Sun N1 Service Provisioning System 5.1 オペレーションとプロビジョニングガイド

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

Part No: 819–4220–10
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 Operation and Provisioning Guide

Part No: 819-1658–10

Revision A





051017@13215

目次

はじめに　**7**

1　コンポーネントの配備　**11**
コンポーネントの配備について　11
直接実行コンポーネントプロシージャーの使用　12
▼ 直接実行コンポーネントプロシージャーを使用する　12
プランの使用法　16
▼ プランを生成する　16
コンポーネントの配備の管理　17
▼ プランを実行する　18
▼ プランの進行状況を表示する　20
▼ 実行中のプランを停止する　21
▼ プランを削除する　21
配備例　22
コンポーネントの更新　22
以前のバージョンのコンポーネントへのダウングレード　23
▼ 以前のバージョンのコンポーネントへダウングレードする　23
サーバー間でのコンポーネントの移行　24
単純コンポーネントのコンテナコンポーネントへのグループ化　24
▼ コンテナコンポーネントを作成する　24

2　比較の実行　**27**
比較について　27
比較のための同期処理　28
モデルとモデルの比較　29
▼ モデルとモデルの比較を作成する　29

モデルとインストールの比較　30
コンポーネントタイプおよびモデルとインストールの比較　32
▼ モデルとインストールの比較を作成する　33
インストールとインストールの比較　35
インストールとインストールの比較でのスクリプトの使用　36
▼ インストールとインストールの比較を作成する　37
比較の管理　39
▼ 比較を実行する　39
▼ 比較を表示する　40
▼ 比較の進行状況を表示する　41
▼ 比較の実行を停止する　41
▼ 比較時に無視するディレクトリやファイルの変更　42

A　配備の問題の障害追跡　**45**
エラーメッセージの説明　45
ログメッセージの説明　48
メモリーの問題　48

用語集　**51**

索引　**59**

例目次

例 2–1 インストールとインストールの比較で使用する prepare ブロック 37
例 2–2 インストールとインストールの比較で使用する cleanup ブロック 37

はじめに

『Sun N1 Service Provisioning System 5.1 オペレーションとプロビジョニングガイド』では、コンポーネントを配備する手順や、ホスト、ホストセット、コンポーネントを比較する手順を紹介します。

対象読者

このマニュアルは、プランを実行に移す作業と、現在のシステムに正しいリソースを確実にインストールする作業を担当する N1 Service Provisioning System ソフトウェアユーザー向けに書かれています。

このマニュアルの構成

第 1 章では、ホストやホストセットにリソースを配布するさまざまな方法について説明します。

第 2 章では、3 種類の比較の作成および実行方法について説明します。

関連マニュアル

Sun N1 Service Provisioning System 5.1 マニュアルセットは、次のマニュアルで構成されています。ここに記載のマニュアルは、すべて docs.sun.com (`http://docs.sun.com`) から入手できます。

表 **P–1** Sun N1 Service Provisioning System5.1 マニュアル

アプリケーション	タイトル
ソフトウェア インストール	『Sun N1 Service Provisioning System 5.1 インストールガイド』
システム管理	『Sun N1 Service Provisioning System 5.1 システム管理者ガイド』
プランおよびコンポーネントの作成	『Sun N1 Service Provisioning System 5.1 プランとコンポーネントの開発者ガイド』
コマンドリファレンス	『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』
XML スキーマリファレンス	『Sun N1 Service Provisioning System 5.1 XML スキーマリファレンスガイド』
リリースノート	『Sun N1 Service Provisioning System 5.1 リリースノート』
プラグイン開発	『Sun N1 Service Provisioning System 5.1 Plug-in Development Guide』

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–2** 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% **su** password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm *filename* と入力します。
『 』	参照する書名を示します。	『コードマネージャー・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

コンポーネントの配備

この章では、プランの概念とコンポーネントの配備方法について簡単に説明します。

コンポーネントの配備について

コンポーネントの配備方法は、次の 3 つの中から選択できます。

- コンポーネントプロシージャー。1 つのコンポーネントと 1 つのホスト (またはホストセット) を使用する単純なシングル操作に適しています。
- プラン
  - 生成されたプラン。1 つのコンポーネントと 1 つのホスト (またはホストセット) を使用する複数の操作に適しています。
  - カスタムプラン。複数のコンポーネントと複数のホストを使用する複雑な操作に適しています。

プロビジョニングシステム では、有効なホストにコンポーネントをいくつでも配備することができます。配備可能なコンポーネントの数は、ホスト上のファイルシステムによってのみ制限されます。ファイルシステムはそれぞれに配備可能なファイルの数を制限しているため、1 つのディレクトリに配備するコンポーネントが多すぎると、そのファイルシステムの制限を超える可能性があります。コンポーネントとファイルの関係は、必ずしも等比ではありません。たとえば、1 つのコンポーネントに 100,000 個のファイルがあるとファイルシステムの制限を越える場合があり、100,000 個のコンポーネントのいずれにもファイルがないとファイルシステムの制限内に収まる場合があります。

ファイルシステムの制限を超えても、コンポーネントが配備されたときに直接障害を引き起こさない場合もあります。ただし、次第に性能が劣化したり、あるとき予測不可能な障害が起こるなどの形で問題が発生する可能性があります。

1 つのディレクトリがサポートできるファイル数の上限は、ファイルシステムによって異なり、オペレーティングシステムの構成に左右されることがあります。ファイルシステムの制限を理解し、単一の大きいディレクトリではなく、より小さい複数のサブディレクトリにコンポーネントを配備することによって、ディレクトリへの過負荷を避けることができます。コンポーネントを配備するときに従来から使われている対策は、1 つのディレクトリに保存するファイル数を 30,000 個以内にすることです。

直接実行コンポーネントプロシージャーの使用

複数のコンポーネントを使用したり、ホスト間での同期が不要な場合は、配備プランは必要ありません。その代わりに、配備するコンポーネントをチェックインしたときに自動生成される直接実行コンポーネントプロシージャーを使用することができます。

▼ 直接実行コンポーネントプロシージャーを使用する

ほとんどのコンポーネントには、インストール、アンインストール、スナップショットの各手続きが含まれています。このほかに、制御手続きが含まれている場合もあります。制御手続きは、インストール済みのコンポーネントを管理する機能を持っています。たとえば、アプリケーションの起動または停止などを行うことができます。

ここでは、ブラウザインタフェースを使用してコンポーネントを直接配備する方法について説明します。この手続きの代わりに、次のコマンドを使用することもできます。

- `pdp.p.genplan` – プランを生成します。
- `pe.p.run` – プランを実行します。

このコマンドの詳細な説明については、『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 9 章「pdb: プランを管理する CLI コマンド」 および 『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 10 章「pe: プランを実行する CLI コマンド」を参照してください。

コンポーネントの開発については、『Sun N1 Service Provisioning System 5.1 プランとコンポーネントの開発者ガイド』を参照してください。

注 – 直接実行コンポーネントプロシージャーによって生成されたプランを削除する必要はありません。これらのプランは、生成元のコンポーネントが削除された時点でシステムから自動的に削除されます。詳細については、『Sun N1 Service Provisioning System 5.1 プランとコンポーネントの開発者ガイド』の「コンポーネントを削除する」を参照してください。

始める前に　直接実行コンポーネントプロシージャーを使用するには、そのコンポーネントが含まれるフォルダに対して Run Component Procedures アクセス許可を持つユーザーグループに属する必要があります。また、このユーザーが、コンポーネントを配備する予定のホストセットに対して、Allow on Host Set アクセス許可を持っている必要があります。

手順

1. ナビゲーションメニューから「**Components**」を選択します。
   「Components」ページが表示されます。
2. コンポーネントの一覧表から、配備するコンポーネントの記述が含まれる行を選択して、「**Details**」をクリックします。
   コンポーネントの「Details」ページが表示されます。
3. 必要に応じて、配備する予定のコンポーネントが表示されるようにフォルダを変更します。
   a. 「**Change Folder**」をクリックします。
   b. コンポーネントを含むフォルダ名を指定して、「**Change to Selected Folder**」をクリックします。
      「Components」ページに、指定したフォルダに含まれているコンポーネントの一覧が表示されます。
4. コンポーネントプロシージャーの表から、実行する手続きを選択して、「**Run**」をクリックします。
   プロビジョニングシステムにより、プランが生成されます。生成されたプランは、`/system/autogen` フォルダに格納されます。
   生成されたプランの「Run」ページが表示されます。
5. 「**Plan Parameters**」領域で、配備するコンポーネントの変数設定を選択します。
   - 当該コンポーネントの変数設定が確立されている場合、メニューから適切な設定を選択します。
   - メニューから設定を選択できない場合は、「**Select From List**」をクリックします。
     「Select Variable Settings From List」ウィンドウが表示されます。

