



Sun N1 Service Provisioning System ユーザーズガイド (OS Provisioning Plug-In 3.0)



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

Part No: 819-6503-10
2006年4月

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

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

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

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

Sun、Sun Microsystems、docs.sun.com、および Java 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、Solaris のロゴマーク、Java Coffee Cup 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に基づくものです。

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

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

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

「ATOK」は、株式会社ジャストシステムの登録商標です。

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

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

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

Unicode は、Unicode, Inc. の商標です。

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

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザ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、OPENLOOK のグラフィカル・ユーザ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 for OS Provisioning Plug-In 3.0

Part No: 819-4458-10

Revision A

目次

はじめに	15
1 OS プロビジョニングの概要	19
OS プロビジョニングの概要	19
機能	20
プロビジョニングプロセスの概要	20
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	21
ソフトウェアアーキテクチャーの全体像	21
ネットワークプロトコル	23
サポートされるシステム	24
システムコンポーネント	24
OS プロビジョニングコンポーネント	25
ブートおよびインストールサーバー	25
ターゲットホストのハードウェアプラットフォーム	26
ターゲットホストのオペレーティングシステム	26
OS プロビジョニングプラグイン用のサンプルコマンド行スクリプト	26
2 OS プロビジョニングプラグインのリリースノート	29
インストール上の問題	29
実行時の問題	29
マルチブートのターゲットホストで Windows のインストールモードが対話モードに切り替わる (6360018)	29
Windows RIS サーバーが DHCP PXE 要求に応答しない (6333612)	30
SPARC: Solaris OS プロビジョニングサーバーの作成プランの Detailed Preflight で無効なエラーが表示される (6376733)	31
Linux ブートおよびインストールサーバーで TFTP サーバーが正しく起動しない (6382810)	31
OS イメージのインポートやプロビジョニングプランが取り消されない (6379004)	32
OS プロビジョニングプラグインのアップグレード後に OS プロビジョニングコンポー	

ネットを表示、管理できない (6391643)	32
OS プロビジョニングプラグインのアップグレード後、OS プロファイルコンポーネントが正しく更新されない (6392983)	32
OS プロビジョニングプラグイン 3.0 へのアップグレード後、Linux または Windows イメージサーバーを再作成できない (6391832)	33
英語以外の Windows OS イメージのプロビジョニングに失敗する (6400419)	34
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)	35
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)	35
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)	35
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)	36
3 OS プロビジョニングの配備環境	37
前提条件	37
基本的な OS プロビジョニング環境	37
ターゲットホスト	38
ネットワーク	39
セキュリティー	41
新しい環境の構成	42
プロセスの概要	42
▼ 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	43
既存の環境の構成	44
ハードウェアおよびソフトウェアの構成	44
ネットワーク環境	44
DHCP サービス	44
ターゲットホスト	45
N1 Service Provisioning System ソフトウェア	45
既存の Solaris JET 環境	45
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	47
OS プロビジョニングプラグインのインストール	47
OS プロビジョニングプラグインの入手	47
N1 SPS への OS プロビジョニングプラグインの追加	49
▼ OS プロビジョニングプラグインをアップグレードする	50
N1 SPS 5.1 リリースでの OS プロビジョニングプラグイン 3.0 の使用	51

▼ 旧バージョンのコンポーネントにアクセスする	52
OS プロビジョニングサーバーの作成	52
▼ 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)	52
▼ 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)	55
サブネットの作成と登録	55
▼ 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)	56
▼ 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)	57
▼ 新しいプロビジョニングサブネットを追加する	57
5 Solaris オペレーティングシステムのプロビジョニング	59
JumpStart Enterprise Toolkit (JET) テクノロジ	59
JET モジュール	59
ゾーンのサポート	60
JET を介した Solaris のインストールプロセス	60
Solaris プロビジョニングプロセスの概要	61
Solaris JET サーバーの設定	61
▼ JET Solaris イメージサーバーを作成する	61
Solaris イメージサーバーのコンポーネント変数	63
JET Solaris イメージサーバーへの Solaris パッチのインポート	64
▼ JET Solaris イメージサーバーに Solaris パッチをインポートする	64
Solaris イメージおよびプロファイルの作成	65
▼ Solaris イメージをインポートする	65
JET による Solaris プロビジョニングプロファイルの作成	68
▼ JET を使用して 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)	68
▼ JET を使用して 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)	71
非 JET Solaris プロビジョニングプロファイルの作成	71
▼ 非 JET 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)	71
▼ 非 JET 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)	74
ターゲットホストへの Solaris OS のインストール	75
▼ 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)	75
▼ ターゲットホストに Solaris OS をインストールする (コマンド行インタフェース)	77
▼ Solaris OS のプロビジョニング状態を確認する	77

▼ Solaris OS のプロビジョニング処理を停止する	77
Solaris イメージとプロファイルの削除	78
▼ Solaris OS イメージを削除する	78
▼ Solaris OS プロファイルを削除する	78
JET Solaris サーバーの管理作業	79
▼ N1 SPS RA インストーラをインポートする	79
▼ ブート params に JET サーバーを使用する	80
custom モジュールの使用法	80
カスタムパッチとパッケージメディアの生成	81
カスタムファイル	82
カスタムスクリプト	83
6 RedHat Linux オペレーティング環境のプロビジョニング	85
RedHat Kickstart テクノロジ	85
RedHat Linux プロビジョニングプロセスの概要	85
RedHat Linux サーバーの設定	86
RedHat Linux OS をプロビジョニングするためのネットワークの準備	86
▼ TFTP アクセス権を設定する	86
▼ RedHat Linux のファイルをコピーする	87
RedHat Linux イメージサーバーの作成	88
▼ RedHat Linux イメージサーバーを作成する (ブラウザインタフェース)	88
▼ RedHat Linux 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)	90
Linux イメージとプロファイルの作成	91
JET による RedHat Linux イメージとプロファイルの作成	91
▼ JET を使用して RedHat 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)	92
▼ JET RedHat 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)	95
非 JET RedHat Linux イメージとプロファイルの作成	96
▼ RedHat Linux OS プロビジョニング用の PXE 構成ファイルを作成する	96
▼ 非 JET RedHat 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)	97
▼ 非 JET RedHat 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)	100
ターゲットホストへの OS のインストール	101
▼ ターゲットホストへ OS をインストールする	101
▼ OS のプロビジョニング状態を確認する	102

RedHat Linux イメージとプロファイルの削除	103
▼ RedHat Linux OS イメージを削除する	103
▼ RedHat Linux OS プロファイルを削除する	103
7 SUSE Linux オペレーティング環境のプロビジョニング	105
SUSE AutoYaST テクノロジー	105
SUSE Linux プロビジョニングプロセスの概要	105
SUSE Linux サーバーの設定	106
SUSE Linux OS をプロビジョニングするためのネットワークの準備	106
▼ TFTP アクセス権を設定する	106
SUSE Linux イメージサーバーの作成	108
▼ SUSE Linux イメージサーバーを作成する (ブラウザインタフェース)	108
▼ SUSE Linux 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)	110
SUSE Linux イメージとプロファイルの作成	111
JET による SUSE Linux イメージとプロファイルの作成	111
▼ JET を使用して SUSE 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)	111
▼ JET SUSE 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)	115
非 JET SUSE Linux イメージとプロファイルの作成	115
▼ SUSE Linux OS プロビジョニング用の PXE 構成ファイルを作成する	115
▼ 非 JET SUSE 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)	116
▼ 非 JET SUSE 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)	119
ターゲットホストへの OS のインストール	120
▼ ターゲットホストへ OS をインストールする	120
▼ OS のプロビジョニング状態を確認する	121
SUSE Linux イメージとプロファイルの削除	121
▼ SUSE Linux OS イメージを削除する	122
▼ SUSE Linux OS プロファイルを削除する	122
8 Windows オペレーティング環境のプロビジョニング	123
Remote Installation Service (RIS) テクノロジー	123
Windows プロビジョニングプロセスの概要	124
Windows サーバーの設定	124
▼ Windows RIS サーバーを設定する	124

▼ 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)	127
▼ 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)	129
Windows イメージサーバーのコンポーネント変数	130
Windows イメージおよびプロファイルの作成	131
▼ Windows RIS サーバー上で Windows イメージを設定する	132
カスタム Windows プロビジョニングプロファイルの作成	133
▼ カスタム 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)	133
▼ カスタム 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)	136
非カスタム Windows プロビジョニングプロファイルの作成	137
▼ Windows イメージの設定情報を構成する	138
▼ 非カスタム 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)	140
▼ 非カスタム 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)	141
Windows プロビジョニングプロファイルの変数	142
ターゲットホストへの Windows OS のインストール	143
▼ ターゲットホストに OS をインストールする	143
Windows のインストール先の管理	144
▼ Windows OS のプロビジョニング状態を確認する	144
▼ Windows OS プロビジョニング処理を停止する	144
▼ N1 SPS にインストール済み Windows ホストを登録する	145
▼ Windows RIS サーバー上のプリステージ状態のホストを削除する	145
9 OS プロビジョニング用ターゲットホスト	147
ターゲットホストタイプ	147
プラットフォームによるターゲットホストタイプ	148
ターゲットホストを定義する作業の例	152
Sun ALOM ターゲットホストの定義	152
▼ 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)	153
▼ 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)	154
Sun IPMI ターゲットホストの定義	155
▼ 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)	155
▼ 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)	157
Sun RSC ターゲットホストの定義	158
▼ Sun RSC ターゲットホストを作成する (ブラウザインタフェース)	158

▼ Sun RSC 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)	160
HP ILO ターゲットホストの定義	161
▼ 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)	161
▼ 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)	163
そのほかの遠隔管理ターゲットホストの定義	163
▼ 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) ..	164
▼ 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)	166
汎用ターゲットホストの定義	167
▼ 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)	167
▼ 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)	169
パスワードの暗号化	169
OS プロビジョニング変数のパスワードの暗号化	169
OS root アクセス権のパスワードの暗号化	170
端末サーバーのサポート	170
10 障害追跡	173
一般的な障害追跡のガイドライン	173
Solaris 関連の問題の解決	175
Linux 関連の問題の解決	176
Windows 関連の問題の解決	177
PXE/DHCP/BINLSVC に関連する問題	178
Remote Installation Service (RIS) に関連する問題	182
GUID に関連する問題	184
A ターゲットホストの変数	187
Sun IPMI ターゲットホストの変数	187
Sun ALOM ターゲットホストの変数	191
Sun RSC ターゲットホストの変数	194
Sun LOM ターゲットホストの変数	197
Sun Mid-Range ターゲットホストの変数	200
HP ILO ターゲットホストの変数	203
そのほかの IPMI ターゲットホストの変数	206
そのほかの遠隔管理ターゲットホストの変数	209
汎用ターゲットホストの変数	215

B Solaris Profile コンポーネントの変数	219
基本 Solaris OS 構成変数	219
base_config モジュール用のコンポーネント変数	219
追加ネットワークインタフェースの構成	227
IP ネットワークマルチパス (IPMP) の構成	228
追加ディスクの構成	229
x86 ベースシステムへの Solaris 9 インストールの自動化	230
▼ Solaris 9 x86 のインストールを自動化する	230
Solaris リモートエージェントのコンポーネント変数	232
一般的な Solaris OS プロビジョニングプラグイン変数	234
C RedHat Linux Profile コンポーネントの変数	237
基本 RedHat Linux OS 構成変数	237
base_config モジュール用のコンポーネント変数	237
RedHat Linux リモートエージェントのコンポーネント変数	251
一般的な RedHat Linux OS プロビジョニングプラグイン変数	253
D SUSE Linux Profile コンポーネントの変数	257
基本 SUSE Linux OS 構成変数	257
base_config モジュール用のコンポーネント変数	257
追加ネットワークインタフェースの構成	278
SUSE Linux リモートエージェントのコンポーネント変数	279
一般的な SUSE Linux OS プロビジョニングプラグイン変数	281
E Microsoft Windows Profile のコンポーネント変数	283
Windows OS コンポーネントの変数	283
F 追加 JET モジュールの作成	289
範囲	289
モジュール設計のガイドライン	289
モジュールの対応範囲	289
モジュールの依存関係	290
モジュールの相互作用	290
モジュールのコーディング	291
モジュールのディレクトリ	291

モジュール構成 (<i>module.conf</i> ファイル)	291
モジュールのインタフェース	292
<i>copy_media</i> インタフェース	292
<i>make_template</i> インタフェース	293
<i>make_client</i> インタフェース	293
<i>begin</i> インタフェース	293
<i>install</i> インタフェース	294
<i>check_client</i> インタフェース	294
ツールキットのサポート関数	295
JET モジュールの追加	295
▼ Flash モジュールを追加する	295
G 高度な構成プロパティ	299
プラグイン提供の DHCP の無効化	299
フォアグラウンドでの配備ジョブの実行	300
新しいターゲットホストの検査	300
索引	303

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	24
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 ..	25
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	25
表 5-1	Solaris イメージサーバーのコンポーネント変数	63
表 8-1	Windows イメージサーバーコンポーネントの変数	130
表 8-2	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	142
表 9-1	プラットフォームによるターゲットホスト	148
表 9-2	端末サーバーの変数と値	171

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 5.2 システム管理者ガイド』
- 『Sun N1 Service Provisioning System 5.2 インストールガイド』
- 『Sun N1 Service Provisioning System 5.2 オペレーションとプロビジョニングガイド』

マニュアルの構成

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、RedHat 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、SUSE 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8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9章では、OS プロビジョニング用として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、RedHat Linux プロビジョニングプロファイルに固有の変数について説明します。

付録Dでは、SUSE Linux プロビジョニングプロファイルに固有の変数について説明します。

付録Eでは、Windows プロビジョニングプロファイルに固有の変数について説明します。

付録F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によって、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。

付録Gでは、プロビジョニングプラン用に高度なプロパティを構成する方法について説明します。

Sun 以外の Web サイト

このマニュアル内で参照している Sun 以外の 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 Sun 以外の Web サイトの可用性については、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 Web サイトでは、以下の追加情報を提供しています。

- マニュアル (<http://jp.sun.com/documentation/>)
- サポート (<http://jp.sun.com/support/>)
- トレーニング (<http://jp.sun.com/training/>)

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。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
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。	machine_name% 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・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。

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が行われたシステム、あるいは現在稼働中のシステムに 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。このプラグインは、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。

- 19 ページの「OS プロビジョニングの概要」
- 20 ページの「機能」
- 20 ページの「プロビジョニングプロセスの概要」
- 21 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」
- 24 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ほか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も対処します。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 のブラウザインタフェースとコマンド行インタフェースを通じて、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、[24 ページの「サポートされるシステム」](#) および [第 3 章](#) で説明しています。
2. [47 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. 52 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 9 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、第 7 章、および第 8 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、RSC、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、そのほか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。

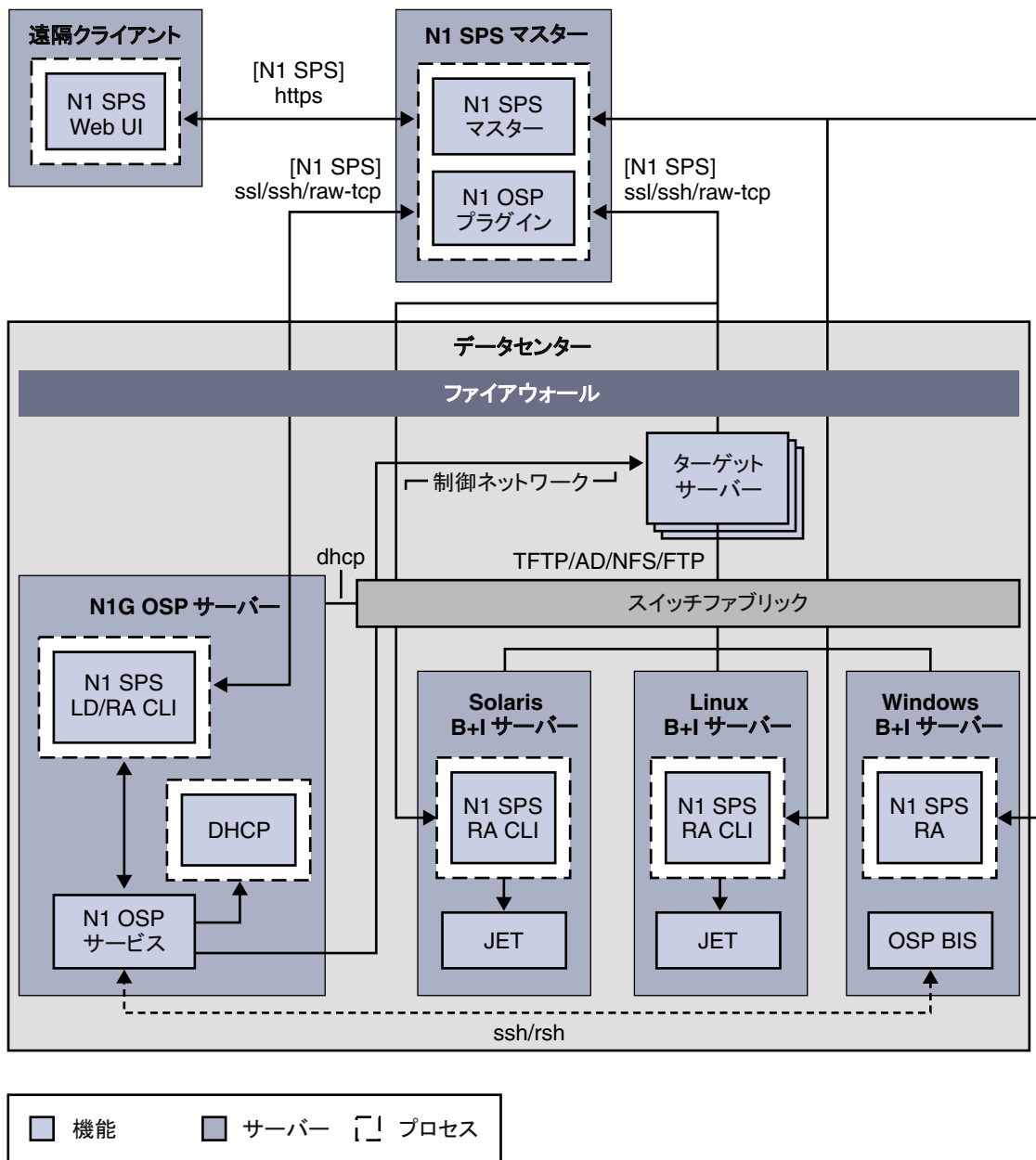


図1-1 ソフトウェアアーキテクチャー図

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。
- N1 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に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 (N1 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、RSC、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 テクノロジーを使用します。
 - 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) テクノロジーを使用します。

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、RSC、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 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)

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	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)

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 8 (SPARC のみ)、Solaris 9 (SPARC および x86)、Solaris 10 (SPARC および x86)、RedHat AS 2.1、RedHat AS 3.0、RedHat AS 4.0、Microsoft Windows Server 2000
OS プロビジョニングサービス	Solaris8 (SPARC のみ)、Solaris 9 (SPARC および x86)、Solaris 10 (SPARC および x86)、RedHat AS 3.0 (32 および 64 ビット)、RedHat AS 4.0 (32 および 64 ビット)
Solaris ブートおよびインストールサーバー	Solaris 9 (SPARC および x86)、Solaris 10 (SPARC および x86)
Linux ブートおよびインストールサーバー	RedHat AS 3.0 (32 および 64 ビット)、RedHat AS 4.0 (32 および 64 ビット)
Windows ブートおよびインストールサーバー	Microsoft Windows 2003 Standard および Enterprise Editions 32 ビット (Service Pack 1 をサポート)

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	JET を介したカスタム JumpStart
RedHat Linux	Kickstart

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 (続き)

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
SUSE Linux	AutoYaST
Microsoft Windows 2003 Server Enterprise Edition 32 ビット (Service Pack 1 をサポート)	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、[147 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。

- SPARC 版 Solaris 8、9、および 10
- x86 版 Solaris 9、および 10
- RedHat AS Linux 3.0 および 4.0 (32 および 64 ビット)
- SUSE Linux 9.0 (32 ビット)
- Microsoft Windows 2000 Server (複数サービスパックをサポート)
- Microsoft Windows 2000 Advanced Server (複数サービスパックをサポート)
- Microsoft Windows Server 2003 Standard および Enterprise Editions (32 および 64 ビット、Service Pack 1 をサポート)
- Microsoft Windows Server 2003 Web Edition (32 ビット、Service Pack 1 をサポート)

OS プロビジョニングプラグイン用のサンプルコマンド行スクリプト

N1 SPS 5.2 DVD の `/plugins/com.sun.n1osp/SampleClis` ディレクトリに、サンプル CLI スクリプトファイルがあります。それらのスクリプトは、次のようなさまざまなタスクを実行します。

- OS プロファイルの作成
- ターゲットホストの作成
- ブートおよびインストールサーバーのインストール
- ターゲットホストへのプロビジョニング

これらのサンプルスクリプトは、実際の環境に固有の CLI コマンドを実行する際の参考にできます。

注-また、Sun では、サンプルスクリプトをサポートしていません。実際の環境に応じて、サンプル内のコマンドおよび引数を自分で編集する必要があります。

◆ ◆ ◆ 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、および既知の問題について説明します。

この章では、次の項目について説明します。

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

インストール上の問題

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。

マルチブートのターゲットホストで **Windows** のインストールモードが対話モードに切り替わる (6360018)

説明: Solaris または Linux OS が動作しているターゲットホストへの Windows OS のプロビジョニングで、システムに Windows OS 用の未使用ディスク領域が十分でない場合は、インストールモードが対話モードに切り替わります。

回避策: 次に示す回避方法のどれか 1 つを選択してください。

- ターゲットホストにプロビジョニングする前に、fdisk ユーティリティを使用し、ターゲットホストから Solaris または Linux パーティションを削除する。

- インストール中にこの問題が発生した場合は、対話画面の指示に従い、DOS FDISK ユーティリティを使用して、既存の未知のパーティションを削除する。

Windows RIS サーバーが DHCP PXE 要求に応答しない (6333612)

説明: Windows RIS サーバー上の最初のインタフェースがプロビジョニングインタフェースとして構成されていない場合、RIS サーバーは、ターゲットホストからの DHCP PXE 要求に応答しません。この問題は、Windows ブートおよびインストールサーバー上の BINLSVC/RIS サーバーが、システムのブート時に構成される最初のインタフェースで待機するために発生します。最初のインタフェースをプロビジョニングインタフェースとして構成していない場合、RIS サーバーはターゲットホストからの DHCP PXE 要求に応答できません。詳細は、178 ページの「PXE/DHCP/BINLSVC に関連する問題」を参照してください。

次のエラーメッセージが表示されます。

```
CLIENT MAC ADDR: 00 09 3D 12 D4 16 GUID: DF5EF8FE E568 11D9 811E 0060B0B37D21
CLIENT IP: 172.0.0.1 MASK: 255.255.255.0 DHCP IP: 172.0.0.3
PXE-E55: ProxyDHCP service did not reply to request on port 4011.
```

PXE-M0F: Exiting Broadcom PXE ROM

回避策: 次に示す回避方法のどれか 1 つを選択してください。

- Windows RIS サーバー上で、プロビジョニングインタフェースだけを使用可能にして、ほかのすべてのインタフェースを使用不可にする。
- Windows RIS サーバー上で、ネットワークサービスがアクセスするネットワーク接続の順序を変更する。次の手順に従ってください。
 1. Windows RIS サーバー上で、「スタート」メニューから「コントロールパネル」ウィンドウを開きます。
 2. 「コントロールパネル」ウィンドウで「ネットワーク接続」をクリックします。
 3. 「ネットワーク接続」画面で「詳細」タブを選択します。
 4. 「詳細設定」ボタンをクリックします。
 5. 「アダプタとバインド」タブをクリックします。
 6. ネットワークサービスがアクセスする接続の順序を変更します。プロビジョニングインタフェースが最初のインタフェースになっていることを確認します。
 7. BINLSVC を停止し、再起動します。この操作は、Services GUI から、または MS-DOS コマンドウィンドウで次のコマンドを入力することによって行うことができます。

```
> net stop binlsvc
> net start binlsvc
```

SPARC: Solaris OS プロビジョニングサーバーの作成プランの Detailed Preflight で無効なエラーが表示される (6376733)

説明: Solaris 9 リリースが動作する SPARC システムへの OS プロビジョニングサーバーの作成で、「Run Detailed Preflight」オプションを選択すると、16 個の無効な警告メッセージが表示されます。

それらの警告メッセージの形式は次のとおりです。

```
Error verifying native procedure during preflight for the command
"/opt/SUNWnlsp/en/N1_Service_Provisioning_System/agent/data/systemcomps/com.sun.nlosp/pkgaddutil
/opt/SUNWnlsp/en/N1_Service_Provisioning_System/agent/data/systemcomps/com.sun.nlosp
adminfile package_name". (026084)
  User "root" does not have execute permissions for command
"/opt/SUNWnlsp/en/N1_Service_Provisioning_System/agent/data/systemcomps/com.sun.nlosp/pkgaddutil".
(026079)
```

