンテキストのパスワードを保存するより優れています。

N1 Service Provisioning System 5.1 における SSL の制限

N1 Service Provisioning System 5.1 の SSL 実装には、次のような制限があります。

- トラストキーストアとプライベートキーストアには同じパスワードを設定する必要があります。また、プライベートキーストア内において、ストア内の各キーのパスワードはストアパスワードと同じでなければなりません。この制限は、キーの作成

に使用された crkeys スクリプトによって適用されます。

- CLI Client アプリケーションのクライアント認証を有効に設定することはできますが、セキュリティーの制限上この設定はサポートされません。CLI Client 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ません。キーストアが作成されている場合は、符号化パスワードを CLI Client プロパティファイルに指定する必要があります。
- N1 Service Provisioning System 5.1 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します。このため、たとえばマスターサーバーがリモートエージェントに接続し、その公開鍵を信頼するとします。リモートエージェントが危険にさらされた場合、CLI Client がクライアント認証を使用するように設定されていれば、リモートエージェントの鍵を使用して、マスターサーバーに対して CLI Client の認証を行えます。同様に、ローカルディストリビュータがリモートエージェントに接続し、リモートエージェントが危険にさらされた場合、ローカルディストリビュータを使用してマスターサーバーにコマンドを発行できます。

  このような問題からマスターサーバーとローカルディストリビュータを保護するには、それらに接続を認められているサーバーからの接続だけを受け入れるようにアプリケーションを設定します。ローカルディストリビュータは、その親ノードからの接続だけを受け入れるようにし、マスターサーバーは指定されている CLI ホストからの接続だけを受け入れるようにします。手順については、第 9 章を参照してください。
- SSH 接続の場合、リモートアプリケーション (ローカルディストリビュータまたはリモートエージェント) は自動的に起動します。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ません。ただし、アプリケーションがキーストアを使用して初期化されている場合は、キーストアの符号化されたパスワードをプロパティファイルに指定する必要があります。
- SSH を使ってマスターサーバーに接続するように CLI Client を構成した場合、CLI Client は、ソケット経由でマスターサーバーに接続する SshProxy アプリケーションを利用してマスターサーバーに接続します。SshProxy は SSL 経由でマスターサーバーに接続できますが、この構成はサポートされていません。
- Windows アプリケーションの場合、符号化されたキーストアパスワードをプロパティファイルで指定する必要があります。

SSL の構成 – 作業概要

次に、SSL を使用するように N1 Service Provisioning System 5.1 を構成するために必要な作業の概要を示します。

1. 使用する SSL 接続を決定します。

   詳細は、95 ページの「N1 Service Provisioning System 5.1 での SSL サポートの概要」を参照してください。
2. crkeys コマンドを使用してキーストアを作成します。

100 ページの「キーストアを作成する」を参照してください。

3. config.properties ファイルを編集して SSL を構成します。

102 ページの「config.properties ファイルを編集して SSL を構成する」を参照してください。

▼ キーストアを作成する

N1 Service Provisioning System 5.1 は、JRE に付属のキーツールユーティリティを使用します。 このキーツールユーティリティは、ユーザーがキーストアを作成できるようにシェルスクリプト crkeys にラップされています。このスクリプトを使用してユーザーは、keytool ユーティリティに正しいパラメータを指定することができます。

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
```

手順 ● キーを生成します。

```
% crkeys -options
```

使用する SSL 接続の種類にもとづいてキーストアを作成する場合、次のオプションを使用します。

-alias *application_hostname*	証明書または鍵ペアの別名を指定します。アプリケーションのホスト名を別名として使用します。別名は、キーストア内で一意でなければなりません。
-mode upstream|downstream	認証を実行しているマシンとの関係で認証されているマシンの場所を指定します。たとえば、ローカルディストリビュータからダウンストリームのリモートエージェントの証明書を生成しているとします。リモートエージェントのモードをダウンストリームに指定します。ローカルディストリビュータのモードをアップストリームに指定します。
-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します。
-delete	エンティティを指定し、鍵のペアまたは証明書をキーストアから削除します。
-epass	プレーンテキストのパスワードを符号化されたパスワードに変換し、出力します。ファイルにパスワードを格納する予定の場合は、符号化されたパスワードを作成します。たとえば、

	config.properties ファイルにキーストアパスワードを格納することを選択した場合は、符号化されたパスワードを指定する必要があります。
-export	エンティティを指定し、自己署名付き証明書を特定のファイルにエクスポートします。
-file *cert_file*	証明書のインポート元またはエクスポート先となるファイルの名前を指定します。
-generate	別名を指定して新しい鍵ペアを生成します。
-help	すべてのオプションを一覧で表示します。
-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します。証明書をインポートする際は、この証明書に記載されたノードのホスト名を別名として使用します。
-keyalg *keyalg*	鍵生成アルゴリズム。デフォルトは RSA です。RSA または DSA を指定できます。
-keysize *keysize*	キーサイズ。デフォルトは 1024 です。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 の倍数を指定できます。
-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で表示します。
-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します。パスワードの長さは、6 文字以上でなければなりません。
-password *password*	キーストアのパスワードを指定します。パスワードを指定しないと、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ます。パスワードの長さは、6 文字以上でなければなりません。
-private	操作の対象として、プライベートキーストアを指定します。
-validity *days_valid*	自己署名付き証明書の有効期間を日数で指定します。
-trust	操作の対象としてトラストキーストアを指定します。

例 8–1 crkeys のコマンド構文

次に、crkeys コマンドの使用例を示します。

公開 / 非公開鍵ペアを生成するには、次を実行します。

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

鍵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、次を実行します。

```
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、次を実行します。

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

鍵または鍵ペアを削除するには、次を実行します。

```
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、次を実行します。

```
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、次を実行します。

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

プレーンテキストのパスワードを符号化されたパスワードに変換し、出力するには、次を実行します。

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

`crkeys` コマンドの使用方法を出力するには、次を実行します。

```
crkeys -help
```

▼ config.properties ファイルを編集して SSL を構成する

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
- *N1SPS5.1-home* /*app*/data/private.store ファイルでプライベートキーストアを検索します。
- *N1SPS5.1-home* /*app*/data/trust.store ファイルでトラストキーストアを検索します。
- 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、次のセキュリティー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します。

  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、109 ページの「SSL 暗号群」を参照してください。
- 接続を求めてくる SSL クライアントをアプリケーションが認証するように指定します。
- 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します。

注 – 認証を有効にするには、アプリケーションをインストールしたあとでキーストアを初期化する必要があります。

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

次に、`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します。サポート対象の SSL 暗号群のリストは、109 ページの「SSL 暗号群」を参照してください。
`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します。
`net.ssl.key.store.pass`		キーストアのパスワード。場合に応じて要求されます。詳細については、以下を参照してください。

注 – net.ssl.key.store.pass パラメータは、N1 Service Provisioning System 5.1 アプリケーションの SSL キーストアパスワードを指定します。このパラメータは、SSL キーストアを使用してアプリケーションを構成するが、アプリケーションの起動時にそのキーストアのパスワードを尋ねるプロンプトを表示させないようにする場合に使用してください。次のような場合には、このパラメータを指定する必要があります。

- システムのブート時に自動的に起動するように N1 Service Provisioning System アプリケーションを設定する場合。
- Windows サーバーでは、N1 Service Provisioning System アプリケーションはキーストアパスワードを求めるプロンプトを表示しません。このため、Windows サーバー上で SSL を使用するように構成されるアプリケーションはすべて、このパラメータを指定する必要があります。
- CLI アプリケーションは、キーストアパスワードを求めるプロンプトを表示しません。このため、SSL を使用するように構成する CLI Client はすべてこのパラメータを指定する必要があります。
- ローカルディストリビュータが SSH 経由でその親に接続される場合、ローカルディストリビュータはパスワードを求めるプロンプトを表示できません。

構成例

例 **8–2** マスターサーバー、ローカルディストリビュータ、リモートエージェント間で認証を使用せずに SSL を構成する方法

1. マスターサーバー、ローカルディストリビュータ、リモートエージェントをインストールします。インストールプログラムが接続の種類を選択するように求めた場合に、SSL を選択します。暗号群を選択するように求められた場合、認証を伴わない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る暗号群を有効にできます。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. ブラウザインタフェースで、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 のローカルディストリビュータを追加します。
5. ローカルディストリビュータ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 のリモートエージェントを追加します。

例 **8–2** マスターサーバー、ローカルディストリビュータ、リモートエージェント間で認証を使用せずに SSL を構成する方法　(続き)

8. リモートエージェントとの接続をテストします。

例 **8–3** SSL サーバー認証を構成する方法

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に `config.properties` ファイルに変更を加える必要はありません。

1. ローカルディストリビュータ用の鍵ペアを生成し、ローカルディストリビュータのプライベートキーストアに格納します。`-mode downstream` を指定します。

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

2. ローカルディストリビュータ上のプライベートキー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-downstream.cert -alias ldhostname-downstream.cr.com
```

3. ローカルディストリビュータの自己署名付き証明書をマスターサーバーにコピーします。
4. 自己署名付き証明書をマスターサーバー のトラストキーストアにインポートします。

```
% server/bin/crkeys -trust -import -file ld-downstream.cert -alias ldhostname-downstream.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 のローカルディストリビュータを追加します。
7. ローカルディストリビュータに対し、CLI `net.gencfg` コマンドを使用して、手動で `transport.config` ファイルを生成します。
8. `transport.config` ファイルをローカルディストリビュータにコピーします。
9. マスターサーバーとローカルディストリビュータが稼働している場合は、これらを停止します。
10. マスターサーバーとローカルディストリビュータを起動します。
11. マスターサーバーとローカルディストリビュータのキーストアのパスワードを入力します。
12. ローカルディストリビュータとの接続をテストします。
13. リモートエージェント用の鍵ペアを生成し、リモートエージェントのプライベートストアに格納します。`-mode downstream` を指定します。