- 別のコンポーネントの変数設定を使用する場合は、「**Import Set**」をクリックします。
  「Import Variable Settings」ウィンドウが表示されます。

  a. 必要に応じて、変数設定のインポート元となるコンポーネントが入ったフォルダへ移動します。

  b. コンポーネントのバージョンを選択します。

  注 – コンポーネントのバージョンが異なると、変数設定も異なる可能性があります。

  ---

  c. 「**Import Variable Settings**」をクリックします。
  インポートされた変数設定が表に追加されます。

  d. 「選択」をクリックします。

- 新しい変数設定を作成する場合は、「**Create Set**」を選択します。
  「Select Variable Settings From List」ウィンドウが表示されます。

  a. 表の最上部のテキストフィールドに、新しい変数設定セットの名前を入力します。

  b. 変更するコンポーネント変数を選択します。
  選択されたセルが強調表示され、テキストフィールドが表示されます。

  c. 新しいコンポーネント変数値を入力します。

  d. 変数設定値の更新が完了したら、「**Save**」をクリックします。
  新しい変数設定が表に追加されます。

  e. 「選択」をクリックします。

6. コンポーネントの配備先を選択します。

- ホストを選択する場合は、「**Target Host**」フィールドにホスト名を入力し、「**Select From List**」をクリックします。
- ホストセットを選択する場合は、「**Target Host Set**」メニューから選択します。

7. 「**Plan Variables**」領域で、設定する変数を構成します。
構成対象の変数の有無は、選択したコンポーネントプロシージャーによって異なります。

8. 「**Options**」領域で、詳細なプリフライトを行うかどうかを選択します。

   このプリフライトでは、リモートエージェント上で実行されるプランの各手順をシミュレートすることにより、以下の処理が正常に行われるかどうかをテストすることができます。

   - リモートインストール
   - `execNative` 呼び出し
   - ファイル変換
   - ファイルの削除 (リソースのアンインストール時)

   詳細なプリフライトを行うと、実際に運用を開始するまでの準備期間が長くなります。

9. **(**省略可能**)** 同時に実行するホストの数を制限する場合は、フィールドにホスト数を指定します。

   あるプランを複数のホストに対して同時に実行した場合、すべてのホストが手順を完了するまで、次の手順に進むことはできません。ホスト数が多すぎると、次の手順に進む前に、ネットワーク接続がタイムアウトになる可能性があります。たとえば、ホスト A が手順 1 を完了したあと、手順 2 の実行前の接続確認 (ping) が行われるまでの時間が長すぎて、ネットワーク接続がタイムアウトになることがあります。

10. **(**省略可能**)** プラン全体の実行時間、またはネイティブ呼び出しの実行時間を制限する場合は、フィールドに数値を入力し、メニューから時間の単位を選択します。

    プランまたはネイティブ呼び出しの実行時間を制限すると、応答しないホストがあるためにプランの実行が途中で停止するのを防ぐことができます。

    プランの実行時の障害について報告する通知規則を設定しておくと、実行途中でタイムアウトになった場合、プランが正常に実行されなかったことを知らせるメッセージを受け取ることができます。

    通知規則の設定については、『Sun N1 Service Provisioning System 5.1 システム管理者ガイド』の「電子メール通知の構成」を参照してください。

11. コンポーネントを配備するかどうかを選択します。

    - 手続きのプリフライトを行うだけでよい場合は、「**Run Preflight Only**」をクリックします。
    - プリフライトの実行後、このチェックでエラーが検出されなかったものと想定して、さらに手続きを実行する場合は、「**Run Plan (Includes Preflight)**」をクリックします。

プランの使用法

プロビジョニングシステムでは、次の 2 種類のプランを実行できます。

- 生成されたプラン
- カスタムプラン

生成されたプラン

プロビジョニングシステムは、複数のコンポーネントプロシージャーから成るプランを自動的に生成することができます。このプランは、そのまま使用することも、XML を使ってより複雑なプランのテンプレートとして保存することもできます。

カスタムプラン

XML でカスタムプランを作成することができます。カスタムプランの作成については、『Sun N1 Service Provisioning System 5.1 プランとコンポーネントの開発者ガイド』の「プランの作成」 and the 『Sun N1 Service Provisioning System 5.1 XML スキーマリファレンスガイド』を参照してください。

▼ プランを生成する

ここでは、ブラウザインタフェースを使用してプランを生成する方法について説明します。次のコマンドでも、プランを生成することができます。

- `pdp.p.genplan` – プランを生成します。
- `pe.p.run` – プランを実行します。

これらのコマンドの詳細な説明については、『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 9 章「pdb: プランを管理する CLI コマンド」を参照してください。

始める前に　プランを生成するためには、プランを生成するコンポーネントが含まれるフォルダに対して、Create、Edit 、Delete のアクセス許可を持つユーザーグループに属する必要があります。

手順　**1.** ナビゲーションメニューから「**Components**」を選択します。

「Components」ページが表示されます。

2. 必要に応じて、配備する予定のコンポーネントが表示されるようにフォルダを変更します。

a. 「**Change Folder**」をクリックします。

b. コンポーネントを含むフォルダ名を指定して、「**Change to Selected Folder**」をクリックします。

「Components」ページに、指定したフォルダに含まれているコンポーネントの一覧が表示されます。

3. コンポーネントの一覧表から、配備するコンポーネントの記述が含まれる行を選択して、「**Details**」をクリックします。

コンポーネントの「Details」ページが表示されます。

4. 「**Component Procedures**」表から、プランに含める手続きを選択します。

5. 「**Component Procedures**」表の下部にある「**Generate Plan With Checked Procedures**」をクリックします。

生成されたプランの「Advanced Edit」が表示されます。

6. 「**Plan**」フィールドに新しいプランの名前を入力します。

7. 「**Check In**」をクリックします。

注 – 次のようなメッセージが表示される場合があります。「Warning - plan names and/or paths differ.」このメッセージは、直前に指定したプラン名が、XML モデルで指定されているプラン名と一致していない場合に表示されます。「Plan」フィールドに入力した名前になるように、XML モデルが更新されます。

---

8. コンポーネントのチェックインを確認します。

a. 指定したプラン名が正しいことを確認します。

b. 必要に応じて、コンポーネントのバージョン番号を大きくします。

c. 「**Continue to Check In**」をクリックします。

新しいプランの「Details」ページが表示されます。

コンポーネントの配備の管理

プランを利用してコンポーネントを配備する場合、処理が完了するまでに相当の時間がかかる可能性があります。コンポーネントが配備されるまでのプロセスをひとめで確認できるようにするには、プランを監視し、実行中のプランや完了したプランに関する情報を表示します。配備の途中でプランを停止することもできます。

- 18 ページの「プランを実行する」
- 20 ページの「プランの進行状況を表示する」

■ 21 ページの「実行中のプランを停止する」

▼ プランを実行する

ここでは、ブラウザインタフェースを使用してプランを実行する方法について説明します。次のコマンドでも、プランを実行することができます。

`pe.p.run` – プランを実行します。

このコマンドの詳細な説明については、『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 10 章「pe: プランを実行する CLI コマンド」を参照してください。

始める前に　プランの変数設定を設定するために必要なインストール情報を準備してください。

プランを実行するには、実行するプランが含まれるフォルダに対して、`Allow on Host Set` 許可を持つユーザーグループに属する必要があります。

手順　**1.** プランの「**Details**」ページに移動します。

2. 「**Run**」をクリックします。
プランの「Run」ページが表示されます。

3. 「**Plan Parameters**」領域で、配備するコンポーネントの変数設定を選択します。

- 当該コンポーネントの変数設定が確立されている場合、メニューから適切な設定を選択します。
- メニューから設定を選択できない場合は、「**Select From List**」をクリックします。
  「Select Variable Settings From List」ウィンドウが表示されます。
  - 別のコンポーネントの変数設定を使用する場合は、「**Import Set**」をクリックします。
    「Import Variable Settings」ウィンドウが表示されます。

    a. 必要に応じて、変数設定のインポート元となるコンポーネントが入ったフォルダへ移動します。

    b. コンポーネントのバージョンを選択します。

    注 – コンポーネントのバージョンが異なると、変数設定も異なる可能性があります。
    ---

    c. 「**Import Variable Settings**」をクリックします。
    インポートされた変数設定が表に追加されます。

d. 「選択」をクリックします。

- 新しい変数設定を作成する場合は、「**Create Set**」を選択します。
  「Select Variable Settings From List」ウィンドウが表示されます。

  a. 表の最上部のテキストフィールドに、新しい変数設定セットの名前を入力します。

  b. 変更するコンポーネント変数を選択します。
  選択されたセルが強調表示され、テキストフィールドが表示されます。

  c. 新しいコンポーネント変数値を入力します。

  d. 変数設定値の更新が完了したら、「**Save**」をクリックします。
  新しい変数設定が表に追加されます。

  e. 「選択」をクリックします。

4. コンポーネントの配備先を選択します。

- ホストを選択する場合は、「**Target Host**」テキストフィールドにホスト名を入力し、「**Select From List**」をクリックします。
- ホストセットを選択する場合は、「**Target Host Set**」メニューから選択します。

