



Sun N1 Service Provisioning System ユーザーズガイドおよびリ リースノート (OS Provisioning Plug-In 2.0)

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

Part No: 819-4230-10
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コービ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イ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イ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ぬものとします。

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 User's Guide and Release Notes for the OS Provisioning Plug-In 2.0

Part No: 819-2884-10

Revision A



051017@13215



目次

はじめに	11
1 OS プロビジョニングの概要	15
OS プロビジョニングの概要	15
機能	16
プロビジョニングプロセスの概要	17
OS プロビジョニングシステムソフトウェアのアーキテクチャー	17
ソフトウェアアーキテクチャーの全体像	17
ネットワークプロトコル	19
サポートされるシステム	20
システムコンポーネント	20
OS プロビジョニングコンポーネント	21
ブートおよびインストールサーバー	21
ターゲットホストのハードウェアプラットフォーム	21
ターゲットホストのオペレーティングシステム	22
2 OS プロビジョニングプラグインのリリースノート	23
インストールの問題	23
実行時の問題	23
Solaris: root パスワードの間違った暗号化により JumpStart エラーが生じる (6245964)	23
Windows ブートおよびインストールサーバーで OS プロビジョニングスクリプトの位置を変更できない (6251010)	24
DHCP 設定の警告がユーザーインターフェイスに報告されない (6248485)	24
EUC 以外のロケールでホストプロビジョン状態によって OS インストールログを表示できない (6255797)	24

3	OS プロビジョニングの配備環境	25
	前提条件	25
	基本的な OS プロビジョニング環境	25
	ターゲットホスト	26
	ネットワーク	26
	セキュリティ	30
	新しい環境の構成	30
	プロセスの概要	30
	▼ マスターサーバーを有効にしてセッション ID を使用する	31
	既存の環境の構成	32
	ハードウェアおよびソフトウェアの構成	32
	ネットワーク環境	32
	DHCP サービス	33
	ターゲットホスト	33
	N1 Service Provisioning System ソフトウェア	33
	既存の Solaris JET 環境	33
4	OS プロビジョニングプラグインのインストールと構成	35
	OS プロビジョニングプラグインのインストール	35
	OS プロビジョニングプラグインの入手	35
	OS プロビジョニングプラグインの N1 SPS への追加	36
	▼ OS プロビジョニングプラグインをアップグレードする	36
	▼ 旧バージョンのコンポーネントにアクセスする	37
	OS プロビジョニングサーバーの作成	38
	▼ OS プロビジョニングサーバーを作成する (ブラウザインタフェース)	38
	▼ OS プロビジョニングサーバーを作成する (コマンド行インタフェース)	39
	サブネットの作成と登録	40
	▼ OS プロビジョニングサーバーのサブネットを特定する (ブラウザインタフェース)	40
	▼ OS プロビジョニングサーバーのサブネットを特定する (コマンド行インタフェース)	41
	▼ 新しいプロビジョニングサブネットを追加する	42
5	Solaris オペレーティングシステムのプロビジョニング	43
	JumpStart Enterprise Toolkit (JET) テクノロジ	43
	JET モジュール	44
	JET を介した Solaris のインストールプロセス	44
	Solaris プロビジョニングプロセスの概要	45

Solaris JET サーバーの設定	45
▼ JET Solaris イメージサーバーを作成する	45
Solaris イメージサーバーのコンポーネント変数	46
Solaris イメージおよびプロファイルの作成	47
▼ Solaris イメージをインポートする	47
▼ Solaris プロビジョニングプロファイルを作成する (ブラウザインタフェース)	49
▼ Solaris プロビジョニングプロファイルを作成する (コマンド行インタフェース)	52
ターゲットホストへの OS のインストール	52
▼ ターゲットホストに OS をインストールする (ブラウザインタフェース)	53
▼ ターゲットホストに OS をインストールする (コマンド行インタフェース)	54
▼ OS のプロビジョニング状態を確認する	54
▼ OS プロビジョニング処理を停止する	55
JET Solaris サーバーの管理作業	55
▼ N1 SPS RA インストーラをインポートする	55
▼ ブート params に JET サーバーを使用する	56
custom モジュールの使用法	57
カスタムパッチとパッケージメディアの生成	58
カスタムファイル	59
カスタムスクリプト	59
6 Linux オペレーティング環境のプロビジョニング	61
Kickstart テクノロジ	61
Linux プロビジョニングプロセスの概要	61
Linux サーバーの設定	62
▼ Linux ファイルをコピーする	62
▼ NFS アクセス権を設定する	62
▼ TFTP アクセス権を設定する	63
▼ Linux イメージサーバーを作成する	65
Linux イメージおよびプロファイルの作成	67
▼ Linux プロビジョニングプロファイルを作成する (ブラウザインタフェース)	67
▼ Linux プロビジョニングプロファイルを作成する (コマンド行インタフェース)	68
ターゲットホストへの OS のインストール	68
▼ ターゲットホストへ OS をインストールする	69
▼ OS のプロビジョニング状態を確認する	69

7	Windows オペレーティング環境のプロビジョニング	71
	Remote Installation Service (RIS) テクノロジ	71
	Windows プロビジョニングプロセスの概要	72
	Windows サーバーの設定	72
	▼ Windows RIS サーバーを設定する	72
	▼ Windows RIS サーバーで OS プロビジョニングスクリプトをインストールする	74
	▼ Windows RIS サーバーに Windows RSH サーバーをインストールする	74
	▼ Windows RIS サーバーに Windows SSH サーバーをインストールする	75
	▼ Windows RIS サーバーで Windows イメージを設定する	75
	▼ Windows イメージの設定情報を構成する	77
	▼ イメージ配備用の OS チューザファイルを構成する	79
	▼ ターゲットホストで BIOS ブート順序を設定する	81
	▼ Windows イメージサーバーを作成する (ブラウザインタフェース)	81
	▼ Windows イメージサーバーを作成する (コマンド行インタフェース)	82
	Windows イメージサーバーのコンポーネント変数	83
	Windows イメージおよびプロファイルの作成	84
	▼ Windows プロビジョニングプロファイルを作成する (ブラウザインタフェース)	84
	▼ Windows プロビジョニングプロファイルを作成する (コマンド行インタフェース)	85
	Windows プロビジョニングプロファイルの変数	85
	ターゲットホストへの OS のインストール	87
	▼ ターゲットホストへ OS をインストールする	87
8	OS プロビジョニング用ターゲットホスト	89
	ターゲットホストタイプ	89
	プラットフォームによるターゲットホストタイプ	90
	ターゲットホストを定義する作業の例	93
	Sun ALOM ターゲットホストの定義	93
	▼ Sun ALOM ターゲットホストを作成する (ブラウザインタフェース)	94
	▼ Sun ALOM ターゲットホストを作成する (コマンド行インタフェース)	95
	Sun IPMI ターゲットホストの定義	96
	▼ Sun IPMI ターゲットホストを作成する (ブラウザインタフェース)	96
	▼ Sun IPMI ターゲットホストを作成する (コマンド行インタフェース)	98
	HP ILO ターゲットホストの定義	99
	▼ HP ILO ターゲットホストを作成する (ブラウザインタフェース)	99
	▼ HP ILO ターゲットホストを作成する (コマンド行インタフェース)	101

そのほかの遠隔管理ターゲットホストの定義	102
▼ そのほかの遠隔管理ターゲットホストを作成する (ブラウザインタフェース)	102
▼ そのほかの遠隔管理ターゲットホストを作成する (コマンド行インタフェース)	104
汎用ターゲットホストの定義	105
▼ 汎用ターゲットホストを作成する (ブラウザインタフェース)	105
▼ 汎用ターゲットホストを作成する (コマンド行インタフェース)	107
パスワードの暗号化	107
OS プロビジョニング変数のパスワードの暗号化	107
OS root アクセス権のパスワードの暗号化	108
9 障害追跡	109
一般的な障害追跡のガイドライン	109
Solaris 関連の問題の解決	111
Linux 関連の問題の解決	112
Windows 関連の問題の解決	114
PXE/DHCP/BINLSVC に関連する問題	114
Remote Installation Service (RIS) に関連する問題	118
GUID に関連する問題	120
A ターゲットホストの変数	121
Sun IPMI ターゲットホストの変数	121
Sun ALOM ターゲットホストの変数	123
Sun LOM ターゲットホストの変数	126
Sun Mid-Range ターゲットホストの変数	128
HP ILO ターゲットホストの変数	130
そのほかの IPMI ターゲットホストの変数	133
そのほかの遠隔管理ターゲットホストの変数	135
汎用ターゲットホストの変数	140
B Solaris Profile コンポーネントの変数	143
基本 OS 構成変数	143
base_config モジュール用のコンポーネント変数	143
追加ネットワークインタフェースの構成	151
IP ネットワークマルチパス (IPMP) の構成	151
追加ディスクの構成	153

N1 SPS リモートエージェントのコンポーネント変数 154
一般的な OS プロビジョニングプラグインの変数 156

C	追加 JET モジュールの作成	159
	範囲	159
	モジュール設計のガイドライン	159
	モジュールの対応範囲	160
	モジュールの依存関係	160
	モジュールの相互作用	160
	モジュールのコーディング	161
	モジュールのディレクトリ	161
	モジュール構成 (<i>module.conf</i> ファイル)	162
	モジュールのインタフェース	163
	copy_media インタフェース	163
	make_template インタフェース	163
	make_client インタフェース	164
	begin インタフェース	164
	install インタフェース	164
	check_client インタフェース	165
	ツールキットのサポート関数	166
	JET モジュールの追加	166
	▼ Flash モジュールを追加する	166

索引 169

表目次

表 1-1	OS プロビジョニングシステムコンポーネント	20
表 1-2	オペレーティングシステムによる OS プロビジョニングコンポーネント	21
表 1-3	オペレーティングシステムによる OS プロビジョニングテクノロジー	21
表 5-1	Solaris イメージサーバーのコンポーネント変数	46
表 7-1	Windows イメージサーバーコンポーネントの変数	83
表 7-2	Windows プロビジョニングプロファイル用のコンポーネント変数	86
表 8-1	プラットフォームによるターゲットホスト	90

はじめに

このマニュアルでは、N1 Service Provisioning System (N1 SPS) 製品によるオペレーティングシステムのプロビジョニング方法を説明します。

対象読者

このマニュアルの主な対象読者は、企業全体でオペレーティングシステムを保守する、N1 SPS ソフトウェアのシステム管理者およびオペレータです。このようなユーザーは、次の内容に関する十分な知識があると想定されています。

- N1 SPS 製品
- 標準的な UNIX™ および Windows のコマンドとユーティリティー
- プロビジョニングするオペレーティングシステムで使用可能な一般的な概念と管理機能

お読みになる前に

N1 SPS ソフトウェアの使用法に関する十分な知識のないユーザーは、次のマニュアルをお読みください。

- 『Sun N1 Service Provisioning System 5.1 システム管理者ガイド』
- 『Sun N1 Service Provisioning System 5.1 インストールガイド』
- 『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』

内容の紹介

第 1 章では、OS プロビジョニング機能の概要を説明します。

第 2 章には、OS プロビジョニング機能に関する既知の問題と最新情報が記載されています。

第 3 章では、OS のプロビジョニングをより簡単にサポートできるよう環境を設定する方法について説明します。

第 4 章では、OS プロビジョニングプラグインをインストールおよび構成する方法について説明します。

第 5 章では、Solaris システムのプロビジョニング方法に関する詳細な手順を説明します。

第 6 章では、Linux システムのプロビジョニング方法に関する詳細な手順を説明します。

第 7 章では、Windows システムのプロビジョニング方法に関する詳細な手順を説明します。

第 8 章では、OS プロビジョニングの適切なターゲットホストについて説明します。

付録 A では、各ターゲットホストタイプに適用される固有の変数について説明します。

付録 B では、Solaris プロビジョニングプロファイルに適用される固有の変数について説明します。

付録 C では、ユーザー固有のカスタムモジュールを作成することで Solaris プロビジョニング機能を拡張する方法について説明します。

第三者の関連 Web サイト

このマニュアル内で参照している第三者の URL は、追加の関連情報を提供します。

注 – このマニュアル内で引用する第三者の Web サイトの可用性について Sun は責任を負いません。Sun は、これらのサイトあるいはリソースに関する、あるいはこれらのサイト、リソースから利用可能であるコンテンツ、広告、製品、あるいは資料に関して一切の責任を負いません。こうしたサイトやリソース上で、またはこれらを経由して利用できるコンテンツ、製品、サービスを利用または信頼したことに伴って発生した (あるいは発生したと主張される) 実際の (あるいは主張される) 損害や損失についても、Sun は一切の責任を負いません。

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/supporttraining/	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せん。

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。	<code>.login</code> ファイルを編集します。 <code>ls -a</code> を使用してすべてのファイルを表示します。 <code>system%</code>

表 P-1 表記上の規則 (続き)

字体または記号	意味	例
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:
<i>AaBbCc123</i>	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 <i>filename</i> 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

OS プロビジョニングの概要

オペレーティングシステム (OS) プロビジョニングプラグインを使用すると、以前にアンインストールされたシステムに Solaris、Linux、および Windows オペレーティングシステムをプロビジョニングできます。プラグインは、異機種システム混在プラットフォームでの OS プロビジョニングを自動化して構成するソリューションを提供します。またプラグインには、インストールの履歴とバージョンを追跡する機能もあります。

この章では、次の項目について説明します。

- 15 ページの「OS プロビジョニングの概要」
- 16 ページの「機能」
- 17 ページの「プロビジョニングプロセスの概要」
- 17 ページの「OS プロビジョニングシステムソフトウェアのアーキテクチャー」
- 20 ページの「サポートされるシステム」

OS プロビジョニングの概要

OS プロビジョニングプラグインソリューションは、「ベアメタル」ハードウェアにオペレーティングシステムをプロビジョニングするコストと複雑さを軽減します。このソリューションは、ハードウェアの再配置を必要とせず、オペレーティングシステムのインストールに関する、手作業のエラーが発生しやすい処理を自動化します。プラグインソリューションは1つのシャシまたはシステムのレベルでのプロビジョニングを自動化するだけでなく、データセンターのプロビジョニングも自動化します。このソリューションにより、システム管理者と一般ユーザーにとっては、配備の柔軟性が高くなります。このソリューションを使用すると、市販のソフトウェアを使用する日常的なシステム管理作業を自動化できます。配備するハードウェア、オペレーティングシステム、または基本テクノロジーに関係なく、プロビジョニングプロセスは同じようなプロセスになります。

OS プロビジョニングプラグインソリューションは N1 Service Provisioning System (N1 SPS) 製品に基づいています。N1 SPS 製品は、一連のコマンド行ツールとグラフィカルユーザーインターフェースを提供しています。プロビジョニングソフトウェアツールにより、ネットワーク上に分散しているベアメタルハードウェアのオペレーティングシステムのプロビジョニングがより容易に行えるようになります。このアプローチを使用することで、オペレーティングシステムのプロビジョニング作業を、ネットワーク上で使用可能なその他のソフトウェアサービスと全く同じようなサービスと見なすことができます。OS プロビジョニング機能は、Web ブラウザまたはコマンド行から起動できます。

システム管理者は、必要なオペレーティングシステムを、サーバーに接続することなく自動的にサーバーにプロビジョニングできます。ほとんどの環境において、OS プロビジョニングプラグインソリューションでは、ターゲットシステムのプロンプト (OBP など) でコマンドを入力したり、電源スイッチのオン/オフを行ったりする必要はありません。この自動化された手作業の不要なインストールは、何百台というマシンにインストールしようとする管理者にとって、非常に便利です。そのため、ホストから要求を開始するのではなく、集中管理された位置からホストにオペレーティングシステムを展開できます。

OS プロビジョニングプラグインソリューションは優れたインフラストラクチャーソフトウェアプロビジョニングソリューションを提供するだけでなく、インストール履歴の追跡、アーカイブ、バージョン管理などの幅広いプロビジョニングの問題に対処します。このソリューションは、イメージのライフサイクル管理とサーバーのライフサイクル管理の両方に対処します。このソリューションは、さまざまなオペレーティングシステムのインストールメカニズムの複雑さを抽象化することによって、共通のオペレーションフローを提供します。

機能

OS プロビジョニングプラグインは、N1 SPS のブラウザインターフェースとコマンド行インターフェースを通じて、次の機能を提供します。

- カスタマイズされた OS プロファイルの情報の作成、管理、および登録
- プロビジョニング可能なホストの情報の作成、管理、および登録
- プロビジョニング可能なホストへのオペレーティングシステムのプロビジョニング
- ブートおよびインストールサーバーの情報の作成、管理、および登録
- OS プロビジョニングサーバーの情報の作成、管理、および登録

プロビジョニングプロセスの概要

オペレーティングシステム (OS) プロビジョニングは、特定のオペレーティングシステムを複数のホストにインストールする作業です。このプロセスは、次のような複数のハイレベルな手順から構成されています。

1. OS プロビジョニングをサポートするようにハードウェアを構成する。推奨される適切な構成と要件の詳細については、20 ページの「サポートされるシステム」および第 3 章で説明されています。
2. 35 ページの「OS プロビジョニングプラグインのインストール」で説明されているように、OS プロビジョニングプラグインをインストールする。
3. 38 ページの「OS プロビジョニングサーバーの作成」で説明されているように、OS プロビジョニングサーバーを作成する。
4. プロビジョニング対象の適切なターゲットホストを定義する。第 8 章を参照してください。
5. ターゲットホストに適切なオペレーティングシステムをプロビジョニングする。第 5 章、第 6 章、および第 7 章を参照してください。

OS プロビジョニングシステムソフトウェアのアーキテクチャー

OS プロビジョニングプラグインは、IPMI、ALOM、LOM、ILO、SC、および telnet などの異なるプロトコルをサポートするさまざまなターゲットホストに、異種オペレーティングシステムをプロビジョニングするフレームワークを提供します。

ソフトウェアアーキテクチャーの全体像

次の図に、OS プロビジョニングプラグインに使用される機能コンポーネントを示します。次の図では、N1 SPS マスターサーバーがファイアウォールの外部にあり、その他のコンポーネントがファイアウォールの内部にあると仮定しています。OS プロビジョニングサーバーは、サブネット、VLAN などから構成されるネットワークレイヤー 2 接続をユーザーが設定すると仮定しています。すべてのネットワーク情報は OS プロビジョニングサーバーに登録されます。

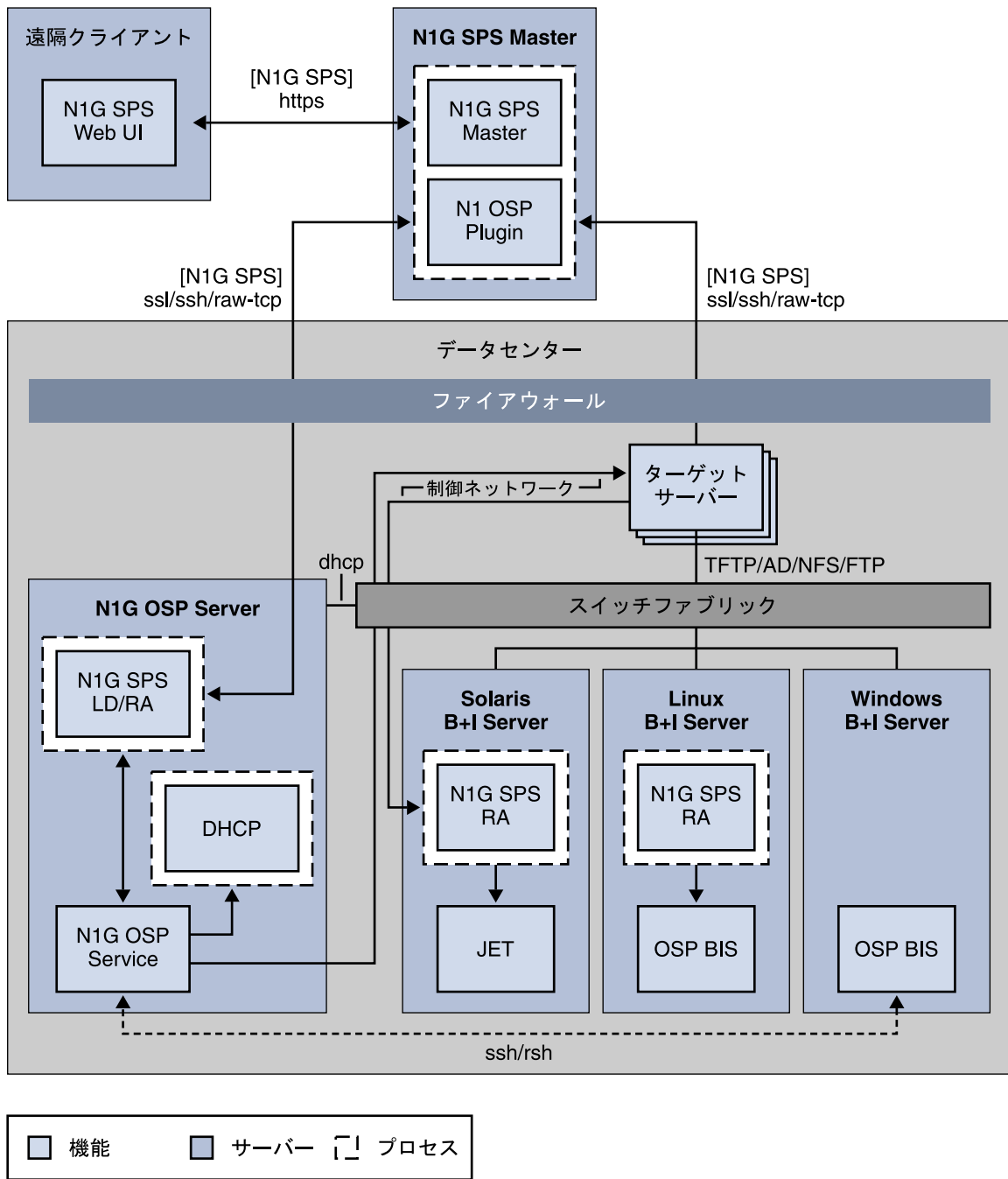


図 1-1 ソフトウェアアーキテクチャー図

この図では、OS プロビジョニングコンポーネント間の関係を説明しています (用語は多少省略して使用)。

- 遠隔クライアント – N1 SPS の遠隔クライアントは、ブラウザインタフェースとコマンド行インタフェースを実行します。遠隔クライアントには、マスターサーバーから独立したシステムを使用できます。
- N1G SPS Master – N1 SPS マスターサーバーは、N1 SPS ソフトウェアのメイン処理エンジンです。
- N1 OSP Plug-In – OS プロビジョニングプラグインは、マスターサーバーにインストールされます。このプラグインは、さまざまなプロトコルをサポートするさまざまなハードウェアプラットフォーム上でオペレーティングシステムをインストールする機能を提供します。
- N1 OSP Server – OS プロビジョニング制御サーバーは、通常、OS プロビジョニングサーバーと呼ばれる、OS プロビジョニングプラグインのメイン処理エンジンです。OS プロビジョニングサーバーは、OS プロビジョニングサービス (N1G OSP Service) を実行し、このサービスが OS プロビジョニングの動作を統合します。OS プロビジョニングサーバーは、適切なネットワーク管理プロトコル (IPMI、ALOM、LOM、ILO、および端末サーバーなど) を使用する制御ネットワークを介してターゲットホストを制御します。制御ネットワーク上のこれらのプロトコルは、電源、ブート、コンソールの各サービスの自動化に使用されます。

OS プロビジョニングサーバーは、幅広いネットワークトポロジ (複数のサブネット、VLAN など) をサポートしています。OS プロビジョニングサーバーには DHCP サーバーが含まれ、関連する IP アドレスおよびそのほかのブート固有の情報をターゲットホストに提供します。

- ブートおよびインストールサーバー – OS 固有のブートおよびインストールサーバーをサポートする次の 3 つのサーバーが示されています。
 - Solaris B + I Server – Solaris ブートおよびインストールサーバーは JumpStart™ Enterprise Toolkit (JET) を使用して、Solaris ディストリビューションメディアとインストールプロファイルのインストールを自動化します。
 - Linux B + I Server – Linux ブートおよびインストールサーバーは Linux Kickstart テクノロジを使用します。
 - Windows B + I Server – Windows ブートおよびインストールサーバーは Windows Remote Installation Services (RIS) テクノロジを使用します。

ブートおよびインストールサーバーは、自動化と監視の目的のため、OS 固有のブートおよびインストールサービスを搭載しています。Linux および Windows のブートおよびインストールサーバーの設定は、OS プロビジョニングプラグインの外部で行う必要があります。Linux システムの場合、N1 SPS リモートエージェント (RA) を手動でインストールする必要があります。Solaris システムの場合、OS プロビジョニングプラグインは RA をインストールおよび構成します。

ネットワークプロトコル

OS プロビジョニングプラグインは次のネットワークプロトコルを使用します。

- DHCP – DHCP は、スイッチファブリックを介して OS プロビジョニングサーバーとターゲットホストの間で IP アドレスとブートメタデータを割り当てるために使用されます。
- ネットワーク管理プロトコル – IPMI、ALOM、LOM、ILO および端末サーバーなどのプロトコルは、OS プロビジョニングサーバーからターゲットホストへの制御ネットワークを介して使用されます。
- TFTP、Active Directory、NFS、および FTP プロトコル – ブートおよびインストールサーバーは適切なプロトコルを使用して通信を行い、またターゲットホストに OS をインストールします。

サポートされるシステム

OS プロビジョニングプラグインソリューションは、オペレーティングシステムとハードウェアプラットフォームのマトリックスをサポートしています。このサポートは次のいくつかのカテゴリに分類されます。

- システムコンポーネント
- OS プロビジョニングコンポーネント
- ブートおよびインストールサーバー
- ターゲットホストのハードウェアプラットフォーム
- ターゲットホストのオペレーティングシステム

システムコンポーネント

次の表に、OS プロビジョニングに使用されるシステムコンポーネントを示します。

表 1-1 OS プロビジョニングシステムコンポーネント

サーバー	N1 SPS コンポーネント	OS プロビジョニングコンポーネント
マスターサーバー	N1 SPS マスター、エージェント、および CLI	OS プロビジョニングプラグイン
OS プロビジョニングサーバー	N1 SPS RA および CLI	OS プロビジョニングサービス
Solaris JET サーバー	N1 SPS RA および CLI	OS プロビジョニングのブートおよびインストールサービス (BIS)
Linux Kickstart サーバー	N1 SPS RA	OS プロビジョニングのブートおよびインストールサービス (BIS)

表 1-1 OS プロビジョニングシステムコンポーネント (続き)

サーバー	N1 SPS コンポーネント	OS プロビジョニングコンポーネント
Windows RIS サーバー	なし	OS プロビジョニングのブートおよびインストールサービス (BIS)

OS プロビジョニングコンポーネント

次の表に、プロビジョニングコンポーネントと、オペレーティングシステムに対するプロビジョニングコンポーネントの関係を示します。

表 1-2 オペレーティングシステムによる OS プロビジョニングコンポーネント

OS プロビジョニングコンポーネント	ホストのオペレーティングシステム
OS プロビジョニングプラグイン	Solaris 9、Solaris 10、RedHat AS 3.0、Windows Server 2000
OS プロビジョニングサービス	Solaris 9 (SPARC)
Solaris ブートおよびインストールサーバー	Solaris 9、Solaris 10
Linux ブートおよびインストールサーバー	RedHat AS 3.0
Windows ブートおよびインストールサーバー	Windows Server 2003

ブートおよびインストールサーバー

次の表に、各オペレーティングシステムに適用されるプロビジョニングテクノロジーを示します。

表 1-3 オペレーティングシステムによる OS プロビジョニングテクノロジー

オペレーティングシステム	プロビジョニングテクノロジー
Solaris 9 および 10	JET を介したカスタム JumpStart
RedHat Linux AS 3.0	Kickstart
Windows 2003	Remote Installation Service (RIS)

ターゲットホストのハードウェアプラットフォーム

サポートされるターゲットホストは、ホストがサポートするプロトコルにより区別されます。サポートされるプラットフォームと関連付けられるターゲットホストタイプの詳細なリストについては、89 ページの「ターゲットホストタイプ」を参照してください。

ターゲットホストのオペレーティングシステム

OS プロビジョニングプラグインは、次のオペレーティングシステムをプロビジョニングできます。

- Solaris 9 および 10 (SPARC および x86)
- RedHat Linux 3.0
- Windows 2000 (Service Pack 適用済み)

第 2 章

OS プロビジョニングプラグインのリリースノート

この章では、OS プロビジョニングプラグインの最新情報と既知の問題を説明します。

この章では、次の内容について説明します。

- [23 ページの「インストールの問題」](#)
- [23 ページの「実行時の問題」](#)

インストールの問題

インストールに関する問題はありません。

実行時の問題

オペレーティングシステムをプロビジョニングする際に次の問題が存在することが判明しています。

Solaris: root パスワードの間違った暗号化により JumpStart エラーが生じる (6245964)

説明: インストール中に次のメッセージが表示され、インストールが対話式になります。

```
root_password=C1z6pK2b6qw=  
syntax error line 2 position 15
```

Solaris OS プロファイルのパスワード変数 `sysidcfg_root_password_base_conf` は暗号化された値を持っています。ただし、ユーザーが指定したパスワードは Solaris 暗号化パスワードではありませんでした。

回避策: Solaris のツールを使用してパスワードを暗号化します。暗号化されたパスワードをユーザーが作成する適切なメカニズムは、パスワードを使用してユーザーを作成することです。/etc/shadow ファイルで暗号化されたパスワードを調べ、それを `sysidcfg_root_password_base_conf` 変数の値として使用します。

Windows ブートおよびインストールサーバーで OS プロビジョニングスクリプトの位置を変更できない (6251010)

説明: Windows ブートおよびインストールサーバーが作成されると、Windows ブートおよびインストールサーバーの OS プロビジョニングスクリプトの位置を変更できません。

回避策: 別の名前を持つ新しい Windows ブートおよびインストールサーバーを再度作成します。

DHCP 設定の警告がユーザーインタフェースに報告されない (6248485)

説明: DHCP 設定が正しくないため、プロビジョニング処理が失敗します。stdout または stderr には、表示されるメッセージがありません。

回避策: 誤った設定により、OS プロビジョニングサブネットが、間違った値を使用して作成されます。OS プロビジョニングサーバーの /var/adm/nlosp* ログファイルで、DHCP エラーがないか調べます。

EUC 以外のロケールでホストプロビジョン状態によって OS インストールログを表示できない (6255797)

説明: OS インストールで指定されたロケールに関係なく、インストールログファイルは常に関連する EUC ロケールになります。リモートエージェントのロケールがこの EUC ロケールとは異なる場合、ロケールが一致しないため、「Status Monitoring」ページから正しくログファイルを表示できません。

回避策: 適切なロケールを使用してサービスポートまたはコンソール (該当する場合) に接続し、ログファイルを直接表示します。

第 3 章

OS プロビジョニングの配備環境

この章では、OS プロビジョニングをサポートする環境を設定するためのガイドラインを説明します。

- 25 ページの「前提条件」
- 30 ページの「新しい環境の構成」
- 32 ページの「既存の環境の構成」

前提条件

OS プロビジョニングプラグインを使用してオペレーティングシステムのプロビジョニングを行うには、システム管理とネットワークの基本を理解する必要があります。また、オペレーティングシステムのプロビジョニングには、マシン間で基本的な IP 接続が存在している必要があります。

基本的な OS プロビジョニング環境

基本的な OS プロビジョニング環境には、次の要件が必要です。

- N1 SPS マスターサーバー — N1 Service Provisioning System ソフトウェアがインストールされ、マスターサーバーとして実行されるよう構成されているシステム。
- OS プロビジョニングサーバー — プラグインを介して OS プロビジョニングサーバーがインストールされる Solaris 9 9/04 またはそれ以降のリリースのシステム。OS プロビジョニングサーバーは DHCP サービスを実行して、ターゲットホストの要求に応答します。
- Solaris ブートおよびインストールサーバー — Solaris システムをプロビジョニングするには、Solaris 9 9/04 またはそれ以降のリリースがインストールされている Solaris ブートおよびインストールサーバーが必要です。

- Linux ブートおよびインストールサーバー — Linux システムをプロビジョニングするには、Linux がインストールされている Linux ブートおよびインストールサーバーが必要です。
- Windows ブートおよびインストールサーバー — Windows システムをプロビジョニングするには、Windows 2003 がインストールされている Windows ブートおよびインストールサーバーが必要です。
- OS プロビジョニングサーバー、ブートおよびインストールサーバー、プロビジョニングターゲットは、ネイティブ LAN、VLAN またはルーターを介して、あるネットワーク内でアクセス可能である必要があります。ネットワークの構成に関する詳細については、適切なネットワークと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。
- プロビジョニングターゲットからの DHCP パケットは、OS プロビジョニングサーバーに到達できる必要があります。

注 - N1 SPS マスターサーバー、OS プロビジョニングサーバー、および Solaris ブートおよびインストールサーバーは、1 つの物理システムであることが可能です。ただし、1 つのシステムでこれら 3 つすべてのサーバーを実行すると、サーバーの負荷が増大し、またサーバーが処理しなければならないネットワークトラフィックが増大します。これらを分離することで、将来の高いスケーラビリティが実現できます。

ターゲットホスト

OS プロビジョニング用のプロビジョニング可能なターゲットシステムを設定する必要があります。OS プロビジョニングサーバーは、MAC アドレス、GUID、遠隔管理接続、およびアクセス情報など、これらのターゲットに関する情報を認識する必要があります。ターゲットの定義方法に関する詳細については、[第 8 章](#)を参照してください。

ネットワーク

OS プロビジョニングプラグインは、幅広いネットワーク構成およびトポロジで動作するよう設計されています。そのため、プラグインはネットワークトポロジを規定したり、スイッチやルーターなどのネットワーク要素を操作したりする必要はありません。ただし、プラグインは次のネットワーク通信の存在に依存します。

- OS プロビジョニングサーバーのプロビジョニングインタフェースと、ターゲットホストのプロビジョニングインタフェースの間のレイヤー 2 接続
- ブートおよびインストールサーバーのプロビジョニングインタフェースと、ターゲットホストのプロビジョニングインタフェースの間の IP 接続
- OS プロビジョニングサーバー、ターゲットホストのネットワーク管理ポート、およびブートおよびインストールサーバーの制御ネットワークインタフェースの間の IP 接続

ネットワークインフラストラクチャーに対するこれらの要件は、OS プロビジョニングサーバーの機能にとって中心的な2つのネットワークタイプのニーズによって課せられるものです。これらのネットワークタイプは制御ネットワークとプロビジョニングネットワークです。

注- アクセスネットワークは、OS プロビジョニングサーバーとブートおよびインストールサーバーへのアクセスに使用されるネットワークです。アクセスネットワークの例としては企業内イントラネットがあります。このネットワークは、OS プロビジョニング機能には必要ありません。セキュリティの観点から、アクセスネットワークを、制御ネットワークおよびプロビジョニングネットワークから分離する必要があります。