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

14. リモートエージェ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。

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

15. リモートエージェントの自己署名付き証明書をローカルディストリビュータにコピーします。
16. 自己署名付き証明書をローカルディストリビュータのトラストストアにインポートします。

例 **8–3** SSL サーバー認証を構成する方法　(続き)

```
% ld/bin/crkeys -trust -import -file ra-downstream.cert -alias rahostname-downstream.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 のリモートエージェントを追加します。
19. リモートエージェントに対し、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。
20. `transport.config` ファイルをリモートエージェントにコピーします。
21. ローカルディストリビュータとリモートエージェントが稼働している場合は、これらを停止します。
22. ローカルディストリビュータとリモートエージェントを起動します。
23. ローカルディストリビュータとリモートエージェントのキーストアのパスワードを入力します。
24. リモートエージェントとの接続をテストします。

例 **8–4** SSL サーバーおよびクライアント認証を構成する方法

1. マスターサーバー、ローカルディストリビュータ、リモートエージェントをインストールします。インストールプログラムが接続の種類を選択するように求めた場合に、SSL を選択します。暗号群を選択するように求められる際に、認証を伴う暗号化を選択します。
2. `config.properties` ファイルを編集して、使用する暗号群と符号化されたキーストアパスワードを含めます。

   すべてのホストで同じキーストアパスワードを使用する必要があります。
3. ローカルディストリビュータ用の鍵ペアを生成し、ローカルディストリビュータのプライベートストアに格納します。`-mode downstream` を指定します。

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

4. マスターサーバー用の鍵ペアを生成し、マスターサーバーのプライベートストアに格納します。`-mode upstream` を指定します。

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

5. ローカルディストリビュータ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-downstream.cert -alias ldhostname-downstream.cr.com
   ```
6. ローカルディストリビュータの自己署名付き証明書をマスターサーバーにコピーします。
7. 自己署名付き証明書をマスターサーバーのトラストストアにインポートします。

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

8. マスターサーバー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-upstream.cert -alias mshostname-upstream.cr.com
   ```

例 **8–4** SSL サーバーおよびクライアント認証を構成する方法　(続き)

9. マスターサーバーの自己署名付き証明書をローカルディストリビュータにコピーします。
10. 自己署名付き証明書をローカルディストリビュータ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
```

11. マスターサーバーとローカルディストリビュータが稼働している場合は、これらを停止します。
12. マスターサーバーとローカルディストリビュータを起動します。
13. マスターサーバーとローカルディストリビュータのキーストアのパスワードを入力します。
14. 新しいホストを作成します。
15. 新しいホストで、接続タイプ SSL のローカルディストリビュータを追加します。
16. ローカルディストリビュータとの接続をテストします。
17. リモートエージェント用の鍵ペアを生成し、リモートエージェントのプライベートストアに格納します。`-mode downstream` を指定します。

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

18. リモートエージェント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-downstream.cert -alias rahostname-downstream.cr.com
```

19. リモートエージェントの自己署名付き証明書をローカルディストリビュータにコピーします。
20. 自己署名付き証明書をローカルディストリビュータのトラストストアにインポートします。

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

21. ローカルディストリビュータ用の鍵ペアを生成し、ローカルディストリビュータのプライベートストアに格納します。`-mode upstream` を指定します。

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

22. ローカルディストリビュータ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-upstream.cert -alias ldhostname-upstream.cr.com
```

23. 手順 21 でエクスポートしたローカルディストリビュータの自己署名付き証明書をリモートエージェントマシンにコピーします。
24. 自己署名付き証明書をリモートエージェントのトラストストアにインポートします。

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

25. ローカルディストリビュータとリモートエージェントが稼働している場合は、これらを停止します。

例 **8–4** SSL サーバーおよびクライアント認証を構成する方法　（続き）

26. ローカルディストリビュータとリモートエージェントを起動します。
27. ローカルディストリビュータとリモートエージェントのキーストアのパスワードを入力します。
28. 新しいホストを作成します。
29. 新しいホストで、接続タイプ SSL のリモートエージェントを追加します。
30. リモートエージェントとの接続をテストします。

例 **8–5** CLI Client とマスターサーバー間の SSL 認証を構成する方法

1. マスターサーバーと CLI Client 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。暗号群を選択するように求められる際に、認証を伴う暗号化を選択します。
2. マスターサーバー用の鍵ペアを生成し、マスターサーバー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 Client 用の鍵ペアを生成し、CLI Client 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. マスターサーバー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. マスターサーバーの自己署名付き証明書を CLI Client にコピーします。
6. 自己署名付き証明書を CLI Client 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 Client 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 Client の自己署名付き証明書をマスターサーバーにコピーします。
9. 自己署名付き証明書をマスターサーバー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. マスターサーバーが実行中の場合は、停止します。
11. マスターサーバーを起動します。
12. マスターサーバーのキーストアのパスワードを入力します。
13. CLI Client で、`config.properties` ファイルを編集し、次の行をこのファイルに含めます。

    `net.ssl.key.store.pass=`*trust-store-password*
14. CLI Client コマンドを実行し、接続を検証します。

SSL 暗号群

暗号群

次に、IBM AIX 以外のすべてのサポートオペレーティングシステムでサポートされる SSL 暗号群を示します。

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
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA
```

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
TLS_DH_anon_WITH_AES_128_CBC_SHA
```

以下は暗号化を伴わない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
```

IBM AIX の暗号群

次に、IBM AIX サーバーでサポートされる SSL 暗号群を示します。

以下の暗号群はすべて、リモートエージェントに使用できます。サーバー認証を要求しない暗号群は、ローカルディストリビュータには使用できません。

以下は、サーバー認証を要求する暗号群です。

```
SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_DES_CBC_SHA
```

```
SSL_RSA_FIPS_WITH_DES_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
```

以下は、サーバー認証を要求しない暗号群です。

注 – サーバー認証を要求しない暗号群は、ローカルディストリビュータには使用できません。

```
SSL_DH_anon_WITH_RC4_128_MD5
SSL_DH_anon_WITH_DES_CBC_SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
```

以下は暗号化を伴わない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 9 章

Java 仮想マシンのセキュリティーポリシーの構成

この章では、N1 Service Provisioning System 5.1 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からの接続しか受け付けず、特定の IP アドレスおよびポート範囲にしか接続しないようにするセキュリティーポリシーの構成方法を示します。

JVM セキュリティーポリシーの構成

各 N1 Service Provisioning System 5.1 アプリケーションの `lib/security/rox.policy` には、Java 仮想マシン (JVM) のセキュリティーポリシーファイルが存在します。このファイルには、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ています。このポリシーファイルのインストール時点では、アプリケーションはどのホストへも接続でき、どのホストからの接続でも受け付けることができます。SSH 接続で CLI Client を使用する場合は、ポリシーファイルを変更して接続をローカルホストだけに制限してください。

これらのアクセス権は、`lib/security/rox.policy` ファイル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機能を制限する場合は、この行を削除し、これよりも制限の厳しいアクセス権を追加します。

SocketPermission のホストパラメータは次のようになります。

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

hostname にはマシンのホスト名を指定し、*IPaddress* にはマシンの 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]
```

セキュリティーポリシーファイルの構文については`http://java.sun.com/j2se/1.4.2/docs/guide/security/PolicyFiles.html`を参照し、「Policy File Syntax」リンクをクリックしてください。