5. 「**Plan Variables**」領域で、設定する変数を構成します。
構成対象の変数の有無は、プランの内容によって異なります。

6. 「**Options**」領域で、詳細なプリフライトを行うかどうかを選択します。
このプリフライトでは、リモートエージェント上で実行されるプランの各手順をシミュレートすることにより、以下の処理が正常に行われるかどうかをテストすることができます。

- リモートインストール
- `execNative` 呼び出し
- ファイル変換
- ファイルの削除 (リソースのアンインストール時)

詳細なプリフライトを行うと、実際に運用を開始するまでの準備期間が長くなります。

7. **(**省略可能**)** 同時に実行するホストの数を制限する場合は、フィールドにホスト数を指定します。
あるプランを複数のホストに対して同時に実行した場合、すべてのホストが手順を完了するまで、次の手順に進むことはできません。ホスト数が多すぎると、次の手順に進む前に、ネットワーク接続がタイムアウトになる可能性があります。たとえば、ホスト A が手順 1 を完了したあと、手順 2 の実行前の接続確認 (ping) が行われるまでの時間が長すぎて、ネットワーク接続がタイムアウトになることがあります。

8. **(**省略可能**)** プラン全体の実行時間、またはネイティブ呼び出しの実行時間を制限する場合は、フィールドに数値を入力し、メニューから時間の単位 **(**分など**)** を選択します。

   プランまたはネイティブ呼び出しの実行時間を制限すると、応答しないホストがあるためにプランの実行が途中でストップするのを防ぐことができます。

   プランの実行時の障害について報告する通知規則を設定しておくと、実行途中でタイムアウトになった場合、プランが正常に実行されなかったことを知らせるメッセージを受け取ることができます。

   通知規則の設定については、『Sun N1 Service Provisioning System 5.1 システム管理者ガイド』の「電子メール通知の構成」を参照してください。

9. コンポーネントを配備するかどうかを選択します。

   - 手続きのプリフライトを行うだけでよい場合は、「**Run Preflight Only**」をクリックします。
   - プリフライトの実行後、このチェックでエラーが検出されなかったものと想定して、さらに手続きを実行する場合は、「**Run Plan (Includes Preflight)**」をクリックします。

▼ プランの進行状況を表示する

ここでは、ブラウザインタフェースを使用してプラン実行の進行状況を表示する方法について説明します。次のコマンドでも、プランの履歴を表示できます。

- `pe.p.la` – 実行中のプラン、完了したプランを一覧表示します。
- `pe.p.lo` – 実行中のプランまたは完了したプランに関する情報を一覧表示します。
- `pe.pi.lo` – プランの実行に使用されたパラメータをリストします。

これらのコマンドの詳細な説明については、『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 10 章「pe: プランを実行する CLI コマンド」を参照してください。

手順 1. ナビゲーションメニューで、「**Application Deployment**」カテゴリを展開し、「**Run History**」をクリックします。

   実行中、完了済みのプランおよびプリフライトがすべて表示されます。「Completed」欄にタイムスタンプが表示されているプランは、現在実行中のプランです。このタイムスタンプは、このプランの実行が開始された時刻を表しています。

2. 特定のプランの詳細情報を表示する場合は、「**Results**」をクリックします。

   「Details」、「Preflight」、「Deployment」の各アイコンを選択すると、プランの詳細、プリフライトの結果、配備の結果というように、表示内容が切り替わります。

「Details」ページに表示される情報は、次のとおりです。

- ターゲットホスト
- このプランのパラメータとして使用された変数設定
- このプランに課された制限
- このプランを実行したユーザー

▼ 実行中のプランを停止する

ここでは、ブラウザインタフェースを使用してプラン実行を停止する方法について説明します。次のコマンドでも、プランの実行を停止することができます。

`pe.p.stop` – 実行中のプランを停止します。

このコマンドの詳細な説明については、『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 10 章「pe: プランを実行する CLI コマンド」を参照してください。

始める前に　プランの実行を停止するには、そのプランの実行を開始したユーザーであるか、`admin` ユーザーグループのメンバーである必要があります。

手順

- 実行中のプランを取り消します。
  - 「**Run Deployment**」ページで、「**Cancel Plan**」をクリックします。
    「Run History」ページに、ユーザーによってプランが取り消されたことを知らせるメッセージが表示されます。
  - 「**Run History**」ページで、実行を停止するプランを選択します。
    a. 「**Action**」列の「**Progress**」をクリックします。
       「Run Deployment」ページが表示されます。
    b. 「**Cancel Plan**」をクリックします。

▼ プランを削除する

ここでは、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生成されていないプランを削除する方法について説明します。

注 – 直接実行コンポーネントプロシージャーから生成されたプランを削除する必要はありません。これらのプランは、生成元のコンポーネントが削除された時点でシステムから自動的に削除されます。詳細については、『Sun N1 Service Provisioning System 5.1 プランとコンポーネントの開発者ガイド』の「コンポーネントを削除する」を参照してください。

次のコマンドを使用してプランを削除することもできます。

- `pe.p.del` – プランを削除します。

このコマンドの詳細な説明については、『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 10 章「pe: プランを実行する CLI コマンド」を参照してください。

始める前に　プランを削除するには、プランを作成したユーザーであるか、または `admin` ユーザーグループのメンバーである必要があります。

手順
1. プランの「**Details**」ページに移動します。
2. 「削除」をクリックします。
   プランを削除するかどうかを確認するよう求められます。
3. 「**Continue to Delete**」をクリックします。
   プランが削除されると、「Plan」の一覧表にプランは表示されません。

配備例

直接実行コンポーネントプロシージャーと生成されたプランを使って、アプリケーションを管理することができます。

- 22 ページの「コンポーネントの更新」
- 23 ページの「以前のバージョンのコンポーネントへのダウングレード」
- 24 ページの「サーバー間でのコンポーネントの移行」
- 24 ページの「単純コンポーネントのコンテナコンポーネントへのグループ化」

コンポーネントの更新

コンポーネントのバージョンを新しくすることができます。新しいコンポーネントには、製品リビジョン、ファイルへの変更点、新規追加ファイルの格納ディレクトリなどの新しい情報が反映されています。

単純コンポーネントを更新する場合、ユーザーは直接実行インストール手続きを実行するだけで済みます。古いバージョンのコンポーネントを新しいバージョンで上書きする処理は、自動的に行われます。

コンポーネント (複数の単純コンポーネントを参照するコンポーネント) を更新する場合、新旧のバージョンに含まれる参照先の単純コンポーネントが同じであるなら、ユーザーは直接実行インストール手続きを実行するだけで済みます。古いバージョンの単純コンポーネントを新しいバージョンで上書きする処理は、自動的に行われます。

複合コンポーネントを更新する場合、新旧のバージョンで参照先の単純コンポーネントが異なっているなら、古いバージョンのコンポーネントをアンインストールして、新しいバージョンをインストールする必要があります。古いバージョンをアンインストールすると、不要になったコンポーネントやこれ以上更新されることがないコンポーネントが除去され、ワークスペースがすっきりします。

以前のバージョンのコンポーネントへのダウングレード

コンポーネントを正しくアップグレードできない場合や、アップグレード後のアプリケーションで問題が発生する場合は、新しいバージョンのコンポーネントの障害追跡を行う間、信頼性の高い旧バージョンのコンポーネントにダウングレードすることができます。

▼ 以前のバージョンのコンポーネントへダウングレードする

始める前に　信頼性の高いバージョンのコンポーネントを再配備するためには、このコンポーネントが含まれるフォルダに対して、Run Component Procedure 許可と Allow on Host Set 許可を持っている必要があります。

手順　**1.** 壊れたコンポーネントをアンインストールします。

a. コンポーネントの「**Details**」ページで、「**Run for the Default**」をクリックします。コンポーネントプロシージャーをアンインストールします。

b. プランの「**Run**」ページで、アンインストールするコンポーネントのホストを選択します。

c. 「**Run Select Installations**」をクリックします。

d. 「**Run Plan (Includes Preflight)**」をクリックします。

壊れたコンポーネントをアンインストールすると、古くなった不要なファイルが、それまでの配備先システムから除去されます。

2. コンポーネントをどのバージョンまでダウングレードするかを選択します。

a. コンポーネントの「**Details**」ページで、「**Version History**」をクリックします。

b. 「**Version History**」ページで、「**Unhide**」をクリックし、「**Details**」をクリックすると、信頼性の高いバージョンが表示されます。

3. この信頼性の高いバージョンのコンポーネントをインストールします。

サーバー間でのコンポーネントの移行

コンポーネントをあるサーバーから別のサーバーへ移行する場合は、現在のホストから除去するコンポーネントをアンインストールして、このコンポーネントを新しいホストにインストールするだけで済みます。