次の図に、ネットワーク環境を示します。

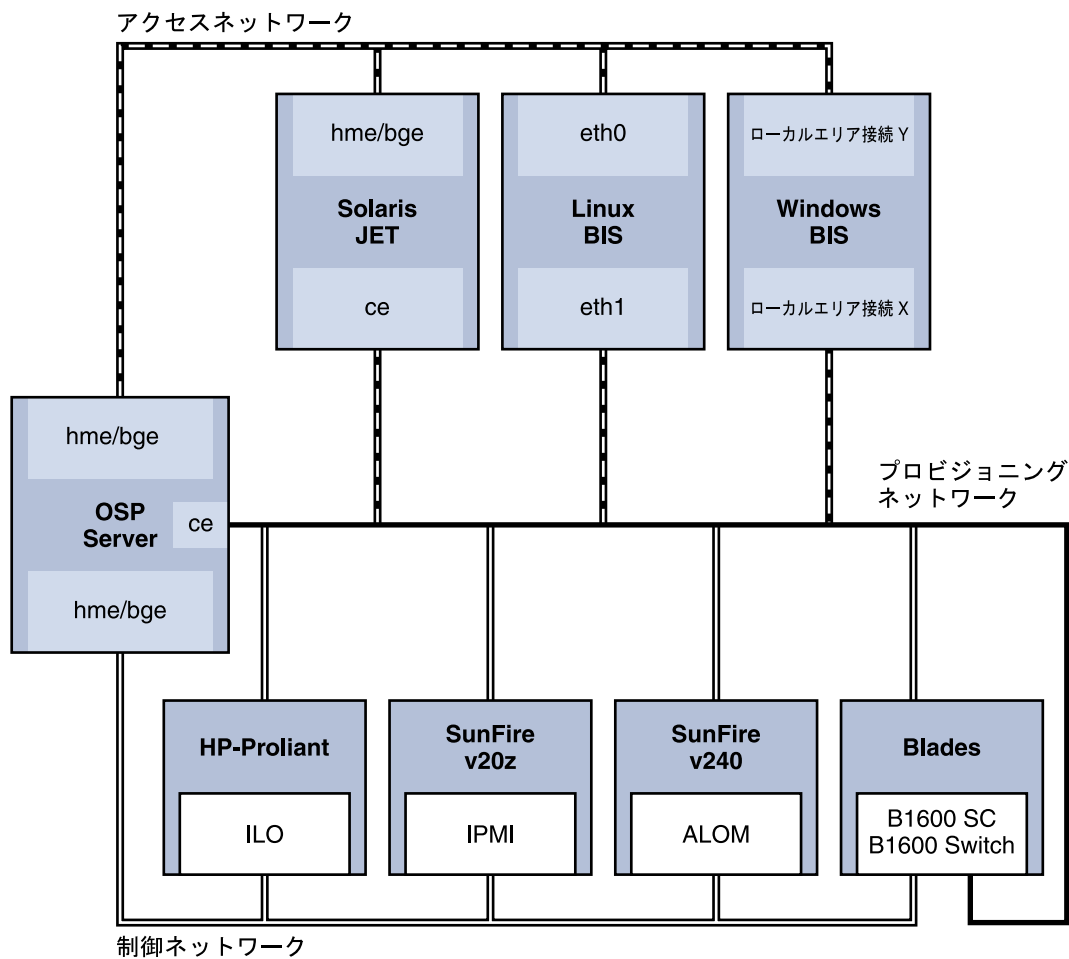


図 3-1 OS プロビジョニング用のネットワーク環境図

プロビジョニングネットワーク

プロビジョニングネットワークは、OS プロビジョニングサーバーのプロビジョニングインタフェース、ターゲットプラットフォームのプロビジョニングインタフェース、および1つまたは複数のブートおよびインストールサーバーのプロビジョニングインタフェースから構成されています。プロビジョニングネットワークは1つまたは複数のサブネットから構成できます。OS プロビジョニングプラグインのインストールは、OS プロビジョニング用の複数のプロビジョニングネットワークの使用をサポートしています。ネットワークベースのプロビジョニングに必要なプロトコルとテクノロジーにより、これらのプロビジョニングネットワークに関する次の要件が規定されます。要件は次の通りです。

- OS プロビジョニングサーバーのプロビジョニングインタフェースおよびターゲットプラットフォームのプロビジョニングインタフェースは、共通のレイヤー 2 ブロードキャストドメインに存在する必要がある。
- ホストにサービスを提供するブートおよびインストールサーバーのプロビジョニングインタフェースには、ターゲットプラットフォームのプロビジョニングインタフェースのサブネットへの IP 接続が必要である。これは、次の 2 つの方法のいずれかで実現できます。
 - ブートおよびインストールサーバーはプロビジョニングサブネットにネットワークインタフェースを持っている。
 - ブートおよびインストールサーバーは、ルーティングによりプロビジョニングサブネットから到達可能なプロビジョニングインタフェースを持っている。

制御ネットワーク

制御ネットワークは、主に次の 2 つの機能のために OS プロビジョニングサーバーにより使用されるネットワークです。

- ターゲットホストのネットワーク管理ポートを管理し、電源の状態、ブートの順序、およびターゲットプラットフォームのコンソールを制御する
- プロビジョニング操作のためにブートおよびインストールサーバーを構成する

制御ネットワークは純粋な IP ネットワークを使用でき、またシリアル/端末サーバー要素を持つ場合もあります。OS プロビジョニングサーバーは、IP ネットワークでブートおよびインストールサーバーと通信します。同時に、ターゲットホストのネットワーク管理ポートとの通信が IP ネットワークまたはシリアルネットワークで行われる場合があります。制御ネットワークは数多くのサブネット上に存在することができます。制御ネットワークの要件は、ブートおよびインストールサーバーとターゲットネットワーク管理ポートが、OS プロビジョニングサーバーから到達可能であることのみです。

スイッチ型ネットワーク

スイッチ型ネットワークの要件は、スイッチを使用する環境では特別な意味を持ちません。スイッチ型ネットワークでは、スイッチ型接続はトランクまたはアクセス (非トランク) モードのいずれかを取ることが可能です。制御ネットワークの場合、OS プロビジョニングサーバーからの IP ルーティングが唯一の要件であるため、スイッチ型接続はアクセスモードになることができます。プロビジョニングネットワークは、プロビジョニングネットワークの設計に応じて、スイッチ型ポートをトランクモードとアクセスモードのいずれかにすることができます。

セキュリティ

OS プロビジョニングプラグインソフトウェアは N1 SPS セキュリティモデルを利用します。異なるサーバー間でのほとんどの通信は、N1 SPS リモートエージェント (RA) を介して行われます。セキュリティ保護された通信を行うように RA を構成します。マスターサーバーと RA の間のセキュリティ保護された通信を有効にする方法の詳細については、ドキュメントを参照してください。

ターゲットの遠隔管理を行うため、OS プロビジョニングサーバーには暗号化されたパスワードが格納されます。パスワードの暗号化の詳細については、[107 ページ](#)の「パスワードの暗号化」を参照してください。

Windows ブートおよびインストールサーバーと通信するためには、RSH サービスまたは SSH サービスのいずれかを有効にする必要があります。OS プロビジョニングサーバーと Windows ブートおよびインストールサーバーの間の通信のセキュリティを保護するには、SSH サービスを使用します。詳細については、[75 ページ](#)の「Windows RIS サーバーに Windows SSH サーバーをインストールする」を参照してください。

新しい環境の構成

Sun Data Center Reference Architecture は、汎用データセンター構成を定義するためのベストプラクティスを取り込み、適用するものです。このアーキテクチャーは、低リスク、低総所有コスト (TCO) で確実かつ迅速に組み立て、テストし、配備することができます。Data Center Reference Architecture Implementations は Sun Data Center Reference Architecture のインスタンス化で、顧客の要件を満たすために、サービスとともに実際のハードウェアとソフトウェアに完全な詳細事項を提供します。Data Center Reference Architecture Implementations は、中小規模および大規模データセンター用に事前に設計およびテストされたコンポーネントのグループで、企業の統合プロジェクトおよび移行プロジェクト用の、運用準備が完了したターゲット環境を提供します。

Sun Data Center Reference Architecture Implementation フレームワークは、SunFire Server、Sun StorEdge™ ストレージレイ、Sun Java™ Enterprise System および Solaris ソフトウェアだけでなく、LAN および SAN インフラストラクチャーからなる柔軟性の高い組み合わせです。詳細については、Sun Data Center Reference Architecture web site を参照してください。

プロセスの概要

1. N1 SPS マスターサーバー、OS プロビジョニングサーバー、ブートおよびインストールサーバー用にハードウェアを準備します。
2. N1 SPS ソフトウェアを入手します。

- 『Sun N1 Service Provisioning System 5.1 インストールガイド』の「N1 Service Provisioning System 5.1 のインストール」で説明されている手順で N1 SPS マスターサーバーをインストールします。
- OS プロビジョニングサーバーに N1 SPS RA および N1 SPS コマンド行インタフェース (CLI) をインストールします。
- Solaris ブートおよびインストールサーバーに N1 SPS RA および N1 SPS CLI をインストールします。
- Linux ブートおよびインストールサーバーに N1 SPS RA をインストールします。
- OS プロビジョニングサーバー、Solaris ブートおよびインストールサーバー、Linux ブートおよびインストールサーバーで RA を準備します。詳細については、『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」を参照してください。

注 - 安全のため、N1 SPS データベースをバックアップします。詳細については、『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。

▼ マスターサーバーを有効にしてセッション ID を使用する

- 手順 1. マスターサーバーの構成ファイルを編集します。
デフォルトでは、このファイルは次の位置にあります。

```
/opt/SUNWn1sps/N1_Service_Provisioning_System_5.1/server/config/config.properties
```

- これが既存の N1 SPS インストールである場合、次の手順に従ってください。
 - 次の形式に似たセッション ID エントリを検索します。
config.allowSessionIDOnHosts=masterserver,biss1
 - イコール記号のあとの値を、OS プロビジョニングサーバーと Solaris ブートおよびインストールサーバーの名前に変更します。
次に例を示します。
config.allowSessionIDOnHosts=myspsserver,sol10bis
- これが N1 SPS の新規インストールである場合、次のような行を追加します。
config.allowSessionIDOnHosts=masterserver,biss1
イコール記号のあとの値には、OS プロビジョニングサーバーと Solaris ブートおよびインストールサーバーの名前が含まれている必要があります。

4. 使用する環境の大域プラン実行タイムアウトを調整します。

config.properties ファイルの次のエントリを変更します。

```
pe.defaultPlanTimeout=12000
pe.nonPlanExecNativeTimeout=12000
```

タイムアウトは秒単位です。タイムアウトは、サイトで予想される最も長いプラン実行処理より長くする必要があります。デフォルトのプランタイムアウトは 30 分 (1800 秒) です。デフォルトのネイティブタイムアウトは 10 分 (600 秒) です。上記の例は任意の値ですが、200 分 (12000 秒) という高い値を設定しています。

5. これらの変更を有効にするには、マスターサーバーを停止して再起動します。

n1sps としてマスターサーバーにログインし、次のコマンドを入力します。

```
# cr_server stop
# cr_server start
```

デフォルトでは、これらのコマンドは次のファイル内にあります。

```
/opt/SUNWn1sps/N1_Service_Provisioning_System_5.1/server/bin
```

既存の環境の構成

OS プロビジョニングプラグインを使用すると、既存のサーバーおよびネットワーク環境に OS をプロビジョニングできます。以降の節では、既存の環境でプラグインをどのように使用するかを詳細に説明します。

ハードウェアおよびソフトウェアの構成

N1 SPS マスターサーバー、OS プロビジョニングサーバー、Solaris ブートおよびインストールサーバー、Linux ブートおよびインストールサーバー、および Windows ブートおよびインストールサーバーをサポートするハードウェアが用意されていることを確認します。適切なシステムの詳細については、[20 ページの「サポートされるシステム」](#)を参照してください。

ネットワーク環境

マスターサーバー、OS プロビジョニングサーバー、およびブートおよびインストールサーバーが IP ネットワークを介して相互に接続できることを確認します。

サーバーを同時にプロビジョニングする十分な帯域幅があることを確認します。帯域幅の要件は、実行する同時プロビジョニング処理の数に応じて異なります。

注 – 同時 OS インストールは大量の帯域幅を必要とし、帯域幅が使用できない場合には障害やタイムアウトが発生する場合があります。問題を回避するには、物理的にトラフィックを分離するか、より多くのブートおよびインストールサーバーを配備します。

DHCP サービス

OS プロビジョニングサーバーは独自の DHCP サービスを使用します。DHCP サービスは、ターゲットにインストール時パラメータとインストール時 IP アドレスを提供するために、プロビジョニング処理中に使用されます。DHCP サービスは、プロビジョニングの対象ではないクライアントには応答しません。そのため、このサブネットでサービスを提供するほかの DHCP サービスが存在する場合、これらのサービスがプロビジョニング処理中にターゲットに応答していないことを確認してください。OS がプロビジョニングされると、ターゲットに応答するよう DHCP を再度アクティブにすることができます。OS プロビジョニングサーバーを同じサブネットに配置するか、ルーティングによって、ターゲットの DHCP パケットが OS プロビジョニングサーバーに到達できることを確認してください。

ターゲットホスト

OS プロビジョニングプラグインは、プロビジョニング中の電源オン/オフのサイクルを自動化できます。ターゲットの遠隔管理インタフェース (存在する場合) を有効にします。ターゲットが遠隔管理をサポートしていない場合は、汎用ターゲットを使用します。ターゲットホストの詳細については、第 8 章を参照してください。

N1 Service Provisioning System ソフトウェア

N1 SPS ソフトウェアがバージョン 5.1 またはそれ以降であることを確認します。

既存の Solaris JET 環境

JumpStart Enterprise Toolkit (JET) テクノロジーを実行している場合、OS プロビジョニングプラグインを使用する前には、JET 製品をアンインストールする必要があります。詳細については、45 ページの「Solaris JET サーバーの設定」を参照してください。

OS プロビジョニングプラグインに付属する SUNWjet パッケージのデフォルトのベースディレクトリは /opt/SUNWjet です。旧バージョンの SUNWjet は、デフォルトのベースディレクトリ /opt/jet を使用していました。

既存の JET パッケージを使用している場合は、45 ページの「Solaris JET サーバーの設定」で説明されている手順で、既存のパッケージをアンインストールしてから JET サーバーを作成します。このプロセスでは以下の作業を実行します。

1. /opt/SUNWjet で OS プロビジョニングプラグインに付属するバージョンの SUNWjet をインストールする。
2. /opt/jet/Products 内のすべての既存の JET 製品モジュールに、/opt/SUNWjet/Products へのシンボリックリンクを作成する。

処理が完了すると、OS プロビジョニングプラグインを使用して新しい Solaris プロファイルを作成する場合、サーバーに以前インストールされていた任意の JET 製品モジュールを名前によって含めることができます。

以前の /opt/jet/Templates および /opt/jet/Clients 領域は変更されません。必要に応じてこれらの領域を参照できます。また値によっては、OS プロビジョニングプラグインを使用して新しい Solaris プロファイルを作成するのに役立つ場合があります。

第 4 章

OS プロビジョニングプラグインのインストールと構成

オペレーティングシステム (OS) プロビジョニングの観点から見ると、インストールは次の 3 つの段階から構成されています。

- 『Sun N1 Service Provisioning System 5.1 インストールガイド』で説明されている N1 SPS ソフトウェアのインストール
- OS プロビジョニングプラグインのインストール
- プロビジョニングするオペレーティングシステム環境のインストール

3 つのインストールのすべての段階には、複数の構成手順もあります。

この章では、次の内容について説明します。

- 35 ページの「OS プロビジョニングプラグインのインストール」
- 38 ページの「OS プロビジョニングサーバーの作成」
- 40 ページの「サブネットの作成と登録」

OS プロビジョニングプラグインのインストール

OS プロビジョニングプラグインの入手

OS プロビジョニングプラグインは、N1 SPS ソフトウェアに対するプラグインとしてパッケージ化されています。プラグインは Java™ アーカイブ (JAR) ファイルにパッケージ化されています。OS プロビジョニングプラグインのプラグインファイルは N1 SPS Supplement CD または Sun Download Center から入手できます。

OS プロビジョニングプラグインは、2 つの異なる JAR ファイルからインポートに使用できます。状況に応じて、適切なファイルを選択してください。

- OS プロビジョニングプラグインを初めてインポートする場合は、`com.sun.nlosp_2.0.jar` ファイルを使用します。
- 旧バージョンの OS プロビジョニングプラグインをすでにインポートしている場合は、`com.sun.nlosp_1.0_2.0.jar` ファイルを使用します。

OS プロビジョニングプラグインの N1 SPS への追加

特定のプラグインを N1 SPS 製品に認識させるには、プラグインをインポートする必要があります。プラグインをインポートするには、『Sun N1 Service Provisioning System 5.1 システム管理者ガイド』の第 5 章「プラグインの管理」で説明されている手順に従います。

1. N1 SPS ブラウザインタフェースのメインウィンドウの「Administrative」セクションで「Plug-ins」をクリックします。
2. 「Plug-ins」ページの「Action」列で「Import」をクリックします。
3. JAR ファイルをダウンロードした場所に移動します。
4. 「Continue to Import」ボタンをクリックします。

インポートが正常に完了すると、プラグインが提供するオブジェクトを示すプラグインの詳細ページが表示されます。

また、コマンド行からプラグインアーカイブファイルをインポートすることもできます。

- 以前に OS プロビジョニングプラグインを使用したことがない場合は、次のコマンドを使用してプラグインをインポートします。

```
% cr-cli -cmd plg.p.add -path com.sun.osp_2.0.jar -u username -p password
```

- 旧バージョンの OS プロビジョニングプラグインがすでにある場合は、次のコマンドを使用してプラグインをインポートします。

```
% cr-cli -cmd plg.p.add -path com.sun.osp_1.0_2.0.jar -u username -p password
```

▼ OS プロビジョニングプラグインをアップグレードする

OS プロビジョニングプラグインをバージョン 1.0 からバージョン 2.0 へアップグレードするには、次の手順に従います。

- 手順 1. 『Sun N1 Service Provisioning System 5.1 インストールガイド』の第 10 章「N1 Service Provisioning System 5.1 へのアップグレード」で説明されている手順に従って、N1 SPS ソフトウェアをバージョン 5.1 にアップグレードします。

2. `com.sun.osp_1.0_2.0.jar` ファイルを取得し、インポートします。
3. OS プロビジョニング (制御) サーバーをアップグレードします。
OS プロビジョニングプラグイン 2.0 ソフトウェアを使用して、既存のすべての OS プロビジョニングサーバーを再度作成する必要があります。38 ページの「OS プロビジョニングサーバーの作成」を参照してください。
4. Solaris のプロビジョニングの場合、JET Solaris イメージサーバーをアップグレードします。
OS プロビジョニングプラグイン 2.0 ソフトウェアを使用して、既存のすべての JET Solaris イメージサーバーを再度作成する必要があります。45 ページの「JET Solaris イメージサーバーを作成する」を参照してください。
5. 既存のすべての Solaris OS プロファイル内の遠隔エージェントの変数を変更します。
プロファイルの `product_version_spsra` 変数は、現在の N1 SPS RA ソフトウェアのバージョンを反映している必要があります。既存のすべてのプロファイルを編集し、`product_version_spsra` 変数の値を 5.1 に変更します。この変数の詳細については、154 ページの「N1 SPS リモートエージェントのコンポーネント変数」を参照してください。

▼ 旧バージョンのコンポーネントにアクセスする

新しいバージョンのプラグインにアップグレードすると、そのプラグインの「Common Tasks」ページが更新され、新しいバージョンのプラグインを使用してインストールされたコンポーネントへのリンクが表示されます。この機能により、最新の機能と改善点を含むコンポーネントを簡単に表示できます。古い機能に依存するコンポーネントは、「Common Tasks」ページからはリンクされません。

以前にインストールしたコンポーネントが見つからない場合、おそらく、そのコンポーネントは旧バージョンのプラグインから作成されたものです。旧バージョンのコンポーネントを表示したり、旧バージョンのコンポーネントで作業するには、次の手順に従います。

- 手順
1. 「Common Tasks」ページで適切なコンポーネントプロシージャをクリックします。
「Component Details」ページが表示されます。
 2. 「Component Details」ページで「Version History」をクリックします。
コンポーネントとバージョンのリストが表示されます。
 3. 適切なリンクをクリックします。
 - コンポーネントを使用した作業を行うには、使用するコンポーネントに該当するバージョン番号または「Details」リンクをクリックします。

- コンポーネントがインストールされている場所を表示するには、「**Where Installed**」をクリックします。

OS プロビジョニングサーバーの作成

OS プロビジョニングプラグインがインポートされていても、OS プロビジョニングサーバーを作成し設定するまでは、オペレーティングシステムをプロビジョニングできません。

▼ OS プロビジョニングサーバーを作成する (ブラウザインタフェース)

この作業が終了すると、OS プロビジョニングソフトウェアが OS プロビジョニングサーバーにインストールされます。また、OS プロビジョニングサーバーには DHCP サービスがインストールされます。

- 手順
1. **N1 SPS** ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**OSP Control Server**」セクションで「**Create**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**Service**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**Service**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 必要に応じて変数を変更します。

変数名	説明
installPath	OS プロビジョニングソフトウェアをインストールするベースディレクトリ。
vhost_name	OS プロビジョニングサーバーを表す仮想ホストの名前。
host_dir	ターゲットホストファイルが存在するディレクトリへのパス。
profile_dir	OS プロファイルが存在するディレクトリへのパス。
subnet_dir	サブネットファイルが存在するディレクトリへのパス。
logs_console_dir	ログとコンソールファイルが存在するディレクトリへのパス。

d. 変数セットを保存します。

e. 「**Plan Parameters**」テーブルの「**Service**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。

5. **OS** プロビジョニングサーバーを作成するホストを選択します。

6. 「**Run Plan (includes preflight)**」をクリックします。

このプランの実行には通常は約 15 分かかります。インストールの進行中、進行状況を参照するには「Plan Run」ウィンドウの「Details」リンクをクリックします。

7. **OS** プロビジョニングサーバーが正常に作成されたことを確認するには、プロビジョニングサーバーのウィンドウの左側にある「**Hosts**」リンクをクリックします。

-osp が付加された仮想ホストの名前が表示されます。たとえば、ホスト masterserver にインストールした場合、仮想ホストは masterserver-osp となります。

▼ OS プロビジョニングサーバーを作成する (コマンド行インタフェース)

手順 ● コマンド行からサーバーを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID "NM:/com/sun/nlosp/untyped/Service-create" \
-tar H:NM:masterserver -comp + -vs + -pto 30 -nto 10
```

注 - ターゲットは OS プロビジョニングサーバーである必要があります。この例では、N1 SPS マスターサーバーが OS プロビジョニングサーバーとして使用されません。

サブネットの作成と登録

OS プロビジョニングサーバーは DHCP サービスを提供します。DHCP サービスは、ターゲットのプロビジョニングに使用されるすべてのサブネット上で待機する必要があります。N1 SPS インタフェースでは、OS プロビジョニングサーバーが DHCP 要求に応答できるよう、ターゲットのプロビジョニングに使用されるサブネットを特定する必要があります。OS プロビジョニングサービスは、プロビジョニング処理中に、プロビジョニングサーバー上の必要なアドレスを使用してインタフェースを作成します (まだ作成されていない場合)。

▼ OS プロビジョニングサーバーのサブネットを特定する (ブラウザインタフェース)

ブラウザインタフェースからサブネットを管理するには、次の手順に従います。

- 手順
1. **N1 SPS** ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**OSP Subnets**」セクションで「**Manage**」をクリックします。
 3. 「**Component Details**」ページの「**Create**」行で「**Run action**」をクリックします。
 4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**Service**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**Service**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。

- c. 必要に応じて変数を変更します。

変数	説明	例
installPath	サブネットアドレス	10.42.42.0
mask	サブネットマスク	255.255.255.0
gateway	サブネットのゲートウェイ	10.42.42.1
host_interface	このサブネットに使用される OS プロビジョニングサーバー上のインタフェース	hme0
host_address	host_interface に割り当てられるホストアドレス	10.42.42.1

注 - OS プロビジョニングソフトウェアは、インタフェースの作成とアドレスの割り当てを、これらの値がまだ設定されていない場合、プロビジョニング処理時に行います。

- d. 変数セットを保存します。
- e. 「Plan Parameters」テーブルの「Service」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
5. このサブネットを適用する仮想ホストを選択します。

ヒント - OS プロビジョニングサーバーの仮想ホストの名前の最後は `-osp` です。

6. 「Run Plan (includes preflight)」をクリックします。

▼ OS プロビジョニングサーバーのサブネットを特定する (コマンド行インタフェース)

コマンド行からサブネットを管理するには、次の手順に従います。

- 手順 1. 変数セットを作成します。
次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/untyped/Subnet -name "subnet1" \  
-u admin -p admin -vars "installPath=10.42.42.0;mask=255.255.255.0;gateway=10.42.42.1; \  

```

```
host_interface=ce8000;host_address=10.42.42.1"
```

変数の詳細については、40 ページの「OS プロビジョニングサーバーのサブネットを特定する (ブラウザインタフェース)」の手順 4 を参照してください。

2. 前の手順で作成した変数セットを使用して、プランを実行し、サブネットを作成します。

次の例のようなコマンドを入力します。

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/nlosp/untyped/Subnet-create \
-tar H:NM:masterserver-osp -comp + -vs subnet1 -pto 30 -nto 10
```

▼ 新しいプロビジョニングサブネットを追加する

複数のサブネットを使用して、オペレーティングシステムをプロビジョニングすることができます。以下の手順では、OS プロビジョニングサーバーに新しいサブネットを追加する方法を説明します。

始める前に OS プロビジョニングサーバー、適用可能なオペレーティングシステムのブートおよびインストールサーバー、およびターゲットホストが同じ IP ネットワーク内に存在することを確認します。

- 手順
1. 40 ページの「OS プロビジョニングサーバーのサブネットを特定する (ブラウザインタフェース)」の説明に従って、新しいサブネットコンポーネントを作成します。

IP 接続に関しては 2 つのオプションがあります。

 - OS プロビジョニングサーバーの新しい IP アドレスを作成する。
 - 既存の IP アドレスを使用するが、DHCP パケットがターゲットから OS プロビジョニングサーバーに到達する経路を作成する。
 2. ターゲットによりブートおよびインストールサーバーに到達できるよう、新しいサブネットのブートおよびインストールサーバー上に新しいインタフェースを作成するか、経路を定義します。
 3. この新しいサブネットで OS メディアを供給するために、ファイルシステム共有が更新されることを確認します。
 4. ターゲットがこのサブネット内に存在することを確認します。
 5. IP アドレスが新しいサブネット内に存在するよう、プロビジョニングするプロファイルを編集します。

第 5 章

Solaris オペレーティングシステムのプロビジョニング

この章では、OS プロビジョニングプラグインを使用して、Solaris オペレーティングシステムをターゲットホストにインストールする方法について説明します。

この章では、次の内容について説明します。

- 43 ページの「JumpStart Enterprise Toolkit (JET) テクノロジー」
- 45 ページの「Solaris プロビジョニングプロセスの概要」
- 45 ページの「Solaris JET サーバーの設定」
- 47 ページの「Solaris イメージおよびプロファイルの作成」
- 52 ページの「ターゲットホストへの OS のインストール」
- 55 ページの「JET Solaris サーバーの管理作業」
- 57 ページの「custom モジュールの使用法」

JumpStart Enterprise Toolkit (JET) テクノロジー

OS プロビジョニングプラグインは、JumpStart™ Enterprise Toolkit (JET) テクノロジーを通じて提供される機能を使用して、Solaris オペレーティングシステム (OS) をプロビジョニングします。JET は、Sun が開発した JumpStart テクノロジーを拡張したもので、ネットワーク経由での Solaris OS のインストールを自動化します。JET を介した OS プロビジョニングプラグインはこの機能をさらに拡張しつつ、複雑さを解消しています。

JET モジュール

JET テクノロジは JumpStart サーバーに製品固有のモジュールを提供し、そのモジュールは Solaris OS およびその他のほかの製品を構造化された状態でインストールします。この構造により、JumpStart 終了スクリプトの「アドホック」スクリプティングを通じて実装可能な機能が拡張されます。

OS プロビジョニングプラグインは、次の 3 つの JET モジュールを提供しています。

- `base_config` – Solaris OS のインストールと構成を行います。`base_config` 変数の詳細については、143 ページの「基本 OS 構成変数」を参照してください。
- `spsra` – Solaris システム上で N1 SPS リモートエージェント (RA) のインストールと構成を行います。`spsra` 変数の詳細については、154 ページの「N1 SPS リモートエージェントのコンポーネント変数」を参照してください。
- `custom` – Solaris パッケージ、パッチ、およびファイルの任意のリストをインストールします。また任意のスクリプトのコレクションを実行できます。カスタムモジュールの詳細については、57 ページの「custom モジュールの使用法」を参照してください。

JET を介した Solaris のインストールプロセス

JumpStart Enterprise Toolkit の構築シーケンスは次のようになります。

1. 標準 Solaris インストールフェーズ
2. 標準 JumpStart 終了スクリプトの呼び出し
3. 個別のモジュール「インストール」スクリプトの呼び出し
4. ターゲットサーバーの再起動
5. (省略可能) プラットフォーム関連のインストール作業 (各レベルのあとに再起動)
6. (省略可能) アプリケーション関連のインストール作業 (各レベルのあとに再起動)
7. (省略可能) 最終インストール作業 (再起動なし)
8. コンソールにログインプロンプトが表示される

最初に再起動したあとの省略可能な手順は、ターゲットサーバーテンプレート内で構成される個別のモジュールに依存します。最初に再起動したあとツールキットが追加の作業を実行することをモジュールが要求するように、モジュールを記述することができます。この要求では、モジュールは作業がプラットフォーム関連領域とアプリケーション関連領域のどちらで行われるか、または追加の再起動が予定されていない場合最後に作業を行う必要があるかどうかを特定できます。

Solaris プロビジョニングプロセスの概要

Solaris オペレーティングシステムをプロビジョニングするには、次のような高度な作業を実行する必要があります。

1. JET サーバーを設定します。
2. Solaris イメージを作成するか、JET サーバーに既存のイメージを添付します。
3. ターゲットホストを作成します。
4. ターゲットホストにイメージをインストールします。

Solaris JET サーバーの設定

Solaris ブートおよびインストールサーバーは JET サーバーです。

▼ JET Solaris イメージサーバーを作成する

N1 SPS のブラウザインタフェースまたはコマンド行インタフェースから JET イメージサーバーを作成できます。コマンド行からサーバーを作成するには、次のコマンドを入力します。

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/nlosp/untyped/Jet-create \  
-tar H:NM:biss1 -comp + -vs + -pto 30 -nto 10
```

ブラウザインタフェースからサーバーを作成するには、次の手順に従います。

- 手順
1. Solaris ブートおよびインストールサーバーにするシステムにログインします。
 2. 現在 **JumpStart Enterprise Toolkit** を使用している場合は、これらのファイルを削除します。
次のコマンドを使用します。# **pkgrm SUNWjet**
 3. N1 SPS ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 4. 「**OS Provisioning Common Tasks**」ページの「**JET Solaris Image Servers**」セクションで「**Create**」をクリックします。
 5. 「**Plans Details**」ページの「**Run**」をクリックします。
 6. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**JET**」コンポーネント行にあるドロップダウンメニューから名前を選択します。

- 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**JET**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 変数を確認し、必要に応じて変数を変更します。
イメージサーバーコンポーネントの変数はごく少数で、それらのほとんどは変更しません。これらの変数とそのデフォルト値のリストは、[46 ページ](#)の「[Solaris イメージサーバーのコンポーネント変数](#)」を参照してください。
 - d. 変数セットを保存します。
 - e. 「**Plan Parameters**」テーブルの「**JET**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
- 7. **JET** イメージサーバーを作成するホストを選択します。
- 8. 「**Run Plan (includes preflight)**」をクリックします。
- 9. **JET** イメージサーバーが正常に作成されたことを確認するには、プロビジョニングサーバーのウィンドウの左側にある「**Hosts**」リンクをクリックします。
-jet が付加された仮想ホストの名前が表示されます。たとえば、ホスト masterserver にインストールした場合、仮想ホストは masterserver-jet となります。

Solaris イメージサーバーのコンポーネント変数

イメージサーバーコンポーネントには次の変数が適用されます。

表 5-1 Solaris イメージサーバーのコンポーネント変数

変数名	説明	デフォルト値
installPath	JET パッケージをインストールする位置。	/opt/SUNWjet
sps_cli	N1 SPS コマンド行インタフェースの位置。	/opt/SUNWn1sps/N1_Service_Provisioning_System_5.1/cli/bin/cr_cli
jetHost	仮想ホスト名に使用する名前。デフォルトでは、仮想 JET ホスト名は、-jet が追加されたターゲットホストのホスト名に設定されています。	:[target]-jet

表 5-1 Solaris イメージサーバーのコンポーネント変数 (続き)

変数名	説明	デフォルト値
jetFolder	JET 固有の情報を格納するフォルダ。デフォルトでは JET フォルダの位置は /com/sun/nlosp/autogen-:[jetHost] で、:[jetHost] は前の手順で定義した仮想 JET ホスト名です。	/com/sun/nlosp/autogen-:[jetHost]
templatePrefix	OS プロビジョニング JET テンプレートの特定に使用する値。	OSP_TEMPLATE_

Solaris イメージおよびプロファイルの作成

基本的な JumpStart の機能を提供するには、適切な Solaris OS メディアのイメージを JET サーバーにインストールする必要があります。続いてそのイメージを、イメージがどのようにインストールされるかを説明するプロファイルに添付する必要があります。

▼ Solaris イメージをインポートする

基本的な JumpStart の機能を提供するには、適切な Solaris OS メディアのイメージを JET サーバーにインストールする必要があります。JET サーバーにインポートされる Solaris のバージョン (複数可) は、ターゲットサーバーにより必要とされるバージョンによって決まります。同時に、同じ JET サーバーに、複数のバージョンの Solaris メディアをインストールできます。

注 - メディアを JET サーバー上のディスクにコピーすることをお勧めしますが、必須ではありません。物理メディアを適切に共有して、その物理メディアからターゲットサーバーをブートすることができます。ただし、物理メディアを使用することによりターゲットサーバーのビルドのパフォーマンスが厳しく制限され、複数のバージョンの Solaris をサポートする能力が最低限に制限されます。

- 手順
1. ローカルドライブから、またはネットワーク上の共有ドライブを使用することで、JET サーバーで **Solaris DVD** をマウントします。
 2. N1 SPS ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。

3. 「OS Provisioning Common Tasks」ページの「Solaris Images」セクションで「Import」をクリックします。
4. 「Plans Details」ページの「Run」をクリックします。
5. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「Plan Parameters」テーブルの「SolarisImage」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「Plan Parameters」テーブルの「SolarisImage」コンポーネント行で「Select from List」をクリックします。
 - a. 「Create Set」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. **version** 変数に **Solaris** のメインバージョン番号を入力します。
 - d. **release** 変数に **Solaris** のマイナーバージョン番号を入力します。
 - e. 残りの変数を確認し、必要に応じてそれらを変更します。
次の表で、追加の変数とそのデフォルト値を説明します。

変数名	説明	デフォルト値
architecture	SPARC や x86 など、物理システムのアーキテクチャー	sparc
installPath	インポートされる Solaris イメージの一意の名前。	Solaris:[version]_: [release]_: [architecture] 次に例を示します。 Solaris9_u7_sparc
image_path	インポートされるイメージが格納される場所へのパス。	/export/osp_image/: [installPath]
image_subnet_address	イメージサーバーのサブネットアドレス。	: [target (/): sys.ipAddress]
image_subnet_mask	イメージサーバーのサブネットマスク	255.255.255.0

変数名	説明	デフォルト値
media_src	イメージファイルへのパス。デフォルトでは、ソフトウェアはファイルが DVD ディスク上に存在すると想定しています。	/cdrom/cdrom0/s0

- f. 変数セットを保存します。
 - g. 「**Plan Parameters**」テーブルの「**SolarisImage**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
6. 「**Plan Details Run**」ページで、イメージのインポート先の **JET** イメージサーバーを選択します。

ヒント - JET イメージサーバーのホスト名の末尾は `-jet` です。

7. 「**Run Plan (includes preflight)**」をクリックします。

▼ Solaris プロビジョニングプロファイルを作成する (ブラウザインタフェース)

始める前に JET サーバーが存在し、そのサーバーで Solaris OS ソフトウェアが使用できることを確認します。JET サーバーの作成方法の詳細については、[45 ページの「Solaris JET サーバーの設定」](#)を参照してください。Solaris イメージの作成方法の詳細については、[47 ページの「Solaris イメージをインポートする」](#)を参照してください。

- 手順
1. **N1 SPS** ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Solaris Images**」セクションで「**Create Profile**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**SolarisImage**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**SolarisImaget**」コンポーネント行で「**Select from List**」をクリックします。

- a. 「**Create Set**」をクリックします。
- b. 変数セットの名前を入力します。
- c. **version** 変数に **Solaris** のメインバージョン番号を入力します。
- d. **release** 変数に **Solaris** のマイナーバージョン番号を入力します。
- e. 残りの変数を確認し、必要に応じてそれらを変更します。
次の表で、追加の変数とそのデフォルト値を説明します。