この例で、*package_name* は、OS プロビジョニングサーバーに追加できないと警告で指摘されているパッケージを表します。

回避策: 無効なエラーメッセージは無視してください。示されたパッケージは OS プロビジョニングサーバーに追加され、サーバーは正しく作成されます。

Linux ブートおよびインストールサーバーで TFTP サーバーが正しく起動しない (6382810)

説明: Solaris ブートおよびインストールサーバーによる Linux OS のプロビジョニングで、TFTP サービスが正しく起動されません。ターゲットホストにプロビジョニングするには、ブートおよびインストールサーバー上の TFTP サービスを手動で起動する必要があります。

回避策: OS ディストリビューション媒体にアクセス可能な状態にし、ブートおよびインストールサーバー上で次の手順を行なってください。

1. IP アドレスを plumb します。
2. NFS share をエクスポートします。
3. TFTP サービスを起動します。

TFTP サービスの起動方法の詳細については、ご使用の Solaris リリースのシステム管理者マニュアルを参照してください。

OS イメージのインポートやプロビジョニングプランが取り消されない (6379004)

説明: OS イメージのインポートプランあるいは OS プロビジョニングプランを取り消しても、バックグラウンドでプラン処理が継続します。この問題は、次の状況で発生します。

- OS イメージのインポート- イメージのインポートを完了する前に OS イメージのインポートプランを取り消すと、プランは取り消されますが、そのインポートが完了するまで、バックグラウンドでインポート処理が継続されます。
- OS プロビジョニング- フォアグラウンドでプロビジョニング処理を行うように OS プロビジョニングを構成していて、OS プロビジョニングが完了する前に OS プロビジョニングプランを取り消すと、バックグラウンドでプロビジョニング処理が継続します。

回避策: 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上で、インポートまたはプロビジョニングプロセスを手動で終了してください。

OS プロビジョニングプラグインのアップグレード後に OS プロビジョニングコンポーネントを表示、管理できない (6391643)

説明: OS プロビジョニングプラグイン 3.0 をアップグレードすると、N1 SPS ブラウザインタフェースで、インストールされた OS プロビジョニングコンポーネントを表示、管理できなくなります。

回避策: 各コンポーネントの詳細ページにある「Version History」リンクを使用すると、旧リリースで作成した OS プロビジョニングコンポーネントの詳細情報を表示できます。詳細は、52 ページの「旧バージョンのコンポーネントにアクセスする」を参照してください。

OS プロビジョニングプラグインのアップグレード後、OS プロファイルコンポーネントが正しく更新されない (6392983)

説明: OS プロビジョニングプラグイン 3.0 へのアップグレード後に OS プロファイルコンポーネントを作成し直すと、現在のプラグインバージョンで廃止された変数の設定が、コンポーネントに含まれていることがあります。その場合、コンポーネントの再作成後、コンポーネント変数が正しく更新されず、そのコンポーネントへの参照を含むプロビジョニングプランで問題が起きることがあります。次のようなメッセージが表示されます。


```
The plan (or preflight) "/system/autogen/Solaris9_7_
sparc.standard-inst-provision_start-1141144282798" finished
with 1 failed host(s). (017034)
```

```
Unable to find accessible variable in component referenced by
substitution variable :[container:sysidcfg_network_interface_base_]. (04011
```

この例のエラーは、OS プロビジョニングプラグイン 3.0 の Solaris OS プロファイルコンポーネントに `sysidcfg_network_interface_base_config` 変数が含まれていないことが原因です。

回避策:次に示す回避方法のどれか 1 つを選択してください。

- OS プロビジョニングプラグインのアップグレード後に、以前の OS プロファイルコンポーネントプロファイルを再作成するのではなく、新規に作成する。
- 以前の OS プロファイルコンポーネントを削除し、有効な変数設定を含む OS プロファイルを新規に作成する。
- 以前の OS プロファイルコンポーネントから非推奨の変数の値を削除する。

OS プロビジョニングプラグイン 3.0 へのアップグレード後、Linux または Windows イメージサーバーを再作成できない (6391832)

説明:OS プロビジョニングプラグイン 3.0 へのアップグレード後の Linux または Windows イメージサーバーの再作成で、`installPath` 変数の値を変更すると、再作成に失敗します。次のようなメッセージが表示されます。

```
Problems encountered during plan run or preflight
```

```
The plan (or preflight) "/com/sun/nlosp/untyped/LinuxServer-create"
finished with 1 failed host(s). (017034)
```

```
This host has already been reserved or already exists. (011220)
```

回避策:次に示す回避方法のどれか 1 つを選択してください。

- 旧バージョンのイメージサーバーの `installPath` 変数値が有効な場合は、`installPath` 変数の値をそのまま変更しないで残す。
- OS イメージサーバーに対する `installPath` 変数の値を変更する必要がある場合は、次の手順の説明に従って OS イメージサーバーを新規に作成する。
 - 88 ページの「RedHat Linux イメージサーバーを作成する (ブラウザインタフェース)」
 - 108 ページの「SUSE Linux イメージサーバーを作成する (ブラウザインタフェース)」

- 124 ページの「Windows RIS サーバーを設定する」

新しい OS イメージサーバーを作成したら、次のプラン変数設定を変更します。

- `installPath` 変数に有効な値を設定します。
- `linuxHost` または `windowsHost` 変数に設定されているプランの仮想ホスト名を変更します。次に例を示します。[`target:sys.hostName`]-1-linux。

英語以外の Windows OS イメージのプロビジョニングに失敗する (6400419)

説明: 英語以外の言語で Windows OS イメージをプロビジョニングしようとする、インストールに失敗します。これは、Microsoft RIS サービスがデフォルトで英語の OSChooser ファイルを探すことが原因です。次のようなメッセージが表示されます。

The system cannot find the path specified.

An error occurred on the server. Please notify your administrator.

E:\RemoteInstall\OSChooser\English\LOGIN.OSC

回避策: 次に示す回避方法のどれか 1 つを選択してください。

- OSChooser ファイルを変更する。次の手順に従ってください。
 1. `multilng.osc` ファイルをエディタで開いて、最初のエントリがイメージの言語を示すようにします。
 2. `multilng.osc` ファイル内の `<TITLE>` タグの前に次の行を追加します。


```
<META ACTION=AUTOENTER>
```
 3. `multilng.osc` ファイルを同じディレクトリ内の `welcome.osc` ファイルにコピーします。
 4. `\OSChooser\language\welcome.osc` をエディタで開いて、`<TITLE>` タグの前に次のテキスト行を挿入します。


```
<META ACTION=AUTOENTER>
```
- 英語以外の言語ディレクトリにある OSChooser ファイルを English ディレクトリにコピーします。

```
% cd RemoteInstall\OSChooser
% copy -r Language English
```

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=Clz6pK2b6qw=  
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/n1osp* 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。

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。

- 37 ページの「前提条件」
- 42 ページの「新しい環境の構成」
- 44 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キング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ている Solaris または RedHat 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。OS プロビジョニングサーバーでサポートされる OS リリースについては、25 ページの「OS プロビジョニングコンポーネント」の一覧を参照してください。

Solaris OS プロビジョニングサーバーの場合は、OS プロビジョニングサーバーを作成する前に Java 1.4.2 リリースをシステムにインストールする必要があります。この Java SE 1.4.2 リリースは、`/usr/bin/java` ディレクトリに置くか、`/usr/bin/java` ディレクトリにリンクさせます。

RedHat OS プロビジョニングサーバーの場合は、OS プロビジョニングサーバーを作成する前に 32 ビット版 Java 1.4.2 および dhcp-3.01-10_EL3 RPM をシステムにインストールする必要があります。この Java 1.4.2 リリースは、`/usr/bin/java` ディレクトリに置か、`/usr/bin/java` ディレクトリにリンクさせます。64 ビット Java バージョンをインストールした場合は、`/opt/SUNWn1osp/lib/libgridos_rhel3_x86_64.so` に `/opt/SUNWn1osp/lib/libgridos.so` をリンクさせます。

- ブートおよびインストールサーバー - ターゲットホストにプロビジョニングする OS イメージを提供するサーバー。環境に応じて、ブートおよびインストールサーバーの種類を選択します。
 - 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 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 または Solaris 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 Service Pack 1 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。

プロビジョニングソフトウェアを使用してターゲットホストにプロビジョニングするには、ブートおよびインストールサーバーのネットワークインタフェースを構成しておく必要があります。

- 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、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。ターゲットの定義については、第 9 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に設計されています。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。

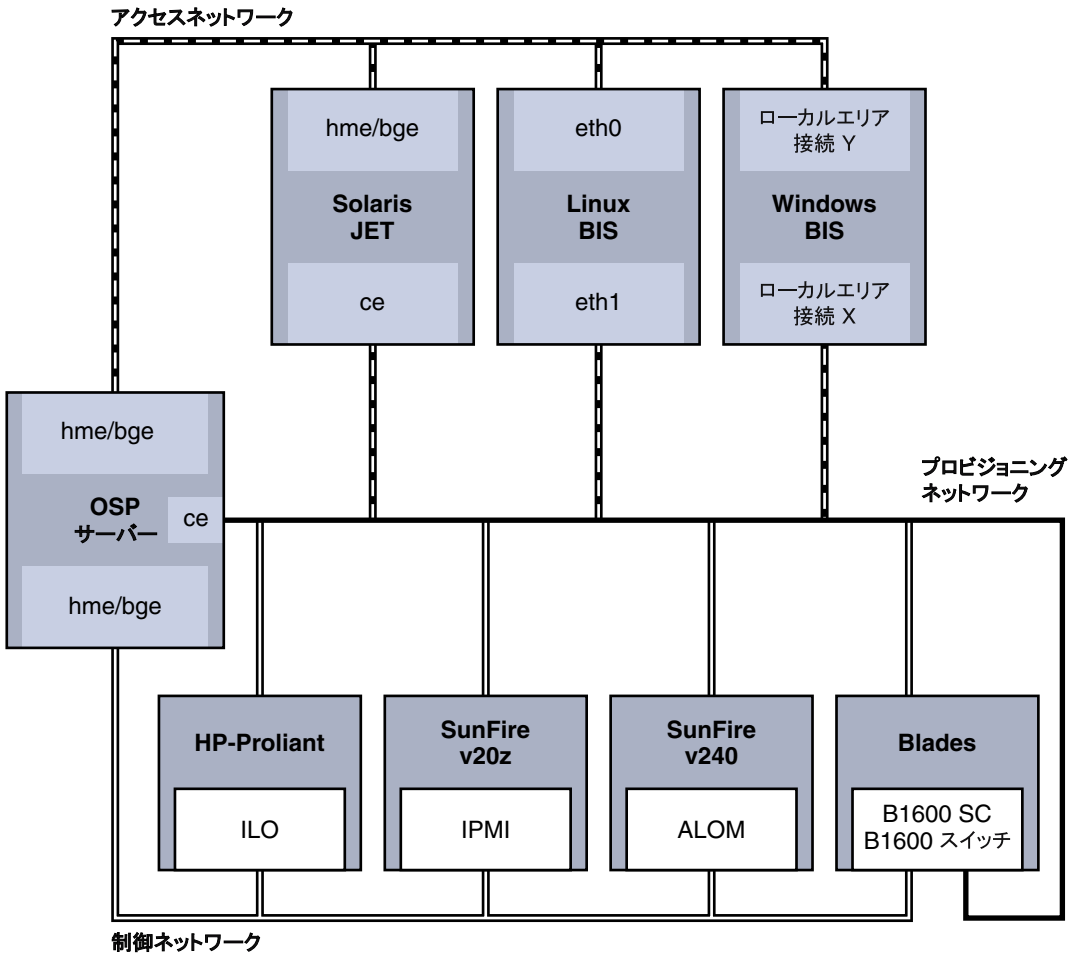


図 3-1 OS プロビジョニング用のネットワーク環境図

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。

セキュリティー

OSプロビジョニングプラグインソフトウェアはN1 SPSセキュリティー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。セキュリティー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ー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、[169 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。[124 ページの「Windows RIS サーバーを設定する」](#)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。
3. 『Sun N1 Service Provisioning System 5.2 インストールガイド』の「Sun N1 Service Provisioning System 5.2 のインストール」の説明に従って、N1 SPS マスターサーバーをインストールします。
4. 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。
5. 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。
6. Linux 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。

7. Windows 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。
8. OS プロビジョニングサーバーと Solaris、Linux、および Windows それぞれのブートおよびインストールサーバーで RA を準備します。詳細は、『Sun N1 Service Provisioning System 5.2 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。『Sun N1 Service Provisioning System 5.2 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

- 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_5.2/server/config/config.properties`
- 2 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。
 - a. 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`
 - b. 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=mypssserver,sol10bis`
- 3 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_5.2/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、[24 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。ブートおよびインストールサーバーのネットワークインタフェースを構成します。

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、このサブネットでは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응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응答するよう DHCP を再度アクティブにすることができません。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。

ネットワークの DHCP サービスを使用してターゲットホストをインストールする場合は、OS プロビジョニングサーバーの提供する DHCP サービスを使用不可にできます。ターゲットホストからの DHCP 要求に응答するよう、DHCP サービスを正しく構成する必要があります。OS プロビジョニングの DHCP サービスを使用不可にする方法については、299 ページの「プラグイン提供の DHCP の無効化」を参照してください。

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 9 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、61 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 では、デフォルトのベースディレクトリとして /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、61 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。このプロセスでは次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。

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。

- N1 SPS ソフトウェアのインストール (『Sun N1 Service Provisioning System 5.2 インストールガイド』で解説)
- 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。

- 47 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」
- 52 ページの「OS プロビジョニングサーバーの作成」
- 55 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の入手の手順は2段階に分かれます。はじめに OS プロビジョニングプラグイン JAR ファイルが含まれたパッケージファイルをシステムに追加する必要があります。そのあとに OS プロビジョニングプラグイン JAR ファイルをインストールします。マスターサーバーにプラグインを追加する手順は、次のとおりです。

1. JAR ファイルを含むファイルを追加します。

OS プロビジョニングプラグインは、N1 Service Provisioning System に対するプラグインとして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用のプラグインファイルは、N1 SPS 5.2 DVD または Sun Download Center (<http://sunsolve.sun.com>) から入手できます。

プラグインパッケージをシステムに追加する方法については、次の手順を参照してください。

- 48 ページの「Solaris への OS プロビジョニングプラグインの追加」
- 48 ページの「Linux への OS プロビジョニングプラグインの追加」
- 49 ページの「Windows への OS プロビジョニングプラグインの追加」

2. JAR ファイルをインポートします。

パッケージファイルをシステムに追加すると、OS プロビジョニングプラグインは2種類の JAR ファイルからインポートできます。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はじめてインポートする場合は、
`n1-basedir/com.sun.n1osp/com.sun.n1osp_3.0.jar` ファイルを入手します。
- 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インポートされている場合は、`n1-basedir/com.sun.n1osp/Upgrade/com.sun.n1osp_2.0_3.0.jar` ファイルを入手します。

プラグイン JAR ファイルのインポート方法については、49 ページの「N1 SPS への OS プロビジョニングプラグインの追加」を参照してください。

Solaris への OS プロビジョニングプラグインの追加

プラグイン製品名プラグインは、SUNWspssosp パッケージに含まれています。

▼ Solaris 用の OS プロビジョニングプラグインパッケージを追加する

- 1 端末ウィンドウで、スーパーユーザーになります。
- 2 プラグインパッケージを含むディレクトリに移動します。
- 3 次のコマンドを入力し、Return キーを押します。

```
# pkgadd -d package_directory SUNWspssosp
```

スタンドアロン用 JAR ファイルは `/opt/SUNWn1sps/plugins/com.sun.n1osp/` ディレクトリにあります。アップグレード用 JAR ファイルは `/opt/SUNWn1sps/plugins/com.sun.n1osp/Upgrade` ディレクトリにあります。

Linux への OS プロビジョニングプラグインの追加

プラグイン製品名プラグインは、`sun-spsosp-3.0-1.noarch.rpm` ファイルに含まれています。

▼ Linux 用の OS プロビジョニングプラグインパッケージを追加する

- 1 端末ウィンドウで、スーパーユーザーになります。
- 2 `sun-spsosp-3.0-1.noarch.rpm` ファイルを含むディレクトリに移動します。
- 3 次のコマンドを入力し、**Return** キーを押します。

```
# rpm -i package_directory sun-spsosp-3.0-1.noarch.rpm
```

スタンドアロン用 JAR ファイルは

`/opt/sun/N1_Service_Provisioning_System/plugins/com.sun.n1osp/` ディレクトリにあります。アップグレード用 JAR ファイルは

`/opt/sun/N1_Service_Provisioning_System/plugins/com.sun.n1osp/Upgrade` ディレクトリにあります。

Windows への OS プロビジョニングプラグインの追加

Windows 用のプラグイン製品名プラグインは、Microsoft Installer (MSI) パッケージファイルの `sun-spsosp-3.0.msi` に含まれています。

▼ Windows 用の OS プロビジョニングプラグイン MSI ファイルを追加する

- 1 `sun-spsosp-3.0.msi` ファイルを含むディレクトリに移動します。
- 2 `sun-spsosp-3.0.msi` ファイルをダブルクリックします。
インストーラの GUI が起動されます。JAR ファイルは、`c:\Program Files\N1 Service Provisioning System\plugins\com.sun.n1osp` ディレクトリにコピーされます。

N1 SPS への OS プロビジョニングプラグインの追加

ある特定のプラグインを N1 SPS 製品に認識させるには、そのプラグインをインポートする必要があります。プラグインをインポートするには、『Sun N1 Service Provisioning System 5.2 システム管理者ガイド』の第 5 章「プラグインの管理」で詳しく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 ファイルのある場所に移動します。
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、プラグインの詳細ページが開いて、そのプラグインが提供するオブジェクトが表示されます。

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.nlosp_3.0.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.nlosp_2.0_3.0.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2.0からバージョン3.0へアップグレードするには、次の手順に従います。

- 1 『Sun N1 Service Provisioning System 5.2 インストールガイド』の第10章「Sun N1 Service Provisioning System 5.2 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5.2にアップグレードします。
- 2 com.sun.nlosp_2.0_3.0.jar ファイルを入手し、インポートします。
OS プロビジョニングプラグインをインストールする方法については、47 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」を参照してください。
- 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3.0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。52 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 環境内のすべてのOS 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3.0ソフトウェアを使用して、既存のすべてのOS イメージサーバーを再度作成する必要があります。
 - JET Solaris イメージサーバーの再作成方法については、61 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。
 - Red Hat Linux イメージサーバーの作成方法については、88 ページの「Red Hat Linux イメージサーバーの作成」を参照してください。

- **JET SUSE Linux** イメージサーバーの作成方法については、[108 ページ](#)の「**SUSE Linux イメージサーバーの作成**」を参照してください。
- **Windows** イメージサーバーの作成方法については、[124 ページ](#)の「**Windows サーバーの設定**」を参照してください。

注- ブートおよびインストールサーバーの再作成では、以前に `installPath` 変数に使用したのと同じ値を使用する必要があります。以前の `installPath` 変数値が有効でなくなっている場合は、有効な `installPath` 変数値を使用して、ブートおよびインストールサーバーを新規に作成する必要があります。

5 既存のすべての 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。既存のプロファイルをエディタで開き、`product_version_spsra` 変数の値を 5.2 に変更します。この変数の詳細については、[232 ページ](#)の「**Solaris 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。

注意事項

- Solaris OS の OS プロビジョニングプラグインパッケージのインストールに失敗した場合は、いったんパッケージを削除して、インストールをやり直してください。
- OS プロビジョニングプラグイン 3.0 で新しい変数を使用する場合は、N1 SPS 環境内にターゲットホストを再作成する必要があります。ターゲットホストの作成方法の詳細は、[第 9 章](#)を参照してください。

既存のターゲットホストの定義をそのまま使用する場合は、OS プロビジョニングプラグイン 3.0 を使用して、それらのホストに引き続きプロビジョニングすることができます。

N1 SPS 5.1 リリースでの OS プロビジョニングプラグイン 3.0 の使用

N1 SPS 5.2 リリースからは、OS プロビジョニングプラグイン 3.0 が N1 SPS 5.2 のパスにデフォルト値を使用するよう変更されています。N1 SPS 5.1 リリースの OS プロビジョニングプラグインを使用している場合は、OS プロビジョニングプラグイン 3.0 が有効になるように、プラン内の `sps_cli` 変数値を編集する必要があります。この後の手順で、この値を編集してください。

- [52 ページ](#)の「**OS プロビジョニングサーバーの作成**」
- [61 ページ](#)の「**Solaris JET サーバーの設定**」
- [88 ページ](#)の「**RedHat Linux イメージサーバーの作成**」
- [108 ページ](#)の「**SUSE Linux イメージサーバーの作成**」

▼ 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、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可能性があります。旧バージョンのコンポーネントを表示または使用するには、次の手順に従います。

- 1 「Common Tasks」ページで、適切なコンポーネントプロシージャーをクリックします。「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。
 - コンポーネントの機能を実行するには、そのバージョン番号をクリックするか、使用するコンポーネントに適用される「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知るには、「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。

- 始める前に
- Solaris OS が動作する OS プロビジョニングサーバーの場合は、プロビジョニングサーバーを作成する前に、システムに Java 2 SE 1.4.2 リリースをインストールする必要があります。この Java 2 SE 1.4.2 リリースは、`/usr/bin/java` ディレクトリに置くか、`/usr/bin/java` ディレクトリにリンクさせます。

- RedHat Enterprise Linux が動作する OS プロビジョニングサーバーの場合は、プロビジョニングサーバーを作成する前に、Java 2 SE 1.4.2 32 ビットリリースと RedHat ディストリビューション用の DHCP RPM をシステムにインストールする必要があります。この Java 2 SE 1.4.2 リリースは、`/usr/bin/java` ディレクトリに置くか、`/usr/bin/java` ディレクトリにリンクさせます。システムに 64 ビット Java バージョンがインストールされている場合は、`/opt/SUNWn1osp/lib/libgridos.so` を `/opt/SUNWn1osp/lib/libgridos_rhel3_x86_64.so` にリンクさせます。OS プロビジョニングサーバーのホスト名がシステム上のループバックアドレスではなく、主インタフェースに解決されることを確認します。
- 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
<code>installPath</code>	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。
<code>vhost_name</code>	OS プロビジョニングサーバーを表す仮想ホストの名前。
<code>host_dir</code>	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。
<code>profile_dir</code>	OS プロファイルが存在するディレクトリへのパス。

変数名	説明
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。
run_provisioning_foreground	<p>プロビジョニング処理をフォアグラウンドで実行するかどうかの指定。</p> <p>この変数を <code>true</code> に設定した場合は、あらゆる配備がフォアグラウンドで実行されます。個々の配備に合わせてこの値をリセットすることはできません。</p> <p>注- この機能を有効にするには、物理ホストのロックを無効にする必要がありますが、この構成はサポートされていません。ホストの物理的なロックを無効にする方法については、300 ページの「フォアグラウンドでの配備ジョブの実行」を参照してください。</p>
java_bin	Java バイナリへのパス。
dhcpd_bin	DHCP デーモンバイナリへのパス。
sps_cli	<p>N1 SPS コマンド行インタフェース (CLI) バイナリの場所。</p> <p>注- N1 SPS 5.1 リリースで OS プロビジョニングプラグイン 3.0 を使用する場合は、環境内の <code>cr_cli</code> ディレクトリの場所を示すようにこの値を編集する必要があります。デフォルト値を使用しないでください。</p>

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。

参照 環境によっては、上記手順で作成、起動される ISC DHCP サーバーではなく、ネットワークの DHCP サーバーの使用が望ましいことがあります。プロビジョニングソフトウェアによって設定された ISC DHCP サーバーを使用不可にする場合は、[299 ページ](#)の「プラグイン提供の DHCP の無効化」を参照してください。

注意事項 OS プロビジョニングサーバーがブートおよびインストールサーバーと同じ物理ホスト上に存在する場合は、ターゲットホストへのプロビジョニングの前にそのシステムのインタフェースを構成する必要があります。OS プロビジョニングサーバーは、ブートおよびインストールサーバーのインタフェースを自動的に構成しません。

▼ 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)

- 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/n1osp/untyped/Service-create" \  
-tar H:NM:masterserver,H:NM:bikickstart -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す。

参照 環境によっては、上記手順で作成、起動される ISC DHCP サーバーではなく、ネットワークの DHCP サーバーの使用が望ましいことがあります。プロビジョニングソフトウェアによって設定された ISC DHCP サーバーを使用不可にする場合は、[299 ページ](#)の「プラグイン提供の DHCP の無効化」を参照してください。

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に、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。

- 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 に割り当てられているホストアドレス	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。