単純コンポーネントのコンテナコンポーネントへのグループ化

複数の単純コンポーネントをグループにまとめて同じホストまたはホストセットに配備する場合に、コンテナコンポーネントを作成できます。単純コンポーネントをグループ化すると、配備に必要な時間が短縮されるだけでなく、一部のコンポーネントの配備漏れも防ぐことができます。

▼ コンテナコンポーネントを作成する

始める前に　コンテナコンポーネントを作成するためには、コンポーネントを作成するフォルダに対して、Create、Edit、Delete のアクセス許可を持っている必要があります。

手順　1. 「**Components**」ページで、新しいコンポーネントを作成します。

a. 「**Component**」フィールドに、新しいコンポーネントの名前を入力します。

b. 「**Type**」フィールドで、「**System#container**」を選択します。

c. 「**Description**」フィールドに、新しいコンポーネントの説明を入力します。

d. 「作成」をクリックします。

2. 「**Referenced Components**」で、コンテナにコンポーネントを追加します。

- 参照するコンポーネントがすでに存在している場合は、「**Reference Existing**」をクリックします。

  「Reference Existing Components in Current Component」ウィンドウが表示されます。

a. コンテナコンポーネント内から参照するコンポーネントを選択し、「**Reference Checked Components**」をクリックします。

コンポーネント変数を含むコンポーネントを参照する場合は、そのコンポーネント変数がコンテナコンポーネントに追加される前に一覧表示されます。

b. 必要に応じて、コンポーネント変数名の横のテキストフィールドで、そのコンポーネント変数のデフォルト値を設定します。

c. 「**Continue**」をクリックして、コンポーネントを参照します。

コンテナコンポーネントの「Edit」ページに戻ります。「Referenced Components」に、参照先コンポーネントが一覧表示されます。

- 参照するコンポーネントが存在しない場合は、「**Check In New**」をクリックします。

  新しいコンポーネントを作成します。

  新しいコンポーネントを作成する方法については、『Sun N1 Service Provisioning System 5.1 プランとコンポーネントの開発者ガイド』の第 2 章「コンポーネント」を参照してください。

  新しいコンポーネントを作成すると、コンテナコンポーネントの「Referenced Components」の一覧に新しいコンポーネントが追加されます。

3. 必要に応じて、コンテナコンポーネントで定義する必要があるコンポーネント変数の値を指定します。

たとえば、コンポーネントが `:[container....]` という変数を持っている場合、コンテナコンポーネントで値を定義する必要があります。

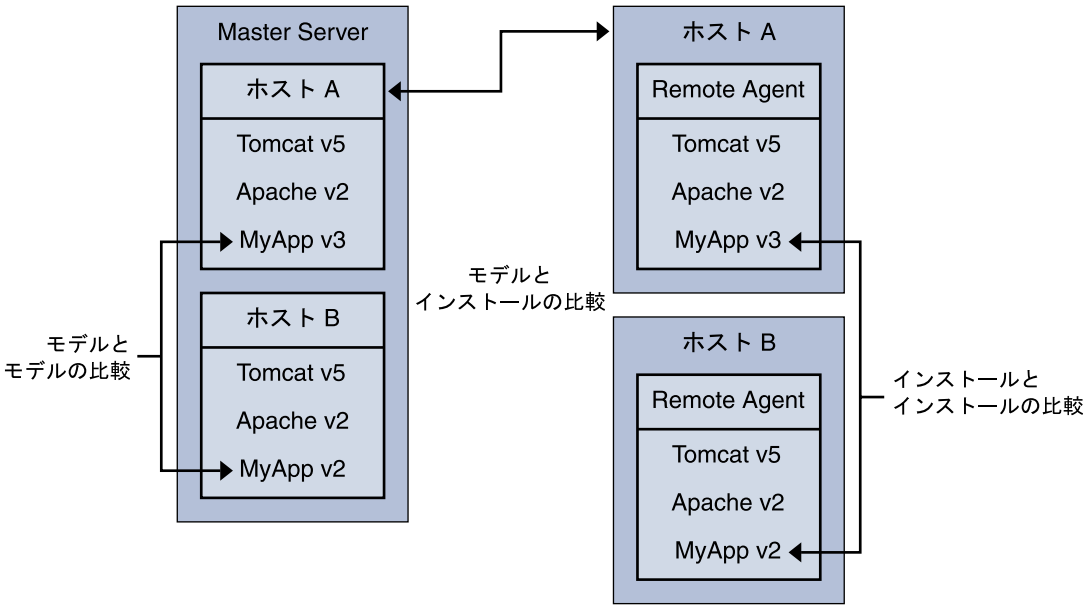
第 2 章

比較の実行

- 29 ページの「モデルとモデルの比較」
- 30 ページの「モデルとインストールの比較」
- 35 ページの「インストールとインストールの比較」
- 39 ページの「比較の管理」

比較について

比較は、インストールやシステム構成の妥当性を検査するために使用されます。現在のホストの構成とその初期構成を比較することで、変更箇所がわかります。1 台以上のホストと参照ホストを比較して、すべてのホストが同じように構成されているかどうかを確認することもできます。



プロビジョニングシステムでは、1 台以上のホストともう 1 台の別のホストとの比較、1 台以上のホストとマスターサーバー内のモデルとの比較、1 台のホストとその初期構成との比較を行うことができます。プロビジョニングソフトウェアを使って、ホスト全体の構成をチェックすることも、チェックを単一のディレクトリに制限することもできます。

N1 Service Provisioning System ソフトウェアでは、次の 3 種類の比較が可能です。

- モデルとモデルの比較
- モデルとインストールの比較
- インストールとインストールの比較

比較のための同期処理

比較のためのタイムスタンプはマスターサーバー内のクロックに基づいているため、すべてのシステムクロックが同じ時刻に同期化されるように、すべての N1 Service Provisioning System ソフトウェアマシンで、何らかの同期化ユーティリティーを実行する必要があります。

モデルとモデルの比較

最も単純な比較は、モデルとモデルの比較です。この比較では、あるホストにインストールされているコンポーネントのリストと別のホストにインストールされているコンポーネントのリストの照合を行います。つまり、プロビジョニングシステムの履歴情報から、これまでにインストールされた内容の確認を行います。

いずれかのコンポーネントリストに帯域外の変更が加えられた場合、この変更は検出されません。モデルとモデルの比較は、マスターサーバーが受信した配備情報に基づいて、マスターサーバー上で行われます。

モデルとモデルの比較の特徴

- 2 台のホストのインストール内容の履歴を比較します。
- データ転送は不要です。
- もっとも実行時間が短い比較です。

モデルとモデルの比較は、次のような場合に使用します。

- 帯域外の変更が加えられていない、2 台のホストの内容を比較する場合。
- インストール時にスナップショットをとっていない、2 台のホストを比較する場合。

▼ モデルとモデルの比較を作成する

ここでは、ブラウザインタフェースを使用して比較を作成する方法について説明します。次のコマンドでも、比較を作成することができます。

`cmp.ds.add` – 新しい比較を作成します。

このコマンドの詳細な説明については、『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 5 章「cmp: 比較を実行する CLI コマンド」を参照してください。

始める前に　新しい比較を作成するためには、比較の Create、Edit、Delete アクセス許可を持っているユーザーグループに所属する必要があります。

手順

1. ナビゲーションメニューの「**Comparisons**」をクリックします。
   「Comparisons」ページが表示されます。
2. 作成する比較の名前と説明を入力して、「**Create**」をクリックします。
   比較の「Details」ページが表示されます。
3. 「**Style of Comparison**」領域の「**Model to Model**」をクリックします。

4. 「**Reference Host**」フィールドに、比較の基準にするモデルのホスト名を指定します。

   ここでは、比較ホスト上で確認する構成を含む参照ホストを選択します。

5. 「**Comparison Host**」フィールドに、参照モデルと比較するモデルのホスト名を入力します。

6. 「**Limits**」領域に、比較の最大実行時間 **(**単位は時間または分**)** を指定します。

   デフォルトでは、比較の最大実行時間は 4 時間です。

   比較の実行時にタイムアウトになった場合、「Comparisons」ページにエラーメッセージが表示されます。

7. 「保存」をクリックします。

モデルとインストールの比較

モデルとモデルの比較では、あるインストールを 1 つの理想的なインストールと比較しますが、モデルとインストールの比較では、インストール済みのコンポーネント (複数可) の配備時の状態と現在の状態を比較します。このとき、モデルとホストを比較するのか、モデルとホストセットを比較するのか、モデルとコンポーネントレベルを比較するのかを指定できます。