変数名	説明	デフォルト値
architecture	SPARC や x86 など、物理システムのアーキテクチャー	sparc
installPath	添付する Solaris イメージ名	Solaris:[version]_[release]_[architecture] 次に例を示します。 Solaris9_u7_sparc
image_path	イメージファイルが存在する場所へのパス	/export/osp_image/[installPath]
image_subnet_addr	イメージサーバーのサブネットアドレス	:[target (/):sys.ipAddress]
image_subnet_mask	イメージサーバーのサブネットマスク	255.255.255.0
media_src	イメージファイルへのパス。デフォルトでは、ソフトウェアはファイルが DVD ディスク上に存在すると想定しています。	/cdrom/cdrom0/s0

- f. 変数セットを保存します。
 - g. 「**Plan Parameters**」テーブルの「**SolarisImage**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
5. 「**Plan Details Run**」ページで、プロファイルの作成先の **JET** イメージサーバーを選択します。

ヒント - JET イメージサーバーのホスト名の末尾は -jet です。

6. 必要に応じてプロファイル名を変更します。

7. 必要に応じてプロファイルの説明を変更します。

8. 使用する **JET** モジュールのリストを更新します。

JET 製品のモジュールの位置は、JET サーバーのディレクトリ
/opt/SUNWjet/Products です。

デフォルトでは、プラン変数 JET Modules Name(s) に、Solaris プロファイルコンポーネントに含める 2 つの JET モジュールがあります。

- **base_config** – Solaris OS のインストールと構成を行います。base_config 変数の詳細については、143 ページの「基本 OS 構成変数」を参照してください。
- **spsra** – Solaris システム上で N1 SPS リモートエージェント (RA) のインストールと構成を行います。spsra 変数の詳細については、154 ページの「N1 SPS リモートエージェントのコンポーネント変数」を参照してください。

base_config は、基本 Solaris インストールを行うための必須 JET モジュールです。そのほかの /opt/SUNWjet/Products モジュールはすべて省略可能です。モジュール名をこのリストに追加することにより、モジュールのインストール関数と構成変数が、新しく作成された Solaris プロビジョニングプロファイルに追加されます。

省略可能な /opt/SUNWjet/Products モジュールの 1 つに、custom モジュールがあります。custom モジュールは、Solaris のパッケージ、パッチ、およびファイルの任意のリストをインストールし、またスクリプトの任意のコレクションを実行できます。詳細については、57 ページの「custom モジュールの使用法」を参照してください。また、そのほかの JET モジュールを作成することもできます。JET モジュールの作成の詳細については、付録 C を参照してください。

9. 「Run Plan (includes preflight)」をクリックします。

10. (省略可能) デフォルトのプロファイル値を変更します。

- a. 新しい **Provision** コンポーネントが含まれる **Solaris** フォルダにナビゲートします。
たとえば次のようなフォルダです。
/com/sun/nlosp/autogen-biss1-jet/provision/
- b. **Solaris** プロビジョニングコンポーネント名をクリックします。
たとえば Solaris10_0205_sparc.standard などです。
- c. デフォルト値を編集します。
Solaris プロファイルの変数の詳細については、付録 B を参照してください。
 - i. 「Details」ページの最下部の「Edit」ボタンをクリックします。
 - ii. 値を変更します。
 - iii. 「Details」ページの最下部の「Check-in」ボタンをクリックします。

▼ Solaris プロビジョニングプロファイルを作成する (コマンド行インタフェース)

手順 1. **SolarisImage** コンポーネントの変数セットを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/untyped/SolarisImage \  
-name "solaris9" -u admin -p admin -vars "version=9;release=u7;architecture=sparc; \  
image_path=/export/install/Solaris9product;image_subnet_addr=10.42.42.2; \  
image_subnet_mask=255.255.255.0;media_src="
```

2. **JET** モジュール情報を指定するには、次のエントリが含まれるファイル
`/tmp/solaris-image` を作成します。

```
standard  
Standard Solaris  
base_config spsra
```

各行の意味は次のとおりです。

- ファイルの最初の行は、ブラウザインタフェースのプラン変数セクションの「Profile Name」フィールドに対応します。
- ファイルの2番目の行は、ブラウザインタフェースのプラン変数セクションの「Profile Description」フィールドに対応します。
- ファイルの3番目の行は、ブラウザインタフェースのプラン変数セクションの「JET Module Name」フィールドに対応します。

3. プランを実行するには、次の例のようなコマンドを入力します。

```
# cat /tmp/solaris-image|cr_cli -cmd pe.p.run -u admin -p admin  
-PID NM:/com/sun/nlosp/untyped/SolarisImage-create-profile  
-tar H:NM:biss1-jet -comp - -vs solaris9 -pto 30 -nto 10
```

ターゲットホストへの OS のインストール

実際のプロビジョニングコンポーネントには、ユーザーが変更する可能性がある変数の長いリストが含まれています。使用可能なすべての変数の完全なリストは、[付録 B](#)にあります。

▼ ターゲットホストに OS をインストールする (ブラウザインタフェース)

次の作業には、ターゲットホストに OS をインストールする際にユーザーが変更する可能性がある変数のサブセットが含まれています。

- 手順
1. **N1 SPS** ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Solaris Images**」セクションで「**Provision OS**」をクリックします。
 3. **provision** コンポーネントが含まれる **Solaris** フォルダにナビゲートします。たとえば `/com/sun/n1osp/autogen-biss1-jet/provision/` などです。
 4. **Solaris** プロビジョニングコンポーネント名を選択します。たとえば `Solaris_9.standard` などです。
 5. 「**Component Details**」ページで **Provision_start:Install** プロシージャの横にある「**Run action**」をクリックします。
 6. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルのコンポーネント名の行にあるドロップダウンメニューから名前を選択します。
 - 既存の変数セットの値を変更するには、「**Plan Parameters**」テーブルのコンポーネント名の行で「**Select from List**」をクリックします。
 - a. 変更する変数セットの列の「**Edit**」ボタンをクリックします。
 - b. 変数を確認し、必要に応じて変数を変更します。
変数とその説明、およびデフォルト値のリストは付録 B にあります。
 - c. すべての値が正しければ「**Save**」ボタンをクリックします。
 - d. 「**Plan Parameters**」テーブルのコンポーネント名の行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルのコンポーネント名の行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 変数の値を確認または変更します。
多くの変数を変更しなければならない場合もありますが、通常は少数の変数を変更します。「**Component Variables**」リストの変数名とプロンプトを使

用すると、変数の適切な入力形式を指定できます。変数とその説明、およびデフォルト値のリストは付録 B にあります。

- d. すべての値が正しければ「**Save**」ボタンをクリックします。
- e. 「**Plan Parameters**」テーブルのコンポーネント名の行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。

7. **OS** のプロビジョニング先であるターゲットホストを選択します。

ヒント – ターゲットホストのホスト名の末尾は `-target` です。

8. 「**Run Plan (includes preflight)**」をクリックします。
プランが完了し、Solaris イメージがターゲットホストにインストールされた時点で、ターゲットホストにログインする必要があります。root ユーザーのパスワードは newroot です。

▼ ターゲットホストに OS をインストールする (コマンド行インタフェース)

- 手順 ● 次の例のようなコマンドを入力します。ただしコンポーネントの名前とターゲットは、使用するコンポーネント名とターゲットに置き換えます。

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID \  
NM:/com/sun/nlosp/autogen-biss1-jet/provision/SolarisProfile-provision-start-Solaris_9.standard \  
  
-tar H:NM:testv240-target -comp + -vs + -pto 60 -nto 60
```

▼ OS のプロビジョニング状態を確認する

- 手順
1. **N1 SPS** ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**OS Provisioning Administration Tasks**」セクションで「**Status**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. 「**Plan Details Run**」ページで、**OS** のプロビジョニング先であるターゲットホストを選択します。
 5. プロビジョニング状態を確認するターゲットホストを選択します。

6. 「Details」リンクをたどり、状態を表示します。

▼ OS プロビジョニング処理を停止する

- 手順
1. N1 SPS ブラウザインタフェースの「Common Tasks」セクションで、「OS Provisioning」を選択します。
 2. 「OS Provisioning Common Tasks」ページの「OS Provisioning Administration Tasks」セクションで「Abort」をクリックします。
 3. 「Plans Details」ページの「Run」をクリックします。
 4. 「Plan Details Run」ページで、OS のプロビジョニング先であるターゲットホストを選択します。
 5. プロビジョニング処理を停止するターゲットホストを選択します。

JET Solaris サーバーの管理作業

▼ N1 SPS RA インストーラをインポートする

クロスプラットフォームインストール (SPARC システムからの x86 プラットフォームへのインストールなど) を有効にするには、アーキテクチャー固有のインストーラを JET Solaris Image サーバーにインポートする必要があります。

- 手順
1. プロビジョニングソフトウェアのページの「Common Tasks」セクションで、「OS Provisioning」を選択します。
 2. 「OS Provisioning Common Tasks」ページの「JET Solaris Image Servers」セクションで「Manage」をクリックします。
 3. 「Component Details」ページで `import_product_media` プロシージャの横にある「Run action」をクリックします。
 4. インストーラをインポートする JET Solaris Image サーバーホストを選択し、「Run Selected Installations」をクリックします。
 5. 「Plans Run」ページで、プラン変数を次のように設定します。

変数	説明	サンプル値
JET Module Name	JET モジュールの名前	spsra
Product Version	N1 SPS インストーラのバージョン	5.1
Architecture	RA インストーラに適用されるアーキテクチャー	sparc
Media Path	RA インストーラが含まれるディレクトリへのパス	/net/myothersys/export/ra

6. 「**Run Plan (includes preflight)**」をクリックします。

注 - インポートされた RA インストーラからインストールするには、RA をインストールする際に使用される Solaris プロファイルで、「Install RA from snapshot (y,n)」の値が n に設定されている必要があります。154 ページの「N1 SPS リモートエージェントのコンポーネント変数」を参照してください。

▼ ブート params に JET サーバーを使用する

デフォルトでは、OS プロビジョニングサーバーにはブート構成用の DHCP ターゲットホスト情報が付属しています。Solaris ターゲットホストを手動で再起動する場合には、代わりに JET サーバーを RARP ブート params サーバーとして使用できます。

- 手順 1. **Generic Target** ホストを作成する際に、**osp_control_service** 変数を **FALSE** に設定します。

Generic Target ホストの作成方法の詳細については、105 ページの「汎用ターゲットホストの定義」を参照してください。この変数を FALSE に設定することにより、ターゲットホストは、OS プロビジョニングのサービスを受けるターゲットホストアドレスから、JET のサービスを受けるターゲットホストアドレスに変更されます。

2. **RARP** ブート **params** サービス用に **JET** サーバーを構成するには、**/opt/SUNWjet/etc/jumpstart.conf** ファイルの **JS_CLIENT_MANAGEMENT** の値を変更します。

たとえば、SPARC システムでは次のように変更します。

```
JS_CLIENT_MANAGEMENT="bootp"
```

注- ターゲットホストをブートする際、ネットワーク経由でターゲットホストをブートおよびインストールするコンソールブートコマンドの例は `boot net - install` です。

custom モジュールの使用法

Solaris Profile の作成時に、Solaris プロファイルコンポーネントに custom モジュールを含めることができます。Solaris プロファイルコンポーネントを作成する場合、プラン変数には JET Module Name(s) が含まれます。

デフォルトでは、Solaris プロファイルコンポーネントには次の 2 つの JET モジュールが含まれています。

- `base_config` - Solaris OS のインストールと構成を行います。`base_config` 変数の詳細については、143 ページの「基本 OS 構成変数」を参照してください。
- `spsra` - Solaris システム上で N1 SPS リモートエージェント (RA) のインストールと構成を行います。`spsra` 変数の詳細については、154 ページの「N1 SPS リモートエージェントのコンポーネント変数」を参照してください。

3 番目に使用可能なモジュールは `custom` で、Solaris のパッケージ、パッチ、およびファイルの任意のリストをインストールし、またスクリプトの任意のコレクションを実行できます。Solaris プロファイルコンポーネントに `custom` モジュールを含めるには、Solaris プロファイルを作成する前に、`custom` を JET Module Name(s) プラン変数に追加します。

`custom` モジュール内の項目がターゲットサーバーにインストールされる際、インストールの順序は常に、パッケージ、パッチ、ファイル、スクリプトの順になります。この順序が適切ではない場合、独自の JET モジュールを作成することもできます。JET モジュールの作成の詳細については、付録 C を参照してください。

`custom` モジュールは、ビルドのさまざまな段階でのパッケージやパッチのインストールに使用できます。44 ページの「JET を介した Solaris のインストールプロセス」を参照してください。

ターゲットサーバーテンプレートを編集する場合、ブートシーケンスのどの時点でパッケージをインストールする必要があるかに応じて、構成変数 `custom_packages_[1-n]` に追加パッケージの名前をリストすることができます。同様に、変数 `custom_patches_[1-n]` でパッチを特定することもできます。

パッケージとパッチは指定された順序でインストールされるため、それらの間の依存関係を満たすようそれらを適切に順序付ける必要があります。数多くのパッケージやパッチがある場合や、複数の異なるターゲットサーバーで共通のセットが頻繁に使用される場合は、独自のモジュールの作成を検討してください。パッケージのインストールにパッチを組み込む必要がある場合も、このアプローチを検討しなければならない場合があります。

製品インストールにインストールされるパッケージが含まれる場合、`package.matrix` ファイルが含まれます。このファイルには、インストールする必要があるパッケージのリストとともに、サポートされるオペレーティングシステムと製品のバージョン番号のリストが含まれています。また、同様に必要なパッチを定義する `patch.matrix` ファイルも存在します。

カスタムパッチとパッケージメディアの生成

`custom` モジュールを使用すると、ターゲットサーバーごとにカスタムパッケージとパッチセットを定義できます。このモジュールは、ツールキットが検索する適切な場所にパッケージとパッチメディアを配置できるようにする、2つのスクリプトも提供します。

パッチとパッケージメディアをコピーする際、ツールキット構成ファイル (`/opt/SUNWjet/etc/jumpstart.conf`) にある `JS_PKG_MEDIA` と `JS_PATCH_MEDIA` の定義に従って、スクリプトはメディアを配置します。別の場所を使用してすべてのメディアを保持するには、スクリプトを実行する前に構成ファイルを変更します。

カスタムパッチは、`/opt/SUNWjet/bin/copy_custom_patches` スクリプトを使用して配置できます。

```
# copy_custom_patches src-dir patch [patch....]
```

このスクリプトは少なくとも2つの引数を取り、最初の引数はパッチが含まれるソースディレクトリです。そのほかの引数はパッチ番号と見なされ、これらはそのディレクトリ内に位置し、それ以降コピーされます。

カスタムパッケージは、`/opt/SUNWjet/bin/copy_custom_packages` スクリプトを使用して転送できます。

```
# copy_custom_packages src-dir arch package [package....]
```

引数はカスタムスクリプトコマンドに使用される引数によく似ていますが、追加の `arch` 引数が含まれています。この引数はパッケージのターゲットアーキテクチャーの定義に使用されます。ターゲットアーキテクチャーは、ターゲットサーバー上の `uname -p` の出力により定義されます。現在、値は `sparc` または `i386` のいずれかで、それぞれ `SPARC` アーキテクチャーと `IA86` アーキテクチャーを表します。

カスタムパッチにはアーキテクチャーの定義は必要ありません。これは、パッケージが使用可能な各アーキテクチャーに対して、独立した番号が付けられたパッチがリリースされるためです。

`custom` モジュールは、同じアーキテクチャーの同じパッケージの複数の異なるバージョンを含むことはできません。この機能が必要である場合は、これらの要件に対応する特定のモジュールを検討してください。付録 C を参照してください。

カスタムファイル

custom モジュールを使用すると、ターゲットサーバーでファイルを作成、上書きしたり、ファイルを追加先にすることができます。ファイルは、次の3つの項目を定義する値により参照されます。

- JumpStart サーバー上の /opt/SUNWjet/Clients 内のターゲットサーバーディレクトリから相対的なソースファイル
- 「a」(追加) または 「o」(上書き) のいずれかである処理のモード
- ターゲットサーバー上の宛先ファイル

たとえば、banana と呼ばれるターゲットサーバーの場合、banana の構築時に、テンプレートのカスタム領域にある次の行は、JumpStart サーバー上のファイル /opt/SUNWjet/Clients/banana/hosts の内容を、マシン banana 上のファイル /etc/hosts に追加します。

```
custom_files="hosts:a:/etc/hosts"
```

ソースファイルは、ターゲットサーバーの構築前に適切に配置する必要があります。ファイルは /opt/SUNWjet ディレクトリツリー内に配置する必要があります。

注 - ソースファイルには絶対パスを使用しないでください。

また、/etc/hosts や /etc/passwd など、ツリーの外部にあるファイルを参照しないでください。

3つのフィールドの中央は、ターゲットサーバー上の宛先ファイルを追加または上書きするかどうかを指定します。多数のターゲットサーバーに対してファイルのセットが共通である場合は、/opt/SUNWjet/Clients ディレクトリ内での保持領域の作成を検討してください。共通ファイルはそのディレクトリ内に配置します。ターゲットサーバーのテンプレートは、ファイルを次のように参照できます。

```
custom_files="../../../common/hosts:a:/etc/hosts ../common/ftpusers:o:/etc/ftpusers"
```

ここでファイルは、各ターゲットサーバー固有のディレクトリの複数コピーではなく、/opt/SUNWjet/Clients/common に配置されます。テンプレートファイルは Bourne シェルスクリプトであるため、通常の手法を使用して行を継続することができます。文字を使用するか、変数 custom_files="{custom_files}" に情報を追加します。

カスタムスクリプト

カスタムスクリプトは、スクリプトのソース位置により定義されます。ツールキットは、ビルド中にターゲットサーバーのスクラッチ領域にスクリプトをコピーし、適切な時点でスクリプトを実行します。

前節で説明したカスタムファイルと同じように、スクリプトのソースは /opt/SUNwjet ディレクトリ内に存在する必要があります。ソースは、共通の保持領域に対する相対参照になることも可能です。

システムが JumpStart サーバーの NFS イメージ上で動作し、実際のターゲットサーバーファイルシステムが \$ROOTDIR (/a) 上にマウントされている場合、カスタムモジュールは最初の再起動の前に、スクリプトを実行するためのプロビジョニングを行いません。

最初の再起動の前にスクリプトを実行するには、付録 C で説明されているモジュールの作成を検討してください。

第 6 章

Linux オペレーティング環境のプロビジョニング

この章では、OS プロビジョニングプラグインを使用して、Linux オペレーティングシステムをターゲットホストにインストールする方法について説明します。

この章では、次の内容について説明します。

- 61 ページの「Kickstart テクノロジー」
- 61 ページの「Linux プロビジョニングプロセスの概要」
- 62 ページの「Linux サーバーの設定」
- 67 ページの「Linux イメージおよびプロファイルの作成」
- 68 ページの「ターゲットホストへの OS のインストール」

Kickstart テクノロジー

OS プロビジョニングプラグインは Linux の kickstart テクノロジーを使用します。

Linux プロビジョニングプロセスの概要

1. Linux イメージサーバーを設定します。
2. Linux イメージを作成するか、イメージサーバーに既存のイメージを添付します。
3. ターゲットホストを作成します。
4. ターゲットホストに OS をインストールします。

Linux サーバーの設定

Linux ブートおよびインストールサーバーには次のソフトウェアが含まれています。

- Linux (RedHat Advanced Server 3.0 でテスト済み)
- TFTP サーバー
- ネットワークファイルシステム (NFS) サーバー
- N1 SPS リモートエージェント (RA) (Linux 版)。詳細については、『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」を参照してください。

▼ Linux ファイルをコピーする

手順 1. **Linux** ファイルを含めるディレクトリを作成します。

注 - 同じサーバーから複数のディストリビューションにサービスを提供するため、各ディストリビューション用に独立したディレクトリを作成します。

2. 最初の **CD-ROM** または **ISO** イメージファイルをマウントします。

次のいずれかの例のようなコマンドを使用します。

```
#mount /dev/cdrom                デフォルトでは /mnt/cdrom にマウント、/etc/fstab  
                                を確認  
#mount /path-to-first-iso /mnt/loop  /mnt/loop が存在すると想定
```

3. **CD-ROM** または **ISO** イメージファイルの内容を、手順 1 で作成したディレクトリにコピーします。

次のいずれかの例のようなコマンドを使用します。

```
#/bin/cp -avf /mnt/cdrom/* /export/RHEL3.0  
#/bin/cp -avf /mnt/loop/* /export/RHEL3.0
```

4. ディストリビューションのすべての **CD-ROM** または **ISO** メディアに対して手順 2 と 3 を繰り返します。

▼ NFS アクセス権を設定する

ディストリビューションは、このブートおよびインストールサーバーからサービスを受ける各サブネットに関して、TFTP を介してアクセス可能である必要があります。

手順 1. `/etc/exports` ファイルを編集し、次の例のような行を追加します。

```
/export 10.42.42.*(ro)
```

2. NFS を有効にし、起動します。

RedHat AS 3.0 に適用される、次のようなコマンドを使用します。

```
#chkconfig nfs on
#service nfs restart
```

▼ TFTP アクセス権を設定する

ディストリビューションは、このブートおよびインストールサーバーからサービスを受ける各サブネットに関して、TFTP を介してアクセス可能である必要があります。

手順 1. `/etc/xinet.d/tftp` ファイルで `server_args` パラメータを探します。

これが、TFTP サーバーにより使用されるルートディレクトリを定義します。ディレクトリが存在しない場合は、ディレクトリを作成します。

注 - この作業では例として `/tftpboot` を使用します。

2. TFTP を有効にし、`xinetd` デーモンを再起動します。

RedHat AS 3.0 で動作する、次の例のようなコマンドを使用します。

```
#chkconfig tftp on
#service xinetd restart
```

3. TFTP ルートディレクトリの下に、次のディレクトリをコピーまたは作成します。

- `/tftpboot/pxelinux.0` - Linux PXE ブートストラップファイルは <http://syslinux.zytor.com/pxe.php> からダウンロードできます。
- `/tftpboot/pxelinux.cfg/` - このディレクトリには PXE 構成ファイルが含まれています。これらのファイルには、各ハードウェアプラットフォームおよび Linux ディストリビューションに固有のパラメータが含まれています。
- `/tftpboot/pxelinux.cfg/v20z-config` - V20z プラットフォームに固有のディレクトリ。
- `/tftpboot/vmlinuz` - ディストリビューションの `/images/pxeboot` にある、インストールカーネル。
- `/tftpboot/initrd.img` - ディストリビューションの `/images/pxeboot` にある、インストール RAM ディスク。

例 6-1 PXE 構成ファイルのサンプル

次の例に、SunFire V20z システムの PXE 構成ファイルを示します。

```
default e130 ks
serial 0 9600
label e130
kernel vmlinuz
append linksleep=30 ksdevice=eth0 console=tty0 console=ttyS1,9600 load_ramdisk=1
initrd=initrd.img network
```

各行の意味は次のとおりです。

- 最初の行 (default) はラベル e130 を使用し、Kickstart インストール (カーネルに ks を渡す) を行うことを指定します。
- 第3行 (label) は、ラベル e130 を定義します。
- 第4行 (kernel) は、読み込むカーネルを特定します。パスは pxelinux.0 に対して相対的です。
- 第5行 (append) には、一連のネットワーク構成パラメータが含まれています。append パラメータはすべて同じ行に存在する必要があります。ここに示す固有のパラメータは次のとおりです。

linksleep=30 は、システムが動作していないと判断する前にカーネルに待機するよう通知します。ネットワークインタフェースの中には、アップ状態になるまで長い時間を要するものもあります。

ksdevice=eth0 は、eth0 インタフェースを介してインストールを実行するよう指定します。このパラメータが省略されている場合、eth0 (の値) が使用されません。

console=tty0 console=ttyS1,9600 は、コンソールにインストール出力を表示するようコンソールパラメータを指定します。パラメータは、ターゲットハードウェアのプラットフォームに応じて異なります。

load_ramdisk は、RAM ディスクに initrd.img を使用することを示しています。RAM ディスクのパスは pxelinux.0 に対して相対的です。

initrd はネットワークインストールを実行することを示しています。

注 - vmlinuz および initrd ファイルは、特定のディストリビューションの特定のバージョンのみで動作します。サーバーを使用して複数のディストリビューションまたはディストリビューションの複数のバージョンを提供する場合、各 vmlinuz と initrd のペアは一意に識別可能である必要があります。次の例に示すように、別のディレクトリを使用するかファイル名を変更することで、一意の識別子を作成できます。

```
/tftpboot/pxelinux.0  
/tftpboot/RHEL2.1/vmlinuz  
/tftpboot/RHEL2.1/initrd  
/tftpboot/RHEL3.0/vmlinuz  
/tftpboot/RHEL3.0/initrd
```

または

```
/tftpboot/pxelinux.0  
/tftpboot/rhel2.1-vmlinuz  
/tftpboot/rhel2.1-initrd  
/tftpboot/rhel3.0-vmlinuz  
/tftpboot/rhel3.0-initrd
```

▼ Linux イメージサーバーを作成する

N1 SPS のブラウザインタフェースまたはコマンド行インタフェースから Linux イメージサーバーを作成できます。コマンド行からサーバーを作成するには、次のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/untyped/LinuxServer \  
-name "linuxserver" -u admin -p admin -vars "boot_server_tftp_root_directory=/tftpboot"  
# cr_cli -cmd pe.p.run -u admin -p admin \  
-PID NM:/com/sun/nlosp/untyped/LinuxServer-create \  
-tar H:NM:bikickstart -comp + -vs linuxserver -pto 30 -nto 10
```

ブラウザインタフェースからサーバーを作成するには、次の手順に従います。

始める前に kickstart ファイルは、/export/kickstart など、ターゲットホストが位置を特定できるように、NFS エクスポートディレクトリの Linux ブートおよびインストールサーバーに配置する必要があります。

- 手順
1. プロビジョニングソフトウェアのページの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Linux Image Servers**」セクションで「**Create**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. このプランに使用する変数を選択します。

- 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**LinuxServer**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
- 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**LinuxServer**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 必要に応じて、`boot_server_tftp_root_directory` 変数を、**Linux** パッケージをインストールする位置に変更します。
デフォルトでは、パスは `/tftpboot` です。
 - d. 必要に応じて、`linuxHost` 変数を、コマンド行インタフェースの位置に変更します。
デフォルトでは、仮想 Linux ホスト名は、`-linux` が追加されたターゲットホストのホスト名に設定されています。



注意 - `installPath` 変数は変更しないでください。

- e. 変数セットを保存します。
 - f. 「**Plan Parameters**」テーブルの「**LinuxServer**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
5. **Linux** イメージサーバーを作成するホストを選択します。
 6. 「**Run Plan (includes preflight)**」をクリックします。
 7. イメージサーバーが正常に作成されたことを確認するには、プロビジョニングサーバーのウィンドウの左側にある「**Hosts**」リンクをクリックします。
`-linux` が付加された仮想ホストの名前が表示されます。たとえば、ホスト `bikickstart` にインストールした場合、仮想ホストは `bikickstart-linux` です。

Linux イメージおよびプロファイルの作成

▼ Linux プロビジョニングプロファイルを作成する (ブラウザインタフェース)

始める前に Linux イメージサーバーが存在し、そのサーバーで Linux オペレーティングシステムソフトウェアが使用できることを確認します。Linux イメージサーバーの作成方法の詳細については、62 ページの「Linux サーバーの設定」を参照してください。

- 手順
1. プロビジョニングソフトウェアのページの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Linux Images**」セクションで「**Create Profile**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**LinuxImage**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**LinuxImage**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 変数を確認し、必要に応じて変数を変更します。
 - d. 変数セットを保存します。
 - e. 「**Plan Parameters**」テーブルの「**LinuxImage**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
 5. 「**Plan Details Run**」ページで、プロファイルの作成先の **Linux** イメージサーバーを選択します。

ヒント - Linux イメージサーバーのホスト名は末尾が `-linux` になります。

6. 「**Run Plan (includes preflight)**」をクリックします。
7. **Linux** イメージサーバーに対して **Linux** プロファイルが作成されたことを確認するには、**N1 SPS** ブラウザインタフェースの左側にある「**Components**」をクリックします。

フォルダ `/com/sun/nlosp/autogen-server-name-linux/provision/` には `provision` コンポーネントがあるはずですが、ここで `server-name` は、変数セットでユーザーが指定した Linux サーバー名です。次に例を示します。
`/com/sun/nlosp/autogen-LinuxRedHat3.0_x86-linux/provision/`

▼ Linux プロビジョニングプロファイルを作成する (コマンド行インタフェース)

- 手順 1. **LinuxImage** コンポーネントの変数セットを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/untyped/LinuxImage \  
-name "redhat3" -u admin -p admin -vars "osp_server=masterserver-osp;version=RedHat3.0; \  
architecture=x86;installPath=Linux:[version]_[architecture]; \  
description=testlinux;min_disk_size=10GB;monitor_install=FALSE; \  
boot_server_name=bikickstart;boot_server_subnet_address=10.42.42.0; \  
boot_server_subnet_ip_address=10.42.42.3;kickstart_file=/export/el30.cfg; \  
pxe_boot_file=linux-install/pxelinux.0;pxe_configuration_file=el30-console"
```

2. プランを実行するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd pe.p.run -u admin -p admin \  
-PID NM:/com/sun/nlosp/untyped/LinuxImage-create-profile \  
-tar H:NM:bikickstart-linux -comp + -vs redhat3 -pto 30 -nto 10
```

ターゲットホストへの OS のインストール

プロファイルは、ターゲットホストに OS イメージをプロビジョニングするために使用されます。これらのプロファイルは、`/com/sun/nlosp/autogen...provision` フォルダにあります。新しい OS イメージが作成 (または添付) されるたびに、新しいプロファイルがプロビジョニングフォルダに追加されます。

▼ ターゲットホストへ OS をインストールする

- 手順
1. プロビジョニングソフトウェアのページの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Linux Images**」セクションで「**Provision OS**」をクリックします。
 3. **provision** コンポーネントが含まれる **Linux** フォルダにナビゲートします。
たとえば次のようなフォルダです。
`/com/sun/nlosp/autogen-bikickstart-linux/provision/LinuxRedHat3.0_x86`
 4. コンポーネント名を選択します。
 5. 「**Component Details**」ページで **Provision_start:Install** プロシージャの横にある「**Run action**」をクリックします。
 6. オペレーティングシステムのプロビジョニング先であるターゲットホストを選択します。

ヒント - ターゲットホストのホスト名の末尾は `-target` です。

7. 「**Run Plan (includes preflight)**」をクリックします。

▼ OS のプロビジョニング状態を確認する

- 手順
1. プロビジョニングソフトウェアのページの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**OS Provisioning Administration Tasks**」セクションで「**Status**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. 「**Plan Details Run**」ページで、オペレーティングシステムのプロビジョニング先であるターゲットホストを選択します。
 5. 「**Run Plan (includes preflight)**」をクリックします。
 6. 「**Details**」リンクをたどり、状態を表示します。

第 7 章

Windows オペレーティング環境のプロビジョニング

この章では、OS プロビジョニングプラグインを使用して、Windows オペレーティングシステムをターゲットホストにインストールする方法について説明します。

この章では、次の内容について説明します。

- 71 ページの「Remote Installation Service (RIS) テクノロジー」
- 72 ページの「Windows プロビジョニングプロセスの概要」
- 72 ページの「Windows サーバーの設定」
- 84 ページの「Windows イメージおよびプロファイルの作成」
- 87 ページの「ターゲットホストへの OS のインストール」

注 - Windows 2000 オペレーティングシステムのプロビジョニングは可能ですが、Windows ブートおよびインストールサーバーは Windows 2003 オペレーティングシステムを実行している必要があります。

Remote Installation Service (RIS) テクノロジー

OS プロビジョニングプラグインは RIS テクノロジーを使用して Windows システムをプロビジョニングします。Remote Installation Service は、RIS サーバー上で実行される独立した設定ルーチンを提供します。RIS は、ネットワークブートを介してアクセス可能なメンテナンスおよび障害追跡ツールを統合する、集中管理された位置を提供します。RIS は、ネットワーク管理者がベースオペレーティングシステムを簡単にインストールしたり、障害が発生したシステムを交換できるようにしたりします。RIS の詳細については、Microsoft のドキュメントを参照してください。

Windows プロビジョニングプロセスの概要

1. Windows イメージサーバーを設定します。
2. Windows イメージを作成するか、イメージサーバーに既存のイメージを添付します。
3. ターゲットホストを作成します。
4. ターゲットホストにイメージをインストールします。

Windows サーバーの設定

Windows ブートおよびインストールサーバーの設定には、次の複数の作業があります。

1. Windows RIS サーバーでの Windows Active Directory および RIS サービスの設定。
2. Windows RIS サーバーでの OS プロビジョニングスクリプトのインストール。
3. Windows RIS サーバーでの Windows Remote Shell Service (RSH) または Secure Shell Service (SSH) のインストール。
4. Windows RIS サーバーでの Windows RIS イメージの設定。
5. 上記のイメージの設定情報 (.sif) ファイルの構成。
6. OS プロビジョニングイメージを配備するための OS チューザファイルの構成。

▼ Windows RIS サーバーを設定する

- 手順
1. **Windows** ブートおよびインストールサーバーに **32 ビットシステム版の Windows 2003 Server Edition** オペレーティングシステムをインストールします。
RIS イメージを保持するため、約 700M バイトのイメージを少なくとも 1 つ格納するのに十分な大きさの、独立した NTFS パーティションを作成します。RIS サーバーにサーバーを作成する際には、このパーティションを特定する必要があります。

注 - Windows ブートおよびインストールサーバーでは、64 ビットシステム版の Windows 2003 Server Edition オペレーティングシステムはサポートされていません。

2. このマシン上で **Active Directory Server** をインストールし、構成します。
このシステムをドメインコントローラに特定します。Active Directory Server の詳細については、Microsoft サポート技術情報 324753 (英語版) を参照してください。
3. このマシンに **RIS** サーバーをインストールし、構成します。
詳細については、Microsoft サポート技術情報 325862 (英語版) を参照してください。
4. 適切な **RIS** サーバーの設定を指定します。
RIS サーバーの RemoteInstall プロパティで、次の 2 つの変数を設定します。
 - Respond to client computers requesting service
 - Do not respond to unknown client computers
5. **DHCP snap-in:- "Manage Authorized Servers"** リストで、**RIS** サーバーの IP アドレス (プロビジョニングサブネットインタフェースの IP アドレス) を追加します。
6. **RIS** サーバーですでに **DHCP** サービスが実行中である場合、すべての種類の **DHCP** サービスを無効にします。
7. **RIS** サーバーのユーザーアカウントを構成します。
RIS サーバーの Active Directory で次の 2 つのユーザーアカウントを作成し、それらを Administrators、Domain Admins、および Domain Users グループに追加します。
 - OS プロビジョニング管理ユーザー - たとえば username n1ospadmin のようにします。
 - OS プロビジョニング RSH/SSH ユーザー - OS プロビジョニングサーバー上の DHCP サービスは root として実行されるため、RSH ユーザー名は常に root である必要があります。SSH ユーザー名には任意の有効なユーザー名を使用できます。
8. **Active Directory Users** および **Computers** スナップインで、ドメインコントローラの「**Delegate control**」ウィザードを実行します。
 - a. 「**Join a Computer to the Domain**」を選択します。
 - b. 前の手順で OS プロビジョニング管理ユーザーに対して定義したユーザー名に、ユーザーを設定します。

9. **Windows** ブートおよびインストールサーバーに **Windows 2000/2003 Resource Kit** ソフトウェアをインストールします。

このソフトウェアは、Windows Resource Kit CD またはオンラインからインストールできます。このパッケージには、OS プロビジョニングスクリプトにより使用される、setupmgr、deploy.cab、xcacIs などのツールが含まれます。

▼ Windows RIS サーバーで OS プロビジョニングスクリプトをインストールする

- 手順 1. 製品メディアからユーザー指定ディレクトリに、次の **Windows OS** プロビジョニングスクリプト、サンプルファイル、およびツールをコピーします。

- /dvd/release/n1osp/1.0/Windows/SampleOSCFiles
- /dvd/release/n1osp/1.0/Windows/SampleSIFFiles
- /dvd/release/n1osp/1.0/Windows/IspScripts
- /dvd/release/n1osp/1.0/Windows/IspTools

ここで、/dvd/release/n1osp/1.0/Windows/ は、製品メディアからこれらのファイルが含まれる Windows ディレクトリへのパスを示します。

2. プロビジョニング時に生成されるカスタマイズされた **SIF** ファイルを格納するフォルダを作成します。
このフォルダを RemInst シェアの下 (たとえば D:\RemoteInstall\folder-name) に配置します。n1osp のように、関連する名前をフォルダに付けます。
3. **OS** プロビジョニング管理ユーザーと **RSH/SSH** ユーザー用に、前の手順で作成したフォルダへの読み取り権と書き込み権を付与します。
4. **OS** プロビジョニング管理ユーザーと **RSH/SSH** ユーザーが、**RemInst** シェアの一時フォルダに対して読み取り権、書き込み権、および実行権を持っていることを確認します。

たとえば D:\RemoteInstall\tmp が一時フォルダです。直前のプロビジョニング処理の結果として BINLSVC により作成された古い一時 SIF ファイルをスクリプトが削除できるように、このアクセス権が必要になります。

アクセス許可を確認または変更するには、フォルダを右クリックし、「プロパティ」、「セキュリティ」、「詳細」タブボタンの順に選択します。設定を継承するよう、すべてのファイルとフォルダのチェックボックスを選択します。

▼ Windows RIS サーバーに Windows RSH サーバーをインストールする

OS プロビジョニングプラグインは、Windows 2003 の WinRsh (Native Windows RSH Service) をサポートしています。

- 手順 1. **Windows** ブートおよびインストールサーバーに **Windows Services for Unix 3.5 (SFU3.5)** ソフトウェアをインストールします。
- このソフトウェアは、Windows Resource Kit CD またはオンラインからインストールできます。このパッケージには Windows Remote Shell Service (RSH) サービスが含まれています。SFU のインストール手順については、Microsoft サポート技術情報 321712 を参照してください。
2. **N1 RSH** ユーザー (つまり **root** ユーザー) として **Windows RSH (WinRsh** サービス) を **RIS** サーバーにインストールし、構成します。
- 詳細は、<http://www.microsoft.com/technet/interopmigration/unix/sfu/sfu35rsh.msp> を参照してください。

注 - ファイルのアクセス権が正しくないと、WinRsh に障害が発生する原因になります。この問題の症状としては、クライアントへ「Access denied」メッセージやイベントビューアのログエントリがあります。この問題を解決するには、.rhosts ACL リストで、ユーザーグループ Administrators/SYSTEM に完全なアクセス権を割り当てます。

次の例に、*system-drive/windows/system32/drivers/etc/.rhosts* ファイルの内容を示します。