- 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/n1osp/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、56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/n1osp/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。次の手順では、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。ブートおよびインストールサーバーのネットワークインタフェースを構成します。

- 1 56 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。

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。

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

JumpStart Enterprise Toolkit (JET) テクノロジ

OS プロビジョニングプラグインは、JumpStart™ Enterprise Toolkit (JET) テクノロジ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、219 ページの「基本 Solaris 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、232 ページの「Solaris 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、80 ページの「`custom` モジュールの使用法」を参照してください。

ゾーンのサポート

OS プロビジョニングプラグインに用意されている JET 機能は Solaris ゾーンを認識し、非大域ゾーンに影響することなく、大域ゾーンにインストールできます。現在、非大域ゾーンは NFS 共有エクスポートをサポートしていないため、非大域ゾーンでは、JET はサポートされません。

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. プロビジョニングプロファイルを作成します。
4. ターゲットホストを作成します。
5. 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/n1osp/untyped/Jet-create \  
-tar H:NM:biss1 -comp - -vs solaris8 + -pto 30 -nto 10
```

solaris8 引数の部分は、適切なリリース値に置き換えてください。

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。

- 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、63 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。
- 別のコンポーネントの変数設定を使用する場合は、「Import Set From Component」をクリックします。

「Import Variable Settings」ウィンドウが表示されます。

 - a. 必要に応じて、変数設定のインポート元となるコンポーネントが入ったフォルダへ移動します。
 - b. コンポーネントのバージョンを選択します。

注-コンポーネントのバージョンが異なると、変数設定も異なる可能性があります。現在のコンポーネントと、変数設定のインポート元のコンポーネントの間で共通に使用されている変数が存在することを確認します。現在のコンポーネントと、変数設定のインポート元のコンポーネントの間で共通に使用されている変数が存在しない場合、変数設定はインポートされません。

- c. 「Import Variable Settings」をクリックします。

変数設定がインポートされ、表形式で表示されます。
- d. 「Plan Details Run」ページの「Variable Settings」ドロップダウンリストから、インポートした変数設定を選択し、「Select」をクリックします。

- ファイルに保存されているコンポーネント変数設定を使用する場合は、次の手順に従います。
 - a. 「**Import Sets from File**」テキストフィールドに、使用する変数設定ファイルへのパスを入力します。
ファイルシステムをブラウズして適切なファイルを探すには、「ブラウズ」ボタンをクリックします。
 - b. 「**Import**」をクリックします。
変数設定がインポートされ、表形式で表示されます。

注-変数設定のインポート元のファイルと、プランで使用するコンポーネントの間で共通に使用している変数が存在しない場合、変数設定はインポートされません。

- c. 「**Plan Details Run**」ページの「**Variable Settings**」ドロップダウンリストから、インポートした変数設定を選択し、「**Select**」をクリックします。

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。

注意事項 OS プロビジョニングサーバーがブートおよびインストールサーバーと同じ物理ホスト上に存在する場合は、ターゲットホストへのプロビジョニングの前にそのシステム上でインタフェースを構成する必要があります。OS プロビジョニングサーバーは、ブートおよびインストールサーバーのインタフェースを自動的に構成しません。

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

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
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_5.2/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
jetFolder	JET 固有の情報を格納するフォルダ。デフォルトでは、JET フォルダは /com/sun/n1osp/autogen-:[jetHost] にあります。:[jetHost] は、上記手順で定義した仮想 JET ホスト名です。	/com/sun/n1osp/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_

JET Solaris イメージサーバーへの Solaris パッチのインポート

Solaris パッチを入手していて、NFS を介して JET Solaris イメージサーバーがそれらのパッチにアクセスできる場合は、JET Solaris イメージサーバーにそれらのパッチをインポートできます。そして、それらのパッチをプロビジョニングプランに追加できます。

▼ JET Solaris イメージサーバーに Solaris パッチをインポートする

- 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「**JET Solaris Image Servers**」セクションで「**Manage**」をクリックします。
- 3 「**Component Details**」ページの「**Component Procedures**」テーブルで、「import_product_patches」のチェックボックスを選択します。
- 4 「**Actions**」列の「import_product_patches」行で、「**Run**」をクリックします。
- 5 「**Plan Details Run**」ページの「**Current Installations**」リストから、パッチのインポート先のブートおよびインストールサーバーを選択します。
- 6 「**Run Selected Installations**」をクリックします。

- 7 「Plan Details Run」 ページで、このプランに使用する変数を指定します。
 - a. 「JET Module Name」 フィールドで `spsra` モジュールを指定します。
 - b. 製品のバージョンを指定します。
たとえば、5.2 というように指定します。
- 8 パッチのアーキテクチャーを指定します。
- 9 「Media Path」 フィールドでパッチへのパスを指定します。
- 10 「Run (includes preflight)」 をクリックします。

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。

ISO イメージだけでなく、`setup_install_server` コマンドによって作成されたイメージをインポートできます。プランの変数設定をエディタで開いて、インポートするイメージを指定します。

始める前に Solaris 9 OS の配備を完全自動化する場合は、追加の構成ファイルを作成して、Solaris 9 インストールイメージに添える必要があります。詳細は、230 ページの「x86 ベースシステムへの Solaris 9 インストールの自動化」を参照してください。

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

変数名	説明	デフォルト値
<code>media_src</code>	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。	<code>/cdrom/cdrom0/s0</code>
<code>ISO_files</code>	<p><code>media_src</code> ディレクトリにある ISO ファイル名のスペース文字区切りのリスト。ファイルは展開 (<code>unzip</code>) されている必要があります。リストの最初の ISO として起動可能な CD を指定します。</p> <p>注 - Solaris OS のクロスプラットフォーム CD ISO イメージをインポートすることはできません。たとえば、SPARC 版 Solaris OS が動作するブートおよびインストールサーバーに Solaris x86 CD ISO ファイルをインポートすることはできません。Solaris SPARC CD ISO ファイルを Solaris x86 ブートおよびインストールサーバーにインポートすることはできません。</p>	<code>sol-10-GA-sparc-v1-iso</code> <code>sol-10-GA-sparc-v2-iso</code> <code>sol-10-GA-sparc-v3-iso</code> <code>sol-10-GA-sparc-v4-iso</code>

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。

JET による Solaris プロビジョニングプロファイルの作成

JumpStart Enterprise Toolkit (JET) は、Solaris OS のプロビジョニングに必要なプロファイルの作成作業を簡単にする一群のユーティリティです。N1 SPS ブラウザインタフェースを使用し、ターゲットホストへの JET による自動インストールを行うのに必要なプロファイルキーワードおよびプロファイル値を生成できます。

JET は使用せずに、手動でプロファイル値を作成する場合は、JET モジュールを使用しない Solaris プロビジョニングプロファイルを作成できます。71 ページの「非 JET Solaris プロビジョニングプロファイルの作成」を参照してください。

▼ JET を使用して 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、61 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、65 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。
 - x86 ターゲットホストに Solaris OS をプロビジョニングする場合は、既存のすべてのパーティションを削除するカスタム JumpStart プロファイルを作成し、`profile_base_config` 変数でそのプロファイルを指定する必要があります。fdisk キーワードを使用して既存のパーティションを削除する JumpStart プロファイルを作成する方法については、『Solaris 10 インストールガイド (カスタム JumpStart/ 上級編)』を参照してください。
- 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/例
<code>architecture</code>	SPARC や x86 など、物理システムのアーキテクチャー	<code>sparc</code> (デフォルト)
<code>installPath</code>	添付する Solaris イメージ名	<code>Solaris:[version]_[release]_[architecture]</code> 次に例を示します。 <code>Solaris9_u7_sparc</code>
<code>image_path</code>	イメージファイルが存在する場所へのパス	<code>/export/osp_image/:[installPath]</code> (デフォルト)
<code>image_subnet_addr</code>	イメージサーバーのサブネットアドレス	<code>:[target(/):sys.ipAddress]</code> (デフォルト)
<code>image_subnet_mask</code>	イメージサーバーのサブネットマスク	<code>255.255.255.0</code> (デフォルト)
<code>media_src</code>	イメージファイルへのパス。	<code>/cdrom/cdrom0/s0</code> (例)
<code>ISO_files</code>	<code>media_src</code> 変数に指定したディレクトリにある ISO ファイルのリスト	<code>sol-10-GA-sparc-v1-iso</code> <code>sol-10-GA-sparc-v2-iso</code> <code>sol-10-GA-sparc-v3-iso</code> <code>sol-10-GA-sparc-v4-iso</code> (例)

- 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 製品のモジュールは、JET サーバーのディレクトリ `/opt/SUNWjet/Products` にあります。

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、[219 ページの「基本 Solaris 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、[232 ページの「Solaris 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、[80 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 F](#)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** フォルダに移動します。
たとえば、`/com/sun/n1osp/autogen-biss1-jet/provision/` などのフォルダです。
- 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。

注 – x86 ターゲットホストに Solaris OS をプロビジョニングする場合は、そのターゲットホスト上の既存のすべてのパーティションを削除するカスタム JumpStart プロファイルを作成する必要があります。`profile_base_config` 変数に、そのプロファイルの場所を指定します。

- 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。

▼ JETを使用して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)

- 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 "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="
```

- 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/SolarisImage-create-profile
-tar H:NM:biss1-jet -comp - -vs solaris10sparc -pto 30 -nto 10
-f /tmp/solaris-profile
```

-u オプションで指定したデフォルトの N1 SPS ユーザー値を書き留めます。N1 SPS Remote Agent ソフトウェアがターゲットホストで起動するには、OS インストール後に、このユーザーが存在する必要があります。

非 JET Solaris プロビジョニングプロファイルの作成

Solaris プロビジョニングプロファイルを手動で作成する場合は、JET モジュールを使用せずに、Solaris プロビジョニングプロファイルを作成できます。この節では、非 JET Solaris プロファイルを作成する方法を説明します。

JET を使用してプロファイルを生成する場合は、[68 ページの「JET による Solaris プロビジョニングプロファイルの作成」](#)を参照してください。

▼ 非 JET 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)

環境によっては、JET を使用して Solaris プロビジョニングプロファイルを生成するのではなく、手動でプロファイルを作成の方が望ましいことがあります。たとえば、N1 SPS 環境にインポートするカスタム JumpStart 環境が存在する場合、その既存の環境の JumpStart ファイルを取り込むには、OS プロビジョニングプロファイルを手動で作成する必要があります。

始める前に Solaris イメージサーバーが存在し、そのサーバーで Solaris 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。Solaris イメージサーバーの作成の詳細については、[61 ページの「Solaris JET サーバーの設定」](#)を参照してください。

- 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「**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「**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. 変数を確認し、必要に応じて変数を変更します。
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/例
<code>architecture</code>	SPARC や x86 など、物理システムのアーキテクチャー	<code>sparc</code> (デフォルト)
<code>installPath</code>	添付する Solaris イメージ名	<code>Solaris:[version]_[release]_[architecture]</code> 次に例を示します。 <code>Solaris9_u7_sparc</code>
<code>image_path</code>	イメージファイルが存在する場所へのパス	<code>/export/osp_image/:[installPath]</code> (デフォルト)
<code>image_subnet_addr</code>	イメージサーバーのサブネットアドレス	<code>:[target(/):sys.ipAddress]</code> (デフォルト)
<code>image_subnet_mask</code>	イメージサーバーのサブネットマスク	<code>255.255.255.0</code> (デフォルト)

変数名	説明	デフォルト値/例
media_src	イメージファイルへのパス。	
ISO_files	media_src 変数に指定したディレクトリにある ISO ファイルのリスト	

- 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 Solaris イメージサーバーを選択します。

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

- 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 「Use the Jet Modules Specified Above」のチェックボックスを選択解除します。
- 9 「Run Plan (includes preflight)」をクリックします。
- 10 Solaris イメージサーバーに対して Solaris 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。
/com/sun/nlosp/autogen-server-name-solaris/provision/ フォルダに、provision コンポーネントが作成されます。server-name は、変数セットに指定した Solaris サーバー名です。次に例を示します。 /com/sun/nlosp/autogen-Solaris9_u7_sparc-solaris/provision/
- 11 (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、 /com/sun/nlosp/autogen-biss1-jet/provision/ などのフォルダです。
 - 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。

注-x86 ターゲットホストに Solaris OS をプロビジョニングする場合は、そのターゲットホスト上の既存のすべてのパーティションを削除するカスタム JumpStart プロファイルを作成する必要があります。profile_base_config 変数には、このプロファイルの場所を指定します。

- 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。

注-プロビジョニングプランに既存の JumpStart 環境を使用する場合は、JumpStart ファイルのディレクトリと IP アドレスをポイントするように、osp_profile_jumpstart_ip_address 変数と osp_profile_jumpstart_directory 変数の値を設定します。

- iii. 「Details」ページの最下部の「Check-in」ボタンをクリックします。

▼ 非 JET 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)

- 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/n1osp/untyped/SolarisImage \
-name "solaris9" -u admin -p admin -vars "version=9;release=u7;architecture=sparc; \
image_path=/export/install/Solaris_9;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、次のエントリを含む /tmp/solaris-profile ファイルを作成します。

```
standard-nojet
Standard Solaris
base_config spsra
false
```

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行目は、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行目は、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。
- ファイルの4行目は、ブラウザインタフェースのプラン変数セクションの「Use the Jet Modules Specified Above」チェックボックスに対応します。

- 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。

```
# 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
-f /tmp/solaris-profile
```

-u オプションで指定したデフォルトの N1 SPS ユーザー値を書き留めます。N1 SPS Remote Agent ソフトウェアがターゲットホストで起動するには、OS インストール後に、このユーザーが存在する必要があります。

ターゲットホストへの **Solaris 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場合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のある変数のサブセットが含まれています。

- 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** フォルダに移動します。
たとえば、/com/sun/nlosp/autogen-biss1-jet/provision/ などのフォルダです。
- 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。

注意事項 x86 版の Solaris 10 1/06 OS をプロビジョニングする場合は、ブートおよびインストールサーバー上に /tftpboot/I86PC.Solaris_10-1 フォルダが存在することを確認します。このフォルダが存在しない場合は、フォルダを作成して、ターゲットホストへのプロビジョニングをやり直します。

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

- ▶ 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
```

▼ **Solaris OS** のプロビジョニング状態を確認する

- 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 「**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。

▼ **Solaris OS** のプロビジョニング処理を停止する

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

Solaris イメージとプロファイルの削除

不要になった OS イメージやプロファイルは、N1 SPS 環境から定期的に削除することを推奨します。

▼ Solaris OS イメージを削除する

- 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「**Solaris Images**」セクションで「**Manage**」をクリックします。
- 3 「**Components Details**」ページで「**Uninstall: Delete**」のチェックボックスを選択します。
- 4 「**Run**」をクリックします。

N1 SPS 環境から Solaris OS イメージが削除されます。「**View link for Solaris Images**」セクションのイメージリストに、削除したイメージが表示されなくなっています。

注- ブートおよびインストールサーバーから、Solaris OS イメージが削除されるわけではありません。

▼ Solaris OS プロファイルを削除する

- 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「**Solaris Images**」セクションで「**Provision OS**」をクリックします。
- 3 「**Components**」ページで、削除するプロファイルのチェックボックスを選択します。
- 4 「**Delete**」をクリックします。

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。

- 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	5.2
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。232 ページの「Solaris 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。

- 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、167 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、219 ページの「基本 Solaris 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、232 ページの「Solaris 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 F 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。60 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の編集では、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のリストを指定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 F 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見て相対的な場所にあるソースファイル
- 「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` or `/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/ftputers:o:/etc/ftputers"
```

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に複数コピーされるのではなく、`/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場合は、[付録 F](#) で説明しているようにモジュールの作成を検討してください。

RedHat 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、RedHat 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。

- 85 ページの「RedHat Kickstart テクノロジー」
- 85 ページの「RedHat Linux プロビジョニングプロセスの概要」
- 86 ページの「RedHat Linux サーバーの設定」
- 91 ページの「Linux イメージとプロファイルの作成」
- 101 ページの「ターゲットホストへの OS のインストール」

RedHat Kickstart テクノロジー

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

RedHat Linux プロビジョニングプロセスの概要

1. RedHat Linux イメージサーバーを設定します。
2. RedHat 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。

RedHat Linux サーバーの設定

OS プロビジョニングプラグインは、RedHat Linux の Kickstart 機能とともに、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、RedHat Linux 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。RedHat Linux サーバーを設定すると、OS をプロビジョニングするための Kickstart 処理を実行する JET モジュールが作成されます。JET についての詳細は、59 ページの「[JumpStart Enterprise Toolkit \(JET\) テクノロジー](#)」を参照してください。

RedHat 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) または Solaris OS
- TFTP サーバー (tftp-server パッケージ)
- NFS サーバー (nfs-utils パッケージ)
- N1 SPS リモートエージェント (RA) (Linux 版)。詳細は、『Sun N1 Service Provisioning System 5.2 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。

RedHat Linux OS をプロビジョニングするためのネットワークの準備

RedHat Linux イメージサーバーを作成する前に、次の作業を行なっておく必要があります。

- ブートおよびインストールサーバー上の RedHat Linux イメージへの TFTP アクセス権の設定
この方法については、86 ページの「[TFTP アクセス権を設定する](#)」を参照してください。
- RedHat Linux OS インストールイメージの作成
この方法については、87 ページの「[RedHat Linux のファイルをコピーする](#)」を参照してください。

▼ TFTP アクセス権を設定する

JET を使用して、RedHat Linux イメージサーバーを作成した場合、TFTP サービスは自動的にプロビジョニング用に構成されます。ただし、その場合でも、ブートおよびインストールサーバー上の TFTP ルートディレクトリに追加のディレクトリおよびファイルを作成またはコピーする必要があります。OS 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。

始める前に ブートおよびインストールサーバーに適切な TFTP サーバーソフトウェアがインストールされていることを確認してください。

- 1 **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。
PXE 構成ファイルの作成方法については、96 ページの「RedHat Linux OS プロビジョニング用の PXE 構成ファイルを作成する」を参照してください。

注 – この作業では、TFTP ルートディレクトリの 1 例として `/tftpboot` を使用します。

- 2 **JET** を使用しない RedHat Linux プロビジョニングプロファイルの場合は、TFTP ルートディレクトリに次のファイルをコピーします。
 - `vmlinuz` – `/images/pxeboot` にあるディストリビューションのインストールカーネル。
 - `initrd.img` – `/images/pxeboot` にあるディストリビューションのインストール RAM ディスク。
 複数リリースの RedHat Linux OS のプロビジョニングを行えるようにする場合は、このファイルに、`vmlinuz_3.0` などの一意の名前を割り当てます。

▼ RedHat Linux のファイルをコピーする

ここでは、製品媒体から RedHat Linux のファイルをコピーすることによって、その OS イメージを作成する手順を説明します。これらのコピーは、RedHat Linux OS または Solaris OS のいずれかが動作するシステムに配置できます。システムの CD-ROM ドライブに物理製品媒体を挿入し、プロビジョニングプラン用のイメージソースとして CD を指定することもできます。RedHat Linux ブートおよびインストールサーバーにコピーを配置する必要はありません。

ブートおよびインストールサーバーではないシステムにコピーを配置する場合は、次の条件が満たされる必要があります。

- JET を使用する RedHat Linux プロビジョニングの場合は、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が NFS 経由でコピーにアクセスできること。
- JET を使用しない RedHat Linux プロビジョニングの場合は、ターゲットホストが NFS 経由でコピーにアクセスできること。

次の手順に従って、RedHat Linux ファイルのコピーを作成してください。

- 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      by default, will mount on /mnt/cdrom, check /etc/fstab
#mount -o loop /path-to-first-iso /mnt/Loop      assumes /mnt/loop exists
```

- 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。

RedHat Linux イメージサーバーの作成

RedHat Linux イメージをコピーしたあと、N1 SPS 環境内に RedHat Linux イメージサーバーを作成する必要があります。ブートおよびインストールサーバーを N1 SPS リモートエージェントにし、ブートおよびインストールサーバーをマスターサーバーに対するリモートエージェントとして追加します。

▼ RedHat Linux イメージサーバーを作成する(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ーバーを作成するには、次の手順に従います。

- 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、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 や boot_server_tftp_root_directory 変数の値は、変更しないでください。

- d. 変数セットを保存します。
 - e. 「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。
- 別のコンポーネントの変数設定を使用する場合は、「Import Set From Component」をクリックします。
「Import Variable Settings」ウィンドウが表示されます。
 - a. 必要に応じて、変数設定のインポート元となるコンポーネントが入ったフォルダへ移動します。
 - b. コンポーネントのバージョンを選択します。

注 - コンポーネントのバージョンが異なると、変数設定も異なる可能性があります。現在のコンポーネントおよびインポートする変数設定があるコンポーネントが共通の変数を使用していることを確認します。変数設定のインポート元のコンポーネントと、プランで使用するコンポーネントの間で共通に使用している変数が存在しない場合、変数設定はインポートされません。

- c. 「Import Variable Settings」をクリックします。
変数設定がインポートされ、表形式で表示されます。
- d. 「Plan Details Run」ページの「Variable Settings」ドロップダウンリストから、インポートした変数設定を選択し、「Select」をクリックします。

- ファイルに保存されているコンポーネント変数設定を使用する場合は、次の手順に従います。
 - a. 「**Import Sets from File**」テキストフィールドに、使用する変数設定ファイルへのパスを入力します。
ファイルシステムをブラウズして適切なファイルを探すには、「ブラウズ」ボタンをクリックします。
 - b. 「**Import**」をクリックします。
変数設定がインポートされ、表形式で表示されます。

注-変数設定のインポート元のファイルと、プランで使用するコンポーネントの間で共通に使用している変数が存在しない場合、変数設定はインポートされません。

- c. 「**Plan Details Run**」ページの「**Variable Settings**」ドロップダウンリストから、インポートした変数設定を選択し、「**Select**」をクリックします。
- 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。

この手順によって、ブートおよびインストールサーバーに SUNWjet と JetRedHat、JetSuse パッケージがインストールされます。SUNWjet パッケージは、TFTP サーバーの設定と、JET ベースディレクトリ (デフォルトでは /opt/SUNWjet) ディレクトリの NFS 共有設定を行います。

注意事項 OS プロビジョニングサーバーがブートおよびインストールサーバーと同じ物理ホスト上に存在する場合は、ターゲットホストへのプロビジョニングの前にそのシステム上でインタフェースを構成する必要があります。OS プロビジョニングサーバーは、ブートおよびインストールサーバーのインタフェースを自動的に構成しません。

▼ RedHat Linux 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)

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ーフェースから RedHat Linux イメージサーバーを作成できます。