モデルとインストールの比較では、スナップショット情報を使って、コンポーネントのインストール直後の状態と現在の状態を比較します。スナップショットとは、インストールされ、ホストへ配備された直後のコンポーネントの状態の記録です。コンポーネントの XML モデルには、このスナップショットの範囲を制御するパラメータが含まれています。たとえば、XML モデルでは、比較時に無視するファイル (ログファイルなど) をリストアップすることができます。

プロビジョニングソフトウェアを使ってモデルとインストールの比較を実行するとき、マスターサーバーは、比較対象の全コンポーネントのスナップショットのリストを検索し、このリストを、コンポーネントがインストールされているホストのリモートエージェントに送信します。スナップショット情報には、比較時に無視するファイルやディレクトリの規則が含まれています。リモートエージェントは、これらの規則を処理します。モデルとインストールの比較を正しく行うためには、マスターサーバーとリモートエージェントのロケールが同じである必要があります。

モデルとインストールの比較の特徴

- インストール済みのコンポーネントのスナップショットと、同じコンポーネントの現在の状態を比較します。
- ホストにインストールされているコンポーネントを個別に、またはいくつかまとめて比較することができます。
- 上位レベルでの比較 (コンポーネントとディレクトリのリストの比較) と下位レベルでの比較 (ファイルの内容の比較) が可能です。

- リモートエージェントでの比較処理
- コンポーネントがリモートエージェントにインストールされるときに、スナップショットを作成している必要があります。

モデルとインストールの比較は、次のような場合に使用します。

- 構成が意図どおりに動作して、その後動作しなくなった場合、帯域外の変更があったと見なすことができます。

コンポーネントタイプおよびモデルとインストールの比較

モデルとインストールの比較では、コンポーネントモデルと特定のコンポーネントのインストールを比較します。この節では、コンポーネントタイプがモデルとインストールの比較に及ぼす影響について説明します。

スナップショットの作成時に、メタベースの現在の状態が XML ファイルにエクスポートされます。モデルとインストールの比較時、メタベースは再度エクスポートされ、元の XML ファイルと比較されます。

IIS タイプの概要
: スナップショットの作成時に、メタベースの現在の状態が XML ファイルにエクスポートされます。モデルとインストールの比較時、メタベースは再度エクスポートされ、元の XML ファイルと比較されます。これらのファイルの差分生成には、標準の XML differentiator が使用されます。

COM+ アプリケーション
: COM+ アプリケーションは、MSI ファイルとしてターゲットシステムに再エクスポートされ、アプリケーションのインストールに使用された MSI ファイルと比較されます。モデルとインストールの比較では、差分があるということ (つまり、2 つのバイナリファイルが異なっているということ) がわかるだけです。差分の詳細まではわかりません。

COM コンポーネント
: COM コンポーネントは、バイナリファイルとして、インストール時に使用されたファイルと比較されます。モデルとインストールの比較では、差分があるということ (つまり、2 つのバイナリファイルが異なっているということ) がわかるだけです。差分の詳細まではわかりません。

H キー
: スナップショットの実行時、レジストリキーの現在の状態 (キーの子を含む) が XML ファイルにエクスポートされます。モデルとインストールの比較時、レジストリキーは再度エクスポートされ、元の XML ファイルと比較されます。これらのファイルの差分生成には、標準の XML differentiator が使用されます。

レジストリ設定
: `*.reg` ファイルでは、モデルとインストールの比較はサポートされていません。インストール時にスナップショットは作成されません。スナップショットがなければ、モデルとインストールの比較を行なっても差分は生成されません。レジストリの差分を比較する場合は、組み込みのレジストリキータイプを使用してください。

MSI ファイルまたは Windows Installer ファイル

Windows Installer ファイルでは、モデルとインストールの比較はサポートされていません。インストール時にスナップショットは作成されません。スナップショットがなければ、モデルとインストールの比較を行なっても差分は生成されません。プロビジョニングシステムはインストール時のアクションを直接認識しないので、キャプチャする必要があるものを特定できません。

Windows バッチファイル
Windows バッチファイルについては、モデルとインストールの比較はサポートされていません。インストール時にスナップショットは作成されません。スナップショットがなければ、モデルとインストールの比較を行なっても差分は生成されません。

Windows スクリプティングホストファイル
WSH ファイルについては、モデルとインストールの比較はサポートされていません。インストール時にスナップショットは作成されません。スナップショットがなければ、モデルとインストールの比較を行なっても差分は生成されません。

WebLogic WAR ファイル
WebLogic WAR ファイルでは、標準のファイル/ディレクトリ方式でモデルとインストールの比較が行われます。

WebLogic Web アプリケーション設定
アプリケーションの関連設定がファイルにエクスポートされます。このファイルと、配備時の設定が含まれたファイルが比較されます。

WebLogic JAR ファイル
WebLogic JAR ファイルでは、標準のファイル/ディレクトリ方式でモデルとインストールの比較が行われます。

WebLogic EJB 設定
EJB の関連する設定がファイルにエクスポートされます。このファイルと、配備時の設定が含まれるファイルが比較されます。

WebLogic EAR ファイル
WebLogic EAR ファイルでは、標準のファイル/ディレクトリ方式でモデルとインストールの比較が行われます。

WebLogic エンタープライズアプリケーション設定
アプリケーションの関連設定がファイルにエクスポートされます。このファイルと、配備時の設定が含まれたファイルが比較されます。

▼ モデルとインストールの比較を作成する

ここでは、ブラウザインタフェースを使用して比較を作成する方法について説明します。次のコマンドでも、比較を作成することができます。

`cmp.ds.add` – 新しい比較を作成します。

このコマンドの詳細な説明については、『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 5 章「cmp: 比較を実行する CLI コマンド」を参照してください。

始める前に　インストールとインストールの比較を正しく行うためには、各ホストのリモートエージェントのロケールが同じである必要があります。

新しい比較を作成するためには、比較の Create、Edit、Delete アクセス許可を持っているユーザーグループに所属する必要があります。

手順　**1.** ナビゲーションメニューの「**Comparisons**」をクリックします。
「Comparisons」ページが表示されます。

2. 作成する比較の名前と説明を入力して、「**Create**」をクリックします。
比較の「Details」ページが表示されます。

3. 「**Style of Comparison**」領域の「**Model to Install**」をクリックします。

4. 比較の詳細レベルを選択します。

- ディレクトリとファイルのプロパティ。両方のホストに同じサイズの同じファイルとディレクトリがあることを確認します。
- ファイルの内容。両方のホストのファイルの内容を比較します。

注 – 比較に使用するコンポーネントに、無視するファイルに関する独自の規則が含まれている場合、これらの規則は、ユーザーがこの比較のために定義した無視の規則に追加して適用されます。

---

5. 比較の範囲を選択します。

- ホストセット。指定したホストセットの前回の状態と現在の状態を比較します。
  a. 「**Scope**」メニューから「**Host Set**」を選択します。
  b. 比較するホストセットを選択します。
- ホスト。指定したホストの前回の状態と現在の状態を比較します。
  a. 「**Scope**」メニューから「**Host**」を選択します。
  b. 比較するホストの名前を入力します。
- コンポーネント。指定したコンポーネントの前回の状態と現在の状態を比較します。
  a. 「**Scope**」メニューから「**Component**」を選択します。
  b. コンポーネントのホストの名前を入力し、「**Apply Change**」をクリックします。
  c. 「**Component (Path)**」メニューからコンポーネントを選択します。

6. **(省略可能)** 比較時に無視するファイルまたはディレクトリの名前を入力します。
   この機能により、ログファイルなどが不要に変更されることがなくなり、比較の失敗を防ぐことができます。

   詳細については、42 ページの「比較時に無視するディレクトリやファイルの変更」を参照してください。

7. 「**Limits**」領域に、比較の最大実行時間 **(**単位は時間または分**)** を指定します。
   デフォルトでは、比較の最大実行時間は 4 時間です。

   比較の実行時にタイムアウトになった場合、「Comparisons」ページにエラーメッセージが表示されます。

8. 比較を保存してあとで実行するか、今すぐ実行するかを選択します。
   - 比較を今は実行しないで保存する場合は、「**Save**」をクリックします。
   - 比較を保存後すぐに実行する場合は、「**Save & Run**」をクリックします。

インストールとインストールの比較

モデルとインストールの比較と同じく、インストールとインストールの比較でも、ファイルのコンテンツレベルで比較することができます。インストールとインストールの比較では、2 台のホストのファイルとディレクトリを比較します。

参照ホストの要素 (ファイル、ディレクトリなど) のリストは、マスターサーバー上で 1 つにまとめられ、比較ホストのリモートエージェントに送信されたあと処理されます。ファイルの種類に関する規則や、無視するディレクトリまたはファイルに関する規則を定義している場合、マスターサーバーは、リスト内の適切な項目を破棄し、残った項目を比較ホストに送信します。その後、マスターサーバーに比較の結果が返されます。