```
-----  
10.5.133.22 root  
-----
```

▼ Windows RIS サーバーに Windows SSH サーバーをインストールする

OS プロビジョニングプラグインは、Windows 2003 用の Cygwin の OpenSSH パッケージをサポートしています。

- 手順 ● **N1 SSH** ユーザーとして、**RIS** サーバーに **Windows 2003** 用の **Cygwin** の **Open Secure Shell Service (OpenSsh** サービス) をインストールし、構成します。
- 詳細については、Cygwin のホームページおよび Open Ssh の Windows インストール手順を参照してください。

▼ Windows RIS サーバーで Windows イメージを設定する

- 手順 1. **CD** またはネットワーク共有から **RIS** サーバー **CIFS** に、**Windows** ディストリビューションをコピーします。
- RiSetup.exe ユーティリティーを使用します。

注 - RIS イメージの作成には MSDN Combo CD を使用しないでください。
Microsoft サポート技術情報 300556 (英語版) で説明されているエラーメッセージが表示されます。この問題を解決するには、作成される OS イメージのリテールバージョンを使用してください。

2. 前の手順で作成された **RIS** イメージに必要なドライバを追加します。
ネットワークアダプタ、SCSI ハードドライブ、ビデオ、モデム用のドライバなど、必ずすべてのドライバを組み込んでください。

RIS イメージにドライバを追加する方法の詳細については、Microsoft サポート技術情報 315279 (英語版)、Microsoft サポート技術情報 246184 (英語版)、Microsoft サポート技術情報 254078 (英語版)、および *information about adding mass storage devices to RIS* を参照してください。

注 - 今までの手順に応じてディストリビューションフォルダを作成します。

詳細については、SampleSIFFiles フォルダのサンプル設定ファイルを参照してください。

3. イメージディストリビューションフォルダの下に、
product-directory/\$OEM\$/\$1/N1ISP/ フォルダを作成します。
product-directory は Windows イメージディストリビューションフォルダで、その下にはカーネルファイルを含むサブフォルダとして「i386」があります。

次に例を示します。

```
D:\RemoteInstall\Setup\English\Images\WinAS_2000\%OEM%\$1\N1ISP
D:\RemoteInstall\Setup\English\Images\WinAS_2000\i386
```

4. **Windows** ブートおよびインストールサーバーの **IspTools** フォルダの **assignIPInfo.exe** ファイルを、前の手順で作成した *product-directory/\$OEM\$/\$1/N1ISP/* フォルダにコピーします。

注 - スクリプトでは、N1ISP フォルダがイメージの一部であることが想定されています。プロビジョニングに必要なツールを、このディレクトリパス *product-directory/\$OEM\$/\$1/N1ISP/* にコピーします。N1ISP フォルダ名は変更しないでください。

ヒント - 74 ページの「Windows RIS サーバーで OS プロビジョニングスクリプトをインストールする」の手順 1 では Windows ブートおよびインストールサーバーの IspTools フォルダを作成しましたが、

ターゲットマシンに RA をインストールする場合は、上記のフォルダに N1 SPS RA バイナリをコピーすることもできます。

次の例に、サンプルディレクトリ構造を示します。

```
D:\RemoteInstall\Setup\English\Images\WIN2003_TRIAL\%OEM%\$1\N1ISP>dir
Volume in drive D is Local Disk
Volume Serial Number is A8D9-4012

Directory of D:\RemoteInstall\Setup\English\Images\WIN2003_TRIAL\%OEM%\$1\N1ISP

02/04/2005  02:39 PM    <DIR>          .
02/04/2005  02:39 PM    <DIR>          ..
02/04/2005  11:34 AM                29,184 assignIPInfo.exe
01/31/2005  03:33 PM           27,595,776 cr_ra_win32_5.0.msi
                2 File(s)      27,624,960 bytes
                2 Dir(s)    23,775,776,768 bytes free
```

▼ Windows イメージの設定情報を構成する

Windows の設定情報は、RIS 構成プロファイルに格納されます。詳細については、SampleSIFFiles フォルダのサンプル設定ファイルを参照してください。

- 手順 1. 各ディストリビューションの構成プロファイルファイルを編集し、以降の手順で説明するパラメータを設定します。

このファイルを編集するには、SetupMgr.exe ユーティリティを使用します。このユーティリティは Windows Resource Kit CD (deploy.cab ファイル) にあります。また、notepad ツールを使用して手作業でこのファイルを編集することもできます。

Windows 2003 イメージの場合、デフォルトの SIF ファイルは RIS サーバーの次の位置にあります。

```
//Reminst/Setup/English/Images/Windows2003/i386/templates/ristndrd.sif
```

2. **ristndrd.sif** 構成ファイルの **[data]** セクションに対して、次の構成パラメータの確認と追加を行います。

```
AutoPartition=1
UnattendedInstall="Yes"
```

3. **ristndrd.sif** 構成ファイルの **[Unattended]** セクションに対して、次の構成パラメータの確認と追加を行います。

```
UnattendMode=FullUnattended
NtUpgrade=No
OverwriteOemFilesOnUpgrade=No
OemPreinstall = Yes
```

4. **ristndrd.sif** 構成ファイルの **[UserData]** セクションに対して、次の構成パラメータの確認と追加を行います。

```
FullName = "Nlisp SunUser"
OrgName = "Sun Microsystems"
ProductKey=XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

注 - FullName および OrgName の値は、使用する環境に適した値に置き換えます。

Windows 2000 Advanced Server など、一部のオペレーティングシステムの配備の際には、ProductKey は機能しません。その代わりに、ProductID 変数を使用します。

5. **ristndrd.sif** 構成ファイルの **[GuiUnattended]** セクションに対して、次の構成パラメータの確認と追加を行います。

```
AdminPassword="sun123"
AutoLogon=Yes
AutoLogonCount=1
EncryptedAdminPassword=NO
```

注 - AutoLogonCount=1 は、インストール後に初めてデフォルトパスワード sun123 を使用して、スーパーユーザー管理者が自動的にターゲットホストにログオンすることを意味します。これは、SIF ファイルの **[GuiRunOnce]** セクションで指定されている provision time コマンドを実行するために必要です。Administrator ユーザーが再度ログインしようとする、パスワード変更するよう求めるプロンプトが表示されます。

6. **ristndrd.sif** 構成ファイルに対して、次のセクションの確認または追加を行います。

```
[Branding]
    BrandIEUsingUnattended=Yes
[Proxy]
    Proxy_Enable=0
    Use_Same_Proxy=0
```

ヒント – 通常は SetupMgr.exe ユーティリティがユーザーのためにこれらのセクションを追加します。

7. イメージに、特定のすべてのプラグアンドプレイドライバを追加します。
Microsoft サポート技術情報 315279 の指示を参照してください。

▼ イメージ配備用の OS チューザファイルを構成する

次の手順では、英語版ユーザーの OS チューザファイルを構成する方法を説明します。この手順は、そのほかの言語では少し異なる場合があります。詳細については、Microsoft の文書を参照してください。

- 手順
1. RIS サーバー上の `//RemoteInstall/OSChooser/i386` ディレクトリの次のファイルをコピーします。
 - a. `startrom.com` ファイルのバックアップコピーを `startrom.bak` に作成します。
 - b. `startrom.n12` ファイルを `startrom.com` にコピーします。
 2. CIW 画面を編集してユーザー操作を単純化します。

次のサブ手順では、次の機能を単純化する変更を行う方法を説明します。

 - すべての必須 `.osc` 画面用の `Autoenter` 機能を追加します。この機能により、インストール段階での CIW 画面とのすべてのユーザー操作が不要になります。
 - `login.osc` ファイルに認証用のユーザー情報を追加します。この手順により、インストール時のユーザー操作が不要になります。`login.osc` ファイルでユーザー情報を自動化することにより、ターゲットホストが認証され、ターゲットホストは RIS サーバーから直接ファイルをダウンロードし、インストールできるようになります。
 - `oschoice.osc` ファイルで、OS プロビジョニングのカスタマイズされたパスをイメージに追加します。この手順は、`oschoice` CIW 画面を、`.sif` ファイルの既知の RIS イメージの位置にポイントします。この値は、プロビジョニング実行時に生成される OS プロビジョニングの一時 `.sif` ファイルへのカスタマイズされたパスを指します。
 - a. 次のファイルのバックアップコピーを作成します。
 - `//RemoteInstall/OSChooser/Welcome.osc`
 - `//RemoteInstall/OSChooser/English/install.osc`
 - `//RemoteInstall/OSChooser/English/login.osc`
 - `//RemoteInstall/OSChooser/English/oschoice.osc`
 - `//RemoteInstall/OSChooser/English/warning.osc`

▼ ターゲットホストで BIOS ブート順序を設定する

手順 1. 次の順序で、ターゲットホスト上の **BIOS** 設定を設定します。

- a. ネットワーク (**PXE** 対応) からのブート
- b. ハードディスクからのブート
- c. **CD** からのブート
- d. フロッピーディスクからのブート

2. **PXE** ネットワークブート用のターゲットホストのプライマリ **NIC MAC** アドレスと、**GUID** 値を特定します。

GUID は、ターゲットホストの **BIOS** 設定、または次のそのほかのソースのいずれかから取得できます。

- コンピュータケース側のラベル。
- コンピュータケース内のラベル。
- クライアントコンピュータの **BIOS**。
- クライアントコンピュータからのネットワークトラフィック。ネットワークユーティリティを使用すると、ネットワークトラフィックをスニッフし、**DHCPDiscover** パケットを特定できます。そのフィールドには 128 ビット、つまり 16 バイトの **GUID/UUID** が含まれています。

注 - **GUID** は、{*ddddddd-dddd-dddd-dddd-aaaaaaaa*} の形式である必要があります。ここで *d* は 16 進数文字です。たとえば

{921FB974-ED42-11BE-BACD-00AA0057B223} のようになります。クライアント **GUID** の有効なエントリは、0 から 9 の数字、小文字のアルファベット文字、大文字のアルファベット文字、およびダッシュに制限されています。スペース、アンダースコア、およびそのほかの特殊記号は **GUID** では使用できません。

▼ Windows イメージサーバーを作成する (ブラウザインタフェース)

- 手順
1. プロビジョニングソフトウェアのページの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Windows Image Servers**」セクションで「**Create**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。

4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**WindowsServer**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**WindowsServer**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 変数を確認し、必要に応じて変数を変更します。
イメージサーバーコンポーネントの変数はごく少数で、それらのほとんどは変更しません。これらの変数とデフォルト値のリストは、[83 ページ](#)の「**Windows イメージサーバーのコンポーネント変数**」を参照してください。
 - d. 変数セットを保存します。
 - e. 「**Plan Parameters**」テーブルの「**WindowsServer**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
5. **Windows** イメージサーバーを作成するホストを選択します。
6. 「**Run Plan (includes preflight)**」をクリックします。
7. イメージサーバーが正常に作成されたことを確認するには、プロビジョニングサーバーのウィンドウの左側にある「**Hosts**」リンクをクリックします。
-windows が付加された仮想ホストの名前が表示されます。たとえば、ホスト masterserver にインストールした場合、仮想ホストは masterserver-windows となります。

▼ Windows イメージサーバーを作成する (コマンド行インタフェース)

コマンド行から Windows イメージサーバーを作成するには、次の手順に従います。

- 手順 1. 変数セットを作成します。
次の例のようなコマンドを使用します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/untyped/WindowsServer -name "winRisServer" \
-u admin -p admin -vars "active_directory_userid=nlispadmin; \
active_directory_password=Clz6pK2b6qw=;boot_server_access_protocol=rsh; \
boot_server_access_userid=root;boot_server_access_password=; \
```

```
ciw_directory=D:\\\\RemoteInstall\\\\OSChooser\\\\English; \
temporary_sif_directory=D:\\\\RemoteInstall\\\\Nlisp;installPath=C:\\\\NlispRisProj"
```

2. プランを実行します。

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/nlosp/untyped/WindowsServer-create \
-tar H:NM:masterserver -comp + -vs winRisServer -pto 30 -nto 10
```

Windows イメージサーバーのコンポーネント変数

表 7-1 Windows イメージサーバーコンポーネントの変数

変数名	説明	デフォルト値
installPath	OS プロビジョニングスクリプトの位置	C:\nlisp\scripts
active_directory_userid	Active Directory のユーザー ID	nlrisroot
active_directory_password	上記でユーザー定義された、暗号化されたパスワード。パスワードの暗号化の詳細については、 107 ページ の「パスワードの暗号化」を参照してください。	
boot_server_access_protocol	ブートサーバーにアクセスするプロトコル	rsh
boot_server_access_userid	ブートサーバーのユーザー ID	root
boot_server_access_password	上記でユーザー定義された、暗号化されたパスワード。パスワードの暗号化の詳細については、 107 ページ の「パスワードの暗号化」を参照してください。	
ciw_directory	D:\RemoteInstall\OSChooser\English など、CIW/osc ファイルのディレクトリ	
temporary_sif_directory	D:\RemoteInstall\nlosp など、RemInst share の一時 sif ファイルの位置	
windowsHost	Windows サーバーを表す仮想ホスト	: [target:sys.hostName] -windows

Windows イメージおよびプロファイルの作成

プロビジョニングプロファイルは、使用する OS ファイルを特定し、またこれらのファイルをどのように配備するかに関する追加情報を含んでいます。

▼ Windows プロビジョニングプロファイルを作成する (ブラウザインタフェース)

始める前に Windows イメージサーバーが存在し、そのサーバーで Windows OS ソフトウェアが使用できることを確認します。Windows イメージサーバーの作成方法の詳細については、72 ページの「[Windows サーバーの設定](#)」を参照してください。

- 手順
1. プロビジョニングソフトウェアのページの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Windows Images**」セクションで「**Create Profile**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**WindowsImage**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**WindowsImage**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 変数を確認し、必要に応じて変数を変更します。
コンポーネント変数テーブルのプロンプトに表示される情報よりも多くの情報が必要である場合は、85 ページの「[Windows プロビジョニングプロファイルの変数](#)」にある、コンポーネントの詳細リストとともにデフォルト値および例を参照してください。
 - d. 変数セットを保存します。
 - e. 「**Plan Parameters**」テーブルの「**WindowsImage**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。

5. 「Plan Details Run」 ページで、イメージの添付先の **Windows** イメージサーバーを選択します。

ヒント – Windows イメージサーバーのホスト名は末尾が `-windows` になります。

6. 「Run Plan (includes preflight)」 をクリックします。
7. **Windows** イメージサーバーに対して **Windows** イメージが添付されたことを確認するには、**N1 SPS** ブラウザインタフェースの左側にある「**Components**」 をクリックします。

Windows provisioning profile のような記述を持つタイプ `com.sun.nlosp#Provision` のコンポーネントが表示されるはずですが、

▼ Windows プロビジョニングプロファイルを作成する (コマンド行インタフェース)

- 手順 1. **WindowsProfile** コンポーネントの変数セットを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/untyped/LinuxImage \  
-name "redhat3" -u admin -p admin -vars "osp_server=masterserver-osp;version=RedHat3.0; \  
architecture=x86;installPath=Linux:[version]_[architecture];description=testlinux; \  
min_disk_size=10GB;monitor_install=FALSE;boot_server_name=bikickstart; \  
boot_server_subnet_address=10.42.42.0; \  
boot_server_subnet_ip_address=10.42.42.3;kickstart_file=/export/el30.cfg; \  
pxe_boot_file=linux-install/pxelinux.0;pxe_configuration_file=el30-console"
```

これらの変数の詳細については、85 ページの「Windows プロビジョニングプロファイルの変数」を参照してください。

2. プランを実行するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/nlosp/untyped/LinuxImage-attach \  
-tar H:NM:bikickstart-linux -comp + -vs redhat3 -pto 30 -nto 10
```

Windows プロビジョニングプロファイルの変数

次の表に、Windows プロビジョニングプロファイルのすべてのコンポーネント変数を示します。デフォルト値を持つ変数に関しては、デフォルトが示されています。デフォルト値を持たない変数に関しては、値の例が示されています。

表 7-2 Windows プロビジョニングプロファイル用のコンポーネント変数

変数名	説明	デフォルトまたはサンプル値
osp_server	OS プロビジョニングサービスを表す仮想サーバー	masterserver-osp (例)
version	Windows のバージョン	2003 (デフォルト)
architecture		x86 (デフォルト)
installPath	Windows イメージの名前	Windows:[version]_[architecture] (デフォルト)
description	説明	
min_disk_size	配備するこのプロファイルの最小ディスクサイズ	10GB (デフォルト)
monitor_install	インストールを監視する必要があるかどうか	False (デフォルト)
boot_server_name	Windows ブートサーバーの名前	
boot_server_subnet_address	ブートサーバーによりサービスが提供されるサブネット	
boot_server_subnet_ip_address	このサブネット上のブートサーバーの IP アドレス	
boot_srv_active_directory_domain	このサブネットの Windows Active Directory ドメイン	winprov.n1lab.sun.com (例)
windows_product_directory	ブートサーバー上の Windows 製品のディレクトリ名	D:\RemoteInstall\Setup\English\Images\Win2003 (例)
sif_file	Software Information File (SIF) の位置	D:\RemoteInstall\Setup\English\Images\Win2003\i386\templates\ristndrd.sif (例)
windows_component_folder	自動生成 Windows Provision コンポーネントフォルダの位置	/com/sun/nlosp/autogen -: [target(...):sys.hostName] -windows (デフォルト)

ターゲットホストへの OS のインストール

プロファイルは、ターゲットホストに OS イメージをプロビジョニングするために使用されます。これらのプロファイルは、/com/sun/nlosp/autogen...provision フォルダにあります。新しい OS イメージが作成 (または添付) されるたびに、新しいプロファイルがプロビジョンフォルダに追加されます。

▼ ターゲットホストへ OS をインストールする

- 手順
1. プロビジョニングソフトウェアのページの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Windows Images**」セクションで「**Provision OS**」をクリックします。
 3. **provision** コンポーネントが含まれる **Windows** フォルダにナビゲートします。
たとえば /com/sun/nlosp/autogen-masterserver-windows/provision/ などです。
 4. コンポーネント名をクリックします。
たとえば Windows2000_x86 などです。
 5. 「**Component Details**」ページで **Provision_start:Install** プロシージャの横にある「**Run action**」をクリックします。
 6. オペレーティングシステムのプロビジョニング先であるターゲットホストを選択します。

ヒント - ターゲットホストのホスト名の末尾は -target です。

7. 「**Run Plan (includes preflight)**」をクリックします。

第 8 章

OS プロビジョニング用ターゲットホスト

オペレーティングシステムをプロビジョニングする場合は、プロビジョニングするハードウェアに関するある程度の知識が必要です。このプロセスを単純化するため、OS プロビジョニングプラグインはターゲットホスト定義のセットを提供しています。OS プロビジョニングプラグインはこの情報を使用して、ターゲットサーバーを遠隔で管理し、電源オン/オフ、コンソール機能の起動、ターゲットのブートモードの設定などの機能を実行します。

この章では、簡単にプロビジョニングできるターゲットホストのタイプを説明し、詳細な例をいくつか説明します。この章では、次の内容について説明します。

- 89 ページの「ターゲットホストタイプ」
- 93 ページの「ターゲットホストを定義する作業の例」
- 107 ページの「パスワードの暗号化」

ターゲットホストタイプ

さまざまなハードウェアタイプを遠隔で管理するために、さまざまなメカニズムが使用されます。その結果、その種類のハードウェアのオペレーティングシステムを効率的に構成するには、固有の情報を知っておく必要があります。OS プロビジョニングプラグインは、OS プロビジョニングターゲットに使用する次のハードウェアグループを提供しています。

- Sun Intelligent Platform Management Interface (IPMI) ターゲット
- Sun Advanced Light Out Management (ALOM) ターゲット
- Sun Light Out Management (LOM) ターゲット
- Sun Mid Range Domain Controller ターゲット
- Hewlett-Packard Integrated Light Out Management (ILO) ターゲット
- そのほかの IPMI ターゲット

- そのほかの遠隔管理ターゲット (B100s や B100x など、上記のカテゴリに属さないもの)
- 遠隔管理をサポートしない汎用ターゲットホスト

各ターゲットホストタイプには、そのホストタイプに適用される固有の変数のセットがあります。各ターゲットホストタイプに適用される変数の詳細については、[付録 A](#) を参照してください。

プラットフォームによるターゲットホストタイプ

次の表に、OS プロビジョニングプラグインがサポートするプラットフォームとターゲットの一部を示します。各ハードウェアプラットフォームに関して、表には次の情報が示してあります。

- テストされているファームウェアまたはサービスプロセッサのバージョン
- これらのプラットフォームで実行する必要がある構成
- OS プロビジョニング変数用に収集する必要がある情報
- 適用されるターゲットホストのタイプ

表 8-1 プラットフォームによるターゲットホスト

ターゲットプラットフォーム	ファームウェアまたはサービスプロセッサ	構成に関する要件	OS プロビジョニングターゲット変数に必要な情報	ターゲットホストタイプ
SunFire v240、v210 ファミリー	Sun Advanced Lights Out Manager 1.3 (ALOM)	<ul style="list-style-type: none"> ■ ALOM ポート (OS プロビジョニングサーバーからルーティング可能) に IP (静的) を提供 ■ telnet を有効化 	<ul style="list-style-type: none"> ■ ALOM ip、ユーザー/パスワード ■ ターゲットプロビジョニングインタフェースの MAC アドレス 	Sun ALOM ターゲット

表 8-1 プラットフォームによるターゲットホスト (続き)

ターゲットプラットフォーム	ファームウェアまたはサービスプロセッサ	構成に関する要件	OS プロビジョニング ターゲット変数に必要な 情報	ターゲットホストタイプ
SunFire v120 ファミ リ	Sun Lights Out Manager のバー ジョン: v3.12 (LOM)	<ul style="list-style-type: none"> ■ LOM ポートを端 末サーバーに接 続 (端末サーバー の IP は OS プロ ビジョニング サーバーから ルーティング可 能) ■ LOM と Console の間で Serial A/LOM を共有 するには、Serial A/LOM と Serial B ポートの両方へ のコンソール接 続を設定 たとえば、 Solaris コマンド 行から次のコマ ンドを入力: <pre>eeprom input-device=ttya eeprom output-device=ttya reboot</pre> 	<ul style="list-style-type: none"> ■ LOM ユーザー /パスワード (存 在する場合) ■ ターゲットプロ ビジョニングイ ンタフェースの MAC アドレス ■ 端末サーバーの 情報 (タイプ、 ip、ターゲット LOM ポートに接 続されたポート) 	Sun LOM ターゲット
SunFire v20z、v40z ファミリ	Sun IPMI v1.5 Service Processor、 バージョン V2.1.0.10	<ul style="list-style-type: none"> ■ Network Management ポート (OS プロ ビジョニング サーバーから ルーティング可 能) に IP (静的) を提供 ■ ssh を有効化 ■ ipmi、ipmi チャンネル「lan」 を有効化 ■ まず PXE 対応 NIC からブート するよう BIOS を 構成 	<ul style="list-style-type: none"> ■ Service Processor の IP ■ ssh ユーザー/パ スワード ■ ipmi パスワード ■ ターゲットプロ ビジョニングイ ンタフェースの MAC アドレス ■ プロビジョニン グインタフェー スの GUID 	Sun IPMI ターゲット

表 8-1 プラットフォームによるターゲットホスト (続き)

ターゲットプラットフォーム	ファームウェアまたはサービスプロセッサ	構成に関する要件	OS プロビジョニング ターゲット変数に必要な 情報	ターゲットホストタイプ
SunFire B100s (sparc)、B100x (x86) ファミリ	Sun Advanced Lights Out Manager for Blade Servers 1.2 (ALOM-B 1.2)	<ul style="list-style-type: none"> ■ ALOM ポート (OS プロビジョニングサーバーからルーティング可能) へ IP (静的) を提供、または ALOM シリアルポートを端末サーバーに接続 (端末サーバーの IP は OS プロビジョニングサーバーからルーティング可能である必要がある) ■ Telnet を有効化 (IP ベースの管理用) 	<ul style="list-style-type: none"> ■ ALOM IP、または端末サーバーの情報 (ip、タイプ、ALOM シリアルポートに接続されるポート) ■ ALOM ユーザー/パスワード ■ ターゲットプロビジョニングインタフェースの MAC アドレス ■ プロビジョニングインタフェースの GUID (B100x のみ) 	Network Management HAL jar power として b1600sc.jar を使用するほかの遠隔管理ターゲット、ブートおよびコンソールサービスはすべてこの HAL によりサポートされる
SunFire v60x、v65x ファミリ	IPMI 対応の Baseboard Management Controller (BMC)	<ul style="list-style-type: none"> ■ LOM/IPMI ポート (OS プロビジョニングサーバーからルーティング可能) へ IP (静的) を提供、または ipmi、ipmi チャンネル「lan」を有効化 ■ まず PXE 対応 NIC からブートするよう BIOS を構成 	<ul style="list-style-type: none"> ■ LOM/IPMI IP ipmi パスワード ■ ターゲットプロビジョニングインタフェースの MAC アドレス ■ プロビジョニングインタフェースの GUID 	Network Management HAL jar 専用 power として ipmi.jar を使用するほかの IPMI ターゲット、サービスはこの HAL によって提供される
SunFire 3800、4800 または 6800 ドメイン	システムコントローラ: ScApp のバージョン: 5.18.1 Build_01 RTOS の バージョン: 41	<ul style="list-style-type: none"> ■ システムコントローラ (OS プロビジョニングサーバーからルーティング可能) に IP (静的) を提供 ■ telnet を有効化 	<ul style="list-style-type: none"> ■ プラットフォーム SC パスワード ■ ドメイン SC パスワード ■ ターゲットプロビジョニングインタフェースの MAC アドレス 	Sun Mid Range ターゲット

表 8-1 プラットフォームによるターゲットホスト (続き)

ターゲットプラットフォーム	ファームウェアまたはサービスプロセッサ	構成に関する要件	OS プロビジョニング ターゲット変数に必要な 情報	ターゲットホストタイプ
HP Proliant DL シ リーズ (iLO ベースの 管理をサポート)	iLO Integrated Lights Out Manager v 1.6.41	<ul style="list-style-type: none"> ■ iLO ポート (OS プロビジョニン グサーバーから ルーティング可 能) に IP (静的) を提供 ■ まず PXE 対応 NIC からブート するよう BIOS を 構成 	<ul style="list-style-type: none"> ■ iLO ip、ユーザー /パスワード ■ ターゲットプロ ビジョニングイ ンタフェースの MAC アドレス ■ プロビジョニン グインタフェー スの GUID 	HP iLO ターゲット
その他のター ゲット	-	まず PXE 対応 NIC からブートするよう BIOS を構成 (x86 の み)	<ul style="list-style-type: none"> ■ ターゲットプロ ビジョニングイ ンタフェースの MAC アドレス ■ プロビジョニン グインタフェー スの GUID (x86 のみ) 	汎用ターゲット (手動 制御)

ターゲットホストを定義する作業の例

ターゲットホストを定義するために従うべき一般的なプロセスは、すべてのターゲットホストタイプで似ています。ただし、タイプごとに詳細は異なります。この節では、すべてではありませんが、一部のターゲットホストタイプの作業例を説明します。

Sun ALOM ターゲットホストの定義

「プロビジョニング」とよく言われる、システムにイメージをインストールする作業の前には、OS プロビジョニングサーバーにそのシステムを通知する必要があります。

▼ Sun ALOM ターゲットホストを作成する (ブラウザインタフェース)

- 手順
1. N1 SPS ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Target Hosts**」セクションの「**Sun ALOM**」セクションで、「**Create**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**SunALOM**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**SunALOM**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 変数の値を確認し、必要に応じて値を変更します。

コンポーネント変数テーブルのプロンプトにより提供される情報よりも多くの情報が必要である場合、[123 ページの「Sun ALOM ターゲットホストの変数」](#)のデフォルト値と例とともに、変数の詳細リストを参照してください。
 - d. 変数セットとプランプロンプトのどちらにパスワードを入力するかを決定します。

パスワードを入力する方法は次の2つがあります。

 - 次の変数の適切な値を入力します。

```
alom_access_userid  
alom_access_password  
terminal_server_userid  
terminal_server_password
```

パスワードの暗号化の詳細については、[107 ページの「パスワードの暗号化」](#)を参照してください。
 - e. 変数セットを保存します。
 - f. 「**Plan Parameters**」テーブルの「**SunALOM**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。

5. 「Plan Details Run」ページから、ターゲットホストを定義するプロビジョニングサーバーを選択します。

ヒント - OS プロビジョニングサーバーのホスト名の末尾は `-osp` です。

6. プランからパスワードを暗号化するには、「Plan Details Run」ページの次の手順に従ってください。
 - a. 「If you are specifying the ALOM password below」の横にあるチェックボックスをクリックします。
 - b. 「Password to Access ALOM」フィールドにパスワードを入力します。入力時にパスワードは暗号化されます。
 - c. 「If you are specifying the terminal server password below」の横にあるチェックボックスをクリックします。
 - d. 「Terminal Server Password」フィールドに端末サーバーのパスワードを入力します。入力時にパスワードは暗号化されます。
7. 「Run Plan (includes preflight)」をクリックします。
8. ターゲットホストが正常に作成されたことを確認するには、**N1 SPS** ブラウザインタフェースの左側にある「Hosts」リンクをクリックします。
-target が付加された仮想ホストの名前が表示されます。たとえば、ホスト test240 のプロビジョニングターゲットを定義した場合、仮想ホストは test240-target となります。

▼ Sun ALOM ターゲットホストを作成する (コマンド行インタフェース)

- 手順 1. SunALOM コンポーネントの変数セットを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/targets/SunALOM \  
-name "testv240" -u admin -p admin -vars "installPath=testv240; \  
description=To test v240;ethernet_mac_address=0:3:ba:53:5b:5d; \  
ethernet_ip_address=10.42.42.100;ethernet_netmask=255.255.255.0;architecture=sun4u; \  
kernel_arch=sun4u;disk_size=20GB;alom_ip_address=10.5.133.150;alom_access_userid=admin; \  
alom_access_password="
```

これらの変数の詳細については、123 ページの「Sun ALOM ターゲットホストの変数」を参照してください。

2. 適切なパスワードアクセス権を設定するには、次のエントリが含まれるファイルを作成します。

```
true
root
false
root
```

各行の意味は次のとおりです。

- ファイルの最初の行は、ブラウザインタフェースのプラン変数セクションの「If you are specifying the ALOM password below」ボックスをチェックすることに対応します。
- ファイルの第2行は、ALOM にアクセスするためのパスワードです。
- ファイルの第3行は、ブラウザインタフェースのプラン変数セクションの「If you are specifying the terminal password below」ボックスをチェックすることに対応します。
- ファイルの第4行は、端末にアクセスするためのパスワードです。

次の手順で思い出せるように、必ず意味のある名前をファイルに付けてください。たとえば、ファイル名を /tmp/testv240-target のようにします。

3. プランを実行し、ターゲットを作成するには、次の例のようなコマンドを入力します。

```
# cat /tmp/ALOM-target | cr cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/nlosp/targets/SunALOM-create -tar H:NM:masterserver-osp \
-comp + -vs testv240 -pto 30 -nto 10 -f /tmp/testv240-target
```

Sun IPMI ターゲットホストの定義

Sun IPMI ターゲットホストタイプは、Sun Fire™ V20z または V60z システムなどのシステムに使用されます。

▼ Sun IPMI ターゲットホストを作成する (ブラウザインタフェース)

この作業では、ターゲットホストは Sun Intelligent Platform Management Interface (IPMI) システムです。特に、この例では V20z ターゲットホストを定義します。

- 手順
1. **N1 SPS** ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Target Hosts**」セクションの「**Sun IPMI Targets**」サブセクションで、「**Create**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。

4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**SunIPMI**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**SunIPMI**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 変数を確認し、必要に応じて変数を変更します。

コンポーネント変数テーブルのプロンプトは、大部分が自明なものです。ただし、詳細情報が必要である場合は、[121 ページの「Sun IPMI ターゲットホストの変数」](#)にある、デフォルト値および例とともに変数の詳細リストを参照してください。
 - d. 変数セットとプランプロンプトのどちらにパスワードを入力するかを決定します。

パスワードを入力する方法は次の 2 つがあります。

 - 次の変数の適切な値を入力します。

```
ipmi_access_userid
ipmi_access_password
ssh_access_userid
ssh_access_password
```

パスワードの暗号化の詳細については、[107 ページの「パスワードの暗号化」](#)を参照してください。

 - コンポーネント変数を空白のままにしておいて、「**plan run**」ページで提供されるプランプロンプトにパスワードを入力します。
 - e. 変数セットを保存します。
 - f. 「**Plan Parameters**」テーブルの「**SunIPMI**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
5. 「**Plan Details Run**」ページから、ターゲットホストを定義するプロビジョニングサーバーを選択します。

ヒント – OS プロビジョニングサーバーのホスト名の末尾は `-osp` です。

6. プランからパスワードを暗号化するには、「**Plan Details Run**」ページの次の手順に従ってください。

- a. 「If you are specifying the IPMI password below」の横にあるチェックボックスをクリックします。
 - b. 「Password to Access IPMI」フィールドにパスワードを入力します。
入力時にパスワードは暗号化されます。
 - c. 「If you are specifying the SSH password below」の横にあるチェックボックスをクリックします。
 - d. 「SSH Password to Access the Remote Management Interface」フィールドにパスワードを入力します。
入力時にパスワードは暗号化されます。
7. 「Run Plan (includes preflight)」をクリックします。
 8. ターゲットホストが正常に作成されたことを確認するには、**N1 SPS** ブラウザインタフェースの左側にある「Hosts」リンクをクリックします。
-target が付加された仮想ホストの名前が表示されます。たとえば、ホスト sunfire-v60x のプロビジョニングターゲットを定義した場合、仮想ホストは sunfire-v20z-target となります。

▼ Sun IPMI ターゲットホストを作成する (コマンド行インタフェース)

- 手順 1. SunIPMI コンポーネントの変数セットを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/targets/SunIPMI -name "v20z" \
-u admin -p admin -vars "installPath=v20zhost;description=test v20z host; \
ethernet_mac_address=00:09:3d:10:94:a6;ethernet_ip_address=10.42.42.110; \
ethernet_netmask=255.255.255.0;architecture=x86;kernel_arch=i86pc; \
guid={3AE45192-159D-11D8-AEF3-001083FDFE90};disk_size=20GB;ipmi_ip_address=10.5.133.134; \
ipmi_access_userid;ipmi_access_password;;ssh_access_userid=admin;ssh_access_password="
```