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

- 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
```

これらのコマンドによって、ブートおよびインストールサーバーに SUNWjet、JetRedHat、および JetSuse パッケージがインストールされます。SUNWjet パッケージは、TFTP サーバーの設定と、JET ベースディレクトリ (デフォルトでは /opt/SUNWjet) ディレクトリの NFS 共有設定を行います。

注意事項 OS プロビジョニングサーバーがブートおよびインストールサーバーと同じ物理ホスト上に存在する場合は、ターゲットホストへのプロビジョニングの前にそのシステム上でインタフェースを構成する必要があります。OS プロビジョニングサーバーは、ブートおよびインストールサーバーのインタフェースを自動的に構成しません。

Linux イメージとプロファイルの作成

RedHat Linux イメージサーバーを作成したあとは、プロビジョニングイメージおよびプロファイルを作成する必要があります。

- JumpStart Enterprise Toolkit (JET) でイメージおよびプロファイルを生成する場合は、91 ページの「JET による RedHat Linux イメージとプロファイルの作成」を参照してください。
- イメージおよびプロファイルを手動で作成する場合は、96 ページの「非 JET RedHat Linux イメージとプロファイルの作成」を参照してください。

JET による RedHat Linux イメージとプロファイルの作成

JumpStart Enterprise Toolkit (JET) は、RedHat Linux OS のプロビジョニングに必要なプロファイルの作成作業を簡単にする一群のユーティリティーです。N1 SPS ブラウザインタフェースを使用して、ターゲットホストの Kickstart インストールを自動的に行うのに必要なプロファイルキーワードおよびプロファイル値を生成できます。

JET を使用せずに、手動でプロファイル値を作成する場合は、JET モジュールを使用しない RedHat Linux イメージおよびプロファイルを作成できます。96 ページの「非 JET RedHat Linux イメージとプロファイルの作成」を参照してください。

▼ JET を使用して RedHat 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 RedHat 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。RedHat Linux イメージサーバーの作成方法の詳細については、86 ページの「RedHat Linux サーバーの設定」を参照してください。

- 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「RedHat 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。
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/例
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 (デフォルト)
version	このコンポーネントで配備する Linux のバージョン	3U4 (例)

変数名	説明	デフォルト値/例
architecture	x86 などの、物理システムのアーキテクチャー 注-この値は、プロファイルコンポーネントおよびOSイメージの名前に使用されません。プロビジョニングソフトウェアによって、この変数が検査、適用されることはありません。	x86 (デフォルト)
installPath	添付する RedHat Linux イメージの名前	Linux:[version]_[architecture] 次に例を示します。 Linux_3U4_x86
description	このプロファイルに関する説明 (任意)	test linux (例)
min_disk_size	RedHat Linux OS のプロビジョニングに必要な最小ディスク容量	10GB (デフォルト)
boot_server_name	RedHat Linux ブートおよびインストールサーバーの名前	masterserver (例)
boot_server_subnet_address	ブートおよびインストールサーバーがサービスを提供するサブネット	10.10.31.0 (例)
boot_server_subnet_ip_address	サブネット上のブートおよびインストールサーバーの IP アドレス	10.10.31.1 (例)
image_subnet_mask	プロビジョニングサブネットのネットマスク	255.255.255.0 (デフォルト)
kickstart_file (非 JET プロファイルのみ)	ブートサーバー上の Kickstart ファイル ks.cfg の絶対パス。 JET 生成プロファイルの場合は、この変数に値を指定しないでください。	なし

変数名	説明	デフォルト値/例
pxe_boot_file (非 JET プロファイルのみ)	ブートサーバー上の pxelinux.0 ブートファイル の、/tftpboot ディレクトリ からの相対パス JET 生成プロファイルの場合 は、この変数に値を指定しな いください。	なし
pxe_configuration_file (非 JET プロファイルのみ)	ブートサーバー上の PXE Linux 構成ファイルの、 pxelinux.cfg ディレクトリか らの相対パス JET 生成プロファイルの場合 は、この変数に値を指定しな いください。	なし
linux_component_folder	自動生成 Linux Provision コン ポーネントフォルダの位置	/com/sun/nlosp/autogen-:[target(..): sys.hostName]-linux (デ フォルト)
image_path (JET プロファイルのみ)	OS イメージへの NFS パス。 非 JET プロファイルの場合 は、この変数に値を指定しな いください。	nfs://10.10.31.1/export/install/RHEL3 (例)
kernel (JET プロファイルのみ)	OS イメージ内のカーネルへ の相対パス。 非 JET プロファイルの場合 は、この変数に値を指定しな いください。	images/pxeboot/vmlinuz (デ フォルト)
initrd (JET プロファイルのみ)	OS イメージ内の initrd への 相対パス。 非 JET プロファイルの場合 は、この変数に値を指定しな いください。	images/pxeboot/initrd.img (デフォルト)

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 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 製品のモジュールは、JET サーバーのディレクトリ `/opt/SUNWjet/Products` にあります。
デフォルトでは、プラン変数 `JET Modules Name(s)` に、RedHat Linux 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` - RedHat Linux 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、237 ページの「基本 RedHat Linux 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、251 ページの「RedHat Linux 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 RedHat Linux 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 RedHat Linux プロビジョニングプロファイルに追加されます。
- 9 「Use the Jet Modules Specified Above」チェックボックスが選択されていることを確認します。
- 10 「Run Plan (includes preflight)」をクリックします。
- 11 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/n1osp/autogen-server-name-linux/provision/` に、`provision` コンポーネントがあります。ここで `server-name` は、変数セットに指定した Linux サーバー名です。次に例を示します。 `/com/sun/n1osp/autogen-Linux3U4_x86-linux/provision/`。

▼ JET RedHat 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)

- 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/n1osp/untyped/LinuxImage \
-name "redhat3" -u admin -p admin -vars "osp_server=masterserver-osp; \
version=RedHat3.0;architecture=x86; \
```

```
installPath=Linux:[version]_[architecture];description=test linux; \
min_disk_size=10GB;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 \
-f /tmp/redhat-profile
```

-u オプションで指定したデフォルトの N1 SPS ユーザー値を書き留めます。OS インストール後に N1 SPS リモートエージェントソフトウェアがターゲットホストで起動するには、このユーザーが存在する必要があります。

非 JET RedHat Linux イメージとプロファイルの作成

RedHat Linux イメージおよびプロファイルを手動で作成する場合は、JET モジュールを使用せずに、RedHat Linux イメージおよびプロファイルを作成できます。この節では、非 JET RedHat Linux イメージとプロファイルの作成方法を説明します。

JET を使用して、イメージおよびプロファイルを作成する場合は、[91 ページの「JET による RedHat Linux イメージとプロファイルの作成」](#)を参照してください。

▼ RedHat Linux OS プロビジョニング用の PXE 構成ファイルを作成する

ターゲットホストに非 JET RedHat Linux イメージを配備するには、手動で PXE 構成ファイルを作成する必要があります。このファイルには、各ハードウェアプラットフォームおよび RedHat 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ます。

- 1 テキストエディタでファイルを作成し、次のテキストを入力します。

```
default el30 ks
serial 0 9600
label el30
kernel vmlinuz
append linksleep=30 ksdevice=eth0 console=ttyS1,9600 load_ramdisk=1 initrd=initrd.img network
```

各行の意味は次のとおりです。

- 先頭行 (default) は、ラベル el30 を使用し、Kickstart インストール (カーネルに ks を渡す) を行うことを示しています。
- 3 行目 (label) は、ラベル el30 の定義です。

- 4行目 (kernel) は、読み込むカーネルの指定です。パスは `pxelinux.0` に対して相対的です。
- 5行目 (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。

`linkslp=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` インタフェースを使用してインストールを行うことを示しています。このパラメータが省略されている場合は、`eth0` が想定されます。

`console=ttyS1,9600` は、コンソールにインストール出力を表示するためのコンソールパラメータを指定しています。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。

- 2 ブートおよびインストールサーバー上の TFTP ルートディレクトリの `pxelinux.cfg` サブディレクトリに、このファイルを保存します。

複数リリースの RedHat Linux OS のプロビジョニングを行えるようにする場合は、このファイルに、`rh30.cfg` などの一意の名前を割り当てます。

▼ 非 JET RedHat 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 RedHat 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。RedHat Linux イメージサーバーの作成方法の詳細については、[86 ページの「RedHat Linux サーバーの設定」](#)を参照してください。

PXE Linux 構成ファイルを作成し、ブートおよびインストールサーバー上の `pxelinux.cfg` ディレクトリにそのファイルを配置します。PXE Linux 構成ファイル例については、[96 ページの「RedHat Linux OS プロビジョニング用の PXE 構成ファイルを作成する」](#)を参照してください。

- 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「RedHat 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。

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/例
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 (デフォルト)
version	このコンポーネントで配備する Linux のバージョン	3U4 (例)
architecture	x86 など、物理システムのアーキテクチャー	x86 (デフォルト)
	注-この値は、プロファイルコンポーネントおよび OS イメージの名前に使用されます。プロビジョニングソフトウェアによって、この変数が検査、適用されることはありません。	
installPath	添付する RedHat Linux イメージの名前	Linux:[version]_[architecture] 次に例を示します。 Linux_3U4_x86
description	このプロファイルに関する説明 (任意)	test linux (例)
min_disk_size	RedHat Linux OS のプロビジョニングに必要な最小ディスク容量	10GB (デフォルト)
boot_server_name	RedHat Linux ブートおよびインストールサーバーの名前	masterserver (例)

変数名	説明	デフォルト値/例
<code>boot_server_subnet_address</code>	ブートおよびインストールサーバーがサービスを提供するサブネット	10.10.31.0 (例)
<code>boot_server_subnet_ip_address</code>	サブネット上のブートおよびインストールサーバーの IP アドレス	10.10.31.1 (例)
<code>image_subnet_mask</code>	プロビジョニングサブネットのネットマスク	255.255.255.0 (デフォルト)
<code>kickstart_file</code> (非 JET プロファイルのみ)	ブートサーバー上の Kickstart ファイル <code>ks.cfg</code> の絶対パス このファイルは、ブートおよびインストールサーバーにアクセスできる必要があります。 JET 生成プロファイルの場合は、この変数に値を指定しないでください。	/tftpboot/v6x/as21.cfg (例)
<code>pxe_boot_file</code> (非 JET プロファイルのみ)	ブートサーバー上の /tftpboot ディレクトリを基準にした <code>pxelinux.0</code> ブートファイルの相対パス JET 生成プロファイルの場合は、この変数に値を指定しないでください。	v6x/pxelinux.0 (例)
<code>pxe_configuration_file</code> (非 JET プロファイルのみ)	ブートサーバー上の <code>pxelinux.cfg</code> ディレクトリを基準にした PXE Linux 構成ファイルの相対パス JET 生成プロファイルの場合は、この変数に値を指定しないでください。	as21-console (例)
<code>linux_component_folder</code>	自動生成 Linux Provision コンポーネントフォルダの位置	/com/sun/n1osp/autogen-: [target(...):sys.hostName]-linux (デフォルト)
<code>image_path</code> (JET プロファイルのみ)	OS イメージへの NFS パス。 非 JET プロファイルの場合は、この変数に値を指定しないでください。	なし

変数名	説明	デフォルト値/例
kernel (JET プロファイルのみ)	OS イメージ内のカーネルへの相対パス。 非 JET プロファイルの場合は、この変数に値を指定しないでください。	なし
initrd (JET プロファイルのみ)	OS イメージ内の initrd への相対パス 非 JET プロファイルの場合は、この変数に値を指定しないでください。	なし

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。

- 「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。

- 「Use the Jet Modules Specified Above」 のチェックボックスを選択解除します。

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

- 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。

/com/sun/nlosp/autogen-*server-name*-linux/provision/ フォルダに、provision コンポーネントがあります。*server-name* は、変数セットに指定した Linux サーバー名です。次に例を示します。 /com/sun/nlosp/autogen-Linux3U4_x86-linux/provision/

▼ 非 JET RedHat 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)

- 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=test linux;min_disk_size=10GB; \  
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 JET モジュール情報を提供するには、次のエントリを含む /tmp/redhat-profile ファイルを作成します。

```
standard-nojet
Standard Linux
base_config
false
```

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行目は、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行目は、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。
- ファイルの4行目は、ブラウザインタフェースのプラン変数セクションの「Use the Jet Modules Specified Above」チェックボックスに対応します。

- 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。

```
# 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
-f /tmp/redhat-profile
```

-u オプションで指定したデフォルトのN1 SPS ユーザー値を書き留めます。OS インストール後にN1 SPS リモートソフトウェアがターゲットホストで起動するには、このユーザーが存在する必要があります。

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

始める前に ブートおよびインストールサーバー上のネットワークインタフェースが適切に構成されていることを確認します。

- 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** フォルダに移動します。
たとえば、/com/sun/n1osp/autogen-bikickstart-linux/provision/LinuxRedHat3.0_x86 などのフォルダです。
- 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

- 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。

RedHat Linux イメージとプロファイルの削除

不要になった OS イメージやプロファイルは、N1 SPS 環境から定期的に削除するとよいでしょう。

▼ RedHat Linux OS イメージを削除する

- 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「**RedHat Linux Images**」セクションで「**Manage**」をクリックします。
- 3 「**Components Details**」ページで「**Uninstall: Delete**」のチェックボックスを選択します。
- 4 「**Run**」をクリックします。

N1 SPS 環境から RedHat Linux OS イメージが削除されます。「**View link for RedHat Linux Images**」セクションのイメージリストから、削除したイメージが削除されています。

注- ブートおよびインストールサーバーから、RedHat Linux OS イメージが削除されるわけではありません。

▼ RedHat Linux OS プロファイルを削除する

- 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「**RedHat Linux Images**」セクションで「**Provision OS**」をクリックします。
- 3 「**Components**」ページで、削除するプロファイルのチェックボックスを選択します。
- 4 「**Delete**」をクリックします。

SUSE 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、SUSE 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。

- 105 ページの「SUSE AutoYaST テクノロジー」
- 105 ページの「SUSE Linux プロビジョニングプロセスの概要」
- 106 ページの「SUSE Linux サーバーの設定」
- 111 ページの「SUSE Linux イメージとプロファイルの作成」
- 120 ページの「ターゲットホストへの OS のインストール」
- 121 ページの「SUSE Linux イメージとプロファイルの削除」

SUSE AutoYaST テクノロジー

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

SUSE Linux プロビジョニングプロセスの概要

1. SUSE Linux イメージサーバーを設定します。
2. SUSE 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。

SUSE Linux サーバーの設定

OS プロビジョニングプラグインは、SUSE Linux の Kickstart 機能とともに、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、SUSE Linux 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。SUSE Linux サーバーを設定する際に、OS をプロビジョニングするための AutoYaST を実行する JET モジュールを作成します。JET についての詳細は、59 ページの「[JumpStart Enterprise Toolkit \(JET\) テクノロジー](#)」を参照してください。

SUSE 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) または Solaris OS
- TFTP サーバー (tftp-server パッケージ)
- NFS サーバー (nfs-utils パッケージ)
- N1 SPS リモートエージェント (RA) (Linux 版)。詳細は、『Sun N1 Service Provisioning System 5.2 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。

SUSE Linux OS をプロビジョニングするためのネットワークの準備

SUSE Linux イメージサーバーを作成する前に、次の作業を行なっておく必要があります。

- ブートおよびインストールサーバー上の SUSE Linux イメージへの TFTP アクセス権の設定
この方法については、106 ページの「[TFTP アクセス権を設定する](#)」を参照してください。
- SUSE Linux OS インストールイメージの作成
この方法については、107 ページの「[SUSE Linux ファイルのコピー](#)」を参照してください。

▼ TFTP アクセス権を設定する

JET を使用して、SUSE Linux イメージサーバーを作成した場合、TFTP サービスは自動的にプロビジョニング用に構成されます。ただし、その場合でも、ブートおよびインストールサーバー上の TFTP ルートディレクトリに追加のディレクトリおよびファイルを作成するか、コピーする必要があります。OS 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。

始める前に ブートおよびインストールサーバーに適切な TFTP サーバーソフトウェアがインストールされていることを確認してください。

- 1 **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。
PXE 構成ファイルの作成方法については、115 ページの「SUSE Linux OS プロビジョニング用の PXE 構成ファイルを作成する」を参照してください。

注 – この作業では、TFTP ルートディレクトリの 1 例として `/tftpboot` を使用します。

- 2 **JET** を使用しない SUSE Linux プロビジョニングプロファイルの場合は、**TFTP** ルートディレクトリに次のファイルをコピーします。
 - `linux` – 1 枚目のディストリビューション CD 上の `boot/loader/` にあるインストールカーネル
 - `initrd` – 1 枚目のディストリビューション CD 上の `boot/loader/` にあるインストール RAM ディスク

複数リリースの SUSE Linux OS のプロビジョニングを行えるようにする場合は、このファイルに、`linux_9` などの一意の名前を割り当てます。

SUSE Linux ファイルのコピー

SUSE Linux OS イメージを作成するには、ブートおよびインストールサーバーか、NFS 経由でネットワークからアクセス可能なほかのシステムに製品媒体のファイルをコピーします。これらのファイルは、SUSE Linux OS または Solaris OS のいずれかが動作するシステムにあります。SUSE Linux ブートおよびインストールサーバーにコピーを配置する必要はありません。

SUSE Linux ファイルをコピーするには、製品 CD の内容を特定のディレクトリ構造に抽出します。SUSE Linux ファイルをコピーする方法については、http://www.suse.com/~ug/AutoYaST_FAQ.html#a6 を参照してください。

ブートおよびインストールサーバーではないシステムにコピーを配置する場合、次の条件を満たす必要があります。

- JET を使用する SUSE Linux プロビジョニングの場合は、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が NFS 経由で OS ファイルのコピーにアクセスできること。
- JET を使用しない SUSE Linux プロビジョニングの場合は、ターゲットホストが NFS 経由で OS ファイルのコピーにアクセスできること。

SUSE Linux イメージサーバーの作成

SUSE Linux イメージをコピーしたあと、N1 SPS 環境内に SUSE Linux イメージサーバーかブートおよびインストールサーバーを作成する必要があります。ブートおよびインストールサーバーを N1 SPS リモートエージェントにし、ブートおよびインストールサーバーをマスターサーバーに対するリモートエージェントとして追加します。

▼ SUSE Linux イメージサーバーを作成する (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ーバーを作成するには、次の手順に従います。

- 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、`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` や `boot_server_tftp_root_directory` 変数の値は、変更しないでください。
-
- d. 変数セットを保存します。
 - e. 「**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。

- 別のコンポーネントの変数設定を使用する場合は、「**Import Set From Component**」をクリックします。
「**Import Variable Settings**」ウィンドウが表示されます。
 - a. 必要に応じて、変数設定のインポート元となるコンポーネントが入ったフォルダへ移動します。
 - b. コンポーネントのバージョンを選択します。

注-コンポーネントのバージョンが異なると、変数設定も異なる可能性があります。現在のコンポーネントおよびインポートする変数設定があるコンポーネントが共通の変数を使用していることを確認します。変数設定のインポート元のコンポーネントと、プランで使用するコンポーネントの間で共通に使用している変数が存在しない場合、変数設定はインポートされません。

- c. 「**Import Variable Settings**」をクリックします。
変数設定がインポートされ、表形式で表示されます。
 - d. 「**Plan Details Run**」ページの「**Variable Settings**」ドロップダウンリストから、インポートした変数設定を選択し、「**Select**」をクリックします。
- ファイルに保存されているコンポーネント変数設定を使用する場合は、次の手順に従います。
 - a. 「**Import Sets from File**」テキストフィールドに、使用する変数設定ファイルへのパスを入力します。
ファイルシステムをブラウズして適切なファイルを探すには、「**ブラウズ**」ボタンをクリックします。
 - b. 「**import**」をクリックします。
変数設定がインポートされ、表形式で表示されます。

注-変数設定のインポート元のファイルと、プランで使用するコンポーネントの間で共通に使用している変数が存在しない場合、変数設定はインポートされません。

- c. 「**Plan Details Run**」ページの「**Variable Settings**」ドロップダウンリストから、インポートした変数設定を選択し、「**Select**」をクリックします。
- 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。

この手順によって、ブートおよびインストールサーバーに SUNWjet、JetRedHat、および JetSuse パッケージがインストールされます。SUNWjet パッケージは、TFTP サーバーの設定と、JET ベースディレクトリ (デフォルトでは /opt/SUNWjet) ディレクトリの NFS 共有設定を行います。

注意事項 OS プロビジョニングサーバーがブートおよびインストールサーバーと同じ物理ホスト上に存在する場合は、ターゲットホストへのプロビジョニングの前にそのシステム上でインタフェースを構成する必要があります。OS プロビジョニングサーバーは、ブートおよびインストールサーバーのインタフェースを自動的に構成しません。

▼ SUSE Linux 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)

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ーフェースから SUSE Linux イメージサーバーを作成できます。

始める前に ターゲットホストの位置を特定できるよう、`autoyast` ファイルは、SUSE Linux ブートおよびインストールサーバー上の NFS エクスポートディレクトリ (例: /export/autoyast) に配置する必要があります。

- ▶ 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
```

これらのコマンドによって、ブートおよびインストールサーバーに SUNWjet と JetRedHat、JetSuse パッケージがインストールされます。SUNWjet パッケージは、TFTP サーバーの設定と、JET ベースディレクトリ (デフォルトでは /opt/SUNWjet) ディレクトリの NFS 共有設定を行います。

注意事項 OS プロビジョニングサーバーがブートおよびインストールサーバーと同じ物理ホスト上に存在する場合は、ターゲットホストへのプロビジョニングの前にそのシステム上でインタフェースを構成する必要があります。OS プロビジョニングサーバーは、ブートおよびインストールサーバーのインタフェースを自動的に構成しません。

SUSE Linux イメージとプロファイルの作成

SUSE Linux イメージサーバーを作成したあとには、プロビジョニングイメージとプロファイルを作成する必要があります。

- JumpStart Enterprise Toolkit (JET) でイメージおよびプロファイルを生成する場合は、[111 ページ](#)の「JET による SUSE Linux イメージとプロファイルの作成」を参照してください。
- イメージおよびプロファイルを手動で作成する場合は、[115 ページ](#)の「非 JET SUSE Linux イメージとプロファイルの作成」を参照してください。

JET による SUSE Linux イメージとプロファイルの作成

JumpStart Enterprise Toolkit (JET) は、SUSE Linux のプロビジョニングに必要なプロファイルの作成作業を簡単にする一群のユーティリティです。N1 SPS ブラウザインタフェースを使用して、ターゲットホストの Kickstart インストールを自動的に行うのに必要なプロファイルキーワードおよびプロファイル値を生成できます。

JET を使用しない方が望ましく、手動でプロファイル値を作成する場合は、JET モジュールを使用しない SUSE Linux イメージおよびプロファイルを作成できます。[115 ページ](#)の「非 JET SUSE Linux イメージとプロファイルの作成」を参照してください。

▼ JET を使用して SUSE 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 SUSE 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。SUSE Linux イメージサーバーの作成方法の詳細については、[106 ページ](#)の「SUSE Linux サーバーの設定」を参照してください。

- 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「SUSE 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。
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/例
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 (デフォルト)
version	このコンポーネントで配備する Linux のバージョン	SLES9 (例)
architecture	x86 など、物理システムのアーキテクチャー 注-この値は、プロファイルコンポーネントおよび OS イメージの名前に使用されません。プロビジョニングソフトウェアによって、この変数が検査、適用されることはありません。	x86 (デフォルト)
installPath	添付する SUSE Linux イメージの名前	SuSeLinux:[version]_[architecture] 次に例を示します。 SuSeLinux_SLES9_x86
description	このプロファイルに関する説明 (任意)	test linux (例)
min_disk_size	SUSE Linux OS のプロビジョニングに必要な最小ディスク容量	10GB (デフォルト)
boot_server_name	SUSE Linux ブートおよびインストールサーバーの名前	masterserver (例)
boot_server_subnet_address	ブートおよびインストールサーバーがサービスを提供するサブネット	10.10.31.0 (例)
boot_server_subnet_ip_address	サブネット上のブートおよびインストールサーバーの IP アドレス	10.10.31.1 (例)

変数名	説明	デフォルト値/例
<code>image_subnet_mask</code>	プロビジョニングサブネットのネットマスク	255.255.255.0 (デフォルト)
<code>distro_root_directory</code>	ブートサーバー上の SUSE Linux ディストリビューションのベースディレクトリの絶対パス	<code>nfs://10.10.31.4/Suse/SLES-9-x86-64/</code> (例)
<code>kernel</code> (JET プロファイルのみ)	OS イメージ内のカーネルへの相対パス。 非 JET プロファイルの場合は、この変数に値を指定しないでください。	<code>boot/loader/linux</code> (デフォルト)
<code>initrd</code> (JET プロファイルのみ)	OS イメージ内の <code>initrd</code> への相対パス 非 JET プロファイルの場合は、この変数に値を指定しないでください。	<code>boot/loader/initrd</code> (デフォルト)
<code>autoyast_file</code> (非 JET プロファイルのみ)	ブートサーバー上の AutoYaST ファイルの絶対パス JET 生成プロファイルの場合は、この変数に値を指定しないでください。	なし
<code>pxe_boot_file</code> (非 JET プロファイルのみ)	ブートサーバー上の <code>pxelinux.0</code> ブートファイルの、 <code>/tftpboot</code> ディレクトリからの相対パス JET 生成プロファイルの場合は、この変数に値を指定しないでください。	なし
<code>pxe_configuration_file</code> (非 JET プロファイルのみ)	ブートサーバー上の PXE Linux 構成ファイルの、 <code>pxelinux.cfg</code> ディレクトリからの相対パス JET 生成プロファイルの場合は、この変数に値を指定しないでください。	なし
<code>SuSelinux_component_folder</code>	自動生成 SUSE Linux Provision コンポーネントフォルダの位置	<code>/com/sun/n1osp/autogen-[target(...):sys.hostName]-linux</code> (デフォルト)

- 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 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 製品のモジュールは、JET サーバーのディレクトリ `/opt/SUNWjet/Products` にあります。
デフォルトでは、プラン変数 `JET Modules Name(s)` に、SUSE Linux 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` - SUSE Linux 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、257 ページの「基本 SUSE Linux 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、279 ページの「SUSE Linux 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 SUSE Linux 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 SUSE Linux プロビジョニングプロファイルに追加されます。

- 9 「Use the Jet Modules Specified Above」 チェックボックスが選択されていることを確認します。
- 10 「Run Plan (includes preflight)」 をクリックします。
- 11 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。
`/com/sun/n1osp/autogen-server-name-linux/provision/` フォルダに、`provision` コンポーネントがあります。`server-name` は、変数セットに指定した Linux サーバー名です。次に例を示します。 `/com/sun/n1osp/autogen-SuSe_SLES9_x86-linux/provision/`。

▼ JET SUSE 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)

- 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/SuseLinuxImage \
-name "SLES9" -u admin -p admin -vars "osp_server=masterserver-osp;version=9; \
architecture=x86;installPath=SuSeLinux:[version]:[architecture]; \
description=SUSE SLES9;min_disk_size=10GB; \
boot_server_name=biautoyast;boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.3; \
distro_root_directory=nfs://10.42.42.3/export/SUSE/SLES-9-x86-64/installroot;\
pxe_boot_file=;pxe_configuration_file=; \
autoyast_file=;kernel=boot/loader/linux;initrd=boot/loader/initrd"
```