インストールとインストールの比較は、同じリモートエージェント上の別々のディレクトリで実行できます。この方法で、同じホスト上の 2 つのディレクトリの内容を比較することができます。

注 – インストールとインストールの比較を正しく行うためには、各ホストのリモートエージェントのロケールが同じである必要があります。

インストールとインストールの比較の特徴

- あるホスト上のコンポーネントの現在の状態と、別のホスト上のコンポーネントの現在の状態を比較します。
- ホストにインストールされているコンポーネントを個別に、またはいくつかまとめて比較することができます。
- 比較ホストのリモートエージェントで比較処理を行います。
- インストール時にコンポーネントのスナップショットが作成されている必要があります。
- リモートエージェントは同じロケールを使用する必要があります。

インストールとインストールの比較は、次のような場合に使用します。

- 2 台のホストのコンポーネントを比較したい場合。

  たとえば、ステージングホスト上のソフトウェアと、運用ホスト上のソフトウェアの比較が可能です。
- 単一ホスト上の 2 つのディレクトリを比較したい場合。

インストールとインストールの比較でのスクリプトの使用

比較対象のコンポーネントのリソースが、パッケージ、データベース、アーカイブ (ZIP ファイルなど) に格納されている場合は、スクリプトを使って、比較の実行前にリソースをエクスポートしたり変換したりすることができます。プロビジョニングソフトウェアで、ファイル内容の解凍やエクスポートなどのネイティブコマンドを実行するには、`execNative` 手順を使用します。

ネイティブコマンドは、`prepare` ブロックや `cleanup` ブロック内で実行できます。これらのブロックは、比較の前後で使用します。これらの使用により、圧縮アーカイブからファイルが抽出され、指定の出力ファイルに出力されます。このファイルは、クリーンアップ時に削除されます。

注 – この `execNative` は、プランやコンポーネントに含まれる `execNative` とは異なり、`prepare` ブロックか `cleanup` ブロックに含まれていなければなりません。また、変数を利用することはできません。

ブラウザインタフェースを使用してインストールとインストールの比較を行うときは、「Include Temporary Files Generated By Scripts During Comparison」というラベルの付いたチェックボックスをオンにします。このチェックボックスをオンにすると、現在のページが更新され、`prepare` ブロックと `cleanup` ブロックの入力フィールドが 1 つずつ表示されます。

適切なチェックボックスをオフにして、比較ホストの `prepare` ブロックと `cleanup` ブロックを置き換えることもできます。

execNative については、『Sun N1 Service Provisioning System 5.1 XML スキーマリファレンスガイド』の第 2 章「コンポーネントと単純プランにより使用される共有スキーマ」を参照してください。

例 **2–1** インストールとインストールの比較で使用する prepare ブロック

この prepare ブロックは、execNative の結果を出力する出力ファイルを指定します。execNative は、/tmp ディレクトリの内容を一覧表示します。

```
<prepare>
  <execNative outputFile="file.txt" dir="temp">
    <exec cmd="ls">
      <arg value="-l"></arg>
      <arg value="/tmp"></arg>
    </exec>
  </execNative></prepare>
```

例 **2–2** インストールとインストールの比較で使用する cleanup ブロック

この cleanup ブロックは、出力ファイル file.txt を /temp ディレクトリから削除します。

```
<cleanup>
  <execNative  dir="/temp/">
    <exec cmd="rm">
      <arg value="-rf"></arg>
      <arg value="file.txt"></arg>
    </exec>
  </execNative></cleanup>
```

▼ インストールとインストールの比較を作成する

ここでは、ブラウザインタフェースを使用して比較を作成する方法について説明します。次のコマンドでも、比較を作成することができます。

cmp.ds.add – 新しい比較を作成します。

このコマンドの詳細な説明については、『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 5 章「cmp: 比較を実行する CLI コマンド」を参照してください。

始める前に　インストールとインストールの比較を正しく行うためには、各ホストのリモートエージェントのロケールが同じである必要があります。

新しい比較を作成するためには、比較の Create、Edit、Delete アクセス許可を持っているユーザーグループに所属する必要があります。

手順　**1.** ナビゲーションメニューの「**Comparisons**」をクリックします。

HTML ユーザーインタフェースに「Comparisons」ページが表示され、データベースにチェックイン済みの比較が一覧表示されます。

2. 作成する比較の名前と説明を入力して、「**Create**」をクリックします。
   比較の「Details」ページが表示されます。
3. 「**Style of Comparison**」領域の「**Install to Install**」をクリックします。
4. 比較の詳細レベルを選択します。
   - ディレクトリとファイルのプロパティ。両方のホストに同じサイズの同じファイルとディレクトリがあることを確認します。
   - ファイルの内容。両方のホストのファイルの内容を比較します。

   注 – 比較に使用するコンポーネントに、無視するファイルに関する独自の規則が含まれている場合、これらの規則は、ユーザーがこの比較のために定義した無視の規則に追加して適用されます。

   ---

5. 参照ホストの名前と、比較ホストと比較するディレクトリのパスを入力します。
   ここでは、比較ホスト上で確認する構成を含む参照ホストを選択します。

   入力したディレクトリによって、階層状のファイル構造のどのレベルまで比較するかが決定します。ツリー構造のトップを入力した場合は、ホスト全体が比較されます。
6. 参照ホストの名前と、この参照ホストと比較するディレクトリのパスを入力します。
7. この比較で使用可能なオプションを選択します。
   - **(省略可能)** 選択したディレクトリの下位のすべてのディレクトリに対して比較を行いたい場合は、「**Include All Subdirectories In Comparison**」を選択します。
   - **(省略可能)** シンボリックリンクを解決して、リンク先も有効であることを確認したい場合は、「**Follow Symbolic Links Found in Directories**」を選択します。
   - **(省略可能)** この比較の前またはあとでスクリプトを実行したい場合は、次のアクションを実行します。

     a. 「**Include Temporary Files Generated by Scripts During Comparison**」をクリックします。
        ページが更新され、generate スクリプト用と cleanup スクリプト用のフィールドが 1 つずつ表示されます。

     b. 適切なフィールドにスクリプトを入力します。
        prepare と cleanup のスクリプトの詳細については、36 ページの「インストールとインストールの比較でのスクリプトの使用」を参照してください。

c. 比較ホストと参照ホストで同じスクリプトを実行しない場合は、該当するオプションを選択解除します。

- 「Generate」チェックボックスをオフにすると、比較ホスト用の generate スクリプトを入力するフィールドが表示されます。
- 「Cleanup」チェックボックスをオフにすると、比較ホスト用の cleanup スクリプトを入力するフィールドが表示されます。

この比較の実行時に無視するリソースを指定したい場合は、42 ページの「比較時に無視するディレクトリやファイルの変更」を参照してください。

8. **(省略可能)** 比較時に無視するファイルまたはディレクトリの名前を入力します。
この機能により、ログファイルなどが不要に変更されることがなくなり、比較の失敗を防ぐことができます。

9. 「**Limits**」領域に、比較の最大実行時間 **(**単位は時間または分**)** を指定します。
デフォルトでは、比較の最大実行時間は 4 時間です。

比較の実行時にタイムアウトになった場合、「Comparisons」ページにエラーメッセージが表示されます。

10. 「保存」をクリックします。

比較の管理

通常、比較の実行には時間がかかります。比較を実行、修正、表示できるのは、比較を作成し終わってからです。

- 39 ページの「比較を実行する」
- 40 ページの「比較を表示する」
- 41 ページの「比較の進行状況を表示する」
- 41 ページの「比較の実行を停止する」
- 42 ページの「比較時に無視するディレクトリやファイルの変更」

▼ 比較を実行する

ここでは、ブラウザインタフェースを使用して比較を実行する方法について説明します。次のコマンドでも、比較を実行することができます。

`cmp.dj.add` – 比較を実行します。

このコマンドの詳細な説明については、『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 5 章「cmp: 比較を実行する CLI コマンド」を参照してください。

注 – 比較で使用するコンポーネントがアップグレードされた場合や、アップグレード以外の何らかの理由でバージョン番号が変更された場合、保存済みの比較が自動的に更新され、新しいバージョンのコンポーネントに対して実行されるようになります。

始める前に　ユーザーの属するユーザーグループが、特定のホストセット上での比較の実行許可を持っている必要があります。

手順　**1.** 「**Comparisons**」ページで、実行する比較を選択します。

2. **(**省略可能**)** 比較基準を変更します。

3. 「**Save & Run**」をクリックします。
比較の「Progress」ページが表示されます。

比較が完了すると、比較の「Results」ページが表示されます。

注 – 比較を実行するために、ほかの作業をストップする必要はありません。まだ比較の実行中であっても、ほかの作業を実行することができます。

---

4. **(**省略可能**)** 比較の結果を別の書式で表示する場合は、「**Print**」または「**Download**」をクリックします。

5. 「終了」をクリックします。

▼ 比較を表示する

ここでは、ブラウザインタフェースを使用して比較を表示する方法について説明します。次のコマンドでも、比較を表示することができます。

- `cmp.ds.la` – すべての比較を表示します。
- `cmp.ds.lo` – 特定の比較の詳細情報を表示します。
  これらのコマンドの詳細な説明については、『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 5 章「cmp: 比較を実行する CLI コマンド」を参照してください。