これらの変数の詳細については、121 ページの「Sun IPMI ターゲットホストの変数」を参照してください。

2. 適切なパスワードアクセス権を設定するには、次のエントリが含まれるファイルを作成します。

```
true
admin
true
admin
```

各行の意味は次のとおりです。

- ファイルの最初の行は、ブラウザインタフェースのプラン変数セクションの「If you are specifying the IPMI password below」ボックスをチェックすることに対応します。
- ファイルの第2行は、IPMI にアクセスするためのパスワードです。
- ファイルの第3行は、ブラウザインタフェースのプラン変数セクションの「If you are specifying the SSH password below」ボックスをチェックすることに対応します。
- ファイルの第4行は、遠隔管理インタフェースにアクセスするためのSSH パスワードです。

次の手順で思い出せるように、必ず意味のある名前をファイルに付けてください。たとえば、ファイル名を /tmp/testv20z-target のようにします。

3. プランを実行し、**V20z** ターゲットを作成するには、次の例のようなコマンドを入力します。

```
# cat /tmp/testv20z-target | cr_cli -cmd pe.p.run -u admin -p admin \  
-PID NM:/com/sun/nlosp/targets/SunIPMI-create -tar H:NM:masterserver-osp \  
-comp + -vs v20z -pto 30 -nto 10 -f /tmp/test-v20z
```

HP ILO ターゲットホストの定義

次の作業に、HP ProLiant シリーズシステムなどの HP ILO ホストを定義する方法を示します。

▼ HP ILO ターゲットホストを作成する (ブラウザインタフェース)

- 手順
1. **N1 SPS** ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Target Hosts**」セクションの「**HP ILO Targets**」サブセクションで、「**Create**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**HPILO**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**HPILO**」コンポーネント行で「**Select from List**」をクリックします。

- a. 「**Create Set**」をクリックします。
- b. 変数セットの名前を入力します。
- c. 変数を確認し、必要に応じて変数を変更します。
コンポーネント変数テーブルのプロンプトは、大部分が自明なものです。ただし、詳細情報が必要である場合は、130 ページの「**HP ILO ターゲットホストの変数**」にある、デフォルト値および例とともに変数の詳細リストを参照してください。
- d. 変数セットとプランプロンプトのどちらにパスワードを指定するかを決定します。
パスワードを指定する方法は次の 2 つがあります。
 - 次の変数の適切な暗号化された値を入力します。


```
ilo_access_userid  
ilo_access_password
```


パスワードの暗号化の詳細については、107 ページの「**パスワードの暗号化**」を参照してください。
 - コンポーネント変数を空白のままにしておいて、「**plan run**」ページで提供されるプランプロンプトにパスワードを入力します。
- e. 変数セットを保存します。
- f. 「**Plan Parameters**」テーブルの「**HPILO**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。

5. 「**Plan Details Run**」ページで、ターゲットホストを定義するプロビジョニングサーバーを選択します。

ヒント – OS プロビジョニングサーバーのホスト名の末尾は `-osp` です。

6. プランからパスワードを暗号化するには、「**Plan Details Run**」ページの次の手順に従ってください。
 - a. 「**If you are specifying the ILO password below**」の横にあるチェックボックスをクリックします。
 - b. 「**Password to Access ILO**」フィールドにパスワードを入力します。
入力時にパスワードは暗号化されます。
7. 「**Run Plan (includes preflight)**」をクリックします。

- ターゲットホストが正常に作成されたことを確認するには、**N1 SPS** ブラウザインタフェースの左側にある「**Hosts**」リンクをクリックします。
-target が付加された仮想ホストの名前が表示されます。たとえば、ホスト hp-proliant-1 のプロビジョニングターゲットを定義した場合、仮想ホストは hp-proliant-1-target となります。

▼ HP ILO ターゲットホストを作成する (コマンド行インタフェース)

- 手順 1. **HPILO** コンポーネントの変数セットを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/targets/HPILO -name "hpProliant" \  
-u admin -p admin -vars "installPath=HP-Proliant;description=Test description; \  
ethernet_mac_address=00:0E:7F:FD:85:26;ethernet_ip_address=10.42.42.20; \  
ethernet_netmask=255.255.255.0;architecture=x86;kernel_arch=i86pc; \  
guid={4D303435-4C47-5037-3239-202020202020};disk_size=20GB;ilo_ip_address=10.5.133.166; \  
ilo_version=1.6.4;ilo_access_userid=admin;ilo_access_password="
```

これらの変数の詳細については、130 ページの「**HP ILO ターゲットホストの変数**」を参照してください。

- 適切なパスワードアクセス権を設定するには、次のエントリが含まれるファイルを作成します。

```
true  
admin123
```

各行の意味は次のとおりです。

- ファイルの最初の行は、ブラウザインタフェースのプラン変数セクションの「If you are specifying the ILO password below」ボックスをチェックすることに対応します。
- ファイルの第 2 行は、ILO にアクセスするためのパスワードです。

次の手順で思い出せるように、必ず意味のある名前をファイルに付けてください。たとえば、ファイル名を /tmp/hp-proliant のようにします。

- プランを実行し、ターゲットホストを作成するには、次の例のようなコマンドを入力します。

```
# cat /tmp/hp-proliant | cr_cli -cmd pe.p.run -u admin -p admin \  
-PID NM:/com/sun/nlosp/targets/HPILO-create -tar H:NM:masterserver-osp -comp + \  
-vs hpProliant -pto 30 -nto 10 -f /tmp/hp-proliant
```

そのほかの遠隔管理ターゲットホストの定義

事前定義されたタイプに一致しないが、SunFire B100s や B100x ブレードなど、そのほかの形式の遠隔管理を有するターゲットホストを特定するには、遠隔管理ターゲットホストタイプを使用します。このオプションでは、ターゲットに関する情報を指定する非常に大きな柔軟性が得られますが、いくつかの変数を入力する必要があります。

このタイプのターゲットに関して収集される遠隔管理情報には2つのセットがあります。遠隔管理情報の最初のセットは `_1` 接尾辞で終わり、2番目のセットは `_2` 接尾辞で終わります。各遠隔ネットワーク管理情報のセットには、ネットワーク管理のタイプに関する情報と、遠隔ネットワーク管理を処理する関連付けられた JAR ファイルが必要です。JAR ファイルは、OS プロビジョニングサーバー上の `/opt/SUNWnlosp/drivers` にあります。各遠隔ネットワーク管理インタフェースは端末サーバー接続を使用できます。この場合、端末サーバーに関する情報を入力する必要があります。ネットワーク管理のタイプに応じて、ネットワーク管理は電源管理、ブートモード、およびコンソールサポートをサポートする場合があります。詳細については、以下の例を参照してください。

▼ そのほかの遠隔管理ターゲットホストを作成する (ブラウザインタフェース)

- 手順
1. **N1 SPS** ブラウザインタフェースの「**Common Tasks**」セクションで、「**OS Provisioning**」を選択します。
 2. 「**OS Provisioning Common Tasks**」ページの「**Target Hosts**」セクションの「**Other Remotely-Managed Targets**」サブセクションで、「**Create**」をクリックします。
 3. 「**Plans Details**」ページの「**Run**」をクリックします。
 4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**Other**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**Other**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. 変数を確認し、必要に応じて変数を変更します。

コンポーネント変数テーブルのプロンプトは、大部分が自明なものです。ただし、詳細情報が必要である場合は、135 ページの「そのほかの遠隔管理ターゲットホストの変数」にある、デフォルト値および例とともに変数の詳細リストを参照してください。

- d. 変数セットとプランプロンプトのどちらにパスワードを指定するかを決定します。

パスワードを指定する方法は次の2つがあります。

- 次の変数の適切な暗号化された値を入力します。

```
netmgmt_password_1
terminal_server_access_password_1
netmgmt_password_2
terminal_server_access_password_2
```

パスワードの暗号化の詳細については、107 ページの「パスワードの暗号化」を参照してください。

- コンポーネント変数を空白のままにしておいて、「plan run」ページで提供されるプランプロンプトにパスワードを入力します。

- e. 変数セットを保存します。

- f. 「Plan Parameters」テーブルの「Other」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。

5. 「Plan Details Run」ページから、ターゲットホストを定義するプロビジョニングサーバーを選択します。

ヒント - OS プロビジョニングサーバーのホスト名の末尾は -osp です。

6. プランからパスワードを暗号化するには、「Plan Details Run」ページの次の手順に従ってください。

- a. 最初の「If you are specifying the password below」の横にあるチェックボックスをクリックします。

- b. 「Password to Access Management Interface 1」フィールドに、最初のネットワーク管理インターフェイスにアクセスするためのパスワードを入力します。入力時にパスワードは暗号化されます。

- c. 最初の「If you are specifying the terminal server password below」の横にあるチェックボックスをクリックします。

- d. 「Password to Access Terminal Server Password 1」フィールドに、最初の端末サーバーにアクセスするためのパスワードを入力します。入力時にパスワードは暗号化されます。

- e. 2つ目の「If you are specifying the password below」の横にあるチェックボックスをクリックします。

- f. 「**Password to Access Management Interface 2**」フィールドに、2つ目のネットワーク管理インタフェースにアクセスするためのパスワードを入力します。
入力時にパスワードは暗号化されます。
 - g. 2つ目の「**If you are specifying the terminal server password below**」の横にあるチェックボックスをクリックします。
 - h. 「**Password to Access Terminal Server Password 2**」フィールドに、2つ目の端末サーバーにアクセスするためのパスワードを入力します。
入力時にパスワードは暗号化されます。
7. 「**Run Plan (includes preflight)**」をクリックします。
8. ターゲットホストが正常に作成されたことを確認するには、**N1 SPS** ブラウザインタフェースの左側にある「**Hosts**」リンクをクリックします。
-target が付加された仮想ホストの名前が表示されます。たとえば、ホスト sunfire-v60x のプロビジョニングターゲットを定義した場合、仮想ホストは sunfire-v60x-target となります。

▼ そのほかの遠隔管理ターゲットホストを作成する (コマンド行インタフェース)

次の CLI の例では、SunFire B100s システムのターゲットホストを作成します。

- 手順 1. **Other** コンポーネントの変数セットを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/targets/Other -name "b100s10" \
-u admin -p admin -vars "installPath=Blade-S10;description=S10 b100s blade target; \
ethernet_mac_address=00:03:ba:29:e4:ea;ethernet_ip_address=10.216.0.11; \
ethernet_netmask=255.255.255.192;architecture=sun4u;kernel_arch=sun4u;disk_size=30GB; \
netmgmt_type_1=b1600sc;netmgmt_version_1=1.0; \
netmgmt_hal_jar_1="file:///opt/SUNWnlosp/drivers/sun/b1600sc.jar"; \
netmgmt_ip_address_1=10.216.0.60;netmgmt_protocol_1=telnet;netmgmt_port_1=s10; \
power_service_true_false_1=true;boot_service_true_false_1=true; \
console_service_true_false_1=true;netmgmt_userid_1=admin;netmgmt_password_1=Clz6pK2b6qw="
```

これらの変数の詳細については、135 ページの「そのほかの遠隔管理ターゲットホストの変数」を参照してください。

2. 適切なパスワードアクセス権を設定するには、次のエントリが含まれるファイルを作成します。

```
false
admin
false
admin
false
```

```
admin
false
admin
```

各行の意味は次のとおりです。

- ファイルの最初の行は、ブラウザインタフェースのプラン変数セクションの最初の「If you are specifying the password below」ボックスに対応します。
- ファイルの第2行は、最初の管理インタフェースに対するパスワードです。
- ファイルの第3行は、ブラウザインタフェースのプラン変数セクションの最初の「If you are specifying the terminal server password below」ボックスに対応します。
- ファイルの第4行は、最初の端末サーバーに対するパスワードです。
- ファイルの第5行は、ブラウザインタフェースのプラン変数セクションの2つ目の「If you are specifying the password below」ボックスに対応します。
- ファイルの第6行は、2つ目の管理インタフェースに対するパスワードです。
- ファイルの第7行は、ブラウザインタフェースのプラン変数セクションの2つ目の「If you are specifying the terminal server password below」ボックスに対応します。
- ファイルの第8行は、2つ目の端末サーバーに対するパスワードです。

次の手順で思い出せるように、必ず意味のある名前をファイルに付けてください。たとえば、ファイル名を /tmp/b100s のようにします。

3. プランを実行し、ターゲットホストを作成するには、次の例のようなコマンドを入力します。

```
# cat /tmp/b100s | cr_cli -cmd pe.p.run -u admin -p admin \  
-PID NM:/com/sun/nlosp/targets/Other-create -tar H:NM:masterserver-osp -comp + \  
-vs b100s10 -pto 30 -nto 10 -f /tmp/b100s
```

汎用ターゲットホストの定義

ワークステーションやデスクトップシステムなど、遠隔管理インタフェースを持たないターゲットの電源を手動でオン/オフにすることができます。これらのターゲットを OS プロビジョニング用に特定するには、「Generic Target Host」オプションを使用します。

▼ 汎用ターゲットホストを作成する (ブラウザインタフェース)

- 手順 1. N1 SPS ブラウザインタフェースの「Common Tasks」セクションで、「OS Provisioning」を選択します。

2. 「**OS Provisioning Common Tasks**」ページの「**Target Hosts**」セクションの「**Generic Targets**」サブセクションで、「**Create**」をクリックします。
3. 「**Plans Details**」ページの「**Run**」をクリックします。
4. このプランに使用する変数を選択します。
 - 既存の変数セットを使用するには、「**Plan Parameters**」テーブルの「**Target**」コンポーネント行にあるドロップダウンメニューから名前を選択します。
 - 新しい変数セットを作成するには、「**Plan Parameters**」テーブルの「**Target**」コンポーネント行で「**Select from List**」をクリックします。
 - a. 「**Create Set**」をクリックします。
 - b. 変数セットの名前を入力します。
 - c. **OSP Control Service** を呼び出すかどうかを決定します。

汎用ターゲットホストに関しては、osp_control_service 変数に特に注意してください。ユーザーがこの値を TRUE に設定し、システムに手動の HAL がある場合、N1 SPS ソフトウェアは停止し、ユーザーが適切な機能を実行するのを待機します。ステータス機能を介してインストールのステータスを監視している場合、OS プロビジョニング制御サービスは、ターゲットホストの電源オフ、再起動、および電源オンの時点をユーザーに通知します。

ターゲットホストが私設ネットワーク上にあり、DHCP サービスにアクセスできない場合は、この変数を FALSE に設定します。
 - d. そのほかの変数を確認し、必要に応じて変数を変更します。

コンポーネント変数テーブルのそのほかのプロンプトは、大部分が自明なものです。ただし、詳細情報が必要である場合は、140 ページの「汎用ターゲットホストの変数」にある、デフォルト値および例とともに変数の詳細リストを参照してください。
 - e. 変数セットを保存します。
 - f. 「**Plan Parameters**」テーブルの「**Target**」コンポーネント行にあるドロップダウンメニューから、直前に保存した変数セットを選択します。
5. 「**Plan Details Run**」ページで、ターゲットホストを定義するプロビジョニングサーバーを選択します。

ヒント - OS プロビジョニングサーバーのホスト名の末尾は -osp です。

6. 「**Run Plan (includes preflight)**」をクリックします。

7. ターゲットホストが正常に作成されたことを確認するには、**N1 SPS** ブラウザインタフェースの左側にある「**Hosts**」リンクをクリックします。
-target が付加された仮想ホストの名前が表示されます。たとえば、ホスト sunfire-v60x のプロビジョニングターゲットを定義した場合、仮想ホストは sunfire-v60x-target となります。

▼ 汎用ターゲットホストを作成する (コマンド行インタフェース)

- 手順 1. **Target** コンポーネントの変数セットを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/targets/Target -name "mysystem" \  
-u admin -p admin -vars "installPath=Generic;target_host=[installPath]-target; \  
description=Test description;ethernet_mac_address=00:0E:7F:FD:85:26; \  
ethernet_ip_address=10.42.42.20;ethernet_netmask=255.255.255.0;architecture=x86; \  
kernel_arch=i86pc;disk_size=20GB"
```

これらの変数の詳細については、140 ページの「汎用ターゲットホストの変数」を参照してください。

2. プランを実行し、ターゲットホストを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID \  
NM:/com/sun/nlosp/targets/Target-create -tar H:NM:masterserver-osp -comp + \  
-vs mysystem -pto 30 -nto 10
```

パスワードの暗号化

Solaris または Linux システムの root アクセス権のパスワードを指定するには、OS プロビジョニングのツールではなくオペレーティングシステムのツールを使用してパスワードを暗号化する必要があります。

OS プロビジョニング変数のパスワードの暗号化

OS プロビジョニング変数のパスワードを暗号化するには、次の 3 つの方法があります。

- OS プロビジョニングサーバーで次のコマンドを実行する。
`/opt/nlosp/sbin/nlosp_encrypter clear-text clear-text` の暗号化されたバージョンが画面に表示されます。

- N1 SPS ブラウザインタフェースを使用して、プラン /com/sun/nlosp/untyped/EncryptPassword にナビゲートし、そのプランでパスワードを入力する。暗号化された値を確認するには、「progress details」リンクをナビゲートし、stdout を表示します。
- N1 SPS サーバーで次のコマンドを入力する。

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/nlosp/untyped/EncryptPassword \
-tar H:NM:masterserver-osp -pto 30 -nto 10
# cr_cli -cmd pe.p.lo -u admin -p admin -ID id-from-the-EncryptPassword-command
# cr_cli -cmd pe.p.en -u admin -p admin -ID last-id-from-the-pe.p.lo-command
```

OS root アクセス権のパスワードの暗号化

Solaris または Linux システムの root アクセス権のパスワードを指定する必要がある場合、OS プロビジョニングのツールではなくオペレーティングシステムのツールを使用してパスワードを暗号化する必要があります。

ツールは、オペレーティングシステムとオペレーティングシステムの特定のバージョンにより異なります。たとえば Solaris 9 システムでは、通常のユーザーは単に /etc/shadow ファイルから root パスワードをコピーします。詳細については、ご使用の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。

第 9 章

障害追跡

この章には、OS プロビジョニングの問題をより簡単に解決できるようにする情報が記載されています。

この章では、次の内容について説明します。

- 109 ページの「一般的な障害追跡のガイドライン」
- 111 ページの「Solaris 関連の問題の解決」
- 112 ページの「Linux 関連の問題の解決」
- 114 ページの「Windows 関連の問題の解決」

一般的な障害追跡のガイドライン

問題: プロビジョンプランは正常に実行されたが、ターゲットに対するプロビジョニングに失敗した。

対処方法: これにはいくつかの問題が考えられます。以下では、プロビジョニング失敗の考えられるいくつかの理由を示します。このリストを使用して問題を特定してください。

1. プロビジョニングログを調べるには、N1 SPS ブラウザインタフェースで「Host Status」情報を表示します。ターゲットで Host Status プランを実行します。stdout と stderr の両方をチェックして、失敗の原因を確認します。
2. OS プロビジョニングサーバーにログインし、`/var/run/nlosp/log` フォルダでログをチェックし、また `/var/run/nlosp/console` フォルダでコンソール出力もチェックします。また、`/var/adm/nlosp*` ファイルのメッセージもチェックします。`/var/adm/nlosp*` ファイルで詳細を表示するには、`/opt/SUNWnlosp/etc/nlosp.properties` ファイルで `n1.isp.core.debuglevel` プロパティの値を変更します。たとえば `n1.isp.core.debuglevel=25` のようにします。
3. ターゲットへのコンソールを取得し、再度プロビジョニングを行って失敗の原因を確認します。

- OS プロビジョニングサーバー、ブートおよびインストールサーバー、およびターゲットホストの間のネットワーク接続をチェックします。OS プロビジョニングサーバーとブートおよびインストールサーバーの間、OS プロビジョニングサーバーとターゲットホストの間、およびブートおよびインストールサーバーとターゲットホストの間のパケットに対して ping/snoop を行います。

注 - OS プロビジョニングサーバーまたはブートおよびインストールサーバーが複数の IP アドレスを持っている場合、プロビジョニングに使用されるアドレスを使用します。

- ターゲットからの DHCP パケットが OS プロビジョニングサーバーに到達していない場合、ターゲットが、DHCP を使用してネットワーク経由でブートするよう構成されているかどうかをチェックします。ターゲットホストに複数のネットワークインタフェースがある場合、ターゲットホストが、ホストプロファイルで指定されているインタフェースを使用し、オペレーティングシステムのブートとインストールを行うようにします。ターゲットのプロビジョニングを再度行い、OS プロビジョニングサーバーの /etc/dhcpd.conf ファイルにターゲットホスト用のエントリがあるかどうかをチェックします。OS プロビジョニングの期間のみターゲットに応答するよう DHCP が構成されている場合は、ターゲットのプロビジョニングを再度行い、DHCP が正しく構成されているかどうかを確認する必要があります。
- Solaris、Linux、および Windows のブートおよびインストールサーバーが正しく設定されているかどうかをチェックします。詳細については、適切な OS のマニュアルを参照してください。NFS (Solaris および Linux の場合) または CIFS (Windows の場合) を使用して OS メディアを共有するよう、ブートおよびインストールサーバーが正しく構成されていることを確認します。OS プロファイルで使用されている IP アドレスと、ブートおよびインストールサーバーで構成されている IP アドレスが一致することを確認します。TFTP サービスがブートおよびインストールサーバーで動作するよう構成されているかどうかをチェックします。
- IP アドレス、パスワードなどの情報の OS プロファイル情報とホストプロファイル情報をチェックします。
- OS がインストールに失敗したりハングする場合は、ネットワーク経由でターゲットをブートするのに必要なドライバが OS プロファイルにあるかどうかをチェックします。詳細については、各 OS のマニュアルを確認してください。

問題: プロファイルの作成時、ホストの作成時、またはターゲットのプロビジョニング時にエラーが発生する。

対処方法: エラーは、プロビジョニングプロセスの複数の時点で発生する可能性があります。次のことをチェックします。

- N1 SPS リモートエージェント (RA) が、ブートおよびインストールサーバーで正しくインストールされていることを確認します。マスターサーバーが RA に到達できることを確認します。詳細については、『Sun N1 Service Provisioning System 5.1 インストールガイド』を参照してください。

2. N1 SPS コマンド行インタフェース (CLI) が OS プロビジョニングサーバーおよび Solaris ブートおよびインストールサーバー上にインストールされていることを確認します。単純な `cr_cli` コマンドを実行します。
3. プランの `stdout` および `stderr` をチェックします。
4. プランおよびコンポーネント変数に有効な値が指定されていることを確認します。

Solaris 関連の問題の解決

問題: Solaris オペレーティングシステムのプロビジョニング処理のシーケンスがわからない。

対処方法: Solaris のシーケンスは次のようになります。

1. ターゲットホスト用に JET サーバーまたは Solaris ブートおよびインストールサーバーが用意されます。
2. OS プロビジョニングサーバー上の DHCP が、ターゲットホストに対して設定されます。
3. ターゲットホストが再起動され、DHCP を使用してネットワーク経由でブートします。
4. ターゲットホストは DHCP 検出パケットをブロードキャストします。
5. OS プロビジョニングサーバー上の DHCP サーバーが DHCP オファーを送信します。
6. ターゲットホストは DHCP 要求パケットをブロードキャストします。
7. OS プロビジョニングサーバー上の DHCP サーバーが DHCP ACK を送信します。
8. ターゲットは TFTP プロトコルを使用して、JET サーバーからブートカーネルを取得します。
9. ターゲットは、NFS 経由で JET サーバーからファイルを取得することで、OS をインストールします。

問題: Solaris イメージをインポートしている間にプランがタイムアウトする。

対処方法: マスターサーバーでプランのデフォルトのタイムアウトを設定します。次の手順を実行します。

1. 次の構成ファイルを編集します。

```
/opt/SUNWnlsp/N1_Service_Provisioning_System_5.1/server/config/config.properties
```

2. 次のプロパティを設定します。

```
pe.nonPlanExecNativeTimeout=12000
pe.defaultPlanTimeout=12000
```

3. マスターサーバーを再起動します。

```
# cr_server stop
# cr_server start
```

問題: Solaris イメージをインポートしている間にプランが失敗する。

対処方法: この問題を分析するには、以下の手順に従います。

1. プランの `stdout` および `stderr` メッセージをチェックします。
2. メディアを保持するのに十分なディスク容量があることを確認します。
3. 変数の値をチェックします。すべてのパスが正しく完全であることを確認します。

問題: `spsra` モジュールでの失敗を示してプロビジョンプランが失敗する。

対処方法: JET サーバーにインストールされている N1 SPS RA のスナップショットを使用するか、RA ディストリビューションを使用して、N1 SPS RA はターゲットにインストールされます。OS プロファイルの `install_ra_from_snapshot_spsra` 値をチェックします。ターゲットホストと JET サーバーのアーキテクチャーが異なる場合は、`install_ra_from_snapshot_spsra` を「n」に設定し、「Jet」コンポーネントを使用して N1 SPS RA を JET サーバーにインストールします。

問題: Solaris x86 をプロビジョニングする際、インストールが対話式になる。

対処方法: OS プロファイルのコンソール変数 `x86_console_base_config` が正しく構成されていることを確認します。v20z ターゲットの場合、これは `ttya` にする必要があります。ブートパーティションサイズが小さいことを示してインストールが失敗する場合、最も可能性の高い原因は、異なるディスクラベルフォーマットを使用する別の OS が以前インストールされていたことです。fdisk ユーティリティーを使用して、ディスクのパーティションを再設定します。

Linux 関連の問題の解決

問題: Linux オペレーティングシステムのプロビジョニング処理のシーケンスがわからない。

対処方法: Linux のシーケンスは次のようになります。

1. ターゲットホスト用に Linux ブートおよびインストールサーバーが用意されます。
2. OS プロビジョニングサーバー上の DHCP が、ターゲットホストに対して設定されます。
3. ターゲットホストが再起動され、DHCP を使用してネットワーク経由でブートします。
4. ターゲットホストは DHCP 検出パケットをブロードキャストします。
5. OS プロビジョニングサーバー上の DHCP サーバーが DHCP オファーを送信します。
6. ターゲットホストは DHCP 要求パケットをブロードキャストします。
7. OS プロビジョニングサーバー上の DHCP サーバーが DHCP ACK を送信します。
8. ターゲットは TFTP プロトコルを使用して、Linux ブートおよびインストールサーバーからブートカーネルを取得します。

9. ターゲットは、NFS 経由で Linux ブートおよびインストールサーバーからファイルを取得することで、OS をインストールします。

問題: インストールが開始するが、ディスクラベルを読み取ることができないというプロンプトがユーザーに表示される。

対処方法: 通常この問題は、Linux がアーキテクチャーのデフォルトとして認識しないディスクラベルフォーマットを使用する別の OS が以前にインストールされていたことを示しています。ユーザーにプロンプトを表示することなく、インストーラにディスクラベルをデフォルトアーキテクチャーに再初期化させるには、kickstart 構成ファイルの clearpart 指令に --initlabel オプションを追加します。

問題: インストールが DHCP を介して IP アドレスを取得できない。

対処方法: 次の解決策を試してください。

- ターゲットが接続されているスイッチで、スイッチポートを portfast に設定します。
- PXE 構成ファイルで linksleep をチェックします。
- kickstart プロパティファイルの InitialBootTimeout をより大きな値に変更します。

問題: ターゲットは DHCP パケットを取得するが、ブートに失敗する。

対処方法: 次の解決策を試してください。

- pxelinux.0 と initrd がターゲットプラットフォームをサポートしているかどうか、およびネットワーク経由でターゲットをブートするのに必要なドライバを持っているかどうかをチェックします。
- OS プロファイルと OS ディストリビューションが Linux ブートおよびインストールサーバーで正しく設定されていることを確認します。
- NFS および TFTP サービスが Linux ブートおよびインストールサーバーで正しく設定されていることを確認します。

問題: コンソールに次のメッセージが表示される。

```
VFS: mounted root (ext2) filesystem
```

対処方法: Linux カーネルは、コンソールを別の場所にリダイレクトしています。PXE 構成ファイルでコンソール設定を変更します。

問題: インストールが対話式モードになる。

対処方法: kickstart ファイルにエラーがないかチェックします。サーバーの IP アドレスとパスが正しく完全であることを確認します。

Windows 関連の問題の解決

問題: Windows オペレーティングシステムのプロビジョニングのシーケンスがわからない。

対処方法: Windows をプロビジョニングするシーケンスは次のようになります。

1. Windows ブートおよびインストールサーバーが、Active Directory でターゲットホストに対して事前準備されます。
2. OS プロビジョニングサーバー上の DHCP が、ターゲットホストに対して設定されます。
3. ターゲットホストが再起動され、DHCP を使用してネットワーク経由でブートします。
4. ターゲットホストは DHCP 検出パケットをブロードキャストします。
5. OS プロビジョニングサーバー上の DHCP サーバー、および Windows ブートおよびインストールサーバー上の BINL が DHCP オファーを送信します。
6. ターゲットは OS プロビジョニングサーバーからの DHCP オファーを選択し、DHCP 要求パケットをブロードキャストします。
7. OS プロビジョニングサーバー上の DHCP サーバー が DHCP ACK を送信します。
8. ターゲットは (PXE ブートサーバーに対して) 再度 DHCP 検出パケットをブロードキャストします。
9. Windows ブートおよびインストールサーバー上の BINL は (PXE に対して) DHCP オファーを送信します。
10. ターゲットは DHCP パケット内の次のサーバーの情報を使用し、TFTP を行なって、Windows ブートおよびインストールサーバーからブートカーネルを取得します。
11. ターゲットは、Windows ブートおよびインストールサーバーから CIFS 経由でファイルを取得することで、テキストモードインストールを行います。
12. ターゲットは再起動します。
13. この時点で DHCP サーバーはクリアされ、ターゲットホストに応答しないため、ターゲットはディスクからブートします。
14. ターゲットは GUI モードインストールを行います。
15. ターゲットは再起動し、SIF ファイルの GuiRunOnce セクションのスクリプトを実行します。

PXE/DHCP/BINLSVC に関連する問題

問題: PXE ROM のバージョンが正しいことを確認する方法は。

対処方法: NetPC またはクライアントコンピュータが ROM ブートを行う場合、PXE (LSA) ROM メッセージが画面に表示されます。クライアントマシンのブートシーケンス時には、表示される PXE ROM コードのバージョンを確認できます。Windows

2000 RIS は、.99c またはそれ以降の PXE ROM をサポートしています。この既存の ROM バージョンで成功しなかった場合、新しいバージョンの PXE ベースの ROM コードを OEM から入手しなければならない場合があります。

問題: クライアントコンピュータが IP アドレスを受信したかどうか、およびリモートインストールサーバーに接触したかどうかを確認する方法は。

対処方法: クライアントコンピュータがブートすると、PXE Boot ROM は読み込みと初期化を開始します。多くの Net PC または PXE ROM ベースのコンピュータでは、次の 4 つの手順のシーケンスが発生します。

注 - シーケンスは、使用中のコンピュータで異なる場合があります。

1. クライアントコンピュータはメッセージ BootP を表示します。このメッセージは、クライアントが DHCP サーバーから IP アドレスを要求していることを示しています。

障害追跡: BootP メッセージがクライアントに到着していなければ、クライアントは IP アドレスを受信していません。次の可能性をチェックしてください。

- DHCP サーバーは使用可能で、サービスが開始されていますか。DHCP および RIS サーバーは、サービスを開始するためには、Active Directory で認証されている必要があります。サービスが開始していて、そのほかの非遠隔、ブート対応クライアントが、このセグメントで IP アドレスを受信していることをチェックします。
 - 非遠隔、ブート対応クライアントなどのそのほかのクライアントコンピュータが、このネットワークセグメントで IP アドレスを受信できていますか。
 - DHCP サーバーでは IP アドレスの有効範囲が定義済みで、起動されていますか。この機能を確認するには「スタート」をクリックし、「プログラム」、「Administrative Tools」の順にポイントし、「DHCP」をクリックします。または、「スタート」をクリックし、「プログラム」、「Administrative Tools」の順にポイントし、「Event Viewer」をクリックする方法もあります。
 - 「System Log for DHCP」のイベントログにエラーメッセージがありますか。
 - クライアントと DHCP サーバーの間に、DHCP パケットの通過を許可しないルーターがありませんか。
2. クライアントが DHCP サーバーから IP アドレスを受信した場合、メッセージは DHCP に変化します。これは、クライアントが正しく IP アドレスをリースし、現在 RIS サーバーとの接触を待機していることを示します。

障害追跡: DHCP メッセージがクライアントに到着していなければ、クライアントは遠隔インストールサーバーから応答を受け取っていません。次の可能性をチェックしてください。

- 遠隔インストールサーバーが使用可能で、(BINLSVC) RIS サービスが開始されていますか。RIS サーバーは、サービスを開始するためには、Active Directory で認証されている必要があります。サービスが開始したことを確認するには、DHCP スナップイン(「スタート」をクリックし、「プログラム」、「Administrative Tools」の順にポイントして「DHCP」をクリック)を使用します。

- そのほかの遠隔ブート対応クライアントが Client Installation ウィザードを受信していますか。受信している場合、このクライアントコンピュータはサポートされていないか、リモートブート ROM 関連の問題がある可能性があります。クライアントコンピュータの PXE ROM のバージョンをチェックします。
- クライアントと遠隔インストールサーバーとの間に、DHCP ベースの要求または応答の通過を許可していないルーターがありませんか。RIS クライアントと RIS サーバーが別のサブネット上にある場合、2つのシステム間のルーターは、RIS サーバーに DHCP パケットを転送するよう構成する必要があります。これは、RIS クライアントは DHCP ブロードキャストメッセージを使用して RIS サーバーを検出するためです。ルーターで DHCP 転送が設定されていない場合、クライアントの DHCP ブロードキャストは RIS サーバーには到達しません。ルーターの構成マニュアルでは、この DHCP 転送プロセスは、DHCP プロキシや IP ヘルパーアドレスと呼ばれる場合もあります。

DHCP の設定を確認するには、「スタート」をクリックし、「プログラム」、「Administrative Tools」の順にポイントし、「Event Viewer」をクリックします。使用する固有のルーターでの DHCP 転送の設定については、そのルーターでの手順を参照してください。

- RIS (BINLSVC)、DNS、または Active Directory に固有の System または Application ログのイベントログに、エラーメッセージがありますか。

3. クライアントは BINL に変化するか、ユーザーに F12 キーを押すよう求めるプロンプトを表示します。このことは、クライアントは RIS サーバーにコンタクトし、最初のイメージファイル OSChooser の TFTP 転送を待機していることを意味します。BINL および TFTP メッセージが表示されない場合がありますが、これは一部のマシンではこのシーケンスがあまりにも早く行われるためです。(注: <reminst_share>\OSChooser\i386 フォルダで startrom.com と startrom.n12 ファイルを交換することにより、F12 キーを押す作業が自動化される。)

障害追跡: クライアントマシンが遠隔インストールサーバーから応答が得られない場合、クライアントはタイムアウトになり、DHCP、BINL、または TFTP のいずれかからファイルを受信しなかったというエラーを表示します。この場合、RIS サーバーはクライアントコンピュータには応答していません。BINLSVC を停止し、再起動します。「スタート」メニューから「ファイル名を指定して実行」をクリックし、「CMD」と入力します。次のコマンドを入力します。Net Stop BINLSVC Net Start BINLSVC

サービスを停止し再起動しようとしたあとにクライアントマシンが応答を受け取らなかった場合、遠隔インストールサーバーの Object プロパティを調べて、正しい設定が設定されていることを確認します。RIS が「Respond to client computers requesting service」および「Do not respond to unknown client computers」に設定されていることを確認します。「スタート」をクリックし、「プログラム」、「Administrative Tools」の順にポイントし、「Event Viewer」をクリックして、RIS サーバーの Event ログをチェックし、DHCP、DNS、または RIS (BINLSVC) に関連するエラーを調べます。

4. この時点でクライアントは、ユーザーに対する Welcome 画面のあいさつとともに、Client Installation ウィザードアプリケーションをダウンロードし、表示しているはずですが。

問題: PXE ベースの遠隔ブート ROM のブート前部分は、セキュリティー保護されているか。

対処方法: いいえ。ROM シーケンスおよび OS インストールまたは複製の全体は、パケットタイプの暗号化、クライアントまたはサーバースプーフィング、またはワイヤレスニッファーベースのメカニズムに関しては、セキュリティー保護されていません。そのため、企業ネットワークで RIS サービスを使用する際には注意してください。ネットワーク上で認証された RIS サーバーのみを許可し、また RIS サーバーのインストールと構成が許可された管理者の数を制御するようにしてください。

問題: ネットワークからのブート時に、ターゲットホストは次のエラーメッセージを表示する。

No proxyDHCP offers were received.

対処方法: クライアントマシンまたはターゲットホストは DHCP サーバーから IP アドレスを取得できません。詳細については、上記の手順 2 を参照してください。次の Microsoft サポート技術情報を参照してください。

- Your RIS client may not be able to obtain an IP address during the PXE restart on Windows Server 2003 and Windows 2000 (英語版)
- Using Dynamic Host Configuration Protocol Options 60, 66, 67 to Direct PXE Clients to RIS Servers May Fail (英語版)
- PXE Clients Do Not Receive IP Address From DHCP Server Across a Router (英語版)

問題: PXE クライアント、DHCP および RIS サーバーはどのようにやりとりを行うか。

対処方法: 次の Microsoft サポート技術情報を参照してください。Description of PXE Interaction Among PXE Client, DHCP, and RIS Server (英語版)

問題: ターゲットホストが、ネットワークからのブート時に次のメッセージを表示する。

ARP Timeout message

対処方法: このエラーメッセージが表示されるのは、クライアントマシンが DHCP サーバーから有効な IP アドレスを取得する場合でも、それが RIS サーバー上の BINL サービスからは無効な PXE ブートサーバーの IP アドレス (プロビジョニングサブネットの RIS サーバーの IP アドレス) である場合です。これは、マルチホーム RIS サーバーとして実行されている場合に、HP-Lpr などの一部の古いマシンで見られます。ただし、マルチホーム RIS サーバーとして構成されている場合であっても、同じベンダーの Hp-Proliant DL 360 G3 シリーズサーバーマシンなどの新しいハードウェアでは、この問題は生じません。エラーメッセージを表示することなく、HP-Lpr などの古いマシンを RIS サーバーとして動作させるには、マシンがマルチホームでないことを確認します。つまりシステムは、プロビジョニングサブネットに存在するインタフェースを 1 つだけ有効にする必要があります。

詳細については、次の Microsoft サポート技術情報を参照してください。A multi-homed RIS server may not answer all clients, and you may receive an error message on PXE clients that are running Windows Server 2003 or Windows 2000 (英語版)

問題: テキストモードのインストールがブートしない。

対処方法: 次の解決策を試してください。

- RIS イメージディストリビューションの設定をチェックします。
- イメージにバンドルされたネットワークアダプタおよび外部記憶装置ドライバに見当たらないものがあるかどうかチェックします。
- ドライバパスの RIS 設定情報ファイルをチェックします。
- <reminst share>/OSChooser/English パスの CIW ファイル、welcome.osc/login.osc/oschoice.osc/install.osc/warning.osc で、無効なユーザー情報や見当たらない AUTOENTER 機能を調べます。
- ris プロパティの InitialBootTimeout をより大きな値に変更します。ドライバ関連の問題の詳細については、RIS ドライバまたはサービスパックの節を参照してください。GUID 関連の問題の詳細については、GUID の節を参照してください。

問題: GUI モードのインストールが対話式モードになる。

対処方法: 次の解決策を試してください。

- Unattended、UserData、GuiRunOnce、および GuiUnattended の各セクションで、データに関する RIS 設定情報ファイルをチェックします。
- sif ファイルでデフォルトの管理者のパスワードと製品キーを指定していることを確認します。CIW および ris SIF ファイルの詳細については、RIS CIW または SIF ファイルの節を参照してください。

問題: 各クライアントで、テキストモードインストールと GUI モードインストールのデフォルトのタイムアウト値を変更する方法は。

対処方法: プロビジョニング操作を開始する前に、N1 OS プロビジョニングサーバーで、ris.properties ファイル (通常は <nlosp folder>/etc/ フォルダの下に存在) の次のプロパティのデフォルトのタイムアウト値を変更したことを確認します。