▼ マスターサーバーの JVM ポリシーを構成する

手順 **1.** **`lib/security/rox.policy`** ファイルを編集します。

2. アプリケーションがすべてのホストに接続し、すべてのホストからの接続を受け付けるように許可する行を削除します。

3. 次の行を追加し、アプリケーションに選択的なアクセス許可を付与します。

```
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 Client がマスターサーバーとの接続に使用するポートです。最初の行は、マスターサーバーがローカル接続または `ssh-proxy` を経由した接続だけをCLI Client に許可するように制限しています。
- *dbport* は、Postgres データベースサーバーのポート番号です。
- *domain* は、ブラウザインタフェース に接続を許可されるホストのドメインです。*httpport* は、ブラウザインタフェース のポート番号です。
- *ipAddress1:port1* と *ipAddress2:port2* は、マスターサーバーに直接接続するリモートエージェントまたはローカルディストリビュータの IP アドレスとポート番号です。

▼ リモートエージェントの JVM ポリシーを構成する

手順 **1.** **`lib/security/rox.policy`** ファイルを編集します。

2. アプリケーションがすべてのホストに接続し、すべてのホストからの接続を受け付けるように許可する行を削除します。

3. 次の行を追加し、アプリケーションにアクセス許可を付与します。

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

ipAddress は、このリモートエージェントが接続するローカルディストリビュータまたはマスターサーバーの IP アドレスです。

参考 ホストに接続するためのアクセス許可の追加

`urltest` などのネットワークアクセスを必要とするステップを含むプランを実行する場合、特定のホストへの接続を許可するアクセス権をこのリモートエージェントに追加したいことがあります。

▼ ローカルディストリビュータの JVM ポリシーを構成する

手順 1. **`lib/security/rox.policy`** ファイルを編集します。

2. アプリケーションがすべてのホストに接続し、すべてのホストからの接続を受け付けるように許可する行を削除します。

3. 次の行を追加し、アプリケーションに選択的なアクセス許可を付与します。

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

- *ipAddress* は、このローカルディストリビュータの親であるローカルディストリビュータまたはマスターサーバーの IP アドレスです。
- *ipAddress1:port1* と *ipAddress2:port2* は、このローカルディストリビュータの子であるリモートエージェントまたはローカルディストリビュータの IP アドレスとポート番号です。

Postgres セキュリティー

Postgres データベースがほかのホストからの接続を受け入れない構成になっていることを確認します。Postgres データベースのデフォルト構成では、UNIX ソケットと `localhost` からの接続を受け入れます。`server/postgres/data/pg_hba.conf` 構成ファイル内のこのデフォルト設定を変更します。また、インストール後、[alter user username with password 'password'] クエリーを使用して、データベースパスワードを変更します。Postgres 構成ファイルに対するこれらの変更を *N1SPS5.1-MasterServer-home*`/config/config.properties` ファイルを使用して行う場合、`db.password` の値を変更する必要があります。

第 10 章

N1 Service Provisioning System 5.1 へのアップグレード

この章では、5.0 バージョンの製品を N1 Service Provisioning System 5.1 へアップグレードする手順を示します。

注 – バージョン 5.0 以前にリリースされたソフトウェアバージョンを使用している場合は、N1 Service Provisioning System 5.1 にアップグレードする前にバージョン 5.0 にアップグレードする必要があります。バージョン 4.0 以前にリリースされたソフトウェアバージョンを使用している場合は、バージョン 4.1 にアップグレードする前にバージョン 4.0 にアップグレードする必要があります。

- 115 ページの「アップグレードの概要」
- 117 ページの「マスターサーバーのアップグレード」
- 121 ページの「リモートエージェントとローカルディストリビュータのアップグレード」

アップグレードの概要

アップグレードの手順は、N1 Service Provisioning System 5.1 がサポートするオペレーティングシステムバージョンで N1 Service Provisioning System 5.0 が稼働しているかどうかによって異なります。以前にサポートされていたオペレーティングシステムでも、新バージョンの N1 Service Provisioning System でサポートされない場合があります。

アップグレードの要件

N1 Service Provisioning System アプリケーションのどれかをアップグレードする場合、5.1 アプリケーションをインストールする新しいサーバーは次の要件を満たしている必要があります。

- オペレーティングシステム – 5.1 アプリケーションを実行するオペレーティングシステムは、5.0 アプリケーションを実行していたオペレーティングシステムと同じタイプでなければなりません。たとえば、Red Hat Linux Advanced Server 2.1 で稼働するマスターサーバーを Red Hat Linux Advanced Server 3.0で稼働するマスターサーバーに移行できます。しかし、Red Hat Linux Advanced Server 2.1 で稼働するマスターサーバーを Solaris OS のいずれかのバージョンで稼働するマスターサーバーに移行することはできません。
- ハードウェアアーキテクチャ – 5.1 アプリケーションを実行するサーバーのハードウェアアーキテクチャは、5.0 アプリケーションを実行していたアーキテクチャと同じでなければなりません。たとえば、Solaris OS で稼働する SPARC ベースサーバーから Solaris OS で稼働するほかの SPARC ベースサーバーにリモートエージェントをアップグレードできます。しかし、Solaris OS で稼働する SPARC ベースサーバーから Solaris OS で稼働する x86 ベースのサーバーにリモートエージェントをアップグレードすることはせきません。
- アプリケーションのユーザー所有権 – 5.1 アプリケーションは、5.0 アプリケーションを所有するユーザーが所有するように設定する必要があります。たとえば、5.0 マスターサーバーをインストールしてユーザー `foo` に所有権を割り当てた場合、5.1 マスターサーバーにもその所有者として `foo` を割り当てる必要があります。

アップグレード – 作業概要

次に、N1 Service Provisioning System 5.0 から N1 Service Provisioning System 5.1 に正しくアップグレードするために必要な作業の概要を示します。

1. アップグレードするサーバーが N1 Service Provisioning System 5.1 を稼働させる上での最小要件を満たしているかどうかを確認します。

   第 2 章を参照してください。
2. 作業を行い、マスターサーバーをアップグレードします。
3. 指示に従って、リモートエージェントとローカルディストリビュータをアップグレードします。

   121 ページの「リモートエージェントとローカルディストリビュータのアップグレード」を参照してください。
4. CLI Clients はアップグレードする必要はありません。5.1 バージョンの CLI Client をインストールし、5.0 バージョンをアンインストールしてください。

マスターサーバーのアップグレード

マスターサーバーアプリケーションのアップグレードは、通常のソフトウェアのようには行われません。まず、新バージョンのマスターサーバーを旧バージョンのマスターサーバーと同じサーバーにインストールします。続いて、旧バージョンのマスターサーバーから新バージョンのマスターサーバーにデータを移行します。

▼ マスターサーバーのデータを移行する

5.0 バージョンのマスターサーバーデータを 5.1 マスターサーバーに移行すると、5.1 バージョンのマスターサーバー内のデータはすべて削除されます。マスターサーバーの両バージョンは、移行作業が完了するまで移行スクリプトによって停止されます。移行作業が進行している間、マスターサーバーは利用できません。

始める前に　アップグレードするサーバーのオペレーティングシステムバージョンが N1 Service Provisioning System 5.1 でサポートされているか確認してください。サポートされているオペレーティングシステムのリストは、25 ページの「オペレーティングシステム要件」を参照してください。

移行前にデータのバックアップをとってください。『Sun N1 Service Provisioning System 5.1 システム管理者ガイド』の第 9 章「バックアップと復元」を参照してください。

手順

1. **5.0** バージョンのマスターサーバーをインストールしたユーザーでマシンにログインします。
2. **5.1** バージョンのマスターサーバーを **5.0** バージョンのマスターサーバーがインストールされているサーバーにインストールします。
   - Linux または UNIX システムを実行している場合は、45 ページの「Linux および UNIX システムに N1 Service Provisioning System 5.1 をインストールする」の説明に従って N1 Service Provisioning System 5.1 マスターサーバーをインストールします。
   - Windows を使用している場合は、次の操作を行います。
     a. Windows Administrative Tools 内の Service アプリケーションを使用して IPC Daemon サービスを停止することにより、5.0 マスターサーバーを停止します。
     b. 5.0 マスターサーバー (具体的には IPC Daemon と Server) を手動で起動するように設定します。
     c. 57 ページの「Windows に N1 Service Provisioning System 5.1 マスターサーバーをインストールする」 の説明に従って、5.1 バージョンのマスターサーバーをインストールします。

5.0 バージョンのマスターサーバーマスターサーバーを所有するユーザーとグループによる所有権を指定して 5.1 マスターサーバーをインストールします。

3. **Linux** および **UNIX** マスターサーバーの場合、移行処理中にデータベースが最適化されないようにしてください。

   データの移行中にデータベースの最適化を開始する cron ジョブがスケジュールされていないかを確認してください。

4. コマンドプロンプトにアクセスします。

   - Linux および UNIX サーバーでは、シェルウィンドウを開いてマスターサーバーを所有するユーザーとしてログインします。
   - Windows では、Command Prompt ウィンドウを開きます。

5. 移行スクリプトが入っているディレクトリに移動します。

   - Linux および UNIX サーバーでは、次のように入力します。

     ```
     % cd /N1SPS5.1-home/server/bin/migrate
     ```

     N1SPS5.1-home には、マスターサーバーをインストールしたディレクトリを指定します。

   - Windows では、次のように入力します。

     ```
     C:\> cd C:\N1SPS5.1-home\server\bin\migrate
     ```

     C:\N1SPS5.1-home には、マスターサーバーをインストールしたディレクトリを指定します。

6. 移行スクリプトを起動します。

   - Linux および UNIX サーバーでは、次のように入力します。

     ```
     % ./migrateMS_5.0-5.1.sh
     ```

   - Windows では、次のように入力します。

     ```
     C:\N1SPS5.1-home\5.1\server\bin\migrate\>.\migrateMS_5.0-5.1.cmd
     ```

7. 画面上に表示される説明に従って移行作業を進めます。

   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、ブラウザインタフェース、およびマスターサーバーのリスナーポート番号は移行されません。N1 Service Provisioning System 5.1 マスターサーバーは、インストール時に指定されたポート番号を使用します。

---

8. ログファイルをチェックし、移行中にエラーが発生していないか確認します。

   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。

9. マスターサーバー ブラウザインタフェースへのアクセスに使用するブラウザのキャッシュを消去します。

   マスターサーバーをアップグレードする前にブラウザセッションを起動した場合は、ブラウザにキャッシュ化されているグラフィックとスタイルシートのために、アップグレートされたブラウザインタフェースが表示されなくなる可能性があります。

10. 『**Sun N1 Service Provisioning System 5.1** システム管理者ガイド』の第 **9** 章「バックアップと復元」の説明に従って、新しいマスターサーバーに移行したデータのバックアップを作成します。

    5.0 マスターサーバーのデータを 5.1 マスターサーバーへ復元することはできません。必要に応じて使用できる正確なバックアップ一式を保持するため、5.1 マスターサーバーデータをバックアップしてください。

11. **(**省略可能**) 5.0** バージョンのマスターサーバーをアンインストールします。

    5.0 バージョンのマスターサーバーを使用しない場合は、第 11 章の説明に従ってアンインストールできます。

マスターサーバーのデータ移行の詳細

この節では、旧バージョンのマスターサーバーから新バージョンのマスターサーバーに移行されるデータについて詳しく説明します。

マスターサーバーのデータ移行

次に、マスターサーバー上で移行されるデータの種類を示します。

表 10–1 移行の概要

マスターサーバー上のデータ	移行対象のデータですか	移行手段
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 Client スクリプト	いいえ	
CLI Client でシリアル化されるオブジェクトの移行	いいえ	
各ホスト上の `config.properties` ファイルに対する変更の移行	はい	このファイルにリストされているプロパティは、120 ページの「プロパティファイルの移行についての詳細」で説明されている方法を使用して移行する

表 **10–1** 移行の概要　(続き)

マスターサーバー上のデータ	移行対象のデータですか	移行手段
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
ロガー構成ファイル	いいえ	
ユーザーインタフェースのカスタマイズ	いいえ	

プロパティファイルの移行についての詳細

5.0 `config.properties` は、5.1 `config.properties` ファイルへ移行されます。この移行では、5.0 ファイルに入っている各プロパティーの値が、5.1 `config.properties` ファイルに入っているプロパティーの値と比較されます。値が同じであれば、そのプロパティは無視されます。値が異なる場合は、5.0 の値が 5.1 の `config.properties` ファイルにコピーされます。プロパティが、5.0 バージョンのファイルに存在するが、5.1 バージョンのファイルには存在しない場合は、5.0 バージョンの値が 5.1 バージョンのファイルに追加されます。次に示すプロパティの値は、5.1 バージョンの `config.properties` ファイルに移行されません。

- `webserver.TomcatHome`
- `rsrc.localrepo`
- `db.port`
- `db.password`
- `db.hostname`
- `db.instancename`
- `db.username`
- `db.type`
- `db.maxconnections`
- `hostdb.ms.ipaddress`
- `hostdb.ms.port`
- `note.mailsubject`
- `net.server.nconn`
- `net.server.type.1`
- `net.server.ip.1`
- `net.server.port.1`
- `net.server.parms.1`
- `note.url`
- `pe.defaultUserToRunAs`
- `hostdb.ms.connectiontype`
- `pe.maxSimulPlans`

5.0 のプロパティーファイルでこれらのプロパティーのいずれかの値を変更した場合は、5.1 `config.properties` ファイルの値を手動で変更する必要があります。

リモートエージェントとローカルディストリビュータのアップグレード

N1 Service Provisioning System 5.1 には、マスターサーバーのブラウザインタフェースを使用してリモートエージェントとローカルディストリビュータをアップグレードできる自動アップグレードユーティリティがあります。

▼ リモートエージェントとローカルディストリビュータをアップグレードする

始める前に　リモートエージェントとローカルディストリビュータをアップグレードする前にマスターサーバーを移行します。

手順

1. **N1 Service Provisioning System 5.1** マスターサーバーのブラウザ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. 「**masterserver**」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。
   ウィンドウが開かれ、アップグレード中のホストの一覧と、アップグレードの進捗状態が表示されます。処理が完了したところで、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** 段階を完了するために、「**Update Entire N1 SPS network...**」ボタンをもう一度クリックします。
   アップグレード中のホストの一覧と、アップグレードの進捗状態が表示されます。処理が完了したところで、各ホストのステータスが「Updated」と表示されます。
7. 「**Close**」ボタンをクリックします。
   以上でアップグレードが完了します。
8. アップグレードしたホストの準備作業を行います。
   アップグレードしたホストでプランを実行する前に、そのホストの準備作業を行う必要があります。ホストの準備作業を行う場合は、『Sun N1 Service Provisioning System 5.1 システム管理者ガイド』の「物理ホストを準備する」の説明に従ってください。

第 11 章

N1 Service Provisioning System 5.1 のアンインストール

この章では、N1 Service Provisioning System 5.1 をアンインストールする方法について説明します。内容は次のとおりです。

- 123 ページの「Linux および UNIX システムでのアプリケーションのアンインストール」
- 126 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」

Linux および UNIX システムでのアプリケーションのアンインストール

N1 Service Provisioning System 5.1 をアンインストールする方法は、アンインストールするアプリケーションと、そのアプリケーションを稼働させているオペレーティングシステムによって異なります。

- Solaris OS 上のマスターサーバーまたは CLI Client をアンインストールする場合は、123 ページの「Solaris OS 上のマスターサーバーまたは CLI Client をアンインストールする」の説明に従ってください。
- Solaris OS 上のリモートエージェントまたはローカルディストリビュータ、あるいは Linux または UNIX サーバー上のアプリケーションのどれかをアンインストールする場合は、124 ページの「Linux および UNIX 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」の説明に従ってください。

▼ Solaris OS 上のマスターサーバーまたは CLI Client をアンインストールする

Solaris OS のマスターサーバーと CLI Client は、パッケージとしてインストールされます。アンインストールスクリプトでは、5.1 バージョンのマスターサーバーまたは CLI Client だけがアンインストールされます。

手順 **1.** アプリケーションをアンインストールするサーバーで、アンインストールするアプリケーションのディレクトリ以外の場所に移動します。

2. アンインストールを開始します。