- 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/SuSeLinuxImage-create-profile \
-tar H:NM:bilinux-linux -comp - -vs SLES9 -pto 30 -nto 10 \
-f /tmp/suse-profile
```

-u オプションで指定したデフォルトの N1 SPS ユーザー値を書き留めます。OS インストール後に N1 SPS Remote Agent ソフトウェアがターゲットホストで起動するには、このユーザーが存在する必要があります。

非 JET SUSE Linux イメージとプロファイルの作成

SUSE Linux イメージおよびプロファイルを手動で作成する場合は、JET モジュールを使用せずに、SUSE Linux イメージおよびプロファイルを作成できます。この節では、非 JET SUSE Linux イメージとプロファイルの作成方法を説明します。

JET を使用して、イメージおよびプロファイルを作成する場合は、[111 ページの「JET による SUSE Linux イメージとプロファイルの作成」](#)を参照してください。

▼ SUSE Linux OS プロビジョニング用の PXE 構成ファイルを作成する

ターゲットホストに非 JET SUSE Linux イメージを配備するには、手動で PXE 構成ファイルを作成する必要があります。このファイルには、各ハードウェアプラットフォームおよび SUSE 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。

- 1 テキストエディタでファイルを作成し、次のテキストを入力します。

```
default sles
serial 0 9600
label sles
kernel linux
append console=ttyS0,9600 load_ramdisk=1 initrd=initrd textmode=1
```

各行の意味は次のとおりです。

- 先頭行 (default) は、ラベル `sles` を使用し、AutoYaST によるインストールを行うことを示しています。
- 3 行目 (label) は、ラベル `sles` の定義です。
- 4 行目 (kernel) は、読み込むカーネルの指定です。パスは `pxelinux.0` に対して相対的です。
- 5 行目 (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。

`console=ttyS1,9600` は、コンソールにインストール出力を表示するためのコンソールパラメータを指定しています。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` 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。

- 2 ブートおよびインストールサーバー上の TFTP ルートディレクトリの `pxelinux.cfg` サブディレクトリに、このファイルを保存します。

複数リリースの SUSE Linux OS のプロビジョニングを行えるようにする場合は、このファイルに、`sles9.cfg` などの一意の名前を割り当てます。

▼ 非 JET SUSE 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 SUSE 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。SUSE Linux イメージサーバーの作成方法の詳細については、[86 ページの「RedHat Linux サーバーの設定」](#)を参照してください。

PXE Linux 構成ファイルを作成し、ブートおよびインストールサーバー上の `pxelinux.cfg` ディレクトリにそのファイルを配置します。PXE Linux 構成ファイル例については、[115 ページの「SUSE Linux OS プロビジョニング用の PXE 構成ファイルを作成する」](#)を参照してください。

- 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「SUSE 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。
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/例
osp_server	OS プロビジョニングサーバーを表す仮想サーバー	masterserver-osp (デフォルト)
version	このコンポーネントで配備する Linux のバージョン	SLES9 (例)
architecture	x86 など、物理システムのアーキテクチャー	x86 (デフォルト)
	注-この値は、プロファイルコンポーネントおよび OS イメージの名前に使用されません。プロビジョニングソフトウェアによって、この変数が検査、適用されることはありません。	
installPath	添付する SUSE Linux イメージの名前	SuSeLinux:[version]_[architecture] 次に例を示します。 SuSeLinux_SLES9_x86
description	このプロファイルに関する説明 (任意)	test linux (例)
min_disk_size	SUSE Linux OS のプロビジョニングに必要な最小ディスク容量	10GB (デフォルト)
boot_server_name	SUSE Linux ブートおよびインストールサーバーの名前	masterserver (例)
boot_server_subnet_address	ブートおよびインストールサーバーがサービスを提供するサブネット	10.10.31.0 (例)
boot_server_subnet_ip_address	サブネット上のブートおよびインストールサーバーの IP アドレス	10.10.31.1 (例)

変数名	説明	デフォルト値/例
<code>image_subnet_mask</code>	プロビジョニングサブネットのネットマスク	255.255.255.0 (デフォルト)
<code>distro_root_directory</code>	ブートサーバー上の SUSE Linux ディストリビューションのベースディレクトリの絶対パス	nfs://10.10.31.4/Suse/SLES-9-x86-64/installroot (例)
<code>kernel</code> (JET プロファイルのみ)	OS イメージ内のカーネルへの相対パス。 非 JET プロファイルの場合は、この変数に値を指定しないでください。	なし
<code>initrd</code> (JET プロファイルのみ)	OS イメージ内の <code>initrd</code> への相対パス 非 JET プロファイルの場合は、この変数に値を指定しないでください。	なし
<code>autoyast_file</code> (非 JET プロファイルのみ)	ブートサーバー上の AutoYaST ファイルの絶対パス JET 生成プロファイルの場合は、この変数に値を指定しないでください。	nfs://10.10.31.4/export/v20z/sles9.xml (例)
<code>pxe_boot_file</code> (非 JET プロファイルのみ)	ブートサーバー上の <code>/tftpboot</code> ディレクトリを基準にした <code>pxelinux.0</code> ブートファイルの相対パス JET 生成プロファイルの場合は、この変数に値を指定しないでください。	v20z/suse/pxelinux.0 (例)
<code>pxe_configuration_file</code> (非 JET プロファイルのみ)	ブートサーバー上の <code>pxelinux.cfg</code> ディレクトリを基準にした PXE Linux 構成ファイルの相対パス JET 生成プロファイルの場合は、この変数に値を指定しないでください。	as21-console (例)
<code>SuSelinux_component_folder</code>	自動生成 SUSE Linux Provisioning コンポーネントフォルダの位置	/com/sun/n1osp/autogen-[target(...):sys.hostName]-linux (デフォルト)

- 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 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 「Use the Jet Modules Specified Above」のチェックボックスを選択解除します。
- 9 「Run Plan (includes preflight)」をクリックします。
- 10 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。
`/com/sun/n1osp/autogen-server-name-linux/provision/` フォルダに、`provision` コンポーネントがあります。`server-name` は、変数セットに指定した Linux サーバー名です。次に例を示します。`/com/sun/n1osp/autogen-SuSeLinuxSLES9_x86-linux/provision/`

▼ 非 JET SUSE 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)

- 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/n1osp/untyped/SuSeLinuxImage \
-name "SLES9x86" -u admin -p admin -vars "osp_server=masterserver-osp;version=SLES9; \
architecture=x86;installPath=SuSeLinux:[version]_[architecture]; \
description=SUSE SLES9;min_disk_size=10GB; \
boot_server_name=bilinux;boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.3; \
autoyast_file=nfs://10.10.31.4/export/v20z/sles9.xml;\
pxe_boot_file=pxelinux.0;pxe_configuration_file=pxesles9"
```
- 2 JET モジュール情報を提供するには、次のエントリを含む `/tmp//suse-profile` ファイルを作成します。

```
standard-nojet
Standard Linux
base_config
false
```

各行の意味は次のとおりです。

- ファイルの先頭行は、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行目は、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行目は、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。
- ファイルの4行目は、ブラウザインタフェースのプラン変数セクションの「Use the Jet Modules Specified Above」チェックボックスに対応します。

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。

```
# cr_cli -cmd pe.p.run -u admin -p admin \  
-PID NM:/com/sun/n1osp/untyped/SuseLinuxImage-create-profile \  
-tar H:NM:bilinux-linux -comp - -vs SLES9x86 -pto 30 -nto 10 \  
-f /tmp/suse-profile
```

-u オプションで指定したデフォルトのN1 SPS ユーザー値を書き留めます。OSインストール後にN1 SPS Remote Agent ソフトウェアがターゲットホストで起動するには、このユーザーが存在する必要があります。

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/n1osp/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

始める前に ターゲットホストのBIOSに正しい日時が設定されていることを確認します。BIOSに過去の日時が設定されていると、suse-build-key RPMのインストールに失敗し、対話モードのインストールに切り替わります。これは、BIOSが、ディストリビューションの使用中のRPM暗号化キーを受け付けられないためです。BIOSからは、暗号化キーが将来の日時に適用されるように見えます。

ブートおよびインストールサーバー上のネットワークインタフェースが適切に構成されていることを確認します。

- 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** フォルダに移動します。
たとえば、`/com/sun/nlosp/autogen-bikickstart-linux/provision/suse/SLES9_x86` などのフォルダです。
- 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

- 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。

SUSE Linux イメージとプロファイルの削除

不要になった OS イメージやプロファイルは、N1 SPS 環境から定期的に削除するとよいでしょう。

▼ SUSE Linux OS イメージを削除する

- 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「**SUSE Linux Images**」セクションで「**Manage**」をクリックします。
- 3 「**Components Details**」ページで「**Uninstall: Delete**」のチェックボックスを選択します。
- 4 「**実行**」をクリックします。

N1 SPS 環境から SUSE Linux OS イメージが削除されます。「**View link for SUSE Linux Images**」セクションのイメージリストから、削除したイメージが削除されています。

注-ブートおよびインストールサーバーから、SUSE Linux OS イメージが削除されるわけではありません。

▼ SUSE Linux OS プロファイルを削除する

- 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「**SUSE Linux Images**」セクションで「**Provision OS**」をクリックします。
- 3 「**Components**」ページで、削除するプロファイルのチェックボックスを選択します。
- 4 「**Delete**」をクリックします。

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。

- 123 ページの「Remote Installation Service (RIS) テクノロジー」
- 124 ページの「Windows プロビジョニングプロセスの概要」
- 124 ページの「Windows サーバーの設定」
- 131 ページの「Windows イメージおよびプロファイルの作成」
- 143 ページの「ターゲットホストへの Windows 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 プロファイルを作成します。
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は、次の作業で構成されます。

1. Windows RIS サーバーを設定する。
2. N1 SPS 環境内に Windows RIS サーバー用のコンポーネントを作成する。

▼ Windows RIS サーバーを設定する

- 1 **32 ビットシステム用の Service Pack 1 付き Windows 2003 Server Standard または Enterprise Edition オペレーティングシステムを、Windows ブートおよびインストールサーバーに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 - Service Pack なしの 32 ビット版 Windows 2003 Server Standard または Enterprise Edition が動作しているマシンがすでに存在する場合は、Windows 2003 Service Pack 1 を別途、サーバーにインストールできます。Microsoft Knowledge Base Article #891128 を参照してください。

- 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 Knowledge Base Article 324753 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 Knowledge Base Article 325862 を参照してください。

注-RIS サーバーのインストールでは、RIS サーバーに Windows 2003 x64 Risetup.exe イメージまたは Windows 2003-Service Pack 1 x86 Risetup.exe イメージを追加します。

Windows Server 2003 SP1 RSETUP イメージが、RIS サーバーにアップロードする最初のイメージの場合は、デフォルトで、OS 選択用の新しい画面 (x8664.osc) が RIS サーバーにコピーされます。

最初の Windows Server 2003 SP1 イメージまたは Windows 2003 x64 を追加する時点で、RIS サーバー上にすでに Windows OS イメージが存在する場合は、インストール時に OS 選択画面を上書き (またはバックアップして上書き) して、新しい画面を取得するかどうかを選択する必要があります。Microsoft Knowledge Base Article #891128 を参照してください。

4 Windows ブートおよびインストールサーバーに Windows 2003 Resource Kit ソフトウェアと Support Tools をインストールします。

このパッケージには、OS プロビジョニングスクリプトの使用する setupmgr や xcacls などのツールが含まれています。これらのツールは、次のいずれかの方法でインストールできます。

- **Windows 2003 Support Tools** をインストールする。

Windows 2003 のインストール中、サポートツールは自動的にインストールされません。インストールプログラムは、OS CD-ROM の \support\tools フォルダにあります。インストールウィザードを起動するには、手動でセットアップファイル (suptools.msi) を開く必要があります。

- **Windows 2003 Resource Kit** ツールをインストールする。

Resource Kit ツールは、Windows 2003 Resource Kit CD に含まれています。または、<http://www.microsoft.com/downloads> から入手することもできます。

5 RIS サーバーに Windows RSH Server または Windows SSH Server をインストールして、構成します。

注-N1 SPS OS プロビジョニングプラグイン 3.0 リリースは、Windows のカスタムプロファイルの配備用としては SSH しかサポートしていません。非カスタムプロファイルの配備用としては RSH と SSH の両方をサポートしています。RIS サーバーと N1 OS プロビジョニングサーバー間の通信の安全性と信頼性の面で、SSH は RSH サービスより優れています。

- **Windows RIS サーバー上に Windows SSH サーバー** をインストールするには、任意のユーザーで、RIS サーバーに **Windows 2003 用 Cygwin® Open Secure Shell Service (OpenSSH サービス)** をインストール、構成します。

詳細については、次の Web サイトを参照してください。

- **Cygwin のホームページ** (<http://www.cygwin.com/>)

- 既存の cygwin パッケージのアンインストール方法については、次の Web サイトを参照してください。 http://cygwin.com/faq/faq_2.html#SEC20
- **Windows RIS** サーバー上に **Windows RSH** サーバーをインストールするには、次の手順に従います。

a. **Windows RIS** サーバー上に **Windows 2003 Resource Kit** および **Windows Services for Unix 3.5 (SFU3.5)** ソフトウェアをインストールします。

Windows 2003 Resource Kit および Windows Services for Unix 3.5 (SFU3.5) ソフトウェアのインストール方法については、Microsoft Knowledge Base Article #321712 を参照してください。

Windows 2003 Resource Kit および Windows Services for Unix 3.5 (SFU3.5) ソフトウェアは Windows Resource Kit CD に含まれています。また、Windows Web サイトから入手することもできます。このソフトウェアには、Windows RSH サービスと、OS プロビジョニングプラグインスクリプトの使用するそのほかのツール (setupmgr、deploy.cab、xcac1s など) が含まれています。

b. root ユーザーで、**RIS** サーバーに **Windows Remote Shell Service (WinRsh サービス)** をインストール、構成します。

この方法については、[Remote Shell Services in SFU](#)

(<http://www.microsoft.com/technet/interopmigration/unix/sfu/sfu35rsh.mspx>) を参照してください。

注-.rhosts ファイルに対するアクセス権が正しく設定されていないと、WinRsh サービスに失敗し、クライアントに「Access denied」メッセージが返されます。この問題の発生を防ぐには、.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。たとえば、c://windows/system32/drivers/etc/ 内の .rhosts ファイルに次のエントリを追加します。

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

編集した .rhosts は、必ず、.rhosts.txt ではなく、.rhosts のまま保存してください。

- 6 **Windows RIS** サーバー上に **N1 SPS** リモートエージェントをインストールします。
- 7 マスターサーバーで、**Windows RIS** サーバー用のリモートエージェントホストを作成します。

▼ 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)

Windows RIS サーバーを設定したあとは、SPS 環境内にそのサーバー用のコンポーネントを作成する必要があります。次の手順に従い、SPS ブラウザユーザーインタフェースでコンポーネントを作成します。

- 1 SPS ブラウザインタフェースにログインします。
- 2 ブラウザインタフェースのメインウィンドウにある「**Common Tasks**」セクションで、「**OS Provisioning**」をクリックします。
「OS Provisioning」ページが表示されます。
- 3 「**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。
- 4 「**Plans Details**」ページの「**Run**」をクリックします。
「WindowsServer-create plan details」ページが表示されます。
- 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「**Variable Settings**」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「**Variable Settings**」列で「**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、[130 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. 変数設定を変更するには、その変数のチェックボックスをクリックします。
チェックボックスを選択すると、そのボックス内にチェックマークが付き、テキストフィールドが表示されます。テキストフィールドに新しい変数値を入力します。
 - e. 変数セットを保存します。

- f. 「Plan Parameters」 テーブルの「Variable Settings」 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。
 - 別のコンポーネントの変数設定を使用する場合は、「Import Set From Component」をクリックします。
 - 「Import Variable Settings」 ウィンドウが表示されます。
 - a. 必要に応じて、変数設定のインポート元となるコンポーネントが入ったフォルダへ移動します。
 - b. コンポーネントのバージョンを選択します。

注-コンポーネントのバージョンが異なると、変数設定も異なる可能性があります。現在のコンポーネントおよびインポートする変数設定があるコンポーネントが共通の変数を使用していることを確認します。変数設定のインポート元のコンポーネントと、プランで使用するコンポーネントの間で共通に使用している変数が存在しない場合、変数設定はインポートされません。

 - c. 「Import Variable Settings」 をクリックします。

変数設定がインポートされ、表形式で表示されます。
 - d. 「Plan Details Run」 ページの「Variable Settings」 ドロップダウンリストから、インポートした変数設定を選択し、「Select」 をクリックします。
 - ファイルに保存されているコンポーネント変数設定を使用する場合は、次の手順に従います。
 - a. 「Import Sets from File」 テキストフィールドに、使用する変数設定ファイルへのパスを入力します。

ファイルシステムをブラウズして適切なファイルを探すには、「ブラウズ」 ボタンをクリックします。
 - b. 「Import」 をクリックします。

変数設定がインポートされ、表形式で表示されます。

注-変数設定のインポート元のファイルと、プランで使用するコンポーネントの間で共通に使用している変数が存在しない場合、変数設定はインポートされません。

 - c. 「Plan Details Run」 ページの「Variable Settings」 ドロップダウンリストから、インポートした変数設定を選択し、「Select」 をクリックします。
- 6 「Target Host」 フィールドに、124 ページの「Windows RIS サーバーを設定する」 で作成した Windows RIS サーバー用に N1 SPS マスターサーバーに作成したホストを指定します。

- 7 「Plan Variables」セクションのテキストフィールドに **Active Directory** ユーザーのパスワードを入力します。
 クリアテキストでパスワードを入力します。パスワードは、Windows のすべてのパスワード要件を満たしている必要があります。詳細は、Microsoft のマニュアルを参照してください。
- 8 「Run Plan (includes preflight)」をクリックします。
 プランは、次の処理を行います。
 - マスターサーバーに Windows RIS サーバーコンポーネントを作成する。
 - Windows RIS サーバー上の、installPath 変数に指定されているディレクトリに NlospRisUtil ディレクトリを作成する。このディレクトリには、次のサブディレクトリが含まれます。
 - NlospScripts
 - NlospTemplates
 - NlospTools
 - SampleOSFiles
 - SampleSIFiles
- 9 Windows RIS 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)

Windows RIS サーバーを設定したあとは、SPS 環境内にそのサーバー用のコンポーネントを作成する必要があります。次の手順に従い、SPS ブラウザーユーザーインタフェースでコンポーネントを作成します。

- 1 (省略可能)RIS サーバーへのアクセスパスワードを暗号化します。
 パスワードを暗号化する方法の詳細については、169 ページの「パスワードの暗号化」を参照してください。
- 2 WindowsServer 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/WindowsServer \  
-name "winRisServer" -u admin -p admin -vars "installPath=C:\\\\N1 ISP\\\\IspScripts; \  
boot_server_domain_name=nllab.west.example.com;active_directory_userid=nlospadmin; \  
boot_server_access_protocol=ssh;boot_server_access_userid=Administrator; \  
\
```

```
boot_server_access_password=yf7813jWUweB4SXBe0JIlw==; \
ris_share_directory=D:\\\\RemoteInstall"
```

WindowsServer 変数セットについては、130 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。

- 3 次のエントリからなる、/tmp/windows-bi-server という名前のファイルを作成します。

```
password4user!
```

上記例では、password4user! には、Active Directory ユーザーのパスワードを指定します。

- 4 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/WindowsServer-create -tar H:NM:risserv \
-comp - -vs winRisServer -pto 30 -nto 10 -f /tmp/windows-bi-server
```

プランは、次の処理を行います。

- マスターサーバーに Windows RIS サーバーコンポーネントを作成する。
- Windows RIS サーバー上の、installPath 変数に指定されているディレクトリに NlospRisUtil ディレクトリを作成します。このディレクトリには、次のサブディレクトリが含まれます。
 - NlospScripts
 - NlospTemplates
 - NlospTools
 - SampleOSCFiles
 - SampleSIFFiles

Windows イメージサーバーのコンポーネント変数

表 8-1 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/例
installPath	N1 OS プロビジョニングスクリプトがインストールされているベースディレクトリの位置。例: C:\Program Files。 この変数で後続のバックスラッシュ (\) を指定しないでください。	C:\(デフォルト)
boot_srserver_domain_name	RIS サーバー用の Windows Active Directory のドメイン名。	winprov.n1lab.sun.com (例)
active_directory_userid	Active Directory のユーザー ID	nlospadmin (デフォルト)

表 8-1 Windows イメージサーバーコンポーネントの変数 (続き)

変数名	説明	デフォルト値/例
boot_server_access_protocol	RIS サーバーへのアクセスに使用するプロトコル (RSH、SSH など)。 注-カスタム Windows RIS サーバードプロファイルを使用する場合は、この変数 boot_server_access_protocol に、RSH を指定することはできません。	ssh (デフォルト)
boot_server_access_userid	RIS サーバーへのアクセスに使用するユーザー ID。RSH プロトコルの場合は、この変数に root を指定します。	デフォルトなし
boot_server_access_password	RIS サーバーへのアクセスに使用する暗号化されたパスワード。この変数は、RIS サーバーが SSH プロトコルを使用する場合、必須です。RSH プロトコルを使用する場合は、省略可能です。 暗号化されたパスワードの作成方法については、 169 ページ の「パスワードの暗号化」を参照してください。	デフォルトなし
ris_share_directory	RIS サーバー上の RemInst share の位置。例: D:\RemoteInstall	デフォルトなし
ris_language	RIS サーバーにインストールされている OSChooser 画面の言語。	English
ciw_directory	CIW および OS Chooser ファイルの位置。例: D:\RemoteInstall\OSChooser\English	:[ris_share_directory] \OSChooser\:[ris_language]
temporary_sif_directory	RemInst share 内の、Sun N1 OS プロビジョニングの一時 sif ファイルの位置。例: D:\RemoteInstall\N1ospSif.	:[ris_share_directory]\N1ospSif
windowsHost	Windows RIS 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 RIS サーバー上で Windows イメージを設定する

- 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 ユーティリティを使用します。RIS イメージの作成時に「Keep old existing OSC files」オプションを選択します。

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 Knowledge Base Article #300556 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。

サービスパックが組み込まれた Windows ディストリビューションの場合は、次のサイトにある指示に従って、RIS イメージを作成します。

- [Service Pack 3 Installation and Deployment Guide for Windows 2000 Server Edition](http://www.microsoft.com/Windows2000/downloads/servicepacks/sp3/spdeploy.htm#the_integrated_installation_fmaj)
(http://www.microsoft.com/Windows2000/downloads/servicepacks/sp3/spdeploy.htm#the_integrated_installation_fmaj)
このサイトには、そのほかの Windows OS バージョン用の類似文書も提供されています。
- [Microsoft Knowledge Base Article #260910](http://support.microsoft.com/kb/260910/EN-US/)
(<http://support.microsoft.com/kb/260910/EN-US/>)
最新のサービスパックの入手方法を解説しています。
- [Download Windows 2000 Service Pack 3](http://www.microsoft.com/Windows2000/downloads/servicepacks/sp3/download.aspx)
(<http://www.microsoft.com/Windows2000/downloads/servicepacks/sp3/download.aspx>)
このサイトから適切なサービスパックをダウンロードします。「Windows 2000 SP3 Network Installation Windows 2000 SP3 Network Installation download」を選択します。このサイトには、そのほかの Windows OS バージョン用の類似のダウンロードページも提供されています。

- 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 イメージへのドライバの追加の詳細については、Microsoft Knowledge Base Article 315279、Microsoft Knowledge Base Article 246184、および Microsoft Knowledge Base Article 254078 を参照してください。

注 - 適切なディストリビューションフォルダを作成します。

詳細は、SampleSIFFiles および SampleDrivers フォルダにあるサンプルのセットアップファイルを参照してください。

- 3 (省略可能) イメージディストリビューションフォルダの下にある `product-directory/OEM/$1/N10SP/` フォルダに N1 SPS リモートエージェントのバイナリをコピーします。

`product-directory` は Windows イメージディストリビューションフォルダで、このフォルダの下に、カーネルファイルを含むサブフォルダとして「i386」または「amd64」があります。

注-このフォルダが存在しない場合は、レイアウトを作成します。

スクリプトでは、N10SP フォルダがイメージに含まれていることが前提です。プロビジョニングに必要なツールを、ディレクトリパス `product-directory/OEM/$1/N10SP/` にコピーします。フォルダ名の N10SP を変更しないでください。