```
ris.InitialBootTimeout  
ris.OsInstallTimeout
```

Remote Installation Service (RIS) に関連する問題

問題: 遠隔インストールサーバーのデバッグモードを有効にする方法は。

対処方法: Microsoft サポート技術情報 236033 (英語版) で説明されている手順に従ってください。

問題: RIS サービスの CIW 画面を自動化する方法は。

対処方法: 次の Microsoft サポート技術情報を参照してください。

- Description of Client Installation Wizard Screens for Remote Installation Services - 268325 (英語版)
- HOW TO: Automate CIW Screens using AUTOENTER - 824184 (英語版)

問題: Setup Information Answer ファイル (.sif ファイル) に関する詳細情報の入手先は。

対処方法: 詳細は、Windows 2000 または 2003 Server Resource Kit CD の deploy.cab ファイルを参照してください。

問題: RIS のインストール中に管理者のパスワードを変更する方法は。

対処方法: 次の Microsoft サポート技術情報を参照してください。How to Set the Administrator Password During RIS Installation - 257948 (英語版)

問題: RIS イメージにドライバを追加する方法は。

対処方法: 次の Microsoft サポート技術情報を参照してください。

RIS インストールにサードパーティの OEM ネットワーク アダプタを追加する方法 - 246184

Windows インストールに OEM プラグ アンド プレイ ドライバを追加する方法 - 254078

問題: RIS イメージにサービスパックをスリップストリームする方法は。

対処方法: Microsoft の Web サイトの次の記事を参照してください。

- 最新の Windows 2000 Service Pack の入手方法
- Windows 2000 Service Pack の Update.exe のスリップストリーム スイッチが RIS サーバー イメージでは機能しない
- サービスパックをバンドルして RIS イメージを作成する方法の詳細は、『Service Pack 3 インストールと導入ガイド』を参照してください。そのほかのサービスパックに関する同じようなガイドは、Windows 2000 - ホーム - ダウンロード - Service Pack にもあります。
- Windows 2000 Service Pack 3 のダウンロードそのほかのサービスパックに関しても、同じようなダウンロードページがあります。

注 - サービスパックをダウンロードして RIS イメージを作成するには、「ネットワークインストール」を選択してください。

問題: テキストモードインストール時に次のエラーメッセージが表示される。

Illegal or Missing File Types Specified in Section SCSI.Name

対処方法: Microsoft サポート技術情報 275334 を参照してください。

問題: RIS イメージをインストールする場合、テキストモードインストール時にエラーメッセージが表示される。エラーメッセージには次の文が含まれます。

Setup Cannot Continue

対処方法: Microsoft サポート技術情報 830751 を参照してください。

問題: テキストモードインストール時に次のエラーメッセージが表示される。

INF File Tmp\<<GUID_number.sif> Is Corrupt or Missing

対処方法: Microsoft サポート技術情報 224830 (英語版) を参照してください。

問題: テキストモードインストール時に次のエラーメッセージが表示される。

The Operating System Image You Selected Does Not Contain the Necessary Drivers

対処方法: Microsoft サポート技術情報 247983 (英語版) を参照してください。

問題: テキストモードインストール時に次のエラーメッセージが表示される。

The Operating System Image You Selected Does Not Contain the Necessary Drivers for Your Network Adapter. Try Selecting a Different Operating System Image. If the Problem Persists, Contact Your System Administrator.

対処方法: Microsoft サポート技術情報 315074 を参照してください。

問題: テキストモード設定の最後に遠隔インストールクライアントがハングする。

対処方法: Microsoft サポート技術情報 226941 (英語版) を参照してください。

問題: RIS 設定が、「Setup is Starting Windows」画面に応答しなくなる。

対処方法: Microsoft サポート技術情報 320865 (英語版) を参照してください。

GUID に関連する問題

問題: Active Directory 内で RIS に使用する事前準備クライアントの GUID または UUID を見つけるには、クライアントコンピュータのどこを探せばいいか。

対処方法: PC98 または Net PC 準拠のクライアントコンピュータの GUID または UUID は、(多くの場合) システム BIOS にあります。OEM には、GUID または UUID に対するシリアル番号のマッピングを含むコンマ区切りファイルまたはスプレッドシートを含むフロッピーディスクを出荷することが推奨されています。これにより、Active Directory 内でクライアントコンピュータの事前準備のスクリプトを作成できるようになります。また OEM には、コンピュータアカウントの識別と事前準備を簡単にするため、コンピュータケースの外側に GUID または UUID を貼り付けることが推奨されています。GUID が上記の場所に見つからない場合、クライアントのネットワークトラフィックをスニッフして、DHCP 検出パケットを特定することができます。DHCP 検出パケット内では 128 ビット、32 バイトの GUID または UUID を見つけることができます。

問題: 2 台のクライアントマシンが同じ GUID 値を持っている。

対処方法: この場合、RIS は各ターゲットホストを Active Directory 内で一意の GUID 値を持つコンピュータオブジェクトとして識別するため、RIS が失敗します。複数のオブジェクトが同じ GUID を持っている場合、RIS クライアントマシンは設定段階の時点でエラーをスローします。次のメッセージが表示されます。

```
BINLSVC found Duplicate GUID accounts on the RIS Server.  
Please contact your system Administrator.
```

この問題を解決するには、継続する前に、RIS サーバーの Active Directory で同じ GUID を持つ古いコンピュータアカウントを削除します。

付録 A

ターゲットホストの変数

各タイプのターゲットホストは、多数の特別な構成変数を持っています。この付録には、各ターゲットホストタイプに適用される変数を説明する一連の表が含まれています。

Sun IPMI ターゲットホストの変数

次のリストに、Sun IPMI ターゲットホストのすべてのコンポーネント変数を示します。デフォルト値を持つ変数に関しては、デフォルトが示されています。デフォルト値を持たない変数に関しては、値の例が示されています。

`installPath`

説明: ターゲットシステムの物理ホスト名。

例: `sunfire-v60x-host`

`target_host`

説明: ターゲットシステムの仮想ホスト名。

デフォルト値: `:[installPath]-target`

`description`

説明: (省略可能) ターゲットの説明。

例: `test v60x`

`ethernet_mac_address`

説明: Mac アドレス。

例: `00:0E:0C:07:E9:31`

`ethernet_ip_address`

説明: IP アドレス。

例: 10.42.42.77

`ethernet_netmask`
説明: ネットワークマスク。

デフォルト値: 255.255.255.0

`architecture`
説明: ターゲットのアーキテクチャー (sun4u、x86 など)。

デフォルト値: x86

`kernel_arch`
説明: カーネルのアーキテクチャー (sun4u、i86pc)。

デフォルト値: i86pc

`disk_size`
説明: ディスクのサイズ。

例: 20GB

`osp_control_service`
説明: 電源オフ、再起動、および電源オン機能を自動化する OSP Control Service を呼び出すかどうか (True、False)。

デフォルト値: TRUE

`sps_ra_parent`
説明: リモートエージェント (RA) の親の名前。

デフォルト値: masterserver

`sps_ra_parent_type`
説明: RA の親が MS (マスターサーバー) と LD (ローカルディストリビュータ) のいずれであるか。

デフォルト値: MS

`sps_ra_host`
説明: RA ホストの名前または IP アドレス。

デフォルト値: : [installPath]

`sps_ra_conn`
説明: RA 接続のタイプ (raw、ssl、ssh)。

デフォルト値: raw

`sps_ra_ssl_auth`
説明: RA が SSL 認証を使用するかどうか (y、n)。

デフォルト値: N

`sps_ra_port`
説明: RA のポート。

デフォルト値: 1131

sps_ra_adv_parms
説明: RA の高度なパラメータ。

guid
説明: {} 内で指定されるターゲットの GUID

注 - GUID は、{ddddddd- dddd-dddd-dddd- ddddddddddd} の形式である必要があります。ここで *d* は 16 進数文字です。

ipmi_ip_address
説明: IPMI の IP アドレス

例: 10.5.133.135

ipmi_access_userid
説明: IPMI にアクセスするためのユーザー ID

ipmi_access_password
説明: IPMI にアクセスするための暗号化されたパスワード

ssh_access_userid
説明: 遠隔管理インタフェースにアクセスするための SSH ユーザー ID。

ssh_access_password
説明: 遠隔管理インタフェースにアクセスするための暗号化された SSH パスワード。

Sun ALOM ターゲットホストの変数

次のリストに、Sun ALOM ターゲットホストのすべてのコンポーネント変数を示します。デフォルト値を持つ変数に関しては、デフォルトが示されています。デフォルト値を持たない変数に関しては、値の例が示されています。

installPath
説明: ターゲットシステムの物理ホスト名

例: sunfire-v60x-host

target_host
説明: ターゲットシステムの仮想ホスト名

デフォルト値: : [installPath] -target

description
説明: (省略可能) ターゲットの説明

例 : test v60x

ethernet_mac_address
説明 : Mac アドレス

例 : 00:0E:0C:07:E9:31

ethernet_ip_address
説明 : IP アドレス

例 : 10.42.42.77

ethernet_netmask
説明 : ネットワークマスク

デフォルト値 : 255.255.255.0

architecture
説明 : ターゲットのアーキテクチャー (sun4u、x86 など)

デフォルト値 : sun4u

kernel_arch
説明 : カーネルのアーキテクチャー (sun4u、i86pc)

デフォルト値 : sun4u

disk_size
説明 : ディスクのサイズ

デフォルト値 : 20GB

osp_control_service
説明 : 電源オフ、再起動、および電源オン機能を自動化する OSP Control Service を呼び出すかどうか (True、False)

デフォルト値 : TRUE

sps_ra_parent
説明 : RA の親の名前

デフォルト値 : masterserver

sps_ra_parent_type
説明 : RA の親が MS と LD のいずれであるか

デフォルト値 : MS

sps_ra_host
説明 : RA ホストの名前または IP アドレス

デフォルト値 : : [installPath]

sps_ra_conn
説明 : RA 接続のタイプ (raw、ssl、ssh)

デフォルト値 : raw

sps_ra_ssl_auth
説明 : RA が SSL 認証を使用するかどうか (y, n)

デフォルト値 : N

sps_ra_port
説明 : RA のポート

デフォルト値 : 1131

sps_ra_adv_parms
説明 : RA の高度なパラメータ

alom_ip_address
説明 : ALOM の IP アドレス

例 : 10.5.133.135

alom_access_userid
説明 : ALOM にアクセスするためのユーザー ID

alom_access_password
説明 : ALOM にアクセスするための暗号化されたパスワード

terminal_server_ip_address
説明 : このシステムに接続された端末サーバーの IP アドレス

例 : 10.5.133.135

terminal_server_port
説明 : 端末サーバーのポート

terminal_server_userid
説明 : 端末サーバーのユーザー ID

terminal_server_password
説明 : 端末サーバーの暗号化されたパスワード

terminal_server_type
説明 : 端末サーバーのタイプ

デフォルト値 : Cisco2621

terminal_server_version
説明 : 端末サーバーのバージョン

terminal_server_hal_jar
説明 : 端末サーバーの Hardware Abstraction Layer (HAL) JAR ファイルへのパス

デフォルト値 : file:///opt/nlosp/drivers/cisco/c2621.jar

Sun LOM ターゲットホストの変数

次のリストに、Sun LOM ターゲットホストのすべてのコンポーネント変数を示します。デフォルト値を持つ変数に関しては、デフォルトが示されています。デフォルト値を持たない変数に関しては、値の例が示されています。

`installPath`

説明: ターゲットシステムの物理ホスト名

例: `sunfire-v60x-host`

`target_host`

説明: ターゲットシステムの仮想ホスト名

デフォルト値: `[installPath]-target`

`description`

説明: (省略可能) ターゲットの説明

例: `test v60x`

`ethernet_mac_address`

説明: Mac アドレス

例: `00:0E:0C:07:E9:31`

`ethernet_ip_address`

説明: IP アドレス

例: `10.42.42.77`

`ethernet_netmask`

説明: ネットワークマスク

デフォルト値: `255.255.255.0`

`architecture`

説明: ターゲットのアーキテクチャー (`sun4u`、`x86` など)

デフォルト値: `sun4u`

`kernel_arch`

説明: カーネルのアーキテクチャー (`sun4u`、`i86pc`)

デフォルト値: `sun4u`

`disk_size`

説明: ディスクのサイズ

デフォルト値: `20GB`

osp_control_service
説明:電源オフ、再起動、および電源オン機能を自動化する OSP Control Service
を呼び出すかどうか (True、False)

デフォルト値: TRUE

sps_ra_parent
説明:RA の親の名前

デフォルト値: masterserver

sps_ra_parent_type
説明:RA の親が MS と LD のいずれであるか

デフォルト値: MS

sps_ra_host
説明:RA ホストの名前または IP アドレス

sps_ra_conn
説明:RA 接続のタイプ (raw、ssl、ssh)

デフォルト値: raw

sps_ra_ssl_auth
説明:RA が SSL 認証を使用するかどうか (y、n)

デフォルト値: N

sps_ra_port
説明:RA のポート

デフォルト値: 1131

sps_ra_adv_parms
説明:RA の高度なパラメータ

disk_type
説明:SCSI または IDE

デフォルト値: scsi

lom_version
説明:LOM のバージョン

lom_hal_jar
説明:LOM HAL JAR ファイルへのパス

デフォルト値:
file:///:[targetableComponent:installPath]/SUNWnlosp/drivers
/sun/lom.jar

terminal_server_ip_address
説明:このシステムに接続された端末サーバーの IP アドレス

例: 10.5.133.135

terminal_server_port
説明: 端末サーバーのポート

terminal_server_userid
説明: 端末サーバーのユーザー ID

terminal_server_password
説明: 端末サーバーのパスワードの暗号化された値

terminal_server_type
説明: 端末サーバーのタイプ

デフォルト値: Cisco2621

terminal_server_version
説明: 端末サーバーのバージョン

terminal_server_hal_jar
説明: 端末サーバーの HALJAR ファイルへのパス

デフォルト値:
file:///:[targetableComponent:installPath]/SUNWnlosp/drivers/
cisco/c2621.jar

power_service_true_false
説明: ネットワーク管理インタフェースが電源オン/オフのサービスを提供するかどうか

デフォルト値: true

boot_service_true_false
説明: ネットワーク管理インタフェースがブートサービスを提供するかどうか
(ブート順: ディスク/ネットワーク)

デフォルト値: true

console_service_true_false
説明: ネットワーク管理インタフェースがコンソールを提供するかどうか

デフォルト値: false

Sun Mid-Range ターゲットホストの変数

次のリストに、Sun Mid-Range ターゲットホストのすべてのコンポーネント変数を示します。SunMidRange コンポーネントは次の変数を使用します。デフォルト値を持つ変数に関しては、デフォルトが示されています。デフォルト値を持たない変数に関しては、値の例が示されています。

installPath
説明: ターゲットシステムの物理ホスト名

例: sunfire-v60x-host

target_host
説明: ターゲットシステムの仮想ホスト名
デフォルト値: [installPath]-target

description
説明: (省略可能) ターゲットの説明
例: test v60x

ethernet_mac_address
説明: Mac アドレス
例: 00:0E:0C:07:E9:31

ethernet_ip_address
説明: IP アドレス
例: 10.42.42.77

ethernet_netmask
説明: ネットワークマスク
デフォルト値: 255.255.255.0

architecture
説明: ターゲットのアーキテクチャー (sun4u、x86 など)
デフォルト値: sun4u

kernel_arch
説明: カーネルのアーキテクチャー (sun4u、i86pc)
デフォルト値: sun4u

disk_size
説明: ディスクのサイズ
デフォルト値: 20GB

osp_control_service
説明: 電源オフ、再起動、および電源オン機能を自動化する OSP Control Service を呼び出すかどうか (True、False)
デフォルト値: TRUE

sps_ra_parent
説明: RA の親の名前
デフォルト値: RA の親の名前

sps_ra_parent_type
説明: RA の親が MS と LD のいずれであるか

デフォルト値 : MS

sps_ra_host
説明 : RA ホストの名前または IP アドレス

sps_ra_conn
説明 : RA 接続のタイプ (raw、ssl、ssh)

デフォルト値 : raw

sps_ra_ssl_auth
説明 : RA が SSL 認証を使用するかどうか (y、n)

デフォルト値 : N

sps_ra_port
説明 : RA のポート

デフォルト値 : 1131

sps_ra_adv_parms
説明 : RA の高度なパラメータ

domain_name
説明 : ドメイン

domainsc_access_userid
説明 : ドメイン SC にアクセスするためのユーザー ID

domainsc_access_password
説明 : ドメイン SC にアクセスするための暗号化されたパスワード

platformsc_ip_address
説明 : プラットフォーム SC の IP アドレス

platformsc_access_userid
説明 : プラットフォーム SC にアクセスするためのユーザー ID

platformsc_access_password
説明 : プラットフォーム SC にアクセスするための暗号化されたパスワード

HP ILO ターゲットホストの変数

次のリストに、HP ILO ターゲットホストのすべてのコンポーネント変数を示します。デフォルト値を持つ変数に関しては、デフォルトが示されています。デフォルト値を持たない変数に関しては、値の例が示されています。

installPath
説明 : ターゲットシステムの物理ホスト名

例 : sunfire-v60x-host

target_host
説明: ターゲットシステムの仮想ホスト名
デフォルト値: [installPath]-target

description
説明: (省略可能) ターゲットの説明
例: test v60x

ethernet_mac_address
説明: Mac アドレス
例: 00:0E:0C:07:E9:31

ethernet_ip_address
説明: IP アドレス
例: 10.42.42.77

ethernet_netmask
説明: ネットワークマスク
デフォルト値: 255.255.255.0

architecture
説明: ターゲットのアーキテクチャー (sun4u、x86 など)
デフォルト値: x86

kernel_arch
説明: カーネルのアーキテクチャー (sun4u、i86pc)
デフォルト値: i86pc

disk_size
説明: ディスクのサイズ
デフォルト値: 20GB

osp_control_service
説明: 電源オフ、再起動、および電源オン機能を自動化する OSP Control Service を呼び出すかどうか (True、False)
デフォルト値: TRUE

sps_ra_parent
説明: RA の親の名前
デフォルト値: masterserver

sps_ra_parent_type
説明: RA の親が MS と LD のいずれであるか
デフォルト値: MS

sps_ra_host
説明: RA ホストの名前または IP アドレス
デフォルト値: [installPath]

sps_ra_conn
説明: RA 接続のタイプ (raw、ssl、ssh)
デフォルト値: raw

sps_ra_ssl_auth
説明: RA が SSL 認証を使用するかどうか (y、n)
デフォルト値: N

sps_ra_port
説明: RA のポート
デフォルト値: 1131

sps_ra_adv_parms
説明: RA の高度なパラメータ

guid
説明: {} 内で指定されるターゲットの GUID

注 - GUID は、{ddddddd- dddd-dddd-dddd- ddddddddd} の形式である必要があります。ここで *d* は 16 進数文字です。

例: {3AE45192-159D-11D8-AEF3-001083FDFE8A}

ilo_ip_address
説明: ILO の IP アドレス
例: 10.5.133.135

ilo_version
説明: ILO のバージョン
デフォルト値: 1.6.4

ilo_access_userid
説明: ILO にアクセスするためのユーザー ID

ilo_access_password
説明: ILO にアクセスするための暗号化されたパスワード

そのほかの IPMI ターゲットホストの変数

次のリストに、そのほかの IPMI ターゲットホストのすべてのコンポーネント変数を示します。デフォルト値を持つ変数に関しては、デフォルトが示されています。デフォルト値を持たない変数に関しては、値の例が示されています。

`installPath`

説明: ターゲットシステムの物理ホスト名

例: `sunfire-v60x-host`

`target_host`

説明: ターゲットシステムの仮想ホスト名

デフォルト値: `:[installPath]-target`

`description`

説明: (省略可能) ターゲットの説明

例: `test v60x`

`ethernet_mac_address`

説明: Mac アドレス

例: `00:0E:0C:07:E9:31`

`ethernet_ip_address`

説明: IP アドレス

例: `10.42.42.77`

`ethernet_netmask`

説明: ネットワークマスク

デフォルト値: `255.255.255.0`

`architecture`

説明: ターゲットのアーキテクチャー (`sun4u`、`x86` など)

デフォルト値: `x86`

`kernel_arch`

説明: カーネルのアーキテクチャー (`sun4u`、`i86pc`)

デフォルト値: `i86pc`

`disk_size`

説明: ディスクのサイズ

デフォルト値: `20GB`

osp_control_service
説明:電源オフ、再起動、および電源オン機能を自動化する OSP Control Service
を呼び出すかどうか (True、False)

デフォルト値: TRUE

sps_ra_parent
説明: RA の親の名前

デフォルト値: masterserver

sps_ra_parent_type
説明: RA の親が MS と LD のいずれであるか

デフォルト値: MS

sps_ra_host
説明: RA ホストの名前または IP アドレス

デフォルト値: : [installPath]

sps_ra_conn
説明: RA 接続のタイプ (raw、ssl、ssh)

デフォルト値: raw

sps_ra_ssl_auth
説明: RA が SSL 認証を使用するかどうか (y、n)

デフォルト値: N

sps_ra_port
説明: RA のポート

デフォルト値: 1131

sps_ra_adv_parms
説明: RA の高度なパラメータ

guid
説明: {} 内で指定されるターゲットの GUID

注 - GUID は、{ddddddd- dddd-dddd-dddd- ddddddddddd} の形式である必要があります。ここで *d* は 16 進数文字です。

例: {3AE45192-159D-11D8-AEF3-001083FDFE8A}

disk_type
説明: ディスクタイプ (SCSI、IDE など)

デフォルト値: scsi

`ipmi_ip_address`
説明: IPMI の IP アドレス

例: 10.5.133.135

`ipmi_access_userid`
説明: IPMI にアクセスするためのユーザー ID

`ipmi_access_password`
説明: IPMI にアクセスするための暗号化されたパスワード

`ipmi_hal_jar`
説明: IPMI HAL JAR ファイルへのパス

デフォルト値:
file:///:[targetableComponent:installPath]/SUNWnlosp/drivers/sun/ipmi.jarf

`ipmi_version`
説明: IPMI のバージョン

デフォルト値: 1.5

`power_service_true_false`
説明: ネットワーク管理インタフェースが電源オン/オフのサービスを提供するかどうか

デフォルト値: true

`boot_service_true_false`
説明: ネットワーク管理インタフェースがブートサービスを提供するかどうか (ディスクに続いてネットワーク)

デフォルト値: false

`console_service_true_false`
説明: ネットワーク管理インタフェースがコンソールサービスを提供するかどうか

デフォルト値: false

そのほかの遠隔管理ターゲットホストの変数

次のリストに、そのほかの遠隔管理ターゲットホストのすべてのコンポーネント変数を示します。Other コンポーネントタイプは次の変数を使用します。ほかのカテゴリに適合しない遠隔管理ターゲットには Other タイプを使用します。デフォルト値を持つ変数に関しては、デフォルトが示されています。デフォルト値を持たない変数に関しては、値の例が示されています。

注 - Other コンポーネントタイプには変数の長いリストがあり、変数の一部は特定のハードウェアタイプには適用されません。変数が適用されない場合は、空白のままにしておきます。

installPath

説明: ターゲットシステムの物理ホスト名

例: sunfire-v60x-host

target_host

説明: ターゲットシステムの仮想ホスト名

デフォルト値: [installPath]-target

description

説明: (省略可能) ターゲットの説明

例: test v60x

ethernet_mac_address

説明: Mac アドレス

例: 00:0E:0C:07:E9:31

ethernet_ip_address

説明: IP アドレス

例: 10.42.42.77

ethernet_netmask

説明: ネットワークマスク

デフォルト値: 255.255.255.0

architecture

説明: ターゲットのアーキテクチャー (sun4u、x86 など)

デフォルト値: x86

kernel_arch

説明: カーネルのアーキテクチャー (sun4u、i86pc)

デフォルト値: i86pc

disk_size

説明: ディスクのサイズ

デフォルト値: 20GB

osp_control_service

説明: 電源オフ、再起動、および電源オン機能を自動化する OSP Control Service を呼び出すかどうか (True、False)

デフォルト値: TRUE

sps_ra_parent
説明: RA の親の名前

デフォルト値: masterserver

sps_ra_parent_type
説明: RA の親が MS と LD のいずれであるか

デフォルト値: MS

sps_ra_host
説明: RA ホストの名前または IP アドレス

デフォルト値: : [ethernet_ip_address]

sps_ra_conn
説明: RA 接続のタイプ (raw、ssl、ssh)

デフォルト値: RAW

sps_ra_ssl_auth
説明: RA が SSL 認証を使用するかどうか (y、n)

デフォルト値: N

sps_ra_port
説明: RA のポート

デフォルト値: 1131

sps_ra_adv_parms
説明: RA の高度なパラメータ

guid
説明: {} 内で指定されるターゲットの GUID

注 - GUID は、{ddddddd- dddd-dddd-dddd- ddddddddd} の形式である必要があります。ここで *d* は 16 進数文字です。

例: {3AE45192-159D-11D8-AEF3-001083FDFE8A}

disk_type
説明: ディスクタイプ (SCSI、IDE など)

デフォルト値: scsi

netmgmt_type_1
説明: ネットワーク管理インタフェースのタイプ (ipmi/alom/lom)

netmgmt_version_1
説明: ネットワーク管理のバージョン

netmgmt_hal_jar_1
説明: ネットワーク管理 HAL JAR ファイルへのパス

デフォルト値:
file:///:[targetableComponent:installPath]/SUNWnlosp/drivers/sun/ipmi.jar

netmgmt_ip_address_1
説明: ネットワーク管理 IP アドレス

netmgmt_protocol_1
説明: ネットワーク管理プロトコル (ipmi/telnet)

デフォルト値: telnet

netmgmt_port_1
説明: ネットワーク管理ポート

power_service_true_false_1
説明: ネットワーク管理インタフェースが電源オン/オフのサービスを提供するかどうか

デフォルト値: true

boot_service_true_false_1
説明: ネットワーク管理インタフェースがブートサービスを提供するかどうか (ディスクに続いてネットワーク)

デフォルト値: true

console_service_true_false_1
説明: ネットワーク管理インタフェースがコンソールサービスを提供するかどうか

デフォルト値: false

netmgmt_userid_1
説明: ネットワーク管理インタフェースにアクセスするためのユーザー ID

netmgmt_password_1
説明: ネットワーク管理インタフェースにアクセスするための暗号化されたパスワード

terminal_server_type_1
説明: 端末サーバーのタイプ

デフォルト値: Cisco2621

terminal_server_version_1
説明: 端末サーバーのバージョン

terminal_server_hal_jar_1
説明: 端末サーバーの HAL JAR ファイルへのパス

デフォルト値:
file:///:[targetableComponent:installPath]/SUNWnlosp/drivers/cisco/c2621.jar

`terminal_server_ip_address_1`
説明: このシステムに接続された端末サーバーの IP アドレス

例: 10.5.133.135

`terminal_server_protocol_1`
説明: 端末サーバーのアクセスプロトコル

デフォルト値: telnet

`terminal_server_port_1`
説明: 端末サーバーのポート

`terminal_server_userid_1`
説明: 端末サーバーのユーザー ID

`terminal_server_password_1`
説明: 端末サーバーの暗号化されたパスワード

`netmgmt_type_2`
説明: ネットワーク管理インタフェースのタイプ (ipmi/alom/lom)

デフォルト値: ipmi

`netmgmt_version_2`
説明: ネットワーク管理のバージョン

`netmgmt_hal_jar_2`
説明: ネットワーク管理 HAL JAR ファイルへのパス

デフォルト値:
file:///:[targetableComponent:installPath]/SUNWnlosp/drivers
/sun/ipmi.jarf

`netmgmt_ip_address_2`
説明: ネットワーク管理 IP アドレス

`netmgmt_protocol_2`
説明: ネットワーク管理プロトコル (ipmi/telnet)

デフォルト値: telnet

`netmgmt_port_2`
説明: ネットワーク管理ポート

`power_service_true_false_2`
説明: ネットワーク管理インタフェースが電源オン/オフのサービスを提供するかどうか

デフォルト値: false

`boot_service_true_false_2`
説明: ネットワーク管理インタフェースがブートサービスを提供するかどうか
(ディスクに続いてネットワーク)

デフォルト値: false

`console_service_true_false_2`
説明: ネットワーク管理インタフェースがコンソールサービスを提供するかどうか
デフォルト値: `false`

`netmgmt_userid_2`
説明: ネットワーク管理インタフェースにアクセスするためのユーザー ID

`netmgmt_password_2`
説明: ネットワーク管理インタフェースにアクセスするための暗号化されたパスワード

`terminal_server_type_2`
説明: 端末サーバーのタイプ
デフォルト値: `Cisco2621`

`terminal_server_version_2`
説明: 端末サーバーのバージョン

`terminal_server_hal_jar_2`
説明: 端末サーバーの HALJAR ファイルへのパス
デフォルト値:
`file:///:[targetableComponent:installPath]/SUNWnlosp/drivers/cisco/c2621.jar`

`terminal_server_ip_address_2`
説明: このシステムに接続された端末サーバーの IP アドレス
例: `10.5.133.135`

`terminal_server_protocol_2`
説明: 端末サーバーのアクセスプロトコル

`terminal_server_port_2`
説明: 端末サーバーのポート

`terminal_server_userid_2`
説明: 端末サーバーのユーザー ID

`terminal_server_password_2`
説明: 端末サーバーの暗号化されたパスワード

汎用ターゲットホストの変数

次のリストに、汎用ターゲットホストのすべてのコンポーネント変数を示します。Target タイプは次の変数を使用します。そのほかのターゲットホストタイプに一致しないホストには、Target タイプを使用します。デフォルト値を持つ変数に関しては、デフォルトが示されています。デフォルト値を持たない変数に関しては、値の例が示されています。

installPath
説明: ターゲットシステムの物理ホスト名

target_host
説明: ターゲットシステムの仮想ホスト名
デフォルト値: : [installPath] -target

description
説明: (省略可能) ターゲットの説明
例: test v60x

ethernet_mac_address
説明: Mac アドレス
例: 00:0E:0C:07:E9:31

ethernet_ip_address
説明: IP アドレス
例: 10.42.42.77

ethernet_netmask
説明: ネットワークマスク
デフォルト値: 255.255.255.0

architecture
説明: ターゲットのアーキテクチャー (sun4u, x86 など)
デフォルト値: sun4u

kernel_arch
説明: カーネルのアーキテクチャー (sun4u, i86pc)
デフォルト値: sun4u

disk_size
説明: ディスクのサイズ
デフォルト値: 20GB

osp_control_service
説明: 電源オフ、再起動、および電源オン機能を自動化する OSP Control Service を呼び出すかどうか
デフォルト値: TRUE

sps_ra_parent
説明: RA の親の名前
デフォルト値: masterserver

sps_ra_parent_type
説明: RA の親が MS と LD のいずれであるか

デフォルト値 : MS

sps_ra_host

説明 : RA ホストの名前または IP アドレス

デフォルト値 : : [ethernet_ip_address]

sps_ra_conn

説明 : RA 接続のタイプ (raw、ssl、ssh)

デフォルト値 : raw

sps_ra_ssl_auth

説明 : RA が SSL 認証を使用するかどうか (y、n)

デフォルト値 : N

sps_ra_port

説明 : RA のポート

デフォルト値 : 1131

sps_ra_adv_parms

説明 : RA の高度なパラメータ

guid

説明 : {} 内で指定されるターゲットの GUID

注 - GUID は、{ddddddd- dddd-ddd-dddd- ddddddddd} の形式である必要があります。ここで *d* は 16 進数文字です。

Solaris Profile コンポーネントの変数

Solaris Profile コンポーネントには各 JET モジュールの変数が含まれています。これらの変数は、次のカテゴリに分類されています。

- 143 ページの「基本 OS 構成変数」
- 154 ページの「N1 SPS リモートエージェントのコンポーネント変数」
- 156 ページの「一般的な OS プロビジョニングプラグインの変数」

基本 OS 構成変数

base_config モジュール用のコンポーネント変数

次のリストで説明している変数は、基本オペレーティングシステム構成を提供します。これらの変数は、JET base_config モジュールに直接対応します。リストには、ターゲット設定可能なコンポーネントの値を使用するいくつかの変数があります。Solaris Profile 変数の目的では、ターゲット設定可能なコンポーネントはターゲットホストです。

ClientArch_base_config

説明: sun4u や x86 などのカーネルアーキテクチャー。デフォルトでは、これはターゲット設定可能なコンポーネントのカーネルアーキテクチャーに設定されています。

デフォルト値: : [targetableComponent:kernel_arch]

ClientEther_base_config

説明: Ethernet MAC アドレス。デフォルトでは、これはターゲット設定可能なコンポーネントの Ethernet MAC アドレスに設定されています。

デフォルト値: : [targetableComponent:ethernet_mac_address]

ClientOS_base_config

説明: プロビジョニングする OS のバージョン。

例: Solaris9_u7_sparc

client_allocation_base_config

説明: このクライアントの構築に使用されるメカニズム。デフォルトでは、`/opt/jet/etc/jumpstart.conf` にあるオプションが使用されます。この特定のクライアントのデフォルトとは異なる動作が必要な場合を除き、この値は空白のままにします。

products_base_config

説明: プロビジョニングする JET モジュール。

例: base_config spsra

sysidcfg_nameservice_base_config

説明: 初期ブート時に構成するサービス名。

デフォルト値: NIS

sysidcfg_network_interface_base_

説明: 初期ブート時に使用するネットワークインタフェース。

デフォルト値: PRIMARY

sysidcfg_ip_address_base_config

説明: 初期ブート時に使用する IP アドレス。デフォルトでは、これはターゲット設定可能なコンポーネントの IP アドレスに設定されています。

デフォルト値: : [targetableComponent:ethernet_ip_address]

sysidcfg_netmask_base_config

説明: 初期ブート時に使用するネットマスク。デフォルトでは、これはターゲット設定可能なコンポーネントのネットマスクに設定されています。

デフォルト値: : [targetableComponent:ethernet_netmask]

sysidcfg_root_password_base_conf

説明: 暗号化された root パスワード。

例: boajrOmU7GFmY は、パスワード「newroot」を暗号化した値です。

sysidcfg_system_locale_base_conf

説明: 初期ブート時に使用するシステムロケール。

例: en_US.ISO8859-1

sysidcfg_timeserver_base_config

説明: 初期ブート用にシステム時刻を取得する場所。空白の場合、システム時刻は JET サーバーから取得されます。また、この変数を `localhost` に設定して、クライアントのハードウェア時計からシステム時刻を取得することもできます。

sysidcfg_timezone_base_config

説明: 初期ブートに使用されるシステムタイムゾーン。

例: US/Pacific

`sysidcfg_terminal_base_config`
 説明: 初期ブート時に設定する端末エミュレータ。
 デフォルト値: vt100

`sysidcfg_security_policy_base_co`
 説明: 初期ブート時に使用する Kerberos セキュリティーポリシー。
 デフォルト値: NONE

`sysidcfg_protocol_ipv6_base_conf`
 説明: 初期ブート時に IPv6 プロトコルを使用するかどうか。
 デフォルト値: no

`sysidcfg_default_route_base_conf`
 説明: Solaris 9 以降の環境で JumpStart 中に使用するルーター IP アドレス。空白の場合、`defaultrouter_base_config` 変数の値を使用しようとします。この変数も空白である場合、また別のネットインタフェースに対しては、JumpStart `sysidcfg` は Jet サーバーからルーター IP を取得します。

`x86_nowin_base_config`
 説明: x86 システムの場合、インストール時に Solaris が Windows を実行しようとするのを防止します。
 デフォルト値: yes

`x86_console_base_config`
 説明: x86 システムでは、クライアントにキーボードとモニターを接続しない場合、コンソールを適切な tty ポートに設定します。この変数を設定することで、シリアルポート経由でインストールを実行できます。b1600、v20z、および v40z システムには、ttya を使用します。lx50、v60x、および v65x システムには、ttyb を使用します。

`x86_disable_acpi_base_config`
 説明: x86 システムでは、どのような値を使用しても ACPI が無効になります。ACPI を無効にすると、割り込みを処理する方法により、インストールプロセスの進行が改善される場合があります。

`x86_safetoreboot_base_config`
 説明: x86 システムでは、システムが自動的に再起動するかどうかを制御します。PXE ブートが一度限りのオプションであり、次の再起動ではディスクからブートする場合、このオプションを yes に設定する必要があります。

`profile_base_config`
 説明: 独自のカスタム JumpStart プロファイルを作成します。
 デフォルトでは、この変数を空白にすると、OS プロビジョニングプラグインは次の 21 個の `profile_*` 変数に基づいて、
`/opt/SUNWjet/Clients/hostname/profile` を作成します。また、『Solaris 10 インストールガイド (カスタム JumpStart/ 上級編)』で説明されている手順で、独

自のカスタム JumpStart プロファイルを作成することもできます。手動で作成したプロファイルを使用するには、`profile_base_config` 変数を、作成したプロファイルの名前に設定します。デフォルトでは、OS プロビジョニングプラグインは、`/opt/SUNWjet/Clients/hostname` ディレクトリのプロファイルを探します。別のディレクトリにあるプロファイルを探すようプラグインに指示するには、`profile_base_config` 変数で絶対パス名を指定します。

`profile_cluster_base_config`

説明: Solaris ソフトウェアグループパッケージ。SUNWCreq は、基本的かつ必要なソフトウェアを特定します。そのほかの選択肢には次のものが含まれます。

- SUNWCuser — ユーザーパッケージ
- SUNWCprog — ユーザーおよび開発者パッケージ
- SUNWCcall — すべてのパッケージ
- SUNWCxall — OEM パッケージを含むすべてのパッケージ (E10K 環境には必須)
- SUNWCrnet — Solaris 10 専用の最小パッケージ

デフォルト値: SUNWCreq

`profile_usedisk_base_config`

説明: OS が読み込まれるブートディスクを定義します。形式 `cnt ndn` またはキーワード `rootdisk` を使用します。値が `rootdisk` である場合、現在のブートディスクが使用されます。

デフォルト値: rootdisk

`profile_dontuse_base_config`

説明: 使用してはならないディスクのコンマ区切りリスト。形式 `cnt ndn` を使用します。この変数が適用されるのは、`profile_usedisk_base_config` が設定されていない場合のみです。

`profile_root_base_config`

説明: ルートの容量 (free、または MB サイズ)

デフォルト値: free

`profile_swap_base_config`

説明: スワップ容量 (MB 単位)。

デフォルト値: 256

`profile_s3_mtpt_base_config`

説明: スライス 3 パーティションに対するマウントパス。

注 - VxVM を使用中で、ブートディスクをミラーのようにする場合は、スライス 3 および 4 を空のままにします。

`profile_s3_size_base_config`

説明: スライス 3 パーティションのサイズ (MB 単位)。

`profile_s4_mtpt_base_config`
説明: スライス 4 パーティションに対するマウントパス。

`profile_s4_size_base_config`
説明: スライス 4 パーティションのサイズ (MB 単位)。

`profile_s5_mtpt_base_config`
説明: スライス 5 パーティションに対するマウントパス。

デフォルト値: /var

`profile_s5_size_base_config`
説明: スライス 5 パーティションのサイズ (MB 単位)。

`profile_s6_mtpt_base_config`
説明: スライス 6 パーティションに対するマウントパス。

デフォルト値: /usr

`profile_s6_size_base_config`
説明: スライス 6 パーティションのサイズ (MB 単位)。

`profile_s7_mtpt_base_config`
説明: スライス 7 パーティションに対するマウントパス。

注 – Solaris Volume Manager (SVM) を使用している場合、デフォルト動作では、スライス 7 を `metastate` データベースの位置として使用します。SVM デフォルト構成を使用している場合は、データにスライス 7 を使用しないでください。

デフォルト値: /opt

`profile_s7_size_base_config`
説明: スライス 7 パーティションのサイズ (MB 単位)。

`profile_additional_disks_base_co`
説明: ブートディスクに加えて使用および構成するディスクのリスト。形式 `c ntndn` を使用します。リストにある各ディスクに対して、マウントポイントとサイズを特定するため、各スライスの変数のセットを定義します。詳細については、153 ページの「追加ディスクの構成」を参照してください。

`profile_add_locales_base_config`
説明: 追加するロケールのコンマ区切りリスト。

例: `fr_FR, ja_JP.UTF-8`

`profile_del_locales_base_config`
説明: 削除するロケールのコンマ区切りリスト。

`profile_add_geos_base_config`
説明: 追加する地域のコンマ区切りリスト。

例: `N_Europe, C_Europe`

profile_del_geos_base_config
説明: 削除する地域のコンマ区切りリスト。

ufs_logging_filesys_base_config
説明: Solaris 7 およびそれ以降のシステムでは、ロギングに使用するマウントポイントのスペース区切りリスト。すべての UFS ファイルシステムでのロギングを有効にするには、キーワード all を使用します。Solaris 9 09/04 では、デフォルトでロギングが有効になっています。

特定のファイルシステムでロギングを無効にするには、マウントポイントの前にハイフンを配置します。すべてのファイルシステムでのロギングを無効にするには、キーワード none を使用します。

注- キーワードとマウントポイントを混在させることはできません。ルートファイルシステム (/) を指定できますが、ルートファイルシステムは all および none キーワードの一部として含まれています。

デフォルト値: all

profile_add_packages_base_config
説明: 追加するパッケージのスペース区切りリスト。

profile_del_packages_base_config
説明: 削除するパッケージのスペース区切りリスト。

profile_add_clusters_base_config
説明: 追加するクラスタパッケージのスペース区切りリスト。

profile_del_clusters_base_config
説明: 削除するクラスタパッケージのスペース区切りリスト。

例: SUNWCpm SUNWCpmx SUNWCdial SUNWCdialx

nfs_mounts_base_config
説明: 遠隔 NFS マウントポイントのスペース区切りリスト。次の例に示すように、マウントターゲットからマウントソースを分離するには、? を使用します。

例: fs?1.1.1.1:/fs

nodename_base_config
説明: /etc/nodename に使用する値 (デフォルトホスト名ではない場合)。

defaultrouter_base_config
説明: /etc/defaultrouter に使用する値。

notrouter_base_config
説明: y に設定すると、IPv4 転送が無効になり、/etc/notrouter ファイルが作成されます。

dns_domain_base_config
説明: /etc/resolv.conf ファイルの DNS ドメインエントリ。

`dns_nameservers_base_config`

説明: /etc/resolv.conf ファイルにある DNS ネームサーバーエントリに使用する IP アドレスのスペース区切りリスト。

`dns_searchpath_base_config`

説明: /etc/resolv.conf ファイルの DNS 検索行に含まれるエントリのリスト。

`dns_disableforbuild_base_config`

説明: 後の時点まで DNS の構成を遅らせます。ビルド環境で DNS が使用できない場合、この変数を `yes` に設定します。

`ntp_servers_base_config`

説明: NTP サーバーの名前または IP アドレスのスペース区切りリスト。最初のサーバーには「prefer」タグが付与されます。このセクションは `server [prefer]` という形式の行を /etc/inet/ntp.conf ファイルに配置します。追加の NTP 制御を行うには、カスタムモジュールを使用して独自のカスタム ntp.conf ファイルを配備します。

`networkifs_base_config`

説明: 定義する追加ネットワークインタフェースのスペース区切りリスト。論理インタフェースには、コロン (:) ではなくアンダースコア (_) を使用します。形式 `cntndn` を使用します。リストにある各インタフェースに対して、変数のセットを定義して、インタフェースのネット名、ネットマスク、ホスト名、および IP アドレスを指定します。この変数の詳細については、151 ページの「追加ネットワークインタフェースの構成」を参照してください。

例: `le1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

説明: IPMP 制御の下で定義されるインタフェースのスペース区切りリスト。リストにある各インタフェースに対して、変数のセットを定義し、インタフェースのネットグループ、モード、`test1`、`test2`、ネットマスク、ホスト名、`log-ip`、`hostname2`、および `log-ip2` を指定します。この変数の詳細については、151 ページの「IP ネットワークマルチパス (IPMP) の構成」を参照してください。

例: `qfe0_qfe4!database-net 1 10.0.0.1 10.0.0.2 24 oracle-db
10.0.0.3 apache 10.0.0.4`

`update_terminal_base_config`

説明: 設定されている場合、`sysidcfg` 端末タイプを `inittab` に適用します。

デフォルト値: `yes`

`enable_savecore_base_config`

説明: 任意の値に設定されている場合、Solaris 2.6 システムのセーブコアを有効にします。

デフォルト値: `yes`

`dumpadm_minfree_base_config`

説明: クラッシュダンプが dump ファイルシステムを埋め尽くさないように、制限を設定します。有効な値については、`dumpadm(1M)` の `-m` オプションを参照してください。

例: 20000k

`noautosshutdown_base_config`

説明: 任意の値に設定すると、電源管理を無効にします。

デフォルト値: `pm_disabled`

`enable_rootlogin_base_config`

説明: 任意の値に設定すると、`telnet/rsh` および `ssh` 両方からのネットワークルートログインを有効にします。

`enable_rootftp_base_config`

説明: 任意の値に設定すると、ルート FTP アクセス権が有効になります。

`shutup_sendmail_base_config`

説明: 設定されている場合、`sendmail` をシャットアップするためのエイリアスホスト名を作成します。

デフォルト値: `yes`

`poweroff_afterbuild_base_config`

説明: 設定されている場合、ビルドが完了すれば、マシンをシャットダウンします。

`dedicated_dump_device_base_config`

説明: 設定されている場合、`dumpadm` ユーティリティーによりパーティションが専用ダンプデバイスとして構成されます。サポートされているオペレーティング環境については、`dumpadm(1M)` のマニュアルページを参照してください。

`enable_altbreak_base_config`

説明: 設定されている場合、代替のブレークシーケンスを有効にします。

`disable_sysid_probe_base_config`

説明: 設定されている場合、最初の再起動で `sysid` をスキップします。この設定は、使用されていないネットワークアダプタの数が多きシステムではプロビジョニングの効率が大幅に上昇します。

デフォルト値: `yes`

`nfsv4_domain_base_config`

説明: 最初の再起動でプロンプトが表示されるのを防ぐよう、NFSv4 ドメインを設定します。設定されていない場合、まず `dns_domain_base_config` でエントリを探し、続いて `/etc/default/nfs` でドメイン値を探します。

`productdir_base_config`

説明: 製品へのパス。空白である場合、`jumpstart.conf` ファイルの情報、および JET サーバーの IP アドレスを使用します。パッケージファイルが JET サーバー上に格納されていない場合、パッケージの位置への NFS スタイルのパスを提供します。

patchdir_base_config

説明: パッチへのパス。空白である場合、jumpstart.conf ファイルの情報、および JET サーバーの IP アドレスを使用します。パッチファイルが JET サーバー上に格納されていない場合、パッチの位置への NFS スタイルのパスを提供します。

追加ネットワークインタフェースの構成

networkifs_base_config 変数は、sysidcfg_network_interface_base 変数を介して構成された最初のネットワークインタフェースに加えて、追加のネットワークインタフェースを構成する手段を提供します。

追加の各インタフェースデバイスに対して、そのインタフェースのネット名、ネットマスク、ホスト名、および IP アドレスを定義する 4 つの変数のグループを指定する必要があります。

次の例に、2 つのインタフェースに対して networkifs_base_config 変数を定義する方法を示します。

- le1
 - ネット名 — netB
 - ネットマスク — 255.255.255.0
 - ホスト名 — myhost-netB
 - インタフェース IP — 192.168.1.0
- hme0
 - ネット名 — netC
 - ネットマスク — 255.255.255.0
 - ホスト名 — myhost-netC
 - インタフェース IP — 192.162.2.0

次の構文を使用して、2 つの追加インタフェースに対して 1 つの networkifs_base_config 変数を定義します。