```
# /N1SPS5.1-home/app_directory/bin/cr_uninstall_app.sh
```

N1SPS5.1-home には、アプリケーションをインストールしたディレクトリを指定します。デフォルトディレクトリは、`/opt/SUNWn1sps/N1_Service_Provisioning_System_5.1` です。*app_directory* には、次に示す値の 1 つを指定します。

- server – マスターサーバーをアンインストールする
- cli – CLI Client をアンインストールする

app には、次に示す値の 1 つを指定します。

- ms – マスターサーバーをアンインストールする
- cli – CLI Client 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、マスターサーバーをアンインストールする場合は ms を、CLI Client をアンインストールする場合は cl を指定します。

注 – `SUNWspsc1` パッケージと `SUNWspsj1` パッケージは、このサーバーにほかのアプリケーションがインストールされている場合は削除されません。たとえば、同じサーバーにマスターサーバーと CLI Client の両方をインストールしてある場合で、マスターサーバーだけをアンインストールするとき、`SUNWspsc` と `SUNWspsj1` パッケージは CLI Client がアンインストールされるまでサーバー上に残されます。

▼ Linux および UNIX 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.** アプリケーションをアンインストールするサーバーで、アンインストールするアプリケーションのディレクトリ以外の場所に移動します。

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. リモートエージェント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許可を変更します。

```
% chmod -R 755 /N1SPS5.1-home/agent/bin/protect
```

N1SPS5.1-home には、リモートエージェントをインストールした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入っているディレクトリ削除します。

   ```
   # rm -r /N1SPS5.1-home/app-directory
   ```

   N1SPS5.1-home には、アプリケーションをインストールしたディレクトリを指定します。UNIX システム上のデフォルトディレクトリは、`/opt/SUNWn1sps/` です。Linux システム上のデフォルトディレクトリは、 `/opt/sun` です。*app-directory* には、次に示す値の 1 つを指定します。

   - server – マスターサーバーをアンインストールする
   - agent – リモートエージェントをアンインストールする
   - cli – CLI Client をアンインストールする
   - ld – ローカルディストリビュータ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で、*N1SPS5.1–home* ディレクトリにアプリケーションディレクトリが残っていないときは、**common/** ディレクトリを削除します。

   ```
   # rm -r N1SPS5.1-home/common
   ```

   以上でアンインストール作業が完了します。

▼ データベースの自動最適化を無効にする

Red Hat Linux マスターサーバーをアンインストールする場合は、データベースを自動的に最適化するようにシステムに指示するエントリを `crontab` ファイルから手動で削除する必要があります。Solaris マスターサーバーのアンインストールスクリプトは、`cronjob` ファイルからこのエントリを自動的に削除します。

手順 1. マスターサーバーを所有するユーザーとして、現在の **crontab** を表示し、出力がファイルになされるように指定します。

   ```
   # crontab -l > newcrontabfile
   ```

2. **newcrontab** ファイルをテキストエディタで開きます。

3. **newcrontab** ファイルから次の行を削除します。

   ```
   MM HH * * * N1SPS5.1-home/server/bin/roxdbcmd vacuumdb -d rox > /dev/null 2> /dev/null
   ```

   N1SPS5.1-home には、マスターサーバーのホームディレクトリを指定します。

4. **newcrontab** ファイルを保存します。

5. **crontab** を更新します。

   ```
   # crontab newcrontabfile
   ```

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 サーバー上のアプリケーションをアンインストールするには、Windows の「コントロール パネル」にある「プログラムの追加と削除」機能を使用します。アンインストールを行う際には、Microsoft Management 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) が開いていない状態にしてください。「サービス」スナップインが開いていると、マスターサーバー、リモートエージェント、またはローカルディストリビュータを正常にアンインストールできないことがあります。

付録 A

インストールと構成のリファレンス

この付録では、N1 Service Provisioning System 5.1 のインストールについて詳しく説明します。内容は次のとおりです。

- 127 ページの「Linux および UNIX システム上の N1 Service Provisioning System 5.1 用の参照データ」
- 133 ページの「Windows 上の N1 Service Provisioning System 5.1 用の参照データ」

Linux および UNIX システム上の N1 Service Provisioning System 5.1 用の参照データ

この節では、Linux および UNIX システム上にインストールされている N1 Service Provisioning System 5.1 についての詳細を示します。内容は次のとおりです。

- 128 ページの「Linux および UNIX システム上の N1 Service Provisioning System 5.1 のディレクトリ構造」
- 130 ページの「Linux および UNIX システムでのデータベースの最適化」
- 131 ページの「Linux および UNIX システム向けのリモートエージェントパラメータサンプル」

Linux および UNIX システム上の N1 Service Provisioning System 5.1 のディレクトリ構造

N1 Service Provisioning System 5.1 をインストールする際には、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。UNIX システム上のデフォルトディレクトリは、 `/opt/SUNWn1sps` です。Linux システム上のデフォルトディレクトリは、 `/opt/sun` です。ホームディレクトリには、インストールプログラムによって次のディレクトリが作成されます。

- `N1_Service_Provisioning_System_5.1`: マスターサーバーと CLI Client のためのソフトウェア格納ディレクトリ
- `N1_Service_Provisioning_System`: ローカルディストリビュータとリモートエージェントのためのソフトウェア格納ディレクトリ

N1 Service Provisioning System 5.1 ソフトウェアは、インストールスクリプトにより、その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。次の表に示されているものを除き、すべてのディレクトリはアクセス権 755 (`rwxr-xr-x`) で作成されます。実行可能ファイルとスクリプト (これらは 755 に設定される) を除き、ほとんどのファイルにはアクセス許可 644 (`rw-r--r`) が割り当てられます。

次に、N1 Service Provisioning System 5.1 のすべてのアプリケーション (マスターサーバー、ローカルディストリビュータ、リモートエージェント、CLI Client) でインストールされる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 JRE のバンドル版コピー
`/common/lib`	一部またはすべてのアプリケーションに共通のライブラリファイル

次に、マスターサーバー用にインストールされるディレクトリを示します。

表 **A–2** マスターサーバー用にインストールされるディレクトリ

ディレクトリ	内容
`/server/config`	マスターサーバーの構成ファイル
`/server/custom`	ユーザーインタフェースのカスタマイズファイル
`/server/data`	マスターサーバーのデータファイル
`/server/bin`	マスターサーバーの実行可能ファイル

表 A–2 マスターサーバー用にインストールされるディレクトリ　(続き)

ディレクトリ	内容
/server/lib	マスターサーバー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/tomcat	Apache Tomcat のバンドル版コピー
/server/webapp	ブラウザインタフェース Web アプリケーション
/server/setup	マスターサーバーの初期化に使用するさまざまなファイル
/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	アクセス許可が 777 に設定された、マスターサーバーの一時ディレクトリ
/server/README	テキストライセンス契約

次に、ローカルディストリビュータ用にインストールされるディレクトリを示します。

表 A–3 ローカルディストリビュータ用にインストールされるディレクトリ

ディレクトリ	内容
/ld/config	ローカルディストリビュータの構成ファイル
/ld/bin	ローカルディストリビュータの実行可能ファイル
/ld/lib	ローカルディストリビュータのライブラリファイル
/ld/data	ローカルディストリビュータ固有のデータ
/ld/data/tmp	アクセス許可が 777 に設定された、ローカルディストリビュータの一時ディレクトリ
/ld/jvm/jre/bin	ローカルディストリビュータの JRE プロキシ
/ld/README	テキストライセンス契約

次に、リモートエージェント用にインストールされるディレクトリを示します。

表 A–4 リモートエージェント用にインストールされるディレクトリ

ディレクトリ	内容
/agent/config	リモートエージェントの構成ファイル
/agent/bin	リモートエージェントの実行可能ファイル

表 **A–4** リモートエージェント用にインストールされるディレクトリ　(続き)

ディレクトリ	内容
/agent/bin/protect	アクセス許可が 100 (--x------) に設定された Jexec ディレクトリ
/agent/bin/protect/jexec	Jexec は、アクセス許可が 4110 に設定された root 権限が必要な場合に使用される
/agent/lib	リモートエージェントのライブラリファイル
/agent/data	リモートエージェント固有のデータ
/agent/work	execNatives の実行用のデフォルトディレクトリ
/agent/data/tmp	アクセス許可が 777 に設定された、リモートエージェントの一時ディレクトリ
/agent/jvm/jre/bin	リモートエージェントの JRE プロキシ
/agent/README	テキストライセンス契約

次に、CLI Client 用にインストールされるディレクトリを示します。

表 **A–5** CLI Client 用にインストールされるディレクトリ

ディレクトリ	内容
/cli/config	CLI の構成ファイル
/cli/bin	CLI の実行可能ファイル
/cli/lib	CLI のライブラリファイル
/cli/data	CLI 固有のデータ
/cli/data/tmp	アクセス許可が 777 に設定された、CLI の一時ディレクトリ
/cli/README	テキストライセンス契約

Linux および UNIX システムでのデータベースの最適化

インストールプログラムは、データベースの最適化を毎日行うかどうかを尋ねるメッセージを表示します。データベースの最適化を毎日行うことを選択すると、インストールスクリプトによって次のコマンドが cronjob ファイルに追加されます。このコマンドは、日次ベースでデータベース最適化を開始する任意の時点で cronjob ファイルに追加できます。