N1 SPS リモートエージェントのバイナリをフォルダにコピーすると、ディレクトリの内容は次の例に示す内容のようになります。

```
D:\RemoteInstall\Setup\English\Images\WIN2003_TRIAL\%OEM%\$1\N10SP>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\N10SP

02/04/2005    02:39 PM    <DIR>        .
02/04/2005    02:39 PM    <DIR>        ..
01/31/2005    03:33 PM    27,595,776  cr_ra_win32_5.2.msi
```

カスタム Windows プロビジョニングプロファイルの作成

プロビジョニングソフトウェアを使用して、配備用のカスタム Windows プロビジョニングプロファイルを作成できます。OS プロビジョニングプラグインは、ターゲットホストに Windows OS をプロビジョニングするために必要なすべての変数およびプランを生成できます。

Windows プロビジョニングプロファイルを手動で作成する場合は、[138 ページ](#)の「Windows イメージの設定情報を構成する」を参照してください。

▼ カスタム 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)

ここでは、OS プロビジョニングのブラウザインタフェースを使用して、カスタム Windows プロビジョニングプロファイルを生成する手順を説明します。

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、[124 ページの「Windows サーバーの設定」](#)を参照してください。

- 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、[142 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。

注 - カスタムプロファイルの場合、`sif_file` 変数には値を指定しません。

- 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 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 「**Create the Custom Profile**」オプションのチェックボックスが選択されていることを確認します。
- 9 「**Run Plan (includes preflight)**」をクリックします。
- 10 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 のような説明の付いた com.sun.n1osp#Provision タイプのコンポーネントが追加されています。
- 11 カスタム Windows プロビジョニングプロファイル用の変数を編集します。
カスタム Windows プロファイル用のコンポーネント変数を編集して、次の項目の値を指定する必要があります。
- OS Provisioning 制御サービス
 - プロダクトキー
 - ユーザ名
 - 組織名
 - タイムゾーン
 - 端末サーバーのプロパティ
 - ターゲットホストの言語とロケール
- a. 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。
- b. 「**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。
- c. 「**Components**」ページで、以前に「**Components**」列で作成したプロファイルをクリックします。
- d. 「**Components Details**」ページで「**Edit**」をクリックします。
- e. 「**Component Details Edit**」ページで、編集する変数の横のチェックボックスをクリックし、「**Value for This Component**」列の変数の値を入力します。
次の変数の値を編集します。
- | | |
|------------------------------------------|-------------------------------------------------------------------------------------------------------|
| <code>osp_profile_control_service</code> | 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を指定します。Windows プロビジョニングの場合、この変数の値は true に設定します。 |
| <code>user_full_name</code> | ターゲットホストのユーザーのフルネームを指定します。 |

<code>organization_name</code>	ターゲットホストが所属する組織の名前を指定します。
<code>product_key</code>	OS ディストリビューションのプロダクトキーを指定します。
<code>timezone</code>	ターゲットホストのタイムゾーンを指定します。
<code>profile_language</code>	ターゲットホストにインストールする言語またはロケールを指定します。
<code>profile_language_group</code>	ターゲットホストにインストールする言語グループを指定します。
<code>install_terminal_server</code>	プロファイルがターゲットホストに端末サーバーをインストールするかどうかを指定します。
<code>terminal_server_allow_connections</code>	ターゲットホストが遠隔接続を許可するかどうかを指定します。
<code>terminal_server_licensing_mode</code>	端末サービスによるクライアントアクセスライセンスの管理方法を指定します。
<code>terminal_server_permissions_setting</code>	端末サーバーユーザーに対するセキュリティモードを選択できます。

f. 「**Check In**」をクリックします。

次の手順 ターゲットホストに Windows OS をインストールするには、BIOS のブート設定を変更して、プロビジョニングプランを実行する必要があります。詳細は、[143 ページの「ターゲットホストへの Windows OS のインストール」](#)を参照してください。

▼ カスタム 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)

- 1 WindowsImage 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/WindowsImage \
-name "windows2003custom" -u admin -p admin -vars \
"osp_server=masterserver-osp;os_name="WIN2003_trial"version=2003; \
installPath="2003customprofile"; \
description=test windows image 2003;min_disk_size=10GB; \
boot_server_name=biwl;boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.4;\
boot_srv_active_directory_domain=n1.west.example.com;\
```



```
windows_product_directory=D:\\\\RemoteInstall\\\\Setup\\\\English
\\\\Images\\\\Win2003_trial"
```

注 -windows_product_directory は、その全内容を 1 行で入力します。表示の目的から、上記の例のエントリは 2 行に分けて表示されています。

これらの変数の詳細については、142 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 次のエントリを含む /tmp/windows-profile という名前のファイルを作成します。

```
custom
Windows 2003 EE x86 Profile
true
```

このファイル内のエントリはそれぞれ次のプラン変数に対応しています。

- custom はプロファイルの名前です。
- Windows 2003 EE x86 Profile はプロファイルの説明です。
- true は、カスタムプロファイルを生成することを示します。

- 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。

```
# cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/n1osp/untyped/WindowsImage-create-profile \
-tar H:NM:rissserver-windows -comp - -vs windows2003custom -pto 30 -nto 10 \
-f /tmp/windows-profile
```

-u オプションで指定したデフォルトの N1 SPS ユーザー値を書き留めます。OS インストール後に N1 SPS Remote Agent ソフトウェアがターゲットホストで起動するには、このユーザーが存在する必要があります。

非カスタム Windows プロビジョニングプロファイルの作成

N1 SPS のブラウザまたはコマンド行インタフェースを使用して、Windows プロビジョニングプロファイルを手動で作成できます。プロビジョニングプロファイルを手動で作成する場合は、プロビジョニングプランで使用する変数に適切な値を指定する必要があります。

OS プロビジョニングプラグインを使用して Windows プロビジョニングプロファイルを作成する場合は、133 ページの「[カスタム Windows プロビジョニングプロファイルを作成する \(ブラウザインタフェース\)](#)」を参照してください。

非カスタム Windows プロビジョニングプロファイルを作成するには、次の作業を行います。

- 138 ページの「Windows イメージの設定情報を構成する」
- 140 ページの「非カスタム 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)」

▼ Windows イメージの設定情報を構成する

ここでは、Windows イメージ用の設定情報を手動で構成する方法を説明します。非カスタム Windows プロビジョニングプロファイルを作成する場合は、この手順を行なっておく必要があります。

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。

- 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。このファイルは、notepad ツールを使用して手動で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] セクションに次の構成パラメータがあるかどうかを確認し、ない場合は追加します。

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

- 3 ristndrd.sif 構成ファイルの [Unattended] セクションに次の構成パラメータがあるかどうかを確認し、ない場合は追加します。

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

- 4 ristndrd.sif 構成ファイルの [UserData] セクションに次の構成パラメータがあるかどうかを確認し、ない場合は追加します。

```
FullName = "N1osp 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。代わりに、ProductID 変数を使用します。

- 5 ristndrd.sif 構成ファイルの [GuiUnattended] セクションに次の構成パラメータがあるかどうかを確認し、ない場合は追加します。

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

注 - AutoLogonCount=1 は、インストール後にはじ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と、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 インストール後に重要なセキュリティアップデートを自動的にダウンロードするように、**Software Information File (sif)** を編集します。

sif ファイルに次の行を追加します。

```
[WindowsFirewall]  
Profiles=WindowsFirewall.TurnOffFirewall  
  
[WindowsFirewall.TurnOffFirewall]  
Mode=0
```

- 7 ristndrd.sif 構成ファイルに次のセクションがあるかどうか確認し、ない場合は追加します。

```
[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。

- 8 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 Knowledge Base article #: 315279 の指示を参照してください。

▼ 非カスタム 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)

ここでは、OS プロビジョニングのブラウザインタフェースを使用し、独自の Windows プロビジョニングプロファイルを手動で作成する手順を説明します。

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の詳細については、[124 ページの「Windows サーバーの設定」](#) および [124 ページの「Windows RIS サーバーを設定する」](#) を参照してください。

- 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 「**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。
 - 5 「**Create Set**」をクリックします。
 - 6 変数セットの名前を入力します。
 - 7 このプランで使用する変数値を指定します。
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、[142 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。
-
- 注 - `sif_file` 変数の値は必須です。
-
- 8 変数セットを保存します。
 - 9 「**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。
 - 10 「**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。

- 11 「Profile Name」 および「Profile Description」 プラン変数の値を入力します。
- 12 「Create Custom Profile」 チェックボックスを選択解除します。
- 13 「Run Plan (includes preflight)」 をクリックします。
- 14 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 のような説明の付いた com.sun.nlosp#Provision タイプのコンポーネントが追加されています。

次の手順 ターゲットホストに Windows OS をインストールするには、BIOS のブート設定を変更して、プロビジョニングプランを実行する必要があります。詳細は、143 ページの「ターゲットホストへの Windows OS のインストール」を参照してください。

▼ 非カスタム 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)

- 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/WindowsImage \
-name "windows2003" -u admin -p admin -vars "osp_server=masterserver-osp;version=2003; \
os_name="WIN2003_trial";description=test windows image 2003;min_disk_size=10GB; \
boot_server_name=biw1;boot_server_subnet_address=10.42.42.0; \
boot_server_subnet_ip_address=10.42.42.4;boot_srv_active_directory_domain=nlisp.sfbay.sun.com; \
windows_product_directory=D:\\\\RemoteInstall\\\\Setup\\\\English\\\\Images\\\\Win2003_trial; \
sif_file=D:\\\\RemoteInstall\\\\Setup\\\\English\\\\Images\\\\Win2003_trial\\\\i386\\\\templates\\\\
Ispristndrd.sif"
```

注 -windows_product_directory は、その全内容を 1 行で入力します。表示の目的から、上記の例のエントリは 3 行に分けて表示されています。

これらの変数の詳細については、142 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 次のエントリを含む /tmp/windows-profile という名前のファイルを作成します。

```
noncustom
Windows 2003 EE x86 Profile
false
```

このファイル内のエントリはそれぞれ次のプラン変数に対応しています。

- noncustom はプロファイルの名前です。
- Windows 2003 EE x86 Profile はプロファイルの説明です。

- false は、カスタムプロファイルを使用しないことを示します。

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。

```
# cr_cli -cmd pe.p.run -u admin -p admin -PID NM:/com/sun/nlosp/untyped/WindowsImage-create-profile \
-tar H:NM:rissserver-windows -comp + -vs windows2003 -pto 30 -nto 1 -f /tmp/windows-profile0
```

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 8-2 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 (デフォルト)
os_name	RIS サーバー上の Windows OS ディストリビューションの名前	Win2003_SP1_x86 (例)
version	Windows のバージョン	2003 (デフォルト)
architecture		x86 (デフォルト)
os_language	Windows OS ディストリビューションの言語	English (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)
boot_server_name	Windows ブートサーバーの名前	Winbiserver1 (例)
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	10.42.42.0 (例)
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	10.42.42.1 (例)
boot_srv_active_directory_domain	このサブネットの Windows Active Directory ドメイン	:[targetableComponent:boot_server_domain_name] (デフォルト)

表 8-2 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 (続き)

変数名	説明	デフォルトまたはサンプル値
windows_product_directory	ブートサーバー上の Windows 製品のディレクトリ名	: [targetableComponent:ris_share_directory] \Setup\:[os_language]\Images\[os_name] (デフォルト)
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 **Windows 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。

▼ ターゲットホストに **OS** をインストールする

始める前に Windows OS をプロビジョニングする前に、インストールするターゲットホストを作成します。詳細は、[第 9 章](#)を参照してください。

インストールするターゲットホストのコンポーネントの変数セットに GUID が指定されていることを確認します。ターゲットホストの変数の詳細については、[付録 A](#)を参照してください。

- 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** フォルダに移動します。
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。

Windows のインストール先の管理

OS プロビジョニングプラグインでは、Windows のインストール先に対して次の管理作業を行うことができます。

- インストール先の状態の確認
- プロビジョニング処理の停止
- N1 SPS リモートエージェントとしてのインストール済みターゲットホストの登録
- N1 SPS 環境からのプレステージ状態のホストの削除

▼ Windows OS のプロビジョニング状態を確認する

- 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「Host 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 「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。

▼ Windows OS プロビジョニング処理を停止する

- 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「Host Admin」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 「Run Plan (includes preflight)」をクリックします。

▼ N1 SPS にインストール済み Windows ホストを登録する

ターゲットホストに Windows OS をインストールしたら、N1 SPS 環境にそのホストを登録します。

- 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「Host Admin」セクションで「Register Host」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、登録す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。

▼ Windows RIS サーバー上のプリステージ状態のホストを削除する

OS プロビジョニングプラグインを使用し、Windows RIS サーバー上の Active Directory からプリステージ状態のホストを削除することができます。

- 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「Windows Image Servers」セクションで「Manage」をクリックします。
- 3 「Components Details」ページの「Component Procedures」テーブルにある「delete_prestage_hosts」の横のチェックボックスをクリックします。
- 4 「Component Procedures」テーブルの「Actions」列で「Run」をクリックします。

- 5 「**Plan Details Run**」ページの「**Current Installations**」リストから、削除するプレステージ状態のホストの名前を選択します。
プレステー状態のホストの名前には、`-target` 接尾辞が付いていません。
- 6 「**Run Selected Installations**」をクリックします。
- 7 「**Details**」リンクをクリックして、削除結果を確認します。

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。

- 147 ページの「ターゲットホストタイプ」
- 152 ページの「ターゲットホストを定義する作業の例」
- 169 ページの「パスワードの暗号化」
- 170 ページの「端末サーバーのサポート」

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 Advanced Remote System Control (RSC) 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。

IP 接続の代替手段として、OS プロビジョニングプラグインでは、Cisco Terminal Server 2621 シリーズや Cyclades Alterpath ACS Terminal Server シリーズの端末サーバー、あるいはコンソントレータタイプを使用したネットワーク管理コントローラとの接続をサポートしています。ターゲットホストに対する変数セットで、使用するターゲットホストに端末サーバーを指定できます。端末サーバーのサポートとそれらの端末サーバー用の変数の詳細については、[170 ページの「端末サーバーのサポート」](#) 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 9-1 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 x4100, x4200 ファミリー	SP バージョン 6464	<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 	<ul style="list-style-type: none"> ■ Service Processor の IP ■ ssh ユーザー/パスワード ■ ipmi 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 	Sun IPMI ターゲット

表 9-1 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 x2100 ファミリー	SP v4.11	<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 	<ul style="list-style-type: none"> Service Processor の IP ipmi 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 	Sun IPMI ターゲット
SunFire v480/v880 ファミリー	RSC 2.2.3、RSCboot	<ul style="list-style-type: none"> RSC 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 RSC 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"> RSC IP、または端末サーバーの情報 (ip、タイプ、RSC シリアルポートに接続されるポート) RSC 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 RSC ターゲット

表 9-1 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	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) 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 	Sun ALOM ターゲット
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

表 9-1 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 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 	<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
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"> ■ 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) に IP (静的) を提供 ■ 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"> ■ IPMI IP の 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

表 9-1 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 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 の IP アドレス、ユーザー ID、およびパスワード ■ ドメイン名、ユーザー ID、およびパスワード ■ 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
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。

SunALOM ターゲットホストの定義

システムにイメージをインストールする (多くの場合「プロビジョニング」と呼ばれる) には、事前にそのシステムについて、OS プロビジョニングサーバーに通知する必要があります。

▼ SunALOM ターゲットホストを作成する(ブラウザインタフェース)

- 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「SunALOM」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、[191 ページの「SunALOM 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、[169 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 Details 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「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. システムが端末サーバーに接続されている場合は、「If you are specifying the terminal server password below.」の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。

▼ SunALOM 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)

- 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/n1osp/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、191 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。

- 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
true
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。

- 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。

```
# 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™ x2100/x4100/x4200/V20z システムをはじめとする、いくつかのシステムファミリ用で使用されます。

▼ 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。具体的に、この例では x2100 ターゲットホストを定義します。

- 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。
- 「**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。
- 「**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、[187 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、[169 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-x2100 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-x2100-target になります。

▼ 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)

- 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/n1osp/targets/SunIPMI -name "x2100" \  
-u admin -p admin -vars "installPath=x2100host;description=test x2100 host; \  
device_type=sunfirex2100;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-AEF3-001083DFE90};disk_size=20GB;ipmi_ip_address=10.5.133.134; \  
ipmi_access_userid;ipmi_access_password;;ssh_access_userid=admin;ssh_access_password= \  
console_service=false"
```

これらの変数の詳細については、187 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、**Sun IPMI** 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。

```
# cr_cli -cmd pe.p.run -u admin -p admin \  
-PID NM:/com/sun/n1osp/targets/SunIPMI-create -tar H:NM:masterserver-osp \  
-comp + -vs x2100 -pto 30 -nto 10 -f /tmp/test-x2100
```

上記の例は、Sun Fire x2100 用です。

Sun RSC ターゲットホストの定義

Sun RSC ターゲットホストタイプは Sun Fire™ v490 または v690 システムなどのシステムに使用されます。

▼ Sun RSC ターゲットホストを作成する (ブラウザインタフェース)

- 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 RSC Targets**」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「**SunRSC**」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「SunRSC」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、194 ページの「Sun RSC ターゲットホストの変数」の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。

```
rsc_access_userid  
rsc_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、169 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 Details 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「SunRSC」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 RSC 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 RSC」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 test490 のプロビジョニングターゲットを定義した場合は、仮想ホストは test490-target になります。

▼ Sun RSC 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)

- 1 SunRSC 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/SunRSC \  
-name "testv490" -u admin -p admin -vars "installPath=testv490; \  
description=To test v490;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;rsc_ip_address=10.5.133.150;rsc_access_userid=admin; \  
rsc_access_password="
```

これらの変数の詳細については、194 ページの「Sun RSC 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 RSC password below」ボックスをチェックすることに相当します。
- ファイルの2行目は、ILO にアクセスするためのパスワードです。

- ファイルの3行目は、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相当します。
- ファイルの4行目は、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/testv490-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/RSC-target | cr_cli -cmd pe.p.run -u admin -p admin \
-PID NM:/com/sun/nlosp/targets/SunRSC-create -tar H:NM:masterserver-osp \
-comp + -vs testv490 -pto 30 -nto 10 -f /tmp/testv490-target
```

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)

- 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、203 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、169 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。

- 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 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。

▼ HPILO 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)

- 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、203 ページの「HPILO 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  
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。

- 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/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もあります。

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機能を処理する関連付けられたハードウェア抽象レイヤー (HAL) JAR ファイルが必要です。HAL JAR ファイルは、OS プロビジョニングサー

パー上の /opt/SUNWn1osp/drivers にあります。各 HALJAR ファイルがサポートする、電源、ブート、コンソールなどのサービスを理解しておく必要があります。たとえば、b1600sc.jar ファイルには、Sun Fire B100s/B100x Blade シリーズに対する電源、ブート、およびコンソールサービスのサポートが含まれています。遠隔管理ホストタイプ用の変数セットを作成する前に、それらのサービス、またそれらのサービスをサポートする HALJAR を理解しておいてください。

各遠隔ネットワーク管理インター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もあります。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。

▼ 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)

- 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、209 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、169 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。
 - 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-b100s-1 のプロビジョニングターゲットを定義した場合は、仮想ホストは sunfire-b100s-1-target になります。

▼ 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。

- 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/n1osp/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/SUNWn1osp/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、209 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/n1osp/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)

- 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の場合は、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。状態監視機能を介してインストールの状態を監視している場合は、OS プロビジョニング制御サービスを使用して、ターゲットホストの電源オフと電源オンのタイミングを把握できます。

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、[215 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)

- 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、215 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 に移動し、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します。

```
# 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。

端末サーバーのサポート

IP 接続の代替手段として、OS プロビジョニングプラグインでは、次の端末サーバー、あるいはコンセントレータタイプを使用したネットワーク管理コントローラとの接続をサポートしています。

- Cisco Terminal Server 2621 シリーズ
製品情報は、<http://www.cisco.com> を参照してください。
- Cyclades Alterpath ACS Terminal Server シリーズ
製品情報は、<http://www.cyclades.com> を参照してください。

OS プロビジョニングプラグインは、これらの端末サーバー/コンセントレータ用のアクセスプロトコルとして telnet プロトコル (Cisco Terminal Server 2621 用) および SSH プロトコル (Cyclades Alterpath ACS Terminal Server 用) をサポートしています。これらの端末サーバーの技術仕様および設定方法については、それぞれのベンダーのマニュアルを参照してください。

OS プロビジョニングプラグインでは、Cisco 2621 が、サーバーネットワーク管理コントローラにバインドするデフォルトの端末サーバーになります。Cyclades Alterpath ACS 端末サーバーを使用するには、ターゲットホストに応じて適切な端末サーバー変数を編集します。表 9-2 は、Cisco 2621 および Cyclades Alterpath ACS 端末サーバーの両方の端末サーバー変数と値の一覧です。

表 9-2 端末サーバーの変数と値

変数 (説明)	Cisco 2621 の変数値	Cyclades Alterpath の変数値
terminal_server_type (端末サーバータイプ)	Cisco2621	cyclades
terminal_server_hal_jar (端末サーバーの HALJAR)	file:///:[targetableComponent:installPath]/SUNWnlosp/drivers/cisco/c2621.jar	file:///:[targetableComponent:installPath]/SUNWnlosp/drivers/cyclades/cycladesTS.jar
terminal_server_port (端末サーバーのポート)	2032 以上のポート値	7000 以上のポート値
terminal_server_protocol (端末サーバーのアクセスプロトコル)	telnet (ssh はサポートされない)	ssh (telnet はサポートされない)

このほかに、ターゲットホストが端末サーバーを使用できるようにするには、次の変数の値を指定する必要があります。

terminal_server_ip_address	ターゲットホストに接続されている端末サーバーの IP アドレスを指定します。
terminal_server_userid	端末サーバー用のユーザー ID を指定します。
terminal_server_password	端末サーバー用の暗号化されたパスワードを指定します。 パスワードを暗号化する方法の詳細については、 169 ページの「パスワードの暗号化」 を参照してください。

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。

- 173 ページの「一般的な障害追跡のガイドライン」
- 175 ページの「Solaris 関連の問題の解決」
- 176 ページの「Linux 関連の問題の解決」
- 177 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。考え得るプロビジョニング失敗原因の一部を、次に示します。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て、`/var/run/nlosp/log` フォルダでログをチェックし、`/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、Solaris ベースの OS プロビジョニングサーバーの場合は、`/var/adm/nlosp*` ファイルのメッセージ、Linux ベースの OS プロビジョニングサーバーの場合は、`/var/log/nlosp*` ファイルのメッセージもチェックします。`/var/adm/nlosp*` ファイルで詳細な情報を表示するには、`/opt/SUNWnlosp/etc/nlosp-core.properties` ファイル内の `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` に変更します。
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なって失敗の原因を確認します。

4. 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。

5. 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。
6. 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。
7. 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。
8. 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。

1. 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 5.2 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と Solaris および Linux 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/SUNWn1sps/N1_Service_Provisioning_System_5.2/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、次の手順に従います。

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: x86 システムでの Solaris 10 1/06 OS のプロビジョニングで、インストールに失敗する。

対処方法: ブートおよびインストールサーバーに /tftpboot/I86PC.Solaris_10-1 フォルダが存在することを確認します。このフォルダが存在しない場合は、フォルダを作成し、ターゲットホストへのプロビジョニングをやり直します。

問題: ヘッドレスターゲットホストへの Solaris OS のプロビジョニングで、インストールが対話形式になり、kdmconfig ユーティリティーが起動される。

対処方法: プロビジョニングプランから SUNWxwssu および SUNWxwscf パッケージを削除します。Solaris OS プロビジョニングプロファイル内の profile_del_packages_base_config 変数の値を SUNWxwssu SUNWxwscf に設定します。

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 DHCPACK 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。

- 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ず。
- 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。
- 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。
- 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。
- 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。
- 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。
- 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。
- 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。
- 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。
- 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。
- ターゲットは再起動します。
- 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。
- ターゲットは GUI モードインストールを行います。
- 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コンタクト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 サーバーとのコンタクト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 および <reminst_share>\OSChooser\amd64 フォルダで、startrom.com ファイルと 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 プロパティをチェックして、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。

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ブスプーフィング、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: 次に示す回避方法のどれか 1 つを選択してください。

- Windows RIS サーバー上で、プロビジョニングインタフェースだけを使用可能にして、ほかのすべてのインタフェースを使用不可にする。
- Windows RIS サーバー上のネットワークサービスがアクセスするネットワーク接続の順序を変更する。次の手順に従ってください。
 1. Windows RIS サーバー上で、「スタート」メニューから「コントロールパネル」ウィンドウを開きます。
 2. 「コントロールパネル」ウィンドウで「ネットワーク接続」をクリックします。
 3. 「ネットワーク接続」画面で「詳細」タブを選択します。
 4. 「詳細設定」ボタンをクリックします。
 5. 「アダプタとバインド」タブをクリックします。
 6. ネットワークサービスがアクセスする接続の順序を変更します。プロビジョニングインタフェースが最初のインタフェースになっていることを確認します。
 7. BINLSVC を一度停止してから再起動します。この操作は、Services GUI を実行、または MS-DOS コマンドウィンドウで次のコマンドを入力することによって行うことができます。

```
> net stop binlsvc  
> net start binlsvc
```

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/<language>) パスの CIW ファイル (welcome.osc/login.osc/oschoice.osc/install.osc/warning.osc/multiling.osc) で、無効なユーザー情報や見当たらない AUTOENTER 機能がないかを調べます。
- 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。

- RIS 設定情報ファイルの、次の各セクションのデータが設定されているかどうかをチェックします。Unattended、UserData、GuiRunOnce、および GuiUnattended。
- sif ファイル内にデフォルトの管理者パスワードと製品キーが指定され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](#).