```
le1!netB 255.255.255.0 myhost-netB 192.168.1.0|hme0!netC 255.255.255.0 myhost-netC 192.162.2.0
```

IP ネットワークマルチパス (IPMP) の構成

IPMP の一般的な情報については、『IP Network Multipathing Administration Guide』を参照してください。

Solaris 8 またはそれ以降の環境における IP マルチパスのデフォルトモードは、自動フェイルバックです。このモードを変更するには、/etc/default/mpathd ファイルを編集します。

インタフェースのペアを使用して IPMP グループを設定できるのは、次のいずれかの構成においてのみです。

- active-standby failover — ipmp モードを s に設定し、論理ホスト名/IP アドレスのペアを 1 つ指定します。
- failover with outbound load spreading — ipmp モードを l に設定し、論理ホスト名/IP アドレスのペアを 1 つ指定します。
- active-active with outbound load spreading — ipmp モードを l に設定し、2 つ目のインタフェース用の別の論理ホスト名/IP アドレスのペアを指定します。

ipmp_networkkifs_base_config 変数は、IPMP の制御下で定義されるインタフェースのリストを定義します。各インタフェースに関して、定義済みの追加インタフェースパラメータに対して変数のセットを含める必要があります。定義する必要があるパラメータを次に示します。

- netgroup — database-net など、IPMP インタフェースグループ名を特定します。
- mode — IPMP モードを指定します。有効な値は次の通りです。
 - s — スタンバイモード。フェイルオーバーにのみ使用されます。このモードでは、テストアドレスが最後に割り当てられます。最初のテストアドレスは、最初の物理アダプタの最初の仮想インタフェース上にあります。2 番目のテストアドレスは、2 番目の物理アダプタ上にあります。
 - l — 負荷の拡散。アクティブ/アクティブモードに使用されます。このモードでは、テストアドレスは最初と 2 番目両方の物理アダプタ上の、最初の仮想インタフェース上に割り当てられます。

物理アダプタにテストアドレスを強制するには、いずれかのモードのパラメータに接尾辞「p」を追加します。



注意 — 一部のアプリケーションを破壊する可能性があるため、物理アダプタへのテストアドレスの強制には注意が必要です。

- test1 — IPMP テストアドレス 1

注 — このアドレスを決してホストファイルに入れしないでください。ホストファイルにあると、このアドレスが使用されます。

- test2 — IPMP テストアドレス 2

注 — このアドレスを決してホストファイルに入れしないでください。ホストファイルにあると、このアドレスが使用されます。

- netmask — IPMP ペアのネットマスク
- hostname — 論理 IP の一意のホスト名

- log-ip — ペアの最初の i/f の論理 IP
- hostname 2 — active-active 構成の場合のみ。論理 IP の一意のホスト名
- log-ip2 — active-active 構成の場合のみ。ペアの 2 つ目の i/f のローカル IP

次の例に、1 つの IPMP インタフェースを定義する ipmp_networkifs_base_config 変数の内容を示します。

```
qfe0_qfe4!database-net 1 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4
```

追加ディスクの構成

profile_additional_disks_base_co 変数は、ルートディスクに加えて、ディスクをフォーマットおよび構成する機能を提供します。各追加ディスクを特定するには、形式 `c ntndn` を使用します。

次に、3 ディスク構成の例を示します。ルートディスクは `c0t0d0` で、変数 `profile_usedisk_base_config` を使用して定義されています。
`profile_additional_disks_base_co` 変数は、2 つの追加ディスク `c0t1d0` および `c0t1d1` を特定します。各ディスクスライスに対して、ディレクトリのサイズと共にディレクトリ名が指定されます。

```
profile_usedisk_base_config = c0t0d0
profile_additional_disks_base_co = c0t1d0s0!/export/mydisk1!5000MB|c0t1d1s0!/export/mydisk2!6000MB
```

これらの変数を定義してから、2 つの追加作業を実行して、名前付きの追加ディスク上でフォーマットおよび使用するスライスを指定する必要があります。

1. JET サーバー上の構成ファイル
`/opt/SUNWjet/etc/osp_jet_variables_to_replace` を使用するように CHANGE 行を追加します。次に例を示します。

```
CHANGE:base_config:profile_additional_disks:profile_disk_<name>_mtpt, profile_disk_<name>_size
```

注 - JET サーバーの構成ファイルを変更することにより、この全体的な編集は、そのファイル内で変更するまで適用されます。この JET サーバー上で `profile_additional_disks_base_co` 変数を使用するすべての Solaris プロファイルは、関連付けられた `provision_start` プランの実行時にこの情報を使用します。

2. `/opt/SUNWjet/etc/populate_client_dir` ファイルに次の行を追加します。

```
base_config_profile_additional_disks="$(echo ${base_config_profile_additional_disks} |
tr ' ' '\n' | sed 's/s[0-7]//' | sort -u)"
```

必ず、`/opt/SUNWjet/etc/populate_client_dir` ファイル内の次のコードセグメントの前に、この行を追加してください。

```
#
# Now check to see if any additional disks need to be defined
```

```
#  
if [ -n "${base_config_profile_additional_disks}" ]; then
```

Solaris プロファイル上で `provision_start` プランを実行すると、適切なマウントポイントとスライスとともに、2つのディスクを構成するために適切なファイルが作成されます。

N1 SPS リモートエージェントのコンポーネント変数

次のリストで説明する変数は、N1 SPS ソフトウェアのリモートエージェント (RA) 構成を提供します。これらの変数は、JET `spsra` モジュールに直接対応します。

`install_spsra`

説明: N1 SPS RA ソフトウェアをインストールするかどうか。

デフォルト値: y

`product_version_spsra`

説明: インストールする RA のバージョン。

デフォルト値: 5.1

`basedir_spsra`

説明: RA ファイルがインストールされるベースディレクトリ。

デフォルト値: /opt/SUNWn1sps

`install_ra_from_snapshot_spsra`

説明: RA ファイルを取得する場所を決定します。JET サーバーをインストールした場合、または JET サーバーを添付した場合、JET サーバー上の RA のスナップショットが自動的に取られます。結果として、この値のデフォルトは y になります。クロスアーキテクチャーインストールの場合は、この変数を n に設定します。この場合、「OS Provisioning common tasks」ページの「JET Solaris Image Servers」セクションにある「Manage」リンクを使用して、JET モジュールの製品メディアをインポートする必要があります。

デフォルト値: y

`username_spsra`

説明: RA インストーラを使用する際に、RA ファイルシステムを所有するユーザー名を指定します。

例: jdoe

`groupname_spsra`

説明: RA インストーラを使用する際に、RA ファイルシステムを所有するグループ名を指定します。

例: staff

setup_user_acct_spsra
説明: username_spsra および groupname_spsra のユーザーおよびグループアカウントを設定するかどうかを決定します。

デフォルト値: n

user_passwd_spsra
説明: username_spsra の暗号化されたパスワード。

例: 6GR9gO1lvRVFw は、パスワード「n1spspwd」の暗号化された値を表します。

userid_spsra
説明: username_spsra のユーザーアカウント ID。

例: 505

groupid_spsra
説明: groupname_spsra のグループアカウント ID。

例: 505

homedir_spsra
説明: RA ファイルのホームディレクトリ。

デフォルト値: /opt/SUNWn1sps

jre_path_spsra
説明: Java Runtime Environment へのパス。値が指定されていない場合は、インストーラにより指定されるパスを使用します。パスが指定されている場合、JRE はそのパスにインストールされている必要があります。インストールされていないと、インストールが失敗します。

setuid_spsra
説明: RA ルートアクセス権を有効にします。

デフォルト値: y

conn_spsra
説明: RA への接続に使用する方式を決定します。デフォルトでは、ターゲット設定可能なコンポーネントに対して定義されている接続方式を使用します。

デフォルト値: :[targetableComponent:sps_ra_conn]

ssl_auth_spsra
説明: ネットワーク接続に使用する SSL 認証。

デフォルト値: 1

port_spsra
説明: ネットワーク接続に使用するポート。デフォルトでは、ターゲット設定可能なコンポーネントに対して定義されているポートを使用します。

注 - SSH 接続を使用するには、ポート番号を 70000 に設定します。

デフォルト値 :: [targetableComponent:sps_ra_port]

adv_params_spsra

説明: ネットワーク接続に使用する高度なパラメータ。デフォルトでは、ターゲット設定可能なコンポーネントに対して定義されている高度なパラメータを使用します。

デフォルト値 :: [targetableComponent:sps_ra_adv_params]

一般的な OS プロビジョニングプラグインの変数

次のリストに示す変数は、OS プロビジョニングプラグインに必要な一般的な変数です。これらの変数は技術上は JET モジュールの一部ではありませんが、OS プロビジョニングプラグインが、JET と N1 SPS ソフトウェアの間のインタフェース接続を行うために必要です。

installPath

説明: Jumpstart Enterprise Toolkit (JET) 仮想ホスト。

例: myhost-jet

osp_target_host

説明: ターゲットホスト名。

デフォルト値 :: [targetableComponent:installPath]

osp_profile_control_service

説明: 電源オフ、再起動、および電源オン機能を自動化する OSP Control Service を呼び出すかどうか (True、False)。

デフォルト値 :: [targetableComponent:osp_control_service]

sps_ra_parent_host

説明: N1 SPS RA の親のホスト名。

デフォルト値 :: [targetableComponent:sps_ra_parent]

sps_ra_parent_host_type

説明: N1 SPS RA の親のホストタイプ (MS または LD)。

デフォルト値 :: [targetableComponent:sps_ra_parent_type]

osp_jet_server
 説明: JET サーバーのホスト名。

例: myserver

osp_profile_architecture
 説明: ターゲットホストのアーキテクチャー (sun4u, x86)。

デフォルト値: : [targetableComponent:architecture]

osp_profile_min_disk_size
 説明: 最小ディスクサイズ。

例: 5GB

osp_profile_description
 説明: プロファイルの説明。

osp_profile_monitor_install
 説明: OS プロビジョニングモニター (TRUE、FALSE)。

デフォルト値: FALSE

osp_profile_boot_server_name
 説明: Solaris ブートサーバー名。

デフォルト値: : [osp_jet_server]

osp_profile_BS_subnet_address
 説明: ブートサーバーによりサービスが提供されるサブネット。

osp_profile_BS_subnet_ip_address
 説明: このサブネット上のブートサーバーの IP アドレス。

例: 199.199.8.6

osp_profile_product_ip_address
 説明: Solaris イメージサーバーの IP アドレス。

例: 199.199.8.6

osp_profile_product_directory
 説明: イメージサーバー上のイメージディレクトリへのパス。

例: /osp_image/Solaris9_u7_sparc

osp_profile_boot_ip_address
 説明: Solaris ブートサーバーの IP アドレス。

例: 199.199.8.6

osp_profile_boot_directory
 説明: Solaris ブートディレクトリ。

デフォルト値: /osp_image/Solaris9_u7_sparc/Solaris_9/Tools/Boot

osp_profile_sysidcfg_ip_address

説明: システム ID 構成の IP アドレス。

例: 199.199.8.6

osp_profile_sysidcfg_directory

説明: システム ID 構成のディレクトリ。

デフォルト値: /opt/SUNWjet/Clients/: [osp_target_host]

osp_profile_jumpstart_ip_address

説明: Jumpstart の IP アドレス。

例: 199.199.8.6

osp_profile_jumpstart_directory

説明: JumpStart のディレクトリ

デフォルト値: /opt/SUNWjet

osp_profile_boot_file_name

説明: Solaris ブートファイル名。

デフォルト値: : [osp_target_host]

osp_profile_nbp_boot_file_name

説明: ネットワークブートファイル名。

デフォルト値: : [osp_target_host]

追加 JET モジュールの作成

範囲

このモジュールは、JET フレームワークと、製品ソフトウェアの実際のインストールを実行するために提供される機能との間の接着剤の働きをします。このモジュールは、テンプレート内の一連の構成オプションで、ターゲット単位でのサーバーオプションを設定できるようにします。サーバーオプションはパラメータに従った製品インストールの実行に使用されます。

テンプレート内にどのオプションが出現するか、またどのように製品インストールを実行するかについては、特別な要件や制限は存在しません。ただし、可能な限りモジュールが互いに共存するようにしたり、不自然な依存関係をなくすようにすべきであるという一定のガイドラインは存在します。

モジュール設計のガイドライン

ツールキットそれ自体がある基本的な原理に基づいて設計されています。そのツールキットとともに当初作成されたモジュールもこのスタイルに従っていました。すべてのモジュール開発者は独自の手法を使用できますが、少なくとも次の事項を考慮し、可能であれば従う必要があります。

モジュールの対応範囲

各モジュールは、なるべく別のモジュールで使用可能な機能を複製することなく、アプリケーションの特定の範囲に対応する必要があります。また、モジュールのサイズを適切にする必要があります。モジュールを適切にサブコンポーネントに分割できる場合、1つの巨大なモジュールの代わりに、より小さなモジュールを作成する必要があります。

次に例を示します。設計者が、ターゲットサーバーインストールにセキュリティーサービスを提供するモジュールを書くことを決定する場合を考えます。このモジュールの一部として、設計者はファイアウォールと、一連の強化スクリプトをインストールすることを決定します。

この場合、設計者は別のモジュールがこれら2つの領域のいずれかにすでに対応しており、そのモジュールを活用できるかどうかを調べる必要があります。また、ファイアウォールと強化スクリプトのいずれかがお互いに分離して別の場所で使用できる場合、それらをまとめるのではなく、2つのモジュールとして作成する方がより柔軟性が高い場合があります。

モジュールの依存関係

各モジュールはそれ自身で完結している必要があり、別のモジュールに依存したり、別のモジュールの存在を想定したりするべきではありません。そのような相互作用が必要である場合(場合によっては実際に意味をなす)、そのモジュールはもう一方のモジュールに依存していると明確に指定する必要があります。このような状況が生じた場合、独立したモジュールに機能が最もよく実現されているかどうか、またはお互いに依存する2つのモジュールが1つの包括的なモジュールでよりよく実現されているかどうかを判断する必要があります。基本的な前提は、意味のある場合は項目をまとめ、多数のより小さなモジュールを、それ自体を目的として作成しないことです。

モジュールの相互作用

状況がモジュールの分離に有利であり、モジュールが(少なくとも一方向で)依存している場合、モジュールは正しい(望ましい)結果が得られるように相互作用を行うようにする必要があります。ツールキットには、モジュールごとにヒントを設定および取得するための非常に単純なメカニズムがあります。モジュールがヒントを取得するかどうかは、モジュールの設計者にかかっていますが、モジュール開発者の間の連携により、モジュールは孤立した状態で正しく動作し、統合した場合により強力なソリューションを提供するようにモジュールを書くことも可能です。

次に例を示します。実際の例では、Sun Cluster 3 製品は Solstice DiskSuite がどのように構成されているかに依存しますが、必ずしも DiskSuite がインストールされている必要はなく、また DiskSuite もクラスタソフトウェアがインストールされている必要はありません。実際、DiskSuite がクラスタマシン以外のマシン上にインストールされていることが非常に重要です。

この場合、DiskSuite に対応するモジュールと、Sun Cluster 製品に対応するモジュールの 2 つの別々のモジュールを用意します。それぞれのモジュールは独立して正しく動作しますが、同じターゲットサーバー構成内で結びつけられた場合、Sun Cluster モジュールは DiskSuite モジュールに影響し、Sun Cluster 3.0 により DiskSuite に課せられた制限に適合します。

この相互作用はモジュールヒントを使用することにより行われ、スタンドアロンシナリオでのそれらの動作は次のように要約できます。

- Sun Cluster モジュールのみがアクティブである場合、Sun Cluster はヒントを設定しますが、DiskSuite は存在せず、ヒントを取得しません。
- DiskSuite のみがアクティブである場合、ヒントを探しますが、ヒントはほかのモジュールにより設定されていないため、通常のデフォルト動作を使用します。

ヒントが実際に何を表現するかは、完全にモジュール開発者にかかっています。モジュール開発者間の緊密な連携により、ヒントを最も効率的に使用できるようになります。可能であれば、そのほかのモジュール開発者が追加インタフェースを活用できるように、モジュールのリリースノート内にヒントを文書化する必要があります。

モジュールのコーディング

モジュール開発者は、JumpStart インストール時に使用可能な標準スクリプト言語を選択する必要があります。たとえば、ターゲットサーバーが使用する NFS ブートイメージにシェルが含まれないため、「バッシュ」が使用できない場合があります。

この問題はすべてのバージョンの Solaris で存在することが認識されているため、可能であれば Bourne Shell を使用する必要があります。最後の手段としてのみ、コンパイル済み言語を検討してください。

モジュールのディレクトリ

各モジュールには、メインツールキットフレームワークがインストールされた場所からの Products サブディレクトリに位置する、独自のディレクトリ構造があります。ディレクトリの名前は、特定のモジュールを参照するためにツールキットにより使用される名前です。

たとえば、モジュール sds (Solstice DiskSuite) は `.../Products/sds/` に存在し、このモジュールのみがそのディレクトリ内に存在するものとその使用法を制御します。次に、注意すべき例外を示します。

ツールキットへのモジュールの登録には追加の相互作用は必要ありません。ディレクトリが存在すれば十分です。Products ディレクトリ内にメインツールキットインストールポイントの外部にあるそのほかの位置へのシンボリックリンクを作成することはお勧めできません。アクセス機能を提供するそのほかの手段が講じられない限り、インストール時にターゲットサーバーはそのようなディレクトリにアクセスできない場合があります。

モジュール構成 (*module.conf* ファイル)

各モジュールはある程度までユーザーにより構成可能なようになっています。これは必須要件ではありませんが、通常は実装されていることが想定されています。

ツールキットは、ユーザーへのモジュール構成の提供に関して非常に単純な見方をしています。ツールキット `make_template` コマンドを使用してテンプレートを作成する場合、1つのフラットファイルが作成されます。このファイルは、コア `base_config` 構成ファイルと、各選択モジュールからの構成ファイルを連結します。モジュールそれぞれ自体のあとで、`.conf` 接尾辞を使用した名前が付けられた構成ファイルを提供することにより、ツールキット `make_template` コマンドは残りの作業を行います。

構成ファイルは単純な Bourne Shell スクリプトです。構成ファイルは、ユーザーに対する変数の形式で、構成可能なオプションを表す必要があります。モジュールの変数名前空間を保存し、あるモジュールが別のモジュールを破壊するのを防ぐため、各変数にはモジュール名とアンダースコアの接頭辞をつける必要があります。

たとえば、モジュール `sds` は、ユーザーに対して、インストールするソフトウェアのバージョンを選択するオプションを表します。構成ファイルの対応する部分は次のようになります。