```
MM HH * * * N1SPS5.1-home/server/bin/roxdbcmd vacuumdb -d rox > /dev/null 2> /dev/null
```

N1SPS5.1-home には、マスターサーバーのホームディレクトリを指定します。

Linux および UNIX システム向けのリモートエージェントパラメータサンプル

マスターサーバーの /server/bin ディレクトリには、マスターサーバーのインストール時にほかのスクリプトと共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。ユーザーはこのファイルを利用して構成内容を選択し、非対話方式でリモートエージェントをインストールできます。以下に、このサンプル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_remote_params.sh,v 1.4 2005/05/19 23:52:07 echiquet Exp $

# CR_RA_INSTALLBASE - the base directory where the
# Remote Agent will be installed.  If the directory
# does not exist, the installer will attempt to create it.
# Defaults to /opt/SUNWn1sps
#
CR_RA_INSTALLBASE=/opt/SUNWn1sps

# CR_RA_OWNER - The owner of the distribution. A pre-existing
# user must be specified.  Defaults to 'n1sps'.
#
CR_RA_OWNER=n1sps

# CR_RA_GROUP - The group owner of the distribution.  A
# pre-existing group name must be specified.   Defaults to 'n1sps'.
#
CR_RA_GROUP=n1sps

# CR_RA_PORT - Port number that the Remote Agent will listen on.
# An integer value between 1024 and 65535 must be specified.  Defaults
# to 1131.
#
CR_RA_PORT=1131

# CR_RA_CTYPE - Parent connection type.  How the parent connects to
# this RA.  One of 'raw' (unencrypted), 'ssh', or 'ssl'.  There is no default.
# This parameter is required.
#
#CR_RA_CTYPE=raw

# CR_RA_CIPHER_TYPE - SSL cipher suite type.  One of '1' (encryption,
# no authentication) or '2' (encryption, with authentication).
# Default is 1, but has no effect for parent connection type of raw or
# ssh.
#
CR_RA_CIPHER_TYPE=1

# CR_RA_CTYPE_CONFIRM - Selection confirmation for an insecure
```

```
# connection type, such as 'raw' or 'ssl with no authentication'. If set to
# anything else than 'true', the installation will fail for such connection
# types.
CR_RA_CTYPE_CONFIRM=false

# CR_RA_INSTALL_JRE - Directive of whether or not a JRE should be
# installed with the Remote Agent for it's use.  Defaults to 'y'.  Valid
# values are 'y' or 'n'.
#
CR_RA_INSTALL_JRE=y

# JRE_HOME - Directive for the location of the JRE installation.  If
# the CR_RA_INSTALL_JRE directive is set to 'y', the installer will
# install the JRE.  In this case, the JRE_HOME value will be
# $CR_RA_INSTALLBASE/common/jre.  If the installer is not going to
# install the JRE, the JRE_HOME should point to where the pre-existing JRE
# is installed.
#
JRE_HOME=$CR_RA_INSTALLBASE/N1_Service_Provisioning_System/common/jre


# CR_RA_SUID - Directive of whether or not the RA should be installed
# with the setuid root privileges.  Valid values are 'y' or 'n'.  This
# only works when the remote installer is run as the root user.
# There is no default. This parameter is required.
#
#CR_RA_SUID=y

# CR_RA_INSTALLER_USER - The user that should perform this install.  This
# is what the remote installer will use to ssh into the remote hosts
# and run the commands as.  It is highly recommended that this be set to
# root, although, it doesn't have to be.  Defaults to the current user.
#
CR_RA_INSTALLER_USER=root

# CR_RA_INSTALLER_WORKDIR - The directory to use to store temporary files.
# The distribution will be copied into this directory so make sure
# that this it has enough space to store the distribution file.  Defaults to
# /tmp
#
CR_RA_INSTALLER_WORKDIR=/tmp

# CR_RA_INSTALLER_LEAVEFILES - Directive of whether or not the temporary
# files should be preserved on the remote host.  Defaults to 'n'.
#
CR_RA_INSTALLER_LEAVEFILES=n

# CR_RA_INSTALLER_HOSTS - List of remote hosts on which the Remote Agent is
# to be installed.  This must contain at least one host name.  This host list
# can also be set in the  environment  variable 'CR_RA_INSTALLER_HOSTS', or
# specified on the command line.  Check the remote agent installer script
# usage message for exactly how this can be done.
#
# Note : The format of the list of hosts is critical.  The list of hosts
# must be separated by a comma (',') and cannot have any spaces in between.
```

```
# It must be in one contiguous string.
#
CR_RA_INSTALLER_HOSTS=""

export CR_RA_INSTALLBASE CR_RA_PORT CR_RA_GROUP CR_RA_OWNER CR_RA_INSTALL_JRE
CR_RA_SUID
export CR_RA_CTYPE CR_RA_CIPHER_TYPE
export CR_RA_INSTALLER_USER CR_RA_INSTALLER_WORKDIR CR_RA_INSTALLER_LEAVEFILES
export CR_RA_INSTALLER_HOSTS  JRE_HOME
```

CR_RA_ALLOWFORWARDVERSION パラメータ

N1 Service Provisioning System 5.1 でサポートされているバージョンよりも高いオペレーティングシステムバージョンに N1 Service Provisioning System 5.1 リモートエージェントをインストールする場合は、パラメータファイルに次のパラメータを追加します。

```
CR_RA_ALLOWFORWARDVERSION=y
```

`CR_RA_ALLOWFORWARDVERSION=y` パラメータを使用すると、インストールプログラムはリモートエージェントがインストールされるオペレーティングシステムがサポートされているかどうかを確認しません。サポートされていないオペレーティングシステムでの N1 Service Provisioning System 5.1 の使用については、Sun サービスは標準サポートを提供しません。



注意 – サポートされていないオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールすると、不確定な動作や予期しない動作が発生することがあります。サポートされていないオペレーティングシステムに対する N1 Service Provisioning System 5.1 のインストールは、テスト目的以外では行わないでください。運用環境では、サポートされていなオペレーティングシステムに N1 Service Provisioning System 5.1 をインストールして使用することは避けてください。