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。

[HOW TO: Add Third-Party OEM Network Adapters to RIS Installations - 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](#)
- [Slipstream Switch for Windows 2000 SP1 Update.exe Does Not Work with RIS Server Images](#)
- 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で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。DCHP 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。

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せん。

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。

このリストは、Sun Fire x2100/x4100/x4200 システムにも対応するように拡張されています。また、IPMI システムタイプを指定するための変数 `device_type` が新しく追加されています。この変数には、次の値を使用できます。

- `sunfirev20z`
- `sunfirev40z`
- `sunfirex2100`
- `sunfirex4100`
- `sunfirex4200`

ブラウザインタフェースの変数プロンプトは、特定のデバイスタイプに対して設定可能な変数であるかどうかを示します。

新しい変数 `console_service` は、IPMI ターゲットホストでコンソールサービスを使用不可にするかどうかを示します。この変数のデフォルト値は `true` です。

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-v20z-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 v20z

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

default_router

説明: ネットワーク接続に使用するデフォルトルーター

例: 10.42.42.1

sysidcfg_default_route

説明: Solaris 配備用のデフォルトルーターの IP アドレス。

例: 10.42.42.1

sysidcfg_network_interface

説明: Solaris 配備用に構成するネットワークインタフェース。

デフォルト値: PRIMARY

networkifs_base_config

説明: Solaris 配備用に構成する追加のネットワークインタフェース。

例 : bge1!netB 255.255.255.0 myhost-netB 192.168.1.0

ipmp_networkifs_base_config

説明 : Solaris 配備用に IP Multipathing を構成します。

例 : qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache
10.0.0.4

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。

linux_network_interface

説明 : ターゲットホストへの Linux プロビジョニングに使用するネットワークインタフェース。この変数は、JET を使用して作成した Linux OS プロファイルに使用します。

デフォルト値 : eth0

networkifs_redhat_base_config

説明: RedHat Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した RedHat Linux OS プロファイルに使用します。

例: `eth1! --ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

networkifs_suse_base_config

説明: SUSE Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した SUSE Linux OS プロファイルに使用します。

例: `th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255`

guid

説明: {} 内で指定されるターゲットの GUID

注 - GUID は、{`ddddddd- dddd-dddd-dddd- dddddddddddd`} 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。

device_type

説明: システムタイプを指定します。例: `sunfirev20z`、`sunfirev40z`、`sunfirex2100`、`sunfirex4100`、`sunfirex4200`

デフォルト値: `sunfirev20z`

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

boot_service

説明: ターゲットホストへのインストールにブートネットオプションを使用するかどうかを指定します (`true/false`)。

デフォルト値: `true`

console_service

説明: ネットワーク管理インタフェースの提供するコンソールを使用するかどうかを指定します (`true/false`)。

デフォルト値: `true`

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。

SunALOM ターゲットホストの変数

次のリストに、SunALOM 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-v240-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 v240

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

default_router

説明: ネットワーク接続に使用するデフォルトルーター

例: 10.42.42.1

sysidcfg_default_route

説明: Solaris 配備用のデフォルトルーターの IP アドレス。

例: 10.42.42.1

sysidcfg_network_interface

説明: Solaris 配備用に構成するネットワークインタフェース。

デフォルト値: PRIMARY

networkifs_base_config

説明: Solaris 配備用に構成する追加のネットワークインタフェース。

例: bge1!netB 255.255.255.0 myhost-netB 192.168.1.0

ipmp_networkifs_base_config

説明: Solaris 配備用に IP Multipathing を構成します。

例: qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache
10.0.0.4

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_protocol`
説明: 端末サーバーとの接続に使用する通信プロトコル。
Cyclades 端末サーバーを使用する場合は、この変数を `ssh` に設定します。
デフォルト値: `telnet`
- `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
Cyclades 端末サーバーを使用する場合は、この変数を
`file:///opt/nlosp/drivers/cyclades/cycladesTS.jar` に設定します。
デフォルト値: `file:///opt/nlosp/drivers/cisco/c2621.jar`

Sun RSC ターゲットホストの変数

次のリストに、Sun RSC 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-v880-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 v880

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 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)

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

default_router

説明: ネットワーク接続に使用するデフォルトルーター

例:10.42.42.1

`sysidcfg_default_route`

説明: Solaris 配備用のデフォルトルーターの IP アドレス。

例:10.42.42.1

`sysidcfg_network_interface`

説明: Solaris 配備用に構成するネットワークインタフェース。

デフォルト値: PRIMARY

`networkifs_base_config`

説明: Solaris 配備用に構成する追加のネットワークインタフェース。

例: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

説明: Solaris 配備用に IP Multipathing を構成します。

例: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`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

rsc_ip_address

説明: RSC の IP アドレス

例: 10.5.133.135

rsc_access_userid

説明: RSC にアクセスするためのユーザー ID

rsc_access_password

説明: RSC 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_protocol

説明: 端末サーバーとの接続に使用する通信プロトコル。

Cyclades 端末サーバーを使用する場合は、この変数を ssh に設定します。

デフォルト値: telnet

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

Cyclades 端末サーバーを使用する場合は、この変数を
file:///opt/nlosp/drivers/cyclades/cycladesTS.jar に設定します。

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-v120-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 v120

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

default_router

説明: ネットワーク接続に使用するデフォルトルーター

例: 10.42.42.1

`sysidcfg_default_route`

説明: Solaris 配備用のデフォルトルーターの IP アドレス。

例: 10.42.42.1

`sysidcfg_network_interface`

説明: Solaris 配備用に構成するネットワークインタフェース。

デフォルト値: PRIMARY

`networkifs_base_config`

説明: Solaris 配備用に構成する追加のネットワークインタフェース。

例: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

説明: Solaris 配備用に IP Multipathing を構成します。

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`

`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 HALJAR ファイルへのパス

デフォルト値:

file:///:[targetableComponent:installPath]/SUNWn1osp/drivers/sun/lom.jar

lom_access_userid

説明: LOM にアクセスするためのユーザー ID

lom_access_password

説明: LOM 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_protocol

説明: 端末サーバーとの接続に使用する通信プロトコル。

Cyclades 端末サーバーを使用する場合は、この変数を ssh に設定します。

デフォルト値: telnet

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

Cyclades 端末サーバーを使用する場合は、この変数を

file:///opt/n1osp/drivers/cyclades/cycladesTS.jar に設定します。

デフォルト値:

file:///:[targetableComponent:installPath]/SUNWn1osp/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-4800-A-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 4800 domain

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

default_router

説明: ネットワーク接続に使用するデフォルトルーター。

例: 10.42.42.1

sysidcfg_default_route

説明: Solaris 配備用のデフォルトルーターの IP アドレス。

例: 10.42.42.1

sysidcfg_network_interface

説明: Solaris 配備用に構成するネットワークインタフェース。

デフォルト値: PRIMARY

networkifs_base_config

説明: Solaris 配備用に構成する追加のネットワークインタフェース。

例: bge1!netB 255.255.255.0 myhost-netB 192.168.1.0

ipmp_networkifs_base_config

説明: Solaris 配備用に IP Multipathing を構成します。

例: qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache
10.0.0.4**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

HPILO ターゲットホストの変数

次のリストに、HPILO 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: `hp-proliant1-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 hp proliant1 host`

`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`

`default_router`

説明: ネットワーク接続に使用するデフォルトルーター。

例: 10.42.42.1

`sysidcfg_default_route`

説明: Solaris 配備時のデフォルトルーターの IP アドレス。

例: 10.42.42.1

`sysidcfg_network_interface`

説明: Solaris 配備用に構成するネットワークインタフェース。

デフォルト値: PRIMARY

`networkifs_base_config`

説明: Solaris 配備用に構成する追加のネットワークインタフェース。

例: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

説明: Solaris 配備用に IP Multipathing を構成します。

例: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`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

linux_network_interface

説明: ターゲットホストへの Linux プロビジョニングに使用するネットワークインタフェース。この変数は、JET を使用して作成した Linux OS プロファイルに使用しません。

デフォルト値: eth0

networkifs_redhat_base_config

説明: RedHat Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した RedHat Linux OS プロファイルに使用します。

例: eth1!--ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static

networkifs_suse_base_config

説明: SUSE Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した SUSE Linux OS プロファイルに使用します。

例: th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255

guid

説明: {} 内で指定されるターゲットの GUID

注 - GUID は、{ddddddd- dddd-dddd-dddd- dddddddddddd} 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

default_router

説明: ネットワーク接続に使用するデフォルトルーター。

例:10.42.42.1

`sysidcfg_default_route`

説明: Solaris 配備用のデフォルトルーターの IP アドレス。

例:10.42.42.1

`sysidcfg_network_interface`

説明: Solaris 配備用に構成するネットワークインタフェース。

デフォルト値: PRIMARY

`networkifs_base_config`

説明: Solaris 配備用に構成する追加のネットワークインタフェース。

例: `bge1!netB 255.255.255.0 myhost-netB 192.168.1.0`

`ipmp_networkifs_base_config`

説明: Solaris 配備用に IP Multipathing を構成します。

例: `qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache 10.0.0.4`

`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

`linux_network_interface`

説明: ターゲットホストへの Linux プロビジョニングに使用するネットワークインタフェース。この変数は、JET を使用して作成した Linux OS プロファイルに使用します。

デフォルト値: `eth0`

`networkifs_redhat_base_config`

説明: RedHat Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した RedHat Linux OS プロファイルに使用します。

例: `eth1! --ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

`networkifs_suse_base_config`

説明: SUSE Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した SUSE Linux OS プロファイルに使用します。

例: `th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255`

`guid`

説明: {} 内で指定されるターゲットの GUID

注 - GUID は、{`dddddddd-dddd-dddd-dddd-dddddddddd`} 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]/SUNWn1osp/drivers/sun/ipmi.jar`

`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。ほかのカテゴリのいずれにも適合せず、かつ電源やブート、コンソール機能を自動化するためのターゲット定義に結合可能な HAL 実装を持つ遠隔管理ターゲットには、Other タイプを使用します。N1 SPS 5.2 製品には、いくつかの HAL 実装が含まれています。たとえば、Sun Fire B100s Blade サーバーは、電源、ブート、およびコンソールサービスをサポートする `file:///:[targetableComponent:installPath]/SUNWn1osp/drivers/sun/b1600sc.jar` 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-b100s-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 b100s blade server`

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`

default_router

説明: ネットワーク接続に使用するデフォルトルーター。

例: `10.42.42.1`

sysidcfg_default_route

説明: Solaris 配備用のデフォルトルーターの IP アドレス。

例: `10.42.42.1`

sysidcfg_network_interface

説明: Solaris 配備用に構成するネットワークインタフェース。

デフォルト値: `PRIMARY`

networkifs_base_config

説明: Solaris 配備用に構成する追加のネットワークインタフェース。

例: bge1!netB 255.255.255.0 myhost-netB 192.168.1.0

ipmp_networkifs_base_config

説明: Solaris 配備用に IP Multipathing を構成します。

例: qfe0_qfe4!database-net l 10.0.0.1 10.0.0.2 24 oracle-db 10.0.0.3 apache
10.0.0.4

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

linux_network_interface

説明: ターゲットホストへの Linux プロビジョニングに使用するネットワークインタフェース。この変数は、JET を使用して作成した Linux OS プロファイルに使用しません。

デフォルト値:eth0

networkifs_redhat_base_config

説明: RedHat Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した RedHat Linux OS プロファイルに使用します。

例: eth1!--ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static

networkifs_suse_base_config

説明: SUSE Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した SUSE Linux OS プロファイルに使用します。

例: th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255

guid

説明: {} 内で指定されるターゲットの GUID

注 - GUID は、{dddddddd-dddd-dddd-dddd-dddddddddd} 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

Cyclades 端末サーバーを使用する場合は、この変数を
file:///opt/n1osp/drivers/cyclades/cycladesTS.jar に設定します。

デフォルト値:

file:///:[targetableComponent:installPath]/SUNWn1osp/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 HALJAR ファイルへのパス

デフォルト値:

file:///:[targetableComponent:installPath]/SUNWn1osp/drivers/sun/ipmi.jar

`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]/SUNWn1osp/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。ほかのいずれかのターゲットホストタイプにも一致せず、かつ電源やブート、コンソール機能を自動化する HAL 実装を持たないホストには、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 ultra 10

`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

`default_router`
説明: ネットワーク接続に使用するデフォルトルーター。

例: 10.42.42.1

`sysidcfg_default_route`
説明: Solaris 配備用のデフォルトルーターの IP アドレス。

例: 10.42.42.1

`sysidcfg_network_interface`
説明: Solaris 配備用に構成するネットワークインタフェース。

デフォルト値: PRIMARY

`networkifs_base_config`
説明: Solaris 配備用に構成する追加のネットワークインタフェース。

例: bge1!netB 255.255.255.0 myhost-netB 192.168.1.0

`ipmp_networkifs_base_config`
説明: Solaris 配備用に IP Multipathing を構成します。

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

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

linux_network_interface

説明: ターゲットホストへの Linux プロビジョニングに使用するネットワークインタフェース。この変数は、JET を使用して作成した Linux OS プロファイルに使用しません。

デフォルト値: eth0

networkifs_redhat_base_config

説明: RedHat Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した RedHat Linux OS プロファイルに使用します。

例: eth1! --ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static

networkifs_suse_base_config

説明: SUSE Linux の配備用に構成する追加のネットワークインタフェース。この変数は、JET を使用して作成した SUSE Linux OS プロファイルに使用します。

例: `th1!static!onboot!1.2.3.4!255.255.255.0!1.2.3.255`

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。

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。

- 219 ページの「基本 Solaris OS 構成変数」
- 232 ページの「Solaris リモートエージェントのコンポーネント変数」
- 234 ページの「一般的な Solaris OS プロビジョニングプラグイン変数」

基本 Solaris 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。

Solaris 10 1/06 x86 リリースをプロビジョニングする場合は、この変数の値を `grub` に設定して、Grub ベースのブートおよびインストールを行えるようにします。

注-`client_allocation_base_config` を `bootp` に設定する場合は、`osp_profile_control_service` を `false` に設定します。

`grub_append_base_config`

説明: Solaris 10 1/06 x86 システムの場合は、GRUB ブートローダーに渡す追加オプションまたは引数を指定します。

`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_config`

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_config`

説明: 暗号化された root パスワード。

例: `boajr0mU7GFmY` は、パスワード「`newroot`」を暗号化した値です。

sysidcfg_system_locale_base_config

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_config

説明: 初期ブート時に使用する Kerberos セキュリティポリシー。

デフォルト値: NONE

sysidcfg_protocol_ipv6_base_config

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_config

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ています。この変数も空白の場合、または別のネットワーク用 interfaces 用の場合は、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は、このオプションを `yes` に設定する必要があります。

x86_disable_kdmconfig_base_config

説明: Solaris x86 システムで、ターゲットホストのキーボード、ディスプレイ、およびマウスを構成するための対話型 `kdmconfig` ユーティリティを無効にします。GRUB ブートローダーを持つ Solaris OS をインストールする場合は、この変数の値を `yes` に設定します。

x86_confflags_base_config

説明: Solaris 9 x86 システムで、`add_install_client` コマンドの `confflags` 属性で使用する引数を指定します。

例: `-f -P /boot/solaris/dca.`

sysidcfg_x86_kdmfile_base_config

説明: Solaris x86 システムで、`sysidcfg` ファイルに付加するキーボード、ディスプレイ、およびマウス構成ファイルの名前を指定します。

例: `:/sysidcfg-addon-file.`

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。

注-x86 ターゲットホストで Solaris OS をプロビジョニングする場合は、既存のすべてのパーティションを削除するカスタム JumpStart プロファイルを作成し、`profile_base_config` 変数でそのプロファイルを指定する必要があります。fdisk キーワードを使用して既存のパーティションを削除する JumpStart プロファイルを作成する方法については、『Solaris 10 インストールガイド (カスタム JumpStart/ 上級編)』を参照してください。

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

- SUNWCall — 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。 *cntndn* の形式、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。 *cntndn* の形式を使用します。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 M バイトサイズ)

デフォルト値: *free*

profile_swap_base_config

説明: スワップ容量 (M バイト単位)。

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 (M バイト単位)。

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 (M バイト単位)。

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 (M バイト単位)。

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(Mバイト単位)。

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(Mバイト単位)。

profile_additional_disks_base_config

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。cntndn の形式を使用します。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、[229 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。

Solaris x86 のヘッドレスターゲットホストでの対話式インストールを回避するには、この値を `SUNWxwssu` `SUNWxwscf` に設定します。

`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せん。

`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] 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。cntrdn の形式を使用します。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、227 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、228 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 l 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、代替のブレイ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 変数を使用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 `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_networkifs_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。
 - `1` — 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 `if` 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 `if` 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 l 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_config` 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、`cnt ndn` の形式を使用します。

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_config` 変数は、2 つの追加ディスク `c0t1d0` および `c0t1d1` を特定します。ディスクスライスごとに、そのディレクトリのサイズとともにディレクトリ名を指定します。

```
profile_usedisk_base_config = c0t0d0
profile_additional_disks_base_config = c0t1d0s0!/export/mydisk1!5000MB|c0t1d1s0!/export/mydisk2!6000MB
```

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。

x86 ベースシステムへの Solaris 9 インストールの自動化

x86 ベースシステムへの Solaris 9 のインストールを完全に自動化するには、Solaris Device Configuration Assistant (DCA) および `kdmconfig` ユーティリティーに情報を提供する追加構成ファイルを作成する必要があります。

▼ Solaris 9 x86 のインストールを自動化する

- 1 次の内容を持つ `dca` という名前のファイルを作成します。

```
IF 0,2="Solaris Device Configuration" THEN
F2
FI
IF *,0="Enter_Continue" THEN
ENTER
FI
IF *,0="Enter_Continue" THEN
ENTER
FI
IF *,0="Enter_Continue" THEN
ENTER
FI
F2 F2
```

- 2 **Solaris 9** インストールイメージ内の `boot/solaris` ディレクトリに、この `dca` ファイルを保存します。
たとえば、`/export/s9_x86/Solaris_9/Tools/Boot/boot/solaris` ディレクトリにファイルを保存します。

- 3 **Solaris 9** インストールイメージの `Solaris_9/Tools/Boot/etc` ディレクトリにある `bootrc` ファイルをエディタで開き、次のような内容にします。

```
#      @(#)bootrc 1.10 01/04/18
#
# bootrc script for installations
#
# ER - This script has been reduced to just start up the kernel boot.
# No user interaction.

# definitions
set DEFAULT_BOOTFILE kernel/unix

# retrieve boot-path constructed by MDB
set bootfile $DEFAULT_BOOTFILE
getprop boot-path bp
getprop boot-args ba
set bootargs "${ba}"

# display current defaults
echo "                                <<< Current Boot Parameters >>>"
Boot path: ${bp}
Boot args: ${ba}

"

        setprop boot-args '- install nowin'
        echo '
<<< starting automatic installation >>>
'
echo "Booting $DEFAULT_BOOTFILE..."
run $DEFAULT_BOOTFILE
```

- 4 次のテキスト内容を持つ `/sysidcfg-addon-file` という名前のファイルを作成します。

```
keyboard=workaround:Unknown
display=workaround:Unknown
pointer=workaround:Unknown
monitor=workaround:Unknown
```

- 5 **N1 SPS** ブラウザインタフェースで次の基本 **Solaris OS** 構成変数を編集します。

<code>x86_confflags_base_config</code>	この変数が Solaris インストールイメージディレクトリ内の <code>dca</code> ファイルを指し示すようにします。たとえば <code>-f -P/boot/solaris/dca</code> です。
<code>sysidcfg_x86_kdmfile_base_config</code>	この変数が、作成した <code>/sysidcfg-addon-file</code> を指し示すようにします。

Solaris OS プロビジョニングプロファイルの変数の作成または編集方法については、71 ページの「[非 JET Solaris プロビジョニングプロファイルを作成する \(ブラウザインタフェース\)](#)」を参照してください。

Solaris 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: 5.2

OS プロビジョニングプラグイン 3.0 によるプロビジョニングに N1 SPS 5.1 リリースを使用する場合は、この変数の値を 5.1 に設定します。

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

注 - install_ra_from_snapshot_spsra は廃止予定で、N1 SPS 5.2 リリースでは非推奨です。

デフォルト値: n

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指定します。

注-プロビジョニング後、ターゲットでユーザーアカウントを使用できない場合は、この変数値を `y` に設定して、プロビジョニング後の再起動でターゲットホスト上の N1 SPS リモートエージェントを起動するようにします。

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: 6GR9g0i1vRVFw 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、インストーラが失敗します。

`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

`setuid_spsra`

説明: リモートエージェントのホスト名または IP アドレス

例: solaris10-target

`conn_spsra`

説明: RA への接続に使用する方式を指定します。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 Solaris OS プロビジョニングプラグイン変数

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)。

注 - client_allocation_base_config を bootp に設定する場合は、osp_profile_control_service を false に設定します。

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_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]`

RedHat Linux Profile コンポーネントの変数

RedHat Linux 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。

- 237 ページの「基本 RedHat Linux OS 構成変数」
- 251 ページの「RedHat Linux リモートエージェントのコンポーネント変数」
- 253 ページの「一般的な RedHat Linux OS プロビジョニングプラグイン変数」

基本 RedHat Linux 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。RedHat Linux 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: 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]

ClientOSName_base_config

説明: プロビジョニングする OS の名前。

例: redhat

ClientOS_base_config

説明: プロビジョニングする OS のバージョン。

例: LinuxRedHat3.0_x86

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

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。

productdir_base_config

説明: RedHat Linux ディストリビューションへのパス。`jumpstart.conf` ファイルの情報と JET サーバーの IP アドレスを使用するには、この値を空白のままにします。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: 適用するパッチへのパス。`jumpstart.conf` ファイルの情報と JET サーバーの IP アドレスを使用するには、この値を空白のままにします。パッチが 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。

kickstart_media_base_config

説明: RedHat Linux 製品媒体の位置。

pxelinux_console_base_config

説明: PXE ブート環境に渡すコンソールパラメータ。

デフォルト値: `ttyS0,9600`

lang_base_config

説明: ターゲットホストにインストールする言語。

例: `en_US`

lang_additional_base_conf

説明: ターゲットホストにインストールする追加言語。

例: `fr_FR`

keyboard_base_conf

説明: ターゲットホストで使用するキーボードマッピング。

デフォルト値: `us`

mouse_base_config

説明: ターゲットホストのマウスタイプ。

デフォルト値: generic3

auth_md5_base_config

説明: MD5 認証を使用するかどうかを指定します。

デフォルト値: yes

auth_shadow_base_config

説明: シADOWパスワードを使用するかどうかを指定します。

デフォルト値: yes

auth_cache_base_config

説明: 認証キャッシュを使用するかどうかを指定します。

デフォルト値: no

auth_nis_enable_base_config

説明: NIS を有効にするかどうかを指定します (yes/no)。

auth_nis_server_base_config

説明: NIS サーバーの名前 (auth_nis_enable_base_config を yes に設定した場合)。

例: nis-1.west.example.com

auth_nis_domain_base_config

説明: NIS ドメインの名前。

例: west.example.com

auth_ldap_enable_base_config

説明: LDAP を有効にするかどうかを指定します (yes/no)。

auth_ldap_enable_auth_base_config

説明: ユーザー認証に LDAP を使用するかどうかを指定します (yes/no)。

auth_ldap_server_base_config

説明: LDAP サーバーの名前。

例: ldap.west.example.com

auth_ldap_basedn_base_config

説明: LDAP サーバーのベースドメイン名。

例: west.example.com

auth_ldap_tls_base_config

説明: LDAP に Transport Layer Security (TLS) プロトコルを使用するかどうかを指定します (yes/no)。

`auth_kerberos5_kdcs_base_config`

説明: Kerberos キーディストリビューションセンター (KDC) のリスト。スペース区切りリストのエントリを含みます。

`auth_kerberos5_realm_base_config`

説明: Kerberos デフォルトレルムの名前。

`auth_kerberos5_adminserver_base_config`

説明: Kerberos 管理サーバーの名前。

`auth_hesiod_enabled_base_config`

説明: Hesiod ネームサービスを有効にするかどうかの指定 (yes/no)。

`auth_hesiod_right_side_base_config`

説明: /etc/hesiod.conf で設定される Hesiod の right side オプションを使用するかどうかを指定します (yes/no)。

このオプションは、LDAP によるベースドメイン名検索の場合と同じように、情報検索時に、DNS 検索対象の名前を決定するときに Hesiod ライブラリによって使用されます。