```
#####  
#  
# Which version of the product is to be installed  
#  
sds_product_version="4.2.1"
```

この例では、デフォルト値 `4.2.1` がすでに構成ファイルに生成されています。これが、モジュールが記述された時点での製品の最も新しいバージョンであったためです。

モジュールのインタフェース

ツールキットは、ターゲットサーバーと JumpStart サーバーの両方において、JumpStart プロセスのライフサイクル中にモジュール内で特定のインタフェースを呼び出します。各インタフェースは、環境変数を通じて供給されるターゲットサーバーのコンテキストを持つ、実行可能なシェルスクリプト (または最悪の場合バイナリ) であると想定されています。

copy_media インタフェース

呼び出される場所 JumpStart サーバー

引数

`<patches|packages> version srcdir destdir arch`

必須/オプション 必須

copy_media スクリプトが呼び出されるのは、ユーザーが copy_product_media または copy_product_patches スクリプトを呼び出してこのモジュールのメディアを管理する場合です。このスクリプトはアプリケーションが配信される形式を理解し、渡されたソースメディアの位置から、サーバー上の適切なメディア位置に対して、コピーを実行する必要があります。この機能を使用することで、メインツールキットを各メディアタイプに対して更新する必要なく、通常とは異なる形式のメディア (tar.gz、zip、bz2 など) を処理できるようになります。また、モジュール開発者が既知の状態ですべてのメディアを配置できるようにもなります。たとえば、ディレクトリツリー全体、または Solaris パッケージのまとまりのみを参照可能にする必要が製品で生じる場合があります。

make_template インタフェース

呼び出される場所 JumpStart サーバー

引数 なし

必須/オプション オプション

管理者がサーバービルドの新しい定義を作成する場合、管理者は /opt/SUNWjet/bin に用意されている最上位レベルコマンド make_template を実行します。この最上位レベル make_template スクリプトは基本ターゲットサーバー構成情報を設定し、存在する場合は各モジュール固有の make_template スクリプトを呼び出します。モジュール固有の make_template スクリプトは、テンプレート上で追加作業を実行できます。たとえば、ユーザーが編集するクライアントごとのデフォルトの生成などです。

make_client インタフェース

呼び出される場所	JumpStart サーバー
引数	なし
必須/オプション	オプション

管理者がインストール用にターゲットサーバーを設定する場合、ツールキットの `bin` ディレクトリに付属する最上位レベルコマンド `make_client` を実行します。この最上位レベル `make_client` スクリプトは基本ターゲットサーバー構成情報を設定し、存在する場合は各モジュール固有の `make_client` スクリプトを呼び出します。モジュール固有の `make_client` スクリプトはターゲットサーバー固有の `/opt/SUNWjet/Clients` ディレクトリで追加作業を実行でき、またモジュールヒントを構成したり、それに応じてターゲットサーバープロファイル `sysidcfg` やそのほかのファイルを変更できます。

begin インタフェース

呼び出される場所	ターゲットサーバー
引数	なし
必須/オプション	オプション

JumpStart プロセスの「begin」段階で、ツールキットはモジュールが `begin` という名前のスクリプトを持っているかどうかを確認し、それが存在する場合は、そのスクリプトが実行されます。テンプレートのモジュール構成セクションで設定されたすべての変数は、スクリプトがアクセスする環境に存在します。

install インタフェース

呼び出される場所	ターゲットサーバー
引数	なし
必須/オプション	必須

`install` スクリプトは、モジュールの中心的存在です。このスクリプトは Solaris のメインのインストールが完了したあとに、「`finish`」スクリプト段階でターゲットサーバー上に呼び出されます。その目的は、特定のアプリケーションのインストールまたは構成を調べることです。

スクリプトそれ自体は、新しくインストールされたターゲットサーバーの最初の再起動の前に呼び出されます。この時点で、ルートディレクトリ (/) は実際には JumpStart サーバーからマウントされた NFS ファイルシステムです。実際のディスクベースのルートディレクトリは、環境変数 \$ROOTDIR (従来は /a に設定されている) の使用を通じて位置が特定されます。

ルートが \$ROOTDIR に位置している場合にアプリケーションをインストールできない場合は、install スクリプトは、ツールキットにより提供されるインストール後機能を使用して、最初の再起動後のそれ以降のインストールをスケジュールする必要があります。最初の再起動後、ターゲットサーバーは実際にそれ自身のディスクからブートし、ルートは実際に「/」になります。

install スクリプトは、ユーザーが提供する構成を取得し、実際のアプリケーションのインストールおよび構成を適切に実行する役割があります。これをどのように実現するかはモジュール開発者の工夫にかかっていますが、パッケージやパッチのインストール、ファイルコピー、メッセージ報告など共通のタスクを支援する数多くのユーティリティー機能がメインツールキットから使用できます。

モジュール install スクリプトが呼び出される前に、テンプレートで定義され、元来は *module.conf* ファイルから生成されたモジュール構成が、シェル環境に読み込まれます。install スクリプトは引数を使用して呼び出されることは想定していませんが、代わりに現在の環境からその構成を取得する必要があります。このテクニックにより、各モジュールインストールスクリプトが異なる数の引数を要求するといった問題を回避できます。

check_client インタフェース

呼び出される場所	ターゲットサーバー
引数	なし
必須/オプション	オプション

モジュール開発者が check_client スクリプトを用いると、テンプレートで指定されている構成オプションに関する基本的なチェックを実行できます。このスクリプトを呼び出すと、テンプレートで設定されている変数を使用して環境が構成され、またスクリプトはインストールエラーを減らすための基本的なチェックを実行できます。

モジュールは、有効なオプションのチェックや、選択したバージョン用のメディアが存在することのチェックを決定できます。提供される機能のレベルは、実装者によって決まります。

ツールキットのサポート関数

メインツールキットには、モジュールにより活用可能な多くの共通関数が用意されています。これにより、コードの再利用が改善され、モジュールがよりシンプルになります。どのような関数を使用できるかを調べる最善の方法は、関数が存在するディレクトリ `/opt/SUNWjet/Utils/lib` を調べる方法です。

JET モジュールの追加

この節では、追加の JET モジュールを Solaris ブートおよびインストール (JET) サーバーに追加する方法について説明します。プロセスは Flash モジュールのコンテキストで説明されていますが、その論理はほかの JET モジュールにも拡張されます。

▼ Flash モジュールを追加する

- 手順
1. **JetFLASH.pkg** を Solaris ブートおよびインストールサーバーにダウンロードします。
 2. **JetFLASH** パッケージをインストールします。
次の例のようなコマンドを使用します。

```
# cat >/tmp/admin-file <<- _EOF
mail=\n
instance=unique
partial=quit
runlevel=ask
idepend=quit
rdepend=nocheck
space=quit
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=/opt/SUNWjet/Products
_EOF
pkgadd -a /tmp/admin-file -d JetFLASH.pkg
```

3. Flash イメージを Solaris ブートおよびインストールサーバーにコピーします。
次の例のようなコマンドを使用します。

```
# telnet solaris-bis-ip-address
# cp flash-archive /export/install/flash/sol10_xall_sparc.flar
```

4. この Flash イメージの Solaris Profile を作成します。

- a. Solaris 10 Flash イメージの変数セットを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd cdb.vs.add -comp NM:/com/sun/nlosp/untyped/SolarisImage \
-name "solaris10sparc" -u admin -p admin -vars "version=10;release=ga;architecture=sparc; \
image_path=/export/install/s10ga-sparc;image_subnet_addr=10.42.42.2; \
image_subnet_mask=255.255.255.0;media_src="
```

- b. Flash 情報を指定するには、次のエントリが含まれるファイル
/tmp/flash-profile を作成します。

```
flash-with-ra
Solaris10 Flash Archive With SPS RA
base_config flash spsra
```

各行の意味は次のとおりです。

- ファイルの最初の行は、ブラウザインタフェースのプラン変数セクションの「Profile Name」フィールドに対応します。
- ファイルの 2 番目の行は、ブラウザインタフェースのプラン変数セクションの「Profile Description」フィールドに対応します。
- ファイルの 3 番目の行は、ブラウザインタフェースのプラン変数セクションの「JET Module Name(s)」フィールドに対応します。

- c. コンポーネントを作成するには、次の例のようなコマンドを入力します。

```
# cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/nlosp/untyped/SolarisImage-create-profile -tar H:NM:biss1-jet \
-comp - -vs solaris10sparc -pto 30 -nto 10 -f /tmp/flash-profile
```

ヒント – また、N1 SPS ブラウザインタフェースを介して Profile コンポーネントを作成することもできます。「Solaris Image: create profile」オプションを使用します。

5. 前の手順で作成した Profile コンポーネントを編集します。

プロファイルの位置は、

/com/sun/nlosp/autogen-masterserver-jet/provision/ です。この作業例では、Profile コンポーネントは

/com/sun/nlosp/autogen-masterserver-jet/provision/Solaris_10.flash
です。

6. **Profile** コンポーネントで、**archive_locations_flash** 変数の値を **Flash** アーカイブを指すよう変更します。
次に例を示します。

```
archive_locations_flash nfs://10.216.0.55/export/install/flash/sol10_xall_sparc.flar
```

7. コンポーネントをチェックインします。
これで上記の Solaris Profile を使用して目的のターゲットホスト上で配備を行う準備ができました。

索引

A

adv_params_spsra 変数, 156
alom_access_password 変数, ALOM ターゲット用, 125
alom_access_userid 変数, ALOM ターゲット用, 125
alom_ip_address 変数, ALOM ターゲット用, 125
ALOM ターゲット変数
 alom_access_password, 125
 alom_access_userid, 125
 alom_ip_address, 125
 architecture, 124
 description, 123
 disk_size, 124
 ethernet_ip_address, 124
 ethernet_mac_address, 124
 ethernet_netmask, 124
 installPath, 123
 kernel_arch, 124
 osp_control_service, 124
 sps_ra_adv_parms, 125
 sps_ra_conn, 124
 sps_ra_host, 124
 sps_ra_parent, 124
 sps_ra_parent_type, 124
 sps_ra_port, 125
 sps_ra_ssl_auth, 125
 target_host, 123
 terminal_server_hal_jar, 125
 terminal_server_ip_address, 125
 terminal_server_password, 125
 terminal_server_port, 125

ALOM ターゲット変数 (続き)

 terminal_server_type, 125
 terminal_server_userid, 125
 terminal_server_version, 125
architecture 変数
 ALOM ターゲット用, 124
 HP ILO ターゲット用, 131
 IPMI ターゲット用, 122
 LOM ターゲット用, 126
 Mid-Range ターゲット用, 129
 そのほかの IPMI ターゲット用, 133
 そのほかの遠隔管理ターゲット用, 136
 汎用ターゲット用, 141

B

base_config 変数
 client_allocation_base_config, 144
 ClientArch_base_config, 143
 ClientEther_base_config, 143
 ClientOS_base_config, 143
 dedicated_dump_device_base_config, 150
 defaultrouter_base_config, 148
 disable_sysid_probe_base_config, 150
 dns_disableforbuild_base_config, 149
 dns_domain_base_config, 148
 dns_nameservers_base_config, 148
 dns_searchpath_base_config, 149
 dumpadm_minfree_base_config, 149
 enable_altbreak_base_config, 150
 enable_rootftp_base_config, 150
 enable_rootlogin_base_config, 150

base_config 変数 (続き)

- enable_savecore_base_config, 149
- ipmp_networkkifs_base_config, 149
- networkkifs_base_config, 149
- nfs_mounts_base_config, 148
- nfsv4_domain_base_config, 150
- noautoshtutdown_base_config, 150
- nodename_base_config, 148
- notrouter_base_config, 148
- ntp_servers_base_config, 149
- patchdir_base_config, 150
- poweroff_afterbuild_base_config, 150
- productdir_base_config, 150
- products_base_config, 144
- profile_add_clusters_base_config, 148
- profile_add_geos_base_config, 147
- profile_add_locales_base_config, 147
- profile_add_packages_base_config, 148
- profile_additional_disks_base_co, 147
- profile_base_config, 145
- profile_cluster_base_config, 146
- profile_del_clusters_base_config, 148
- profile_del_geos_base_config, 147
- profile_del_locales_base_config, 147
- profile_del_packages_base_config, 148
- profile_dontuse_base_config, 146
- profile_root_base_config, 146
- profile_s3_mtpt_base_config, 146
- profile_s3_size_base_config, 146
- profile_s4_mtpt_base_config, 146
- profile_s4_size_base_config, 147
- profile_s5_mtpt_base_config, 147
- profile_s5_size_base_config, 147
- profile_s6_mtpt_base_config, 147
- profile_s6_size_base_config, 147
- profile_s7_mtpt_base_config, 147
- profile_s7_size_base_config, 147
- profile_swap_base_config, 146
- profile_usedisk_base_config, 146
- shutup_sendmail_base_config, 150
- sysidcfg_default_route_base_conf, 145
- sysidcfg_ip_address_base_config, 144
- sysidcfg_nameservice_base_config, 144
- sysidcfg_netmask_base_config, 144
- sysidcfg_network_interface_base_, 144
- sysidcfg_protocol_ipv6_base_conf, 145
- sysidcfg_root_password_base_conf, 144
- sysidcfg_security_policy_base_co, 145

base_config 変数 (続き)

- sysidcfg_system_locale_base_conf, 144
- sysidcfg_terminal_base_config, 145
- sysidcfg_timeserver_base_config, 144
- sysidcfg_timezone_base_config, 144
- ufs_logging_filesys_base_config, 148
- update_terminal_base_config, 149
- x86_console_base_config, 145
- x86_disable_acpi_base_config, 145
- x86_nowin_base_config, 145
- x86_safetoreboot_base_config, 145

basedir_spsra 変数, 154

boot_service_true_false_1 変数, そのほかの遠隔管理ターゲット用, 138

boot_service_true_false_2 変数, そのほかの遠隔管理ターゲット用, 139

boot_service_true_false 変数
LOM ターゲット用, 128
そのほかの IPMI ターゲット用, 135

C

client_allocation_base_config 変数, 144

ClientArch_base_config 変数, 143

ClientEther_base_config 変数, 143

ClientOS_base_config 変数, 143

conn_spsra 変数, 155

console_service_true_false_1 変数, そのほかの遠隔管理ターゲット用, 138

console_service_true_false_2 変数, そのほかの遠隔管理ターゲット用, 139

console_service_true_false 変数
LOM ターゲット用, 128
そのほかの IPMI ターゲット用, 135

D

dedicated_dump_device_base_conf 変数, 150

defaultrouter_base_config 変数, 148

description 変数
ALOM ターゲット用, 123
HP ILO ターゲット用, 131
IPMI ターゲット用, 121
LOM ターゲット用, 126

description 変数 (続き)
そのほかの IPMI ターゲット用, 133
そのほかの遠隔管理ターゲット用, 136
汎用ターゲット用, 141

disable_sysid_probe_base_config 変数, 150

disk_size 変数
ALOM ターゲット用, 124
HP ILO ターゲット用, 131
IPMI ターゲット用, 122
LOM ターゲット用, 126
Mid-Range ターゲット用, 129
そのほかの IPMI ターゲット用, 133
そのほかの遠隔管理ターゲット用, 136
汎用ターゲット用, 141

disk_type 変数
LOM ターゲット用, 127
そのほかの IPMI ターゲット用, 134
そのほかの遠隔管理ターゲット用, 137

dns_disableforbuild_base_config 変数, 149

dns_domain_base_config 変数, 148

dns_nameservers_base_config 変数, 148

dns_searchpath_base_config 変数, 149

domain_name 変数, Mid-Range ターゲット用, 130

domainsc_access_password 変数,
Mid-Range ターゲット用, 130

domainsc_access_userid 変数, Mid-Range
ターゲット用, 130

dumpadm_minfree_base_config 変数, 149

E

enable_altbreak_base_config 変数, 150

enable_rootftp_base_config 変数, 150

enable_rootlogin_base_config 変数, 150

enable_savecore_base_config 変数, 149

ethernet_ip_address 変数
ALOM ターゲット用, 124
HP ILO ターゲット用, 131
IPMI ターゲット用, 121
LOM ターゲット用, 126
Mid-Range ターゲット用, 129
そのほかの IPMI ターゲット用, 133
そのほかの遠隔管理ターゲット用, 136
汎用ターゲット用, 141

ethernet_mac_address 変数
ALOM ターゲット用, 124
HP ILO ターゲット用, 131
IPMI ターゲット用, 121
LOM ターゲット用, 126
Mid-Range ターゲット用, 129
そのほかの IPMI ターゲット用, 133
そのほかの遠隔管理ターゲット用, 136
汎用ターゲット用, 141

ethernet_netmask 変数
ALOM ターゲット用, 124
HP ILO ターゲット用, 131
IPMI ターゲット用, 122
LOM ターゲット用, 126
Mid-Range ターゲット用, 129
そのほかの IPMI ターゲット用, 133
そのほかの遠隔管理ターゲット用, 136
汎用ターゲット用, 141

G

groupid_spsra 変数, 155

groupname_spsra 変数, 154

guid 変数
HP ILO ターゲット用, 132
IPMI ターゲット用, 123
そのほかの IPMI ターゲット用, 134
そのほかの遠隔管理ターゲット用, 137

guid 変数, 汎用ターゲット用, 142

H

homedir_spsra 変数, 155

HP ILO ターゲット変数
architecture, 131
description, 131
disk_size, 131
ethernet_ip_address, 131
ethernet_mac_address, 131
ethernet_netmask, 131
guid, 132
ilo_access_password, 132
ilo_access_userid, 132
ilo_ip_address, 132
ilo_version, 132
installPath, 130

HP ILO ターゲット変数 (続き)

- kernel_arch, 131
- osp_control_service, 131
- sps_ra_adv_parms, 132
- sps_ra_conn, 132
- sps_ra_host, 131
- sps_ra_parent, 131
- sps_ra_parent_type, 131
- sps_ra_port, 132
- sps_ra_ssl_auth, 132
- target_host, 130

I

- ilo_access_password 変数, HP ILO ターゲット用, 132
- ilo_access_userid 変数, HP ILO ターゲット用, 132
- ilo_ip_address 変数, HP ILO ターゲット用, 132
- ilo_version 変数, HP ILO ターゲット用, 132
- install_ra_from_snapshot_spsra 変数, 154
- install_spsra 変数, 154
- installPath 変数
 - ALOM ターゲット用, 123
 - HP ILO ターゲット用, 130
 - IPMI ターゲット用, 121
 - LOM ターゲット用, 126
 - Mid-Range ターゲット用, 128
 - Solaris OS プロビジョニング用, 156
 - そのほかの IPMI ターゲット用, 133
 - そのほかの遠隔管理ターゲット用, 136
 - 汎用ターゲット用, 141
- ipmi_access_password 変数
 - IPMI ターゲット用, 123
 - そのほかの IPMI ターゲット用, 135
- ipmi_access_userid 変数
 - IPMI ターゲット用, 123
 - そのほかの IPMI ターゲット用, 135
- ipmi_hal_jar 変数, そのほかの IPMI ターゲット用, 135
- ipmi_ip_address 変数
 - IPMI ターゲット用, 123
 - そのほかの IPMI ターゲット用, 134
- ipmi_version 変数, そのほかの IPMI ターゲット用, 135

IPMI ターゲット変数

- architecture, 122
- description, 121
- disk_size, 122
- ethernet_ip_address, 121
- ethernet_mac_address, 121
- ethernet_netmask, 122
- guid, 123
- installPath, 121
- ipmi_access_password, 123
- ipmi_access_userid, 123
- ipmi_ip_address, 123
- kernel_arch, 122
- osp_control_service, 122
- sps_ra_adv_parms, 123
- sps_ra_conn, 122
- sps_ra_host, 122
- sps_ra_parent, 122
- sps_ra_parent_type, 122
- sps_ra_port, 122
- sps_ra_ssl_auth, 122
- ssh_access_password, 123
- ssh_access_userid, 123
- target_host, 121

ipmp_networkifs_base_config 変数, 149

J

- JAR ファイル, 35
- Java アーカイブ, 「JAR ファイル」を参照
- JET
 - 既存のユーザー, 45
 - モジュール, 44
- jre_path_spsra 変数, 155
- JumpStart, 「JET」を参照
- JumpStart Enterprise Toolkit, 「JET」を参照

K

- kernel_arch 変数
 - ALOM ターゲット用, 124
 - HP ILO ターゲット用, 131
 - IPMI ターゲット用, 122
 - LOM ターゲット用, 126
 - Mid-Range ターゲット用, 129
 - そのほかの IPMI ターゲット用, 133

kernel_arch 変数 (続き)
そのほかの遠隔管理ターゲット用, 136
汎用ターゲット用, 141

L

lom_hal_jar 変数, LOM ターゲット用, 127
lom_version 変数, LOM ターゲット用, 127
LOM ターゲット変数
architecture, 126
boot_service_true_false, 128
console_service_true_false, 128
description, 126
disk_size, 126
disk_type, 127
ethernet_ip_address, 126
ethernet_mac_address, 126
ethernet_netmask, 126
installPath, 126
kernel_arch, 126
lom_hal_jar, 127
lom_version, 127
osp_control_service, 126
power_service_true_false, 128
sps_ra_adv_parms, 127
sps_ra_conn, 127
sps_ra_host, 127
sps_ra_parent, 127
sps_ra_parent_type, 127
sps_ra_port, 127
sps_ra_ssl_auth, 127
target_host, 126
terminal_server_hal_jar, 128
terminal_server_ip_address, 127
terminal_server_password, 128
terminal_server_port, 127
terminal_server_type, 128
terminal_server_userid, 128
terminal_server_version, 128

M

Mid-Range ターゲット変数
architecture, 129
disk_size, 129
domain_name, 130

Mid-Range ターゲット変数 (続き)

domainsc_access_password, 130
domainsc_access_userid, 130
ethernet_ip_address, 129
ethernet_mac_address, 129
ethernet_netmask, 129
installPath, 128
kernel_arch, 129
osp_control_service, 129
platformsc_access_password, 130
platformsc_access_userid, 130
platformsc_ip_address, 130
sps_ra_adv_parms, 130
sps_ra_conn, 130
sps_ra_host, 130
sps_ra_parent, 129
sps_ra_parent_type, 129
sps_ra_port, 130
sps_ra_ssl_auth, 130
target_host, 129

N

netmgmt_hal_jar_1 変数, そのほかの遠隔管理ターゲット用, 137
netmgmt_hal_jar_2 変数, そのほかの遠隔管理ターゲット用, 139
netmgmt_ip_address_1 変数, そのほかの遠隔管理ターゲット用, 138
netmgmt_ip_address_2 変数, そのほかの遠隔管理ターゲット用, 139
netmgmt_password_1 変数, そのほかの遠隔管理ターゲット用, 138
netmgmt_password_2 変数, そのほかの遠隔管理ターゲット用, 140
netmgmt_port_1 変数, そのほかの遠隔管理ターゲット用, 138
netmgmt_port_2 変数, そのほかの遠隔管理ターゲット用, 139
netmgmt_protocol_1 変数, そのほかの遠隔管理ターゲット用, 138
netmgmt_protocol_2 変数, そのほかの遠隔管理ターゲット用, 139
netmgmt_type_1 変数, そのほかの遠隔管理ターゲット用, 137
netmgmt_type_2 変数, そのほかの遠隔管理ターゲット用, 139

netmgmt_userid_1 変数, そのほかの遠隔管理ターゲット用, 138
netmgmt_userid_2 変数, そのほかの遠隔管理ターゲット用, 140
netmgmt_version_1 変数, そのほかの遠隔管理ターゲット用, 137
netmgmt_version_2 変数, そのほかの遠隔管理ターゲット用, 139
networkifs_base_config 変数, 149
nfs_mounts_base_config 変数, 148
nfsv4_domain_base_config 変数, 150
noautoshtutdown_base_config 変数, 150
nodename_base_config 変数, 148
notrouter_base_config 変数, 148
ntp_servers_base_config 変数, 149

O

osp_control_service 変数
ALOM ターゲット用, 124
HP ILO ターゲット用, 131
IPMI ターゲット用, 122
LOM ターゲット用, 126
Mid-Range ターゲット用, 129
そのほかの IPMI ターゲット用, 133
そのほかの遠隔管理ターゲット用, 136
汎用ターゲット用, 141
osp_jet_server 変数, 156
osp_profile_architecture 変数, 157
osp_profile_boot_directory 変数, 157
osp_profile_boot_file_name 変数, 158
osp_profile_boot_ip_address 変数, 157
osp_profile_boot_server_name 変数, 157
osp_profile_BS_subnet_address 変数, 157
osp_profile_BS_subnet_ip_address 変数, 157
osp_profile_control_service 変数, 156
osp_profile_description 変数, 157
osp_profile_jumpstart_directory 変数, 158
osp_profile_jumpstart_ip_address 変数, 158
osp_profile_min_disk_size 変数, 157
osp_profile_monitor_install 変数, 157
osp_profile_nbp_boot_file_name 変数, 158

osp_profile_product_directory 変数, 157
osp_profile_product_ip_address 変数, 157
osp_profile_sysidcfg_directory 変数, 158
osp_profile_sysidcfg_ip_address 変数, 157
osp_target_host 変数, 156
OS プロビジョニングプラグイン, インポート, 36

P

patchdir_base_config 変数, 150
platformsc_access_password 変数, Mid-Range ターゲット用, 130
platformsc_access_userid 変数, Mid-Range ターゲット用, 130
platformsc_ip_address 変数, Mid-Range ターゲット用, 130
port_spsra 変数, 155
power_service_true_false_1 変数, そのほかの遠隔管理ターゲット用, 138
power_service_true_false_2 変数, そのほかの遠隔管理ターゲット用, 139
power_service_true_false 変数
LOM ターゲット用, 128
そのほかの IPMI ターゲット用, 135
poweroff_afterbuild_base_config 変数, 150
product_version_spsra 変数, 154
productdir_base_config 変数, 150
products_base_config 変数, 144
profile_add_clusters_base_config 変数, 148
profile_add_geos_base_config 変数, 147
profile_add_locales_base_config 変数, 147
profile_add_packages_base_config 変数, 148
profile_additional_disks_base_co 変数, 147
profile_base_config 変数, 145
profile_cluster_base_config 変数, 146
profile_del_clusters_base_config 変数, 148

profile_del_geos_base_config 変数, 147
profile_del_locales_base_config 変数, 147
profile_del_packages_base_config 変数, 148
profile_dontuse_base_config 変数, 146
profile_root_base_config 変数, 146
profile_s3_mtpt_base_config 変数, 146
profile_s3_size_base_config 変数, 146
profile_s4_mtpt_base_config 変数, 146
profile_s4_size_base_config 変数, 147
profile_s5_mtpt_base_config 変数, 147
profile_s5_size_base_config 変数, 147
profile_s6_mtpt_base_config 変数, 147
profile_s6_size_base_config 変数, 147
profile_s7_mtpt_base_config 変数, 147
profile_s7_size_base_config 変数, 147
profile_swap_base_config 変数, 146
profile_usedisk_base_config 変数, 146

S

setuid_spsra 変数, 155
setup_user_acct_spsra 変数, 155
shutup_sendmail_base_config 変数, 150
Solaris OS プロビジョニング変数
installPath, 156
osp_jet_server, 156
osp_profile_architecture, 157
osp_profile_boot_directory, 157
osp_profile_boot_file_name, 158
osp_profile_boot_ip_address, 157
osp_profile_boot_server_name, 157
osp_profile_BS_subnet_address, 157
osp_profile_BS_subnet_ip_address, 157
osp_profile_control_service, 156
osp_profile_description, 157
osp_profile_jumpstart_directory, 158
osp_profile_jumpstart_ip_address, 158
osp_profile_min_disk_size, 157
osp_profile_monitor_install, 157
osp_profile_nbp_boot_file_name, 158
osp_profile_product_directory, 157
osp_profile_product_ip_address, 157
osp_profile_sysidcfg_directory, 158
osp_profile_sysidcfg_ip_address, 157
osp_target_host, 156

Solaris OS プロビジョニング変数 (続き)
sps_ra_parent_host, 156
sps_ra_parent_host_type, 156
sps_ra_adv_parms 変数
ALOM ターゲット用, 125
HP ILO ターゲット用, 132
IPMI ターゲット用, 123
LOM ターゲット用, 127
Mid-Range ターゲット用, 130
そのほかの IPMI ターゲット用, 134
そのほかの遠隔管理ターゲット用, 137
汎用ターゲット用, 142
sps_ra_conn 変数
ALOM ターゲット用, 124
HP ILO ターゲット用, 132
IPMI ターゲット用, 122
LOM ターゲット用, 127
Mid-Range ターゲット用, 130
そのほかの IPMI ターゲット用, 134
そのほかの遠隔管理ターゲット用, 137
汎用ターゲット用, 142
sps_ra_host 変数
ALOM ターゲット用, 124
HP ILO ターゲット用, 131
IPMI ターゲット用, 122
LOM ターゲット用, 127
Mid-Range ターゲット用, 130
そのほかの IPMI ターゲット用, 134
そのほかの遠隔管理ターゲット用, 137
汎用ターゲット用, 142
sps_ra_parent_host_type 変数, 156
sps_ra_parent_host 変数, 156
sps_ra_parent_type 変数
ALOM ターゲット用, 124
HP ILO ターゲット用, 131
IPMI ターゲット用, 122
LOM ターゲット用, 127
Mid-Range ターゲット用, 129
そのほかの IPMI ターゲット用, 134
そのほかの遠隔管理ターゲット用, 137
汎用ターゲット用, 141
sps_ra_parent 変数
ALOM ターゲット用, 124
HP ILO ターゲット用, 131
IPMI ターゲット用, 122
LOM ターゲット用, 127
Mid-Range ターゲット用, 129
そのほかの IPMI ターゲット用, 134

sps_ra_parent 変数 (続き)
そのほかの遠隔管理ターゲット用, 137
汎用ターゲット用, 141

sps_ra_port 変数
ALOM ターゲット用, 125
HP ILO ターゲット用, 132
IPMI ターゲット用, 122
LOM ターゲット用, 127
Mid-Range ターゲット用, 130
そのほかの IPMI ターゲット用, 134
そのほかの遠隔管理ターゲット用, 137
汎用ターゲット用, 142

sps_ra_ssl_auth 変数
ALOM ターゲット用, 125
HP ILO ターゲット用, 132
IPMI ターゲット用, 122
LOM ターゲット用, 127
Mid-Range ターゲット用, 130
そのほかの IPMI ターゲット用, 134
そのほかの遠隔管理ターゲット用, 137
汎用ターゲット用, 142

spsra 変数
adv_params_spsra, 156
basedir_spsra, 154
conn_spsra, 155
groupid_spsra, 155
groupname_spsra, 154
homedir_spsra, 155
install_ra_from_snapshot_spsra, 154
install_spsra, 154
jre_path_spsra, 155
port_spsra, 155
product_version_spsra, 154
setuid_spsra, 155
setup_user_acct_spsra, 155
ssl_auth_spsra, 155
user_passwd_spsra, 155
userid_spsra, 155
username_spsra, 154

ssh_access_password 変数, IPMI ター
ゲット用, 123

ssh_access_userid 変数, IPMI ターゲッ
ト用, 123

ssl_auth_spsra 変数, 155

Sun Data Center Reference Architecture, 30

sysidcfg_default_route_base_conf 変
数, 145

sysidcfg_ip_address_base_config 変
数, 144

sysidcfg_nameservice_base_config 変
数, 144

sysidcfg_netmask_base_config 変数, 144

sysidcfg_network_interface_base_ 変
数, 144

sysidcfg_protocol_ipv6_base_conf 変
数, 145

sysidcfg_root_password_base_conf 変
数, 144

sysidcfg_security_policy_base_co 変
数, 145

sysidcfg_system_locale_base_conf 変
数, 144

sysidcfg_terminal_base_config 変
数, 145

sysidcfg_timeserver_base_config 変
数, 144

sysidcfg_timezone_base_config 変
数, 144

T

target_host 変数
ALOM ターゲット用, 123
HP ILO ターゲット用, 130
IPMI ターゲット用, 121
LOM ターゲット用, 126
Mid-Range ターゲット用, 129
そのほかの IPMI ターゲット用, 133
そのほかの遠隔管理ターゲット用, 136
汎用ターゲット用, 141

terminal_server_hal_jar_1 変数, そのほ
かの遠隔管理ターゲット用, 138

terminal_server_hal_jar_2 変数, そのほ
かの遠隔管理ターゲット用, 140

terminal_server_hal_jar 変数
ALOM ターゲット用, 125
LOM ターゲット用, 128

terminal_server_ip_address_1 変数, そ
のほかの遠隔管理ターゲット用, 138

terminal_server_ip_address_2 変数, そ
のほかの遠隔管理ターゲット用, 140

terminal_server_ip_address 変数
ALOM ターゲット用, 125
LOM ターゲット用, 127

terminal_server_password_1 変数, そのほかの遠隔管理ターゲット用, 139

terminal_server_password_2 変数, そのほかの遠隔管理ターゲット用, 140

terminal_server_password 変数
ALOM ターゲット用, 125
LOM ターゲット用, 128

terminal_server_port_1 変数, そのほかの遠隔管理ターゲット用, 139

terminal_server_port_2 変数, そのほかの遠隔管理ターゲット用, 140

terminal_server_port 変数
ALOM ターゲット用, 125
LOM ターゲット用, 127

terminal_server_protocol_1 変数, そのほかの遠隔管理ターゲット用, 139

terminal_server_protocol_2 変数, そのほかの遠隔管理ターゲット用, 140

terminal_server_type_1 変数, そのほかの遠隔管理ターゲット用, 138

terminal_server_type_2 変数, そのほかの遠隔管理ターゲット用, 140

terminal_server_type 変数
ALOM ターゲット用, 125
LOM ターゲット用, 128

terminal_server_userid_1 変数, そのほかの遠隔管理ターゲット用, 139

terminal_server_userid_2 変数, そのほかの遠隔管理ターゲット用, 140

terminal_server_userid 変数
ALOM ターゲット用, 125
LOM ターゲット用, 128

terminal_server_version_1 変数, そのほかの遠隔管理ターゲット用, 138

terminal_server_version_2 変数, そのほかの遠隔管理ターゲット用, 140

terminal_server_version 変数
ALOM ターゲット用, 125
LOM ターゲット用, 128

U

ufs_logging_filesys_base_config 変数, 148

update_terminal_base_config 変数, 149

user_passwd_spsra 変数, 155

userid_spsra 変数, 155

username_spsra 変数, 154

X

x86_console_base_config 変数, 145

x86_disable_acpi_base_config 変数, 145

x86_nowin_base_config 変数, 145

x86_safetoreboot_base_config 変数, 145

こ

構成のベストプラクティス, 30

そ

そのほかの IPMI ターゲット変数

- architecture, 133
- boot_service_true_false, 135
- console_service_true_false, 135
- description, 133
- disk_size, 133
- disk_type, 134
- ethernet_ip_address, 133
- ethernet_mac_address, 133
- ethernet_netmask, 133
- guid, 134
- installPath, 133
- ipmi_access_password, 135
- ipmi_access_userid, 135
- ipmi_hal_jar, 135
- ipmi_ip_address, 134
- ipmi_version, 135
- kernel_arch, 133
- osp_control_service, 133
- power_service_true_false, 135
- sps_ra_adv_parms, 134
- sps_ra_conn, 134
- sps_ra_host, 134
- sps_ra_parent, 134
- sps_ra_parent_type, 134
- sps_ra_port, 134
- sps_ra_ssl_auth, 134
- target_host, 133

そのほかの遠隔管理ターゲット変数

- architecture, 136

その他の遠隔管理ターゲット変数 (続き)

boot_service_true_false_1, 138
boot_service_true_false_2, 139
console_service_true_false_1, 138
console_service_true_false_2, 139
description, 136
disk_size, 136
disk_type, 137
ethernet_ip_address, 136
ethernet_mac_address, 136
ethernet_netmask, 136
guid, 137
installPath, 136
kernel_arch, 136
netmgmt_hal_jar_1, 137
netmgmt_hal_jar_2, 139
netmgmt_ip_address_1, 138
netmgmt_ip_address_2, 139
netmgmt_password_1, 138
netmgmt_password_2, 140
netmgmt_port_1, 138
netmgmt_port_2, 139
netmgmt_protocol_1, 138
netmgmt_protocol_2, 139
netmgmt_type_1, 137
netmgmt_type_2, 139
netmgmt_userid_1, 138
netmgmt_userid_2, 140
netmgmt_version_1, 137
netmgmt_version_2, 139
osp_control_service, 136
power_service_true_false_1, 138
power_service_true_false_2, 139
sps_ra_adv_parms, 137
sps_ra_conn, 137
sps_ra_host, 137
sps_ra_parent, 137
sps_ra_parent_type, 137
sps_ra_port, 137
sps_ra_ssl_auth, 137
target_host, 136
terminal_server_hal_jar_1, 138
terminal_server_hal_jar_2, 140
terminal_server_ip_address_1, 138
terminal_server_ip_address_2, 140
terminal_server_password_1, 139
terminal_server_password_2, 140
terminal_server_port_1, 139

その他の遠隔管理ターゲット変数 (続き)

terminal_server_port_2, 140
terminal_server_protocol_1, 139
terminal_server_protocol_2, 140
terminal_server_type_1, 138
terminal_server_type_2, 140
terminal_server_userid_1, 139
terminal_server_userid_2, 140
terminal_server_version_1, 138
terminal_server_version_2, 140

て

データセンターの構成, 30

は

汎用ターゲット変数

architecture, 141
description, 141
disk_size, 141
ethernet_ip_address, 141
ethernet_mac_address, 141
ethernet_netmask, 141
guid, 142
installPath, 141
kernel_arch, 141
osp_control_service, 141
sps_ra_adv_parms, 142
sps_ra_conn, 142
sps_ra_host, 142
sps_ra_parent, 141
sps_ra_parent_type, 141
sps_ra_port, 142
sps_ra_ssl_auth, 142
target_host, 141

ふ

プラグインのインポート, 36

プラグインファイル

インポート, 36
の位置, 35