Windows 上の N1 Service Provisioning System 5.1 用の参照データ

この節では、Windows 上にインストールされている N1 Service Provisioning System 5.1 についての詳細を示します。内容は次のとおりです。

- 134 ページの「Windows 上の N1 Service Provisioning System 5.1 のディレクトリ構造」
- 136 ページの「Cygwin」
- 136 ページの「Windows インストールスクリプトによって行われる処理」

Windows 上の N1 Service Provisioning System 5.1 のディレクトリ構造

N1 Service Provisioning System 5.1 をインストールする際には、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。デフォルトディレクトリは、次のいずれかです。

- `C:\Program Files\N1 Service Provisioning System\5.1`: マスターサーバーと CLI Client のためのソフトウェア格納ディレクトリ
- `C:\Program Files\N1 Service Provisioning System` : ローカルディストリビュータとリモートエージェントのためのソフトウェア格納ディレクトリ

N1 Service Provisioning System 5.1 ソフトウェアは、インストールスクリプトにより、その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。次に、N1 Service Provisioning System 5.1 のすべてのアプリケーション (マスターサーバー、ローカルディストリビュータ、リモートエージェント、CLI Client) でインストールされる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アプリケーションに共通のライブラリファイル

次に、マスターサーバー用にインストールされるディレクトリを示します。

表 **A–7** マスターサーバー用にインストールされるディレクトリ

ディレクトリ	内容
`\server\config`	マスターサーバーの構成ファイル
`\server\data`	マスターサーバーのデータファイル
`\server\bin`	マスターサーバーの実行可能ファイル
`\server\lib`	マスターサーバー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`	ブラウザインタフェース Web アプリケーション

表 **A–7** マスターサーバー用にインストールされるディレクトリ　(続き)

ディレクトリ	内容
\server\setup	マスターサーバーの初期化に使用するさまざまなファイル
\server\data\tmp	アクセス許可が 777 に設定された、マスターサーバーの一時ディレクトリ
\server\README	テキストライセンス契約

次に、ローカルディストリビュータ用にインストールされるディレクトリを示します。

表 **A–8** ローカルディストリビュータ用にインストールされるディレクトリ

ディレクトリ	内容
\ld\config	ローカルディストリビュータの構成ファイル
\ld\bin	ローカルディストリビュータの実行可能ファイル
\ld\lib	ローカルディストリビュータのライブラリファイル
\ld\data	ローカルディストリビュータ固有のデータ
\ld\data\tmp	ローカルディストリビュータの一時ディレクトリ
\ld\jvm\jre\bin	ローカルディストリビュータの JRE プロキシ
\ld\README	テキストライセンス契約

次に、リモートエージェント用にインストールされるディレクトリを示します。

表 **A–9** リモートエージェント用にインストールされるディレクトリ

ディレクトリ	内容
\agent\config	リモートエージェントの構成ファイル
\agent\bin	リモートエージェントの実行可能ファイル
\agent\lib	リモートエージェントのライブラリファイル
\agent\data	リモートエージェント固有のデータ
\agent\work	execNatives の実行用のデフォルトディレクトリ
\agent\data\tmp	リモートエージェントの一時ディレクトリ
\agent\jvm\jre\bin	リモートエージェントの JRE プロキシ
\agent\README	テキストライセンス契約

次に、CLI Client 用にインストールされるディレクトリを示します。

表 **A–10** CLI Client 用にインストールされるディレクトリ

ディレクトリ	内容
\cli\config	CLI の構成ファイル
\cli\bin	CLI の実行可能ファイル
\cli\lib	CLI のライブラリファイル
\cli\data	CLI 固有のデータ
\cli\data\tmp	アクセス許可が 777 に設定された、CLI の一時ディレクトリ
\cli\README	テキストライセンス契約

Cygwin

UNIX および Linux システムで稼働しているアプリケーションとの相互運用を促進するため、Windows バージョンのソフトウェアには 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 によって Windows 向けに開発された UNIX 環境です。Cygwin は 2 つの部分から構成されます。その 1 つは、実質的な UNIX API 機能を提供する UNIX エミュレーション層として機能する DLL (cygwin1.dll) です。もう 1 つは、UNIX から移植されたツール群です。これらのツールは、UNIX/Linux の外観と操作性を実現します。Cygwin DLL は、Windows 95 移行のすべての Windows に対応しています (ただし、ベータ版、リリース候補、ix86 バージョンと、Windows CE を除く)。

Windows インストールスクリプトによって行われる処理

Windows マスターサーバーのインストールスクリプトは、次の処理を行います。

- ユーザーが指定するディレクトリにすべてのインストールコンテンツをコピーする
- 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マスターサーバー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 に必要な構成ファイルを生成するスクリプトを実行する

Windows ローカルディストリビュータのインストールスクリプトは、次の処理を行います。

- ユーザーが指定するディレクトリにインストールコンテンツをコピーする
- 通信プロトコルとして SSL を選択した場合、SSL に必要な構成ファイルを生成するスクリプトを実行する
- ローカルディストリビュータ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ローカルディストリビュータを起動する設定にしている場合、ローカルディストリビュータを起動する

Windows リモートエージェントのインストールスクリプトは、次の処理を行います。

- ユーザーが指定するディレクトリにインストールコンテンツをコピーする
- 通信プロトコルとして SSL を選択した場合、SSL に必要な構成ファイルを生成するスクリプトを実行する
- リモートエージェント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 Client のインストールスクリプトは、次の処理を行います。

- ユーザーが指定するディレクトリにインストールコンテンツをコピーする
- 通信プロトコルとして SSL を選択した場合、SSL に必要な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 5.1 のインストールと構成に関連したトラブルシューティング情報を挙げます。

- 139 ページの「Linux および UNIX システムのインストール時の問題」
- 140 ページの「Microsoft Windows にインストールする際の問題」
- 141 ページの「SSH 接続」

Linux および UNIX システムのインストール時の問題

IBM AIX にインストールする際のパッケージ破損エラー (6279820)

プロビジョニングシステムを IBM AIX 5.2 または 5.3 システムにインストールするときに、次のエラーが表示される場合があります。

```
Checking the integrity of the package.
Error! This package seems to be corrupted.
Use another copy of the package.
```

回避策: IBM AIX オペレーティングシステムに問題があると、インストールが失敗する原因となります。次のパッチのいずれかをインストールして、問題を修正します。

- IBM AIX 5.2 システムでは、IBM APAR IY68428 パッチをインストールします。このパッチについては、`http://www-1.ibm.com/support/docview.wss?uid=isg1IY68428`を参照してください。

- IBM AIX 5.3 システムでは、IBM APAR IY67843 パッチをインストールします。このパッチについては、http://www-1.ibm.com/support/docview.wss?uid=isg1IY67843を参照してください。

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]
```

AIX マシンに JRE をインストールする際には、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。キャッシングの際に、これらのライブラリはディスク上でロックされます。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。

新バージョンの JRE をインストールしないでください。JRE をインストールするプロンプトが表示された場合には、no を選択し、マシンにすでにインストールされている JRE のパスを入力してください。

Microsoft Windows にインストールする際の問題

Windows にインストールする際のエラー

Windows サーバーにインストールする際には、次のメッセージが表示されます。

```
/!\ Internal Error 2755.
```

このエラーは、MSI パッケージの保存先であるディレクトリに対して書き込み許可を持っていない場合に表示されます。インストールを行うためには、インストールプログラムを実行するユーザーの書き込み許可が含まれるようにディレクトリのアクセス許可を変更し、インストールを再開します。

SSH 接続

マスターサーバーが中間のローカルディストリビュータを経由してローカルディストリビュータに接続できない

「Host Details」ページでマスターサーバー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、マスターサーバー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、マスターサーバーと接続先マシン間の一部または全部の中間ローカルディストリビュータに対して、`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マスターサーバーへ移動しながら、問題が発生したマシンと個々のアップストリームローカルディストリビュータとの接続をテストしてください。問題が発生したマシンに正常に接続できるローカルディストリビュータの中で最も近いローカルディストリビュータについては、`transport.config` ファイルと、そのすべてのダウンストリームローカルディストリビュータを再生成します。`transport.config` ファイルは、CLI クライアントの `net.gencfg` コマンドを使って生成できます。

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 5.1 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

始める前に　`ssh-agent` を使用している場合は、`ssh-agent` を起動に使用したセッションからこの作業を行います。

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

- アップストリームマシンのすぐ隣りのダウンストリームマシンをテストするには、次のコマンドを使用します。

  ```
  # ssh target-IPaddress ls -l
  ```

  target-IPaddress には、テストするダウンストリームマシンのうちもっとも遠くのマシンの IP アドレスを指定します。

- `ssh-agent` を使用している場合に、`ssh-agent` を実行しているマシンのすぐ隣りではないダウンストリームマシンをテストするには、次のコマンドを使用します。

```
# 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ローカルディストリビュータマシンの 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` はマスターサーバー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マシンのうちもっとも遠くのマシンの 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マシンとダウンストリームマシンの中間にあるマシンの IP アドレスを指定します。