手順　**1.** ナビゲーションメニューの「**Comparisons**」をクリックします。
「Comparisons」ページが表示されます。

2. **(**省略可能**)** 特定の比較の詳細情報を表示したい場合は、調べたいプランが含まれている行の「**Details**」をクリックします。
比較の「Details」ページが表示されます。

▼ 比較の進行状況を表示する

ここでは、ブラウザインタフェースを使用して比較実行の進行状況を表示する方法について説明します。次のコマンドでも、比較の履歴を表示できます。

- `cmp.dj.la` – 実行中および完了した比較をリストします。
- `cmp.dj.lo` – 特定の比較の状態を表示します。

これらのコマンドの詳細な説明については、『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 5 章「cmp: 比較を実行する CLI コマンド」を参照してください。

手順

1. ナビゲーションメニューで、「**Application Deployment**」カテゴリを展開し、「**Run History**」をクリックします。
   実行中の比較、完了した比較がすべて一覧表示されます。実行中の比較の「Completed」列には、プランの実行開始時刻が表示されます。
2. 特定のプランの詳細情報を表示する場合は、「**Results**」をクリックします。

▼ 比較の実行を停止する

ここでは、ブラウザインタフェースを使用して比較を実行する方法について説明します。次のコマンドでも、比較を実行することができます。

`cmp.dj.del` – 比較を停止します。

このコマンドの詳細な説明については、『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 5 章「cmp: 比較を実行する CLI コマンド」を参照してください。

始める前に　比較の実行を停止するには、その比較処理を開始したユーザーか、`admin` ユーザーグループのメンバーである必要があります。

手順

1. 「**Run History**」ページで、処理を停止する比較を選択します。
2. 「**Action**」列の「**Progress**」をクリックします。
3. 「**Cancel Comparison**」をクリックします。

注 – 実行中の比較を途中で取り消すと、全比較の一覧ページに戻ります。場合によっては、一覧が更新され、比較処理が停止したことがわかるまで、多少時間がかかることがあります。

▼ 比較時に無視するディレクトリやファイルの変更

比較を行うときは、比較の詳細レベルや比較対象とするホストまたはコンポーネントを指定できるだけでなく、比較対象から除外するファイルやディレクトリも指定できます。たとえば、重要でない差分が大量に含まれるリソースがある場合に、比較対象から除外します。この指定内容を保存し、別の比較定義で再利用することも可能です。

次のコマンドでも、比較を変更することができます。

`cmp.ds.mod` – 比較を変更します。

このコマンドの詳細な説明については、『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 5 章「cmp: 比較を実行する CLI コマンド」を参照してください。

始める前に　比較を編集するためには、比較の Create、Edit、Delete アクセス許可を持っているユーザーグループに所属する必要があります。

手順　**1.** 必要に応じて、変更するモデルとインストールの比較、またはインストールとインストールの比較を選択します。

比較の「Details」ページが表示されます。

2. 「**Directories and Files to Ignore During Comparisons**」の表に変更を加えます。

- 除外リストにディレクトリやファイルを追加します。

  a. 比較の「**Details**」ページに、除外するディレクトリまたはファイルの名前を入力し、「**Add**」をクリックします。

  ワイルドカード (*) を使って、除外するディレクトリまたはファイルを表す文字列を入力することもできます。たとえば、すべてのログファイルを無視する場合は、`*.log` と入力します。

  注 – ワイルドカードを使用しても、使用しなくても、ファイル名までの完全なパターンを指定してください。ディレクトリまでしか指定しないと、そのディレクトリの下位のファイルは除外されません。下位ファイルも除外対象に含める場合は、ディレクトリ名の末尾にアスタリスク (*) を指定する必要があります。たとえば、`/usr/apache/logs` ディレクトリ内のすべてのファイルを除外対象とする場合は、`/usr/apache/logs*` のように指定する必要があります。

  ---

  b. 「追加 **(Add)**」をクリックします。

  指定した内容が除外対象のディレクトリとファイルのリストに追加されます。

- 除外リストからディレクトリまたはファイルを削除します。

a. 削除するディレクトリまたはファイルを選択します。

注 – 大域的な除外規則は、削除することも無効にすることもできません。

---

b. 解除 をクリックします。

3. 「保存」をクリックします。

付録 A

配備の問題の障害追跡

この付録では、N1 Service Provisioning System ソフトウェアの一般的なエラーメッセージとログメッセージについて説明します。

- 45 ページの「エラーメッセージの説明」
- 48 ページの「ログメッセージの説明」

エラーメッセージの説明

`This component has not been installed on any hosts yet`

説明: アンインストールしようとしているコンポーネントがホストにインストール済みであることを、マスターサーバーが認識しない場合に発生するエラーです。マスターサーバーは、システム内の各コンポーネントのインストール状態を管理しています。コンポーネントがホストにインストールされたときの履歴がマスターサーバーのデータベースに残っていないと、このコンポーネントをアンインストールすることができません。また、このコンポーネントを制御することもできません。

対処方法: この問題を修正するには、次のようにします。

1. コンポーネントがホストにインストールされていることを確認します。
2. このコンポーネントがホストにインストールされている場合は、空のインストールブロックを使ってコンポーネントの `markOnly` インストールを実行し、マスターサーバーデータベースを更新します。構文は次のとおりです。

```
<installSteps name="markOnly" />
```

`017007 Plan timed out on hosts`

説明: プランのタイムアウトパラメータの設定値が小さすぎて、プランの実行に必要な時間が足りません。

例: たとえば、プランに含まれる手順に 100 秒間の待ち時間が設定されているのに、プランのタイムアウトパラメータが 60 秒に設定されている場合、このプランは待ち時間の終了前にタイムアウトになります。

対処方法: プランのタイムアウトパラメータには、プランの実行を完了できることを考慮しながら、適切な値を設定してください。デフォルトのタイムアウト値は 30 分です。プランの実行時間が 30 分以上になる可能性がある場合は、このプランのタイムアウトパラメータを変更してください。

`017031 Unable to generate configuration file` *filename* `for host` *hostname*

説明: 構成テンプレートに、コンポーネントの `<varList>` に宣言されていない変数が含まれています。

例: たとえば、`/ant/config.properties` 構成テンプレートを配備するとします。この構成テンプレートに、`/ant/config.properties` コンポーネントの `<varList>` 内に宣言されていない `:[port]` 変数が含まれている場合、次のエラーが表示されます。

- `:[port]` variable is undefined (040201)
- Configuration file `/ant/config.properties` cannot be generated (017031)
- Plan fails because configuration file cannot be generated (017034)

対処方法: このエラーを防ぐには、構成テンプレートで使用する変数がコンポーネントの `<varList>` に宣言されていることを確認してください。構成テンプレートをチェックインすると、ファイルからプロビジョニングシステム変数が自動的にスキャンされ、コンポーネントの `<varList>` に追加されます。

`017074 java.io.IOException: Not enough space`

説明: リモートエージェントの物理メモリーが不足しているため、リモートエージェントを起動できません。ディスク容量に関するエラーメッセージが表示されますが、実際には、物理メモリーの問題が原因で発生するエラーです。

例: `<execNative>` 手順を 1 つ含むプランを実行した場合、マスターサーバーログファイル `server/bin/cr_server.out` に次のエラートレースが表示されます。

```
:NestedTrace: java.io.IOException: Not enough space
   at java.lang.UNIXProcess.forkAndExec(Native Method)
   at java.lang.UNIXProcess.(UNIXProcess.java:54)
   at java.lang.Runtime.execInternal(Native Method)
   at java.lang.Runtime.exec(Runtime.java:546)
```

ブラウザインタフェースには、次のエラーメッセージが表示されます。

```
Problems encountered during plan run or preflight

The plan (or preflight)
"/system/autogen/ldap_cmd-inst-default-1087486030740" finished with 1
failed host(s). (017034)

Error executing the exec native command "/usr/bin/pfksh". (017074)
Not enough space
```

```
plan : /system/autogen/ldap_cmd-inst-default-1087486030740
version : 1.11 version history | run history
```