情報の入力を求めるプロンプトが表示されたら必要情報を入力し、テストをもう一度行います。

情報の入力を求める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. **`priority="debug"`** という指定でログを記録するため、アップストリームマシンの **`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。

```
<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** で使用したコマンドを使って、再度接続をテストします。

コマンド行および `SSH.STDERR` ログファイルに出力されたログを確認します。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。

ダウンストリームアプリケーションの起動に使用した SSH コマンドについてアップストリーム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。ログメッセージに示された問題を修正し、再度テストを行います。

ログファイルに問題が現れていない場合は、アップストリームマシンはダウンストリームマシンに正常に接続している可能性がありますが、アプリケーションは正常に起動していません。次の手順に進みます。

4. **`ROX`** ログファイルをチェックし、ダウンストリームマシン上でアプリケーションの起動時にエラーが発生していないか確認します。

- Solaris OS システムでは、`/var/tmp/ROX` *appnumbers*`.log` ファイルを確認します。
- ほかのすべてのシステムでは、`/tmp/ROX` *appnumbers*`.log` ファイルを確認します。

app には、テストするダウンストリームマシン上のアプリケーションを指定します。リモートエージェントには Agent、ローカルディストリビュータには Dist、CLI Client には Proxy を指定してください。*numbers* には、ファイル名の一部としてランダムに生成された数値を指定します。

5. ログファイル内で見つかったエラーを修正します。

用語集

抽象コンポーネント
ほかのコンポーネントを拡張するためのベースコンポーネントとしてのみ機能するコンポーネント。抽象コンポーネントはインストールすることはできません。また、抽象的な子要素を宣言できるのは抽象コンポーネントのみです。

呼び出し互換性
システムサービスコンポーネントの互換性タイプの 1 つ。この互換性は、API 互換性やインタフェース互換性とも呼ばれます。

カテゴリ
複数のフォルダに保存されたグループオブジェクトをグループ化できる一般的なクラス。

子コンポーネント
コンテナコンポーネントによって参照されるコンポーネント。包含コンポーネントとも呼ばれます。

*container component*も参照してください。

比較
ホストとコンポーネントモデル間の相違点を検索し、特定する機能。N1 Service Provisioning System は、次に示す 3 種類の比較をサポートします。

- モデルとモデル – マスターサーバー上に保存された 2 つのホストの配備リポジトリと履歴が検査され、差分が報告されます。
- モデルとインストール – ホストにインタフェース済みであるとマスターサーバーから報告された内容とホストに実際にインストールされている内容が比較され、差分が報告されます。
- インストールとインストール – 2 つのホストのファイルシステムの内容が検査され、差分が報告されます。

コンポーネント
アプリケーションを定義するソース情報の論理グループ。コンポーネントは、ソース情報の管理方法を指定する命令も含みます。

コンポーネントの XML 表現には、以下のものが含まれます。

- アプリケーションが使用するリソースの一覧
- インストール手順

- アンインストール手順
- 依存関係

コンポーネントの互換性	コンポーネントを別のコンポーネントで安全に置換できること。N1 Service Provisioning System は、コンポーネントの互換性として呼び出しの互換性とインストールの互換性の 2 種類をサポートします。
コンポーネントの継承	コンポーネントがほかのコンポーネントから属性と動作を取得する手段。コンポーネントは、その作成時に、関連付けられたコンポーネントタイプから変数、スナップショット、手続きなどを継承します。
コンポーネントの手続き	インストール、アンインストール、管理、スナップショットのキャプチャなど、コンポーネントの配備を制御するコンポーネント内のプログラム。管理手続きは、制御ブロックで定義されます。
コンポーネントのリポジトリ	コンポーネントとそのリソースが登録される、マスターサーバー上の場所。
コンポーネントタイプ	ほかのコンポーネントによって再使用できる動作をカプセル化する特殊なコンポーネント。コンポーネントは、拡張によって特定のコンポーネントタイプの動作を継承できます。
コンポーネント変数	ユーザー定義が可能な、名前と値の組み合わせ。コンポーネント変数は、外部のオブジェクトからコンポーネントの一部にアクセスして設定できるようにするために使用されます。
複合コンポーネント	ほかのコンポーネント (単純コンポーネントまたは複合コンポーネント) に対する参照だけが含まれるコンポーネント。複合コンポーネントはリソースを含むことはできません。
複合プラン	サブプラン (単純サブプランまたは複合サブプラン) だけから成るプラン。　異なるターゲットセットで各サブプランが実行される可能性があるため、複合プランを直接のターゲットとすることはできません。
構成生成エンジン	置換変数参照を適切な変数設定値に置換する、マスターサーバー上のソフトウェアエンジン。このエンジンは、ユーザーがプランを実行してコンポーネントを配備する際に、ホストリポジトリおよびコンポーネントリポジトリとやりとりしながら値を解決します。
包含コンポーネント	ほかのコンポーネントによって参照されるコンポーネント。
コンテナコンポーネント	ほかのコンポーネントに対する参照を含むコンポーネント。
control	配備されたアプリケーションの制御に使用できるコンポーネントによって定義される手続き。たとえばアプリケーションの起動や停止などを制御できます。制御サービスとも呼ばれます。
配備	コンポーネントに対して行われる、プランまたはコンポーネント手続きを実行します。コンポーネントのライフサイクルとしては、インストール、アンインストール、アプリケーション管理などがあります。
直接実行手続き	ブラウザインタフェースを使用してコンポーネントから直接実行できるコンポーネント手続きです。

ダウンストリーム	N1 Service Provisioning System 5.1 のネットワーク階層では、マスターサーバーまでの階層的距離がより遠いサーバー。たとえば、マスターサーバーはダウンストリームをローカルディストリビュータに接続します。このローカルディストリビュータに接続されるリモートエージェントはすべて、ローカルディストリビュータから見てダウンストリームになります。
`execNative` 呼び出し	プランまたはコンポーネントの XML からカスタムスクリプトに対して行われるオプション呼び出しです。
実行プラン	*plan*を参照してください。
拡張	コンポーネントタイプによって定義された変数と手続きをコンポーネントが継承するように、コンポーネントタイプをベースにしてコンポーネントを作成すること。コンポーネントは、そのコンポーネントタイプで定義された変数値と手続き定義を上書きできます。
最終コンポーネント	ほかのコンポーネントによって拡張できないコンポーネント。
フォルダ	コンポーネント、プラン、サブフォルダをまとめることができる、ディレクトリに似たコンテナ。コンテナにはアクセス許可を設定できます。
ゴールドサーバー	アプリケーションを構成するファイル、ディレクトリなどのリソースが格納された参照サーバー。これらのリソースは、ゴールドサーバーによってマスターサーバーにチェックイン (アップロード) されます。
ホスト	N1 Service Provisioning System によって管理されるサーバー。
ホストセット	1 つ以上の属性 (物理的な場所や機能グループなど) を共有するホストを論理的にグループ化したもの。ホストセットはユーザー定義が可能です。ホストセットを使用すると、セット内の全ホストのアプリケーションを簡単にすばやく更新できます。また、2 つのホスト間で「モデルとインストールの比較」を実施するのにも利用できます。
ホストタイプ	ユーザー定義が可能な一連の共通属性によってバインドされる、サーバーの基本クラス。ホストタイプを使用することで、ホストを論理グループとして分類し、ホスト検索の効率を高めることができます。
ホスト検索	ホストリポジトリに対して行われるクエリー。この結果、指定されたものに一致する属性を持つホストの一覧が表示されます。ホスト検索を行うことで、ホストタイプが同じであるホスト、同じアプリケーションを実行しているホスト、同じサブネットマスクが設定されているホストなどの一覧を作成できます。
インストール互換性	コンポーネントタイプの互換性タイプの 1 つです。この互換性は、構造上の互換性とも呼ばれます。
Java Runtime Environment (JRE)	ランタイム環境を再配布するユーザーや開発者向けの Java Development Kit (JDK®) のサブセット。Java Runtime Environment は、Java 仮想マシン (JVM)、Java コアクラス、サポートファイルから構成されます。

Java 仮想マシン (JVM)	Java Runtime Environment (JRE) の一部であり、バイトコードを解釈します。
Jython	オブジェクト指向の高度な動的言語 Python の実装の 1 つ。Java プラットフォームとシームレスに統合されます。Jython の前身である JPython は、100% ピュア Java として認定されています。
ラベル	N1 Service Provisioning System のバージョン番号という域を越えてコンポーネントバージョンをマークする手段。たとえば、コンポーネントバージョン番号はコンポーネントのバージョンを示すために使用できます。一方ラベルは、コンポーネントが表現するアプリケーションのバージョンを示すために使用できます。
ローカルディストリビュータ **(Local Distributor)**	サーバーにインストールされるアプリケーション。ローカルディストリビュータアプリケーションは、N1 Service Provisioning System 内のほかのサーバー間のリンクとして次のように機能します。 ■ マスターサーバーとリモートエージェントの間のリンク ■ マスターサーバーとほかのローカルディストリビュータの間のリンク ■ ローカルディストリビュータとリモートエージェントの間のリンク ローカルディストリビュータには帯域幅の効率と速度を最大化する効果があるほか、制限された環境をナビゲートするための安全なネットワーク接続も提供できます。
マスターサーバー **(Master Server)**	N1 Service Provisioning System を管理する目的でサーバー上にインストールされるアプリケーション。マスターサーバーアプリケーションは、N1 Service Provisioning System によって管理された任意のデータセンター環境に接続できます。マスターサーバーは、一元的なデータストレージ、データ処理、ユーザーインタフェースなどを実現します。
モデル化	N1 Service Provisioning System で配備するアプリケーションに相当するコンポーネントやプランを作成する作業。
入れ子コンポーネント	インストール時にそのコンテナコンポーネントにだけそのサービスを提供できる包含コンポーネント。入れ子になった包含コンポーネントを利用するとコンテナコンポーネントに必要な緻密な機能単位を定義できますが、ほかのコンポーネントには有用ではありません。
ネットワークプロトコル	ネットワーク上のデバイス間でデータを転送する方法。N1 Service Provisioning System 5.1 は、TCP/IP、SSH、および SSL を使用します。
通知電子メール	システムイベント、管理イベント、カスタムイベントなどの発生を知らせるために N1 Service Provisioning System によって送信される電子メールです。システム管理者は、電子メールを送信するタイミングと電子メールの送信先アドレスを決定するための規則を指定します。

通知規則	電子メール通知を送信するかどうかを決定するために N1 Service Provisioning System が使用する基準。システム管理者は、電子メール通知を送信するタイミングを決定するための基準を定義します。
親コンポーネント	ほかのコンポーネントに対する参照を含むコンポーネント。コンテナコンポーネントとも呼ばれます。 *contained component*も参照してください。
物理ホスト	ネットワークに接続される物理サーバー。プロビジョニングシステムでは、物理ホストはリモートエージェントまたはローカルディストリビュータとして機能できます。
プラン	1 つ以上のコンポーネントの操作に使用される一連の命令。1 つ以上のプランで一連の命令が共有されるように、複数のプランから 1 つのプランが作成される場合もあります。
Plan executor	プリフライトと配備を行う、マスターサーバー上のソフトウェアエンジンです。
プリフライト	シミュレートされた UNIX 環境に対するプランのシミュレートを実行します。配備に影響を与える可能性のある (潜在的) エラーを見つけて、報告します。プリフライトは配備前に必ず行われますが、単独の処理として実行することもできます。
手続き **(**プロシージャ**)**	*component procedure*を参照してください。
プロビジョニングシステム	サーバーにインストールされ、N1 Service Provisioning System 5.1 として機能するソフトウェアアプリケーションです。
リモートエージェント **(Remote Agent)**	コンポーネントの配備先となる任意のサーバーにインストールされる、N1 Service Provisioning System 内のアプリケーション。リモートエージェントアプリケーションは、ソフトウェアのインストール、サービスの制御、マスターサーバーへ配布する情報の収集などの作業を管理します。
リソース	プランの実行時にホストに配備されるファイル。 このファイルは、ディレクトリやシンボリックリンクといった別の種類のファイルの場合もあります。
リソース記述子ファイル	単純なコンポーネントのリソースを構成するファイルやディレクトリで使用するために、所有者、グループ、およびアクセス権の設定を指定する XML ファイルです。リソース記述子ファイルを使用することで、コンポーネントのチェックイン時に決定されるアクセス権を書き換えられます。
サーバー	リソースを管理し、クライアントにサービスを提供するコンポーネント。N1 Service Provisioning System 5.1 では、サーバーは、N1 Service Provisioning System アプリケーションの 1 つがインストールされているコンピュータを指します。

セッション	ユーザーのログイン時に開始される一定の時間枠。1 つのセッションは、ユーザーがログアウトするか、非アクティブ状態のためにセッションが期限切れとなるまで継続します。論理的には、セッションは特定のユーザーの認証済み資格情報を意味します。セッションは、再認証を行うことなく関連する一連の要求でユーザーを識別するために使用されます。
セッション変数	ユーザーセッションに関連付けられた変数。ユーザーは、ログインセッションごとにセッション変数の値を変更できます。後続のセッションで再使用できるように、セッション変数値を確実に保存することも可能です。
単純コンポーネント	単一のリソースが入ったコンポーネント。単純コンポーネントは、ほかのコンポーネントに対する参照を含むことはできません。
単純プラン	特定のターゲットサーバー群で実行される一連のステップ。単純プランにはほかのプランは含まれず、ほかのプランを呼び出すこともありません。
スナップショット	配備時にホストに保存されるリソースのキャプチャ。スナップショットは、ホストと、マスターサーバー上のそのモデルを比較する場合に使用されます (モデルとインストールの比較)。
ステップ	プランまたはコンポーネントの一部である命令。
置換変数	プラン、コンポーネント、または構成ファイルで出現する変数であり、配備時に構成生成エンジンによって置換されます。
システムサービス	該当するホストが用意される際にそれらのホストすべてに自動的に配備されるコンポーネント。システムサービスは、ユーティリティ制御と、ほかのコンポーネントによって使用できるリソースを定義します。
ターゲット可能コンポーネント	インストールされている場合にほかのコンポーネントの配備ターゲットとして機能するホストを作成するコンポーネント。ターゲット可能コンポーネントがアンインストールされる際、そのコンポーネントが作成したホストは自動的に削除されます。
トップレベルコンポーネント	インストールされる場合に、プランによって直接インストールされているかのように任意のコンポーネントから使用できる包含コンポーネント。トップレベルの包含コンポーネントは、コンテナコンポーネントやその他のコンポーネントによって使用されるサービスを定義します。
アップストリーム	N1 Service Provisioning System ネットワーク階層では、マスターサーバーにより近いサーバー。たとえば、マスターサーバーはローカルディストリビュータから見てアップストリームになります。ローカルディストリビュータは、そのローカルディストリビュータに接続されたどのリモートエージェントから見てもアップストリームになります。
変数	*component variable*を参照してください。

変数設定	1 つ以上のコンポーネント変数のデフォルト値を上書きするために使用できる変数値の集まり。コンポーネント変数には、使用する変数設定に基づいてさまざまな値を指定できます。ユーザーは、プランの実行時に使用するように変数設定を指定します。
仮想ホスト	ほかのサービスのホストとして機能するサービス。たとえば、仮想ホストは Web アプリケーションのホストとして機能するアプリケーションサーバーになることができます。
XML スキーマ	プランとコンポーネントを作成するために N1 Service Provisioning System によって使用される言語。

索引

C

CLI (Command Line Interface) Client
- アップグレード，116
- アプリケーションのディレクトリ構造，130, 136
- アンインストール，123-124, 124-125, 126
- インストール，45-47, 60-61
- オペレーティングシステム要件，25-31
- 実行，54-55
- 説明，20-21
- ハードウェア要件，31-32

`config.properties` ファイル, SSL のための
- 編集，102-104

H

HTTP, インストール中の選択，39-40
HTTPS
- HTTP からの再構成，74-76
- インストール後のキーストアファイルの更新，72-73
- インストール中の選択，39-40
- キーストアパスワードの作成，70-72
- キーストアパスワードの符号化，72-73
- キーストアファイルの作成，70-72

J

Java Virtual Machine，80-81
jexec，80-81
JRE (Java Runtime Environment)，35-36
JRE (Java Runtime Environment) (続き)
- トラブルシューティング，140

Jython，38

L

Linux, システム設定，27-28

M

Master Server, アップグレード，117-120

R

raw (TCP/IP), インストール中の選択，38-39
Remote Agent
- インストール
  - リモート，51-54

S

Solaris, システム設定，26-27
SSH (Secure Shell)
- jexec，80-81
- インストール中の選択，38-39
- 概要，78-81
- 構成作業，81
- コマンド，93-94
- システム要件，33

SSH (Secure Shell) (続き)
説明，22
その他のセキュリティー，80-81
トラブルシューティング，141-143
SSL (Secure Socket Layer)
config.properties ファイルの編集，102-104
暗号群
説明，95-96
リスト，109-110
インストール中の選択，38-39
概要，95-99
キーストアの作成，100-102
構成作業，99-104
構成例，104-108
説明，22
SUSE Linux, nolocks，27-28

W

Web ブラウザの要件，32-33

あ

アップグレード
概要，116
手順，117-120, 121
アプリケーションの起動，「アプリケーションの実行」を参照
アプリケーションの実行，54-55
アンインストール，123-125, 126
暗号群
説明，95-96
リスト，109-110

い

移行，「アップグレード」を参照
インストール
HTTPS，39-40
JRE (Java Runtime Environment)，35-36
インストール前の意思決定，35-43
概要，17-18
セキュリティー構成，38-40
対話方式，45-47, 57-60, 60-61

お

オペレーティングシステムの構成，26-28

き

キーストア
作成，100-102
説明，96-98
キーストアパスワード
インストール後のパスワードの構成，72-73
作成，70-72
キーストアパスワードの作成，70-72
キーストアファイル
インストール後の手動更新，72-73
作成，70-72
キーストアファイルの作成，70-72

し

システム要件
Linux システムの設定，27-28
Solaris システムの設定，26-27
Web ブラウザ，32-33
オペレーティングシステムパッチ，28-31
オペレーティングシステムの構成，26-28
オペレーティングシステムのバージョン，25-31
ハードウェア，31-32

せ

製品の概要，19-21
セキュリティー
JVM セキュリティーポリシー，111-113
概要，21-22
データベース，113

て

ディレクトリ構造
Linux および UNIX システム，128-130
Windows，134-136

は
ハードウェア要件，31-32

ひ
非対話方式によるのインストール, リモート
　エージェント，48-51
非対話方式のインストール
　Windows の変数値，66-67
　指示，62-63
　リモートエージェントサンプルパラメータ
　　ファイル，131-133

ま
マスターサーバー
　アプリケーションのディレクトリ構
　　造，128-129, 134-135
　アンインストール，123-124, 124-125, 126
　インストール，45-47, 57-60
　オペレーティングシステム要件，25-31
　システム要件，31
　実行，54-55
　説明，19

り
リモートインストール
　Remote Agent，51-54
　Windows の変数値，66-67
　手順，63-66
リモートエージェント
　Windows の変数値，66-67
　アップグレード，121
　アプリケーションのディレクトリ構
　　造，129-130, 135
　アンインストール，126
　インストール
　　対話方式，45-47, 60-61
　　非対話方式，48-51, 62-63
　　リモート，63-66
　オペレーティングシステム要件，25-31
　サンプルパラメータファイル，131-133
　実行，54-55
　説明，20
リモートエージェント (続き)
　ハードウェア要件，31-32
リモートエージェントt, アンインストー
　ル，124-125

ろ
ローカルディストリビュータ
　アップグレード，121
　アプリケーションのディレクトリ構造，129,
　　135
　アンインストール，124-125, 126
　インストール，45-47, 60-61
　オペレーティングシステム要件，25-31
　実行，54-55
　説明，19-20
　ハードウェア要件，31-32
ロック使用不可，27-28