対処方法: 物理メモリーの問題を修正するには、cr_agent スクリプトで、最小 JVM™ [1] ヒープサイズを小さくします。次の手順を実行します。

1. リモートエージェントの実行サーバーで、リモートエージェントスクリプトが格納されているディレクトリに移動します。

   # **cd /***N1SPS50-home***/agent/bin**

   N1SPS50-home は、リモートエージェントのインストールディレクトリです。

2. JVM ヒープサイズのプロパティを変更します。

   CR_JVM_MEMORY=*-Xms32m -Xmx256m*

   -Xms 32m は最小ヒープサイズ設定です。

3. サーバーを再起動します。

030005 Expecting "0" subplan component builds in component build array "0".

説明: このエラーは、CLI からプランを実行しているときに発生します。cr_cli コマンドを使ってプランを実行する際、-comp + オプションや -vs + オプションを指定すると、コンポーネントはまったくインストールされず、上記のようなエラーメッセージが表示されます。

例: たとえば、次の cr_cli プラン実行コマンドを実行したとします。

```
% cr_cli -cmd pe.p.run -u bmclean -p password  \
-PID NM:/app/Weblogic/RollingWeblogicRestart \
-tar H:NM:dev-prf30 -pto 600s -nto 600s -comp + -vs +
```

この場合、次のエラーが表示されます。

```
Expecting "0" subplan component builds in component build array "0". (030005)
```

プロビジョニングシステムのブラウザインタフェースを使用したときは、同じターゲットホストを指定して、同じプランを正常に実行することができました。このプランの XML コンテンツは次のとおりです。

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated by N1 SPS -->
<executionPlan xmlns="http://www.sun.com/schema/SPS"
name="RollingWeblogicRestart" version="4.1"
description="Restart Weblogic instances on target hosts in series"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" path="/app/Weblogic"
xsi:schemaLocation="http://www.sun.com/schema/SPS plan.xsd">

<simpleSteps executionMode="SERIES">
 <call blockName="restartAppServer">
  <installedComponent name="WeblogicUtil" path="/app/Weblogic"/>
 </call>
```

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。

```
</simpleSteps>
</executionPlan>
```

このプランでは、コンポーネントはまったくインストールされません。以前にインストールされたコンポーネントに対する制御 (呼び出しブロック) が呼び出されるだけです。したがって、コンポーネントや変数セットを指定する必要はありません。

対処方法: この問題を防ぐには、コンポーネントをインストールしないプランの実行時には、コマンド行に `-comp +` オプションや `-vs +` オプションを指定しないようにします。

`040201 Substitution variable` *`:target:port`* `is undefined`

説明: ターゲットホストのホスト属性からコンポーネント内のターゲット属性を得ることができないときに発生するエラーです。

例: たとえば、`/ant/config.properties` 構成テンプレートを配備するとします。コンポーネントには、デフォルト値が `:[target:port]` の変数を持つ `varList` があります。配備先ホストには、*port* 変数の値を求めるホスト属性がありません。

対処方法: この問題を防ぐには、配備先のホストに、コンポーネント内のターゲット変数の値を求めるホスト属性セットがあることを確認してください。

ログメッセージの説明

この節では、配備の実行時に遭遇する可能性がある、ログに記録された問題について説明します。

メモリーの問題

`java.lang.OutOfMemoryError`

説明: 配備中にマスターサーバーがメモリー不足になりました。このメモリー不足エラーは、マスタサーバーのログファイルで確認できることがあります。また、同様のエラーメッセージをブラウザインタフェースで確認できることもあります。このエラーは、大規模で複雑なコンポーネントを配備したり、複数のホストに対して同時に配備したりすることが原因で発生する場合があります。

大規模で複雑な配備の例:

- 参照されるコンポーネントの数が 100 を超えるコンテナを配備する場合。
- サイズの大きいファイルを多数持つ `system#directory` コンポーネントを配備する場合。サイズが 250k を超えるファイルが、サイズの大きいファイルとみなされます。
- 多数のプランを多数のホストに配備する場合。

  たとえば、4 つ以上のプランを 4 つ以上のホストセットに配備し、かつ各ホストセットが 8 つ以上のホストで構成され、各プランが 100 を超えるリソースを配備する場合です。

対処方法: このエラーを防ぐ方法は 2 つあります。まず、ホストとコンポーネントの数を減らして、より簡単な配備を実行することができます。次に、マスターサーバー内の物理 RAM を増やしてから、`CR_MAX_HEAP_SIZE` 値を増やすことができます。



注意 – `CR_MAX_HEAP_SIZE` 値を増やす前に、システムに十分な RAM があることを確認してください。

新しい `CR_MAX_HEAP_SIZE` 値を確立するには、システムのメモリーが不足した際に使用可能だった未使用の物理 RAM の量を特定します。`CR_MAX_HEAP_SIZE` の最大値は、未使用の物理 RAM よりも 100Mb 少ない値にしてください。

値を設定するには、`CR_MAX_HEAP_SIZE` 値を次のように変更して、*/N1SPS-home*`/server/bin/cr_server` スクリプトを修正します。

```
CR_MAX_HEAP_SIZE=${CR_MAX_HEAP_SIZE:-new-heap-size}
```

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。プロビジョニングソフトウェアは、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。プロビジョニングソフトウェアは 2 種類のコンポーネントの互換性をサポートしています。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。
コンポーネントプロシージャー	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。
制御	ユーザのすばやい注意を要求するメッセージです。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プロシージャーを実行します。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プロシージャーです。
ダウンストリーム	N1 Service Provisioning System ソフトウェア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	プロビジョニングソフトウェアが管理するサーバー。
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)	実行環境の再配布を望むユーザーや開発者を対象にした、Java Development Kit (JDK®) のサブセットです。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	プロビジョニングソフトウェアのバージョン番号という域を超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	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、プロビジョニングソフトウェア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。
マスタサーバー	プロビジョニングソフトウェアを管理するサーバー上にインストールされてい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、プロビジョニングソフトウェアによって管理される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	プロビジョニングソフトウェアで配備するアプリケーションを表すコンポーネントおよびプランを作成すること。
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 ソフトウェアでは、TCP/IP、SSH、SSL が使用されます。

通知メール	システム、管理、またはカスタムイベントが発生したことを知らせるため、プロビジョニングソフトウェアが送信するメール。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	プロビジョニングソフトウェアが、電子メール通知が送信されたかどうかを特定するために使用する基準。システム管理者は、通知メールをいつ送信するかを決定する基準を定義します。
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 componen*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た場合に、N1 Service Provisioning System ソフトウェアを構成するソフトウェアアプリケーション
リモートエージェント	コンポーネントの配備先であるプロビジョニングソフトウェアの任意のサーバーにインストールするアプリケーション。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用に使用する、所有者、グループ、およびアクセス許可設定を指定する XML ファイル。リソース記述子ファイルを使用すると、コンポーネントのチェックイン時に決定されたアクセス許可を無効にすることができます。

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 ソフトウェアでは、プロビジョニングソフトウェアアプリケーションがインストールされているコンピュータをサーバーと呼びます。
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。
上位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	プロビジョニングソフトウェア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	プロビジョニングソフトウェアがプランやコンポーネントの作成において使用する言語。

索引

C

`cleanup` ブロック，36-37

P

`prepare` ブロック，36-37

い

インストールとインストールの比較，35-39
　作成，37-39
　スクリプトの使用，36-37
　　prepare と cleanup，36-37

え

エラーメッセージ, 説明，45-48

か

回避策，45-48

こ

コンポーネント, 配備，11-17
コンポーネントプロシージャー, 実行，12

さ

削除, プラン，21-22
作成
　生成されたプラン，16-17
　比較
　　インストールとインストール，37-39
　　モデルとインストール，33-35
　　モデルとモデル，29-30
差分，「比較」を参照
差分設定，「比較」を参照

し

実行
　「実行」を参照
　コンポーネントプロシージャー，12
　生成されたプラン，16, 18-20
　比較，39-40
　プラン，18-20
自動生成されたプラン，「生成されたプラン」を参照
障害追跡
　エラーメッセージ，45-48
　メモリーの問題，48-49
　ログメッセージ，48-49
除外，「無視」を参照

す

スナップショット，30-35

せ
生成されたプラン
　作成，16-17
　実行，16

ち
直接実行手続き, 使用，12

て
停止, 比較，41-42
ディレクトリやファイルの無視, 比較，42-43

と
取り消し，「停止」を参照

は
配備, コンポーネント，11-17

ひ
比較，27-28
　インストールとインストール，35-39
　作成
　　インストールとインストール，37-39
　　モデルとインストール，33-35
　　モデルとモデル，29-30
　実行，39-40
　停止，41-42
　ディレクトリやファイルの無視，42-43
　表示，40
　ホストの同期化，28
　モデルとインストール，30-35
　モデルとモデル，29-30
比較設定，「比較」を参照
表示, 比較，40

ふ
プラン
　削除，21-22
　作成
　　生成された，16-17
　実行，18-20

ほ
ホストの同期化, 比較，28

め
メモリーの問題, 障害追跡，48-49

も
モデルとインストールの比較，30-35
　作成，33-35
　スナップショット，30-35
モデルとモデルの比較，29-30
　作成，29-30
問題，「エラーメッセージ」を参照

ろ
ログメッセージ，48-49