`auth_hesiod_left_side_base_config`

説明: /etc/hesiod.conf で設定される Hesiod の left side オプションを使用するかどうかを指定します (yes/no)。

このオプションは、LDAP によるベースドメイン名検索の場合と同じように、情報検索時に、DNS 検索対象の名前を決定するときに Hesiod ライブラリによって使用されません。

`auth_smb_servers_base_config`

説明: Server Message Block サーバーの名前。SMB は、コンピュータ間でのファイル、プリンタ、シリアルポート、および通信アブストラクション (名前付きパイプ、メールスロットなど) の共有に関するプロトコルです。

`auth_smb_workgroup_base_config`

説明: SMB ワークグループの名前。

`root_password_encrypted_base_config`

説明: `root_password_base_config` 変数のターゲットホストの root パスワードが暗号化されているかどうかを指定します (yes/no)。

デフォルト値: no

`root_password_base_config`

説明: ターゲットホストの root パスワード。 `root_password_encrypted_base_config` 変数が yes に設定されている場合は、この変数に暗号化したパスワードを設定します。

デフォルト値: newroot

`timezone_base_config`

説明: ターゲットホストのタイムゾーン。

デフォルト値: US/Pacific

timezone_utc_base_config

説明: タイムゾーンが協定世界時で設定されているかどうかを指定します (yes/no)。

デフォルト値: no

firewall_base_config

説明: ターゲットホストでファイアウォールを有効にするかどうかを指定します (enabled/disabled)。

デフォルト値: disabled

firewall_trusted_base_config

説明: すべてのトラフィックがファイアウォールの通過を許可されている、ネットワークインタフェースのスペース区切りリスト。

例: eth0 eth1

firewall_incoming_base_config

説明: ポート番号およびサービス (ssh, http, dhcp, telnet, smtp, ftp など) のスペース区切りのリスト。service:protocol の形式でサービスの種類と関連するプロトコルを指定することもできます。

例: imap:tcp 1234:udp ssh ftp

bootloader_name_base_config

説明: ターゲットホストをブートし、インストールするブートローダーの名前 (grub/lilo)。

デフォルト値: grub

bootloader_location_base_config

説明: ブートローダーの位置。

デフォルト値: mbr

bootloader_append_base_config

説明: ブートローダーがカーネルに渡す、カーネルパラメータのスペース区切りリスト。

例: hdd=ide-scsi ide=nodma

bootloader_lilo_lba32_base_config

説明: bootloader_name_base_config 変数に lilo が設定されている場合、この変数はブートローダーが lba32 を使用するかどうかを指定します。

デフォルト値: yes

bootloader_lilo_linear_base_config

説明: bootloader_name_base_config 変数に lilo が設定されている場合、この変数はブートローダーが linear を使用するかどうかを指定します。

デフォルト値: yes

`bootloader_grub_password_encrypted_base_config`

説明: `bootloader_name_base_config` 変数に `grub` が設定されている場合、この変数は、`root_password_base_config` 変数の GRUB root パスワードが暗号化されているかどうかを指定します (yes/no)。

例: no

`bootloader_grub_password_base_config`

説明: ターゲットホストの GRUB root パスワード。

`bootloader_grub_password_encrypted_base_config` 変数が `yes` に設定されている場合は、この変数に暗号化したパスワードを設定します。

`xconfig_card_base_config`

説明: ターゲットホストのビデオカード。

`xconfig_videoram_base_config`

説明: ターゲットホストのビデオ RAM。

`xconfig_monitor_base_config`

説明: ターゲットホストのモニターモデル。

`xconfig_resolution_base_config`

説明: ターゲットホストのモニターの解像度。

`xconfig_depth_base_config`

説明: ターゲットホストのモニターの発色数。

`xconfig_desktop_base_config`

説明: ターゲットホストで使用するデスクトップソフトウェアを指定します。

`bootproto_base_config`

説明: ターゲットホストのブートプロトコル。

デフォルト値: dhcp

`network_interface_base_config`

説明: プロビジョニング中に使用するネットワークインタフェース。

デフォルト値: :[targetableComponent:linux_network_interface]

`ip_base_config`

説明: ターゲットホストの IP アドレス。

デフォルト値: :[targetableComponent:ethernet_ip_address]

`hostname_base_config`

説明: ターゲットホストのホスト名。

デフォルト値: :[targetableComponent:installPath]

`netmask_base_config`

説明: プロビジョニング中に使用するネットマスク。

デフォルト値: :[targetableComponent:ethernet_netmask]

gateway_base_config

説明: プロビジョニング中に使用するゲートウェイ。

デフォルト値: [targetableComponent:default_router]

nameservers_base_config

説明: プロビジョニング中に使用するネームサーバーのスペース区切りリスト。

networkifs_base_config

説明: 構成する追加のネットワークインタフェースを指定します。この変数の値は、ターゲットホスト変数 `networkifs_redhat_base_config` から算出されます。

例: `l eth1! --ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

partition_clear_base_config

説明: プロビジョニング中に内容を消去するパーティションを指定します。

例: `all`

partition_zerombr_base_config

説明: マスターブートレコード (MBR) の内容を消去するかどうかを指定します。

デフォルト値: `yes`

partition_1_disk_base_config

説明: 1 番目のパーティションのデバイスを指定します。

デフォルト値: `sda`

注-パーティション変数は、パーティション1から順に定義してください。パーティション1から始めなかった場合は、インストールが対話式に切り替わります。

partition_1_mountpoint_base_config

説明: 1 番目のパーティションのマウントポイントを指定します。

デフォルト値: `/diag`

partition_1_fstype_base_config

説明: 1 番目のパーティションのファイルシステムの種類を指定します。

デフォルト値: `vfat`

partition_1_size_base_config

説明: 1 番目のパーティションのサイズ (M バイト単位) を指定します。

デフォルト値: `60`

partition_1_maxsize_base_config

説明: 1 番目のパーティションの最大サイズ (M バイト単位) を指定します。

partition_1_grow_base_config

説明: パーティションの拡張を許可するかどうかを指定します (yes/no)。

`partition_1_primary_base_config`

説明: 強制的にプライマリパーティションにするかどうかを指定します (yes/no)。

デフォルト値: yes

`partition_2_disk_base_config`

説明: 2 番目のパーティションのデバイスを指定します。

デフォルト値: sda

`partition_2_mountpoint_base_config`

説明: 2 番目のパーティションのマウントポイントを指定します。

デフォルト値: /boot

`partition_2_fstype_base_config`

説明: 2 番目のパーティションのファイルシステムの種類を指定します。

デフォルト値: ext3

`partition_2_size_base_config`

説明: 2 番目のパーティションのサイズを指定します (M バイト単位)。

デフォルト値: 100

`partition_2_maxsize_base_config`

説明: 2 番目のパーティションの最大サイズを指定します (M バイト単位)。

`partition_2_grow_base_config`

説明: パーティションの拡張を許可するかどうかを指定します (yes/no)。

`partition_2_primary_base_config`

説明: 強制的にプライマリパーティションにするかどうかを指定します (yes/no)。

デフォルト値: yes

`partition_3_disk_base_config`

説明: 3 番目のパーティションのデバイスを指定します。

デフォルト値: sda

`partition_3_mountpoint_base_config`

説明: 3 番目のパーティションのマウントポイントを指定します。

デフォルト値: /

`partition_3_fstype_base_config`

説明: 3 番目のパーティションのファイルシステムの種類を指定します。

デフォルト値: ext3

`partition_3_size_base_config`

説明: 3 番目のパーティションのサイズを指定します (M バイト単位)。

デフォルト値: 10000

`partition_3_maxsize_base_config`

説明:3番目のパーティションの最大サイズを指定します(Mバイト単位)。

`partition_3_grow_base_config`

説明:パーティションの拡張を許可するかどうかを指定します(yes/no)。

`partition_3_primary_base_config`

説明:強制的にプライマリパーティションにするかどうかを指定します(yes/no)。

`partition_4_disk_base_config`

説明:4番目のパーティションのデバイスを指定します。

デフォルト値:sda

`partition_4_mountpoint_base_config`

説明:4番目のパーティションのマウントポイントを指定します。

デフォルト値:swap

`partition_4_fstype_base_config`

説明:4番目のパーティションのファイルシステムの種類を指定します。

デフォルト値:swap

`partition_4_size_base_config`

説明:4番目のパーティションのサイズを指定します(Mバイト単位)。

デフォルト値:1024

`partition_4_maxsize_base_config`

説明:4番目のパーティションの最大サイズを指定します(Mバイト単位)。

`partition_1_grow_base_config`

説明:パーティションの拡張を許可するかどうかを指定します(yes/no)。

`partition_4_primary_base_config`

説明:強制的にプライマリパーティションにするかどうかを指定します(yes/no)。

`partition_5_disk_base_config`

説明:5番目のパーティションのデバイスを指定します。

`partition_5_mountpoint_base_config`

説明:5番目のパーティションのマウントポイントを指定します。

`partition_5_fstype_base_config`

説明:5番目のパーティションのファイルシステムの種類を指定します。

`partition_5_size_base_config`

説明:5番目のパーティションのサイズを指定します(Mバイト単位)。

`partition_5_maxsize_base_config`

説明:5番目のパーティションの最大サイズを指定します(Mバイト単位)。

`partition_5_grow_base_config`

説明:パーティションの拡張を許可するかどうかを指定します(yes/no)。

`partition_5_primary_base_config`

説明: 強制的にプライマリパーティションにするかどうかを指定します (yes/no)。

`partition_6_disk_base_config`

説明: 6番目のパーティションのデバイスを指定します。

`partition_6_mountpoint_base_config`

説明: 6番目のパーティションのマウントポイントを指定します。

`partition_6_fstype_base_config`

説明: 6番目のパーティションのファイルシステムの種類を指定します。

`partition_6_size_base_config`

説明: 6番目のパーティションのサイズを指定します (Mバイト単位)。

`partition_6_maxsize_base_config`

説明: 6番目のパーティションの最大サイズを指定します (Mバイト単位)。

`partition_6_grow_base_config`

説明: パーティションの拡張を許可するかどうかを指定します (yes/no)。

`partition_6_primary_base_config`

説明: 強制的にプライマリパーティションにするかどうかを指定します (yes/no)。

`partition_7_disk_base_config`

説明: 7番目のパーティションのデバイスを指定します。

`partition_7_mountpoint_base_config`

説明: 7番目のパーティションのマウントポイントを指定します。

`partition_7_fstype_base_config`

説明: 7番目のパーティションのファイルシステムの種類を指定します。

`partition_7_size_base_config`

説明: 7番目のパーティションのサイズを指定します (Mバイト単位)。

`partition_7_maxsize_base_config`

説明: 7番目のパーティションの最大サイズを指定します (Mバイト単位)。

`partition_7_grow_base_config`

説明: パーティションの拡張を許可するかどうかを指定します (yes/no)。

`partition_7_primary_base_config`

説明: 強制的にプライマリパーティションにするかどうかを指定します (yes/no)。

`partition_8_disk_base_config`

説明: 8番目のパーティションのデバイスを指定します。

`partition_8_mountpoint_base_config`

説明: 8番目のパーティションのマウントポイントを指定します。

`partition_8_fstype_base_config`

説明: 8番目のパーティションのファイルシステムの種類を指定します。

partition_8_size_base_config

説明: 8 番目のパーティションのサイズを指定します (M バイト単位)。

partition_8_maxsize_base_config

説明: 8 番目のパーティションの最大サイズを指定します (M バイト単位)。

partition_8_grow_base_config

説明: パーティションの拡張を許可するかどうかを指定します (yes/no)。

partition_8_primary_base_config

説明: 強制的にプライマリパーティションにするかどうかを指定します (yes/no)。

raid_1_device_base_config

説明: 使用する RAID デバイスの名前 (md0、md1 など) を指定します。RAID デバイスの範囲は md0 ~ md7 で、それぞれ 1 回使用できます。

例: md0

raid_1_level_base_config

説明: 1 番目の RAID デバイスのレベル (0、1、または 5) を指定します。

例: 1

raid_1_mountpoint_base_config

説明: 1 番目の RAID デバイスのマウントポイントを指定します。

例: raid.01

raid_1_meta_devices_base_config

説明: 1 番目の RAID デバイスのメタデバイスのスペース区切りリストを指定します。これらのメタデバイスは、`partition_n mountpoint_base_config` 変数に `raid.number` を指定することによって、`partition_` 変数で作成します (例: `partition_1_mountpoint_base_config=raid.01`)。

`raid.01` および `raid.02` のパーティション名は、JET `base_config` モジュールを使用して作成可能です。`raid_1_level_base_config` 変数を使用して、RAID レベルを 1 に設定する場合は、同じサイズのパーティションを少なくとも 2 つ指定する必要があります。`raid_1_level_base_config` 変数を使用して、RAID レベルを 5 に設定する場合は、同じサイズのパーティションを少なくとも 3 つ指定する必要があります。

raid_2_device_base_config

説明: 使用する RAID デバイスの名前 (md0、md1 など) を指定します。RAID デバイスの範囲は md0 ~ md7 で、それぞれ 1 回使用できます。

例: md1

raid_2_level_base_config

説明: 2 番目の RAID デバイスのレベル (0、1、または 5) を指定します。

例: 1

raid_2_mountpoint_base_config

説明: 2 番目の RAID デバイスのマウントポイントを指定します。

例:raid.02

`raid_2_meta_devices_base_config`

説明:2番目のRAIDデバイスのメタデバイスのスペース区切りリストを指定します。これらのメタデバイスは、`partition_n mountpoint_base_config`変数に`raid.number`を指定することによって、`partition_`変数で作成します(例:`partition_2_mountpoint_base_config=raid.02`)。

`raid.01`および`raid.02`のパーティション名は、JET `base_config`モジュールを使用して作成可能です。`raid_1_level_base_config`変数を使用して、RAIDレベルを1に設定した場合は、同じサイズのパーティションを少なくとも2つ指定する必要があります。`raid_1_level_base_config`変数を使用して、RAIDレベルを5に設定した場合は、同じサイズのパーティションを少なくとも3つ指定する必要があります。

`append_kernel_cmdline_base_config`

説明:PXEブート時のカーネルに対する追加のコマンド行オプション。

`clusters_base_config`

説明:インストールするパッケージグループを指定します。

デフォルト値:Everything

`packages_add_base_config`

説明:ターゲットホストにインストールする追加パッケージを含むスペース区切りリスト。

`packages_remove_base_config`

説明:ターゲットホストから削除する追加パッケージを含むスペース区切りリスト。

`packages_resolvedeps_base_config`

説明:プロビジョニング中のパッケージの依存関係を解消するかどうかを指定します(yes/no)。

`packages_ignoredeps_base_config`

説明:プロビジョニング中のパッケージの依存関係を無視するかどうかを指定します(yes/no)。

`packages_ignoremissing_base_config`

説明:存在しないパッケージを無視して、プロビジョニングを継続するかどうかを指定します(yes/no)。

`enable_services_base_config`

説明:ターゲットホスト上で有効にするXinetdサービスのスペース区切りリスト。

`postscript_chroot_base_config`

説明:別のルートディレクトリでJETポストスクリプトを実行するかどうかを指定します(yes/no)。

`bootloader_grub_redirect_fix_base_config`

説明:シリアルコンソールではなく、コンソールにインストール出力をリダイレクトするかどうかを指定します(yes/no)。

一部ハードウェアでは、インストーラによって、ブートローダーにシリアルまたはコンソールディスプレイが正しく設定されません。ハードウェアが GRUB ブートローダーにコンソールへのリダイレクトを正しく設定しない場合は、この変数を `yes` に設定します。

`no_autoreboot_base_config`

説明: インストール後にリブートするかどうかを指定します (`yes/no`)。

この変数を `no` に設定した場合は、システムがリブートするまで、ポストインストールスクリプトは実行されません。

`script_1_pre_base_config`

説明: JET サーバー上のプリインストールスクリプトの位置を指定します。

`script_1_pre_interpreter_base_config`

説明: プリインストールスクリプトのインタプリタを指定します。

`script_1_post_base_config`

説明: JET サーバー上のポストインストールスクリプトの位置を指定します。

`script_1_post_interpreter_base_config`

説明: ポストインストールスクリプトのインタプリタを指定します。

`script_1_post_nochroot_base_config`

説明: 代替ルートディレクトリの `jail` 内でポストインストールスクリプトを実行するかどうかを指定します。

`partitionadd_base_config`

説明: パーティション 9 から始まるパーティションをターゲットホストに追加します。変数値の構文は次のとおりです。

`n!disk!mountpoint!fstype!size!maxsize!grow!primary`

ここでの各要素の意味は次のとおりです。

`n` には、パーティション番号を指定します。

`disk` には、パーティションのデバイスを指定します。

`mountpoint` には、パーティションのマウントポイントを指定します。

`fstype` には、パーティションのファイルシステムの種類を指定します。

`size` には、パーティションのサイズを M バイト単位で指定します。

`maxsize` には、パーティションの最大サイズを指定します。

`grow` は、パーティションの拡張を許可するかどうかを指定します。

`primary` は、このパーティションを強制的にプライマリパーティションにするかどうかを指定します。

変数値には、すべての追加パーティションの情報を含めます。たとえば、9番目と10番目のパーティションを作成する場合は、この変数に、次の例のような値を設定します。

```
9!sda!/export1!ext3!1000!!!!|10!sda!/export10!jfs!100!2000!yes!!
```

この値によって、次の変数セットが作成されます。

```
base_config_partition_9_disk=sda
base_config_partition_9_mountpoint=/export1
base_config_partition_9_fstype=ext3
base_config_partition_9_size=1000
base_config_partition_9_maxsize=
base_config_partition_9_grow=
base_config_partition_9_primary=

base_config_partition_10_disk=sda
base_config_partition_10_mountpoint=/export10
base_config_partition_10_fstype=jfs
base_config_partition_10_size=100
base_config_partition_10_maxsize=2000
base_config_partition_10_grow=yes
base_config_partition_10_primary=
```

raidadd_base_config

説明: デバイス 3 から始まる RAID デバイスをターゲットホストに追加します。変数値の構文は次のとおりです。

```
n!device!level!mountpoint!metadevices
```

ここでの各要素の意味は次のとおりです。

n には、RAID ボリューム番号を指定します。

device には、RAID ボリュームのデバイスを指定します。

level には、RAID ボリュームの RAID レベルを指定します。

mountpoint には、RAID ボリュームのマウントポイントを指定します。

metadevices には、RAID ボリュームに作成するメタデバイスを指定します。

変数値には、すべての追加パーティションの情報を含めます。raidadd_base_config 変数の基本構文は、partitionadd_base_config 変数の構文と同じです。

scriptpreadd_base_config

説明: 番号 2 から始まる追加のプリインストールスクリプトを OS プロビジョニングファイルに追加します。変数値の構文は次のとおりです。

```
n!pre!pre_interpreter
```

ここでの各要素の意味は次のとおりです。

n には、プリインストールスクリプト番号を指定します。

pre には、プリインストールスクリプトの名前を指定します。

pre_interpreter には、プリインストールスクリプトに使用するインタプリタを指定します。

変数値には、使用するすべての追加プリインストールスクリプトの情報を含めます。*scriptpreadd_base_config* 変数の基本構文は、*partitionadd_base_config* 変数の構文と同じです。

scriptpostadd_base_config

説明: 番号 2 から始まる追加のポストインストールスクリプトを OS プロビジョニングファイルに追加します。変数値の構文は次のとおりです。

n!post!post_interpreter!post_nochroot

ここでの各要素の意味は次のとおりです。

n には、ポストインストールスクリプト番号を指定します。

post には、ポストインストールスクリプトの名前を指定します。

post_interpreter には、ポストインストールスクリプトに使用するインタプリタを指定します。

post_nochroot は、ポストインストールスクリプトを、代替ルートディレクトリの jail 内で実行するかどうかを指定します (yes/no)。

変数値には、使用するすべての追加ポストインストールスクリプトの情報を含めます。*scriptpostadd_base_config* には、*partitionadd_base_config* 変数と同じ構文を使用します。

RedHat Linux 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: 5.2

OS プロビジョニングプラグイン 3.0 によるプロビジョニングに N1 SPS 5.1 リリースを使用する場合は、この変数の値を 5.1 に設定します。

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

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指定します。

注-プロビジョニング後、ターゲットからユーザーアカウントを使用できない場合は、この変数値を `y` に設定して、プロビジョニング後のリポートでターゲットホストで N1 SPS リモートエージェントを起動するようにします。

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: 6GR9g0i1vRVFw 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、インストーラが失敗します。

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

host_spsra

説明: リモートエージェントのホスト名または IP アドレス

デフォルト値: :[targetableComponent:sps_ra_host]

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: :[targetableComponent:sps_ra_ssl_auth]

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_parms]

一般的な RedHat Linux OS プロビジョニングプラグイン変数

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で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]`

`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 (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_boot_server_name`

説明: RedHat Linux 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: RedHat Linux 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/RedHat/RHEL3-U4

osp_profile_boot_ip_address

説明: RedHat Linux ブートサーバーの IP アドレス。

例: 199.199.8.6

SUSE Linux Profile コンポーネントの変数

SUSE Linux 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。

- 257 ページの「基本 SUSE Linux OS 構成変数」
- 279 ページの「SUSE Linux リモートエージェントのコンポーネント変数」
- 281 ページの「一般的な SUSE Linux OS プロビジョニングプラグイン変数」

基本 SUSE Linux 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。SUSE Linux 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: 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]

ClientOSName_base_config

説明: プロビジョニングする OS の名前。

例: redhat

ClientOS_base_config

説明: プロビジョニングする OS のバージョン。

例: LinuxRedHat3.0_x86

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

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。

productdir_base_config

説明: SUSE Linux ディストリビューションへのパス。`jumpstart.conf` ファイルの情報と JET サーバーの IP アドレスを使用するには、この値を空白のままにします。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: ターゲットホストにプロビジョニングするパッチへのパス。インストールするパッチが JET SUSE イメージサーバーにある場合は、この値を空白のままにしておきます。パッチが JET SUSE イメージサーバーにない場合は、NFS アクセス可能なシステムにある、それらのパッチの位置への絶対パスを指定します。

デフォルト値: null

suse_media_base_config

説明: SUSE Linux 製品媒体の位置。

pxelinux_console_base_config

説明: PXE ブート環境に渡すコンソールパラメータ。

例: ttyS0,9600

append_kernel_cmdline_base_config

説明: PXE ブート時にカーネルに渡す追加のコマンド行オプション。

nis_domain_base_config

説明: NIS ドメインの名前。

nis_broadcast_base_config

説明: NIS サーバーからネームサービスデータをブロードキャストするかどうかを指定します (yes/no)。

デフォルト値: no

nis_broken_server_base_config

説明: SUSE Linux インストールの <nis> セクションにある `nis_broken_server` 変数に対応します (yes/no)。

nis_local_only_base_config

説明: SUSE Linux インストールの <nis> セクションにある `nis_local_only` 変数に対応します (yes/no)。

nis_options_base_config

説明: SUSE Linux インストールの <nis> セクションにある `nis_options` 変数に対応します (yes/no)。

nis_servers_base_config

説明: NIS サーバーのスペース区切りリスト。

nis_start_autofs_base_config

説明: SUSE Linux インストールの <nis> セクションにある `start_autofs` 変数に対応します (yes/no)。

nis_start_nis_config

説明: NIS サービスを開始するかどうかを指定します (yes/no)。

デフォルト値: no

nis_other_1_domain_base_config

説明: 追加の NIS ドメインを指定します。

nis_other_1_broadcast_base_config

説明: 追加の NIS サーバーから名前サービスデータをブロードキャストするかどうかを指定します (yes/no)。

nis_other_1_servers_base_config

説明: 追加の NIS サーバーを指定するコンマ区切りリスト。

ldap_v2_base_config

説明: LDAP v2 認証を使用するかどうかを指定します (true/false)。

デフォルト値: false

ldap_start_base_config

説明: LDAP サービスを開始するかどうかを指定します (true/false)。

デフォルト値: false

ldap_server_base_config

説明: LDAP サーバーの名前。

ldap_basedn_base_config

説明: LDAP サーバーのベースドメイン名。

ldap_tls_base_config

説明: LDAP に Transport Layer Security (TLS) プロトコルを使用するかどうかを指定します (true/false)。

デフォルト値: false

runlevel_base_config

説明: ターゲットホストの実行レベルを指定します。

デフォルト値: 3

security_console_shutdown_base_config

説明: コンソールのシャットダウン動作を指定します。

デフォルト値: reboot

security_cwd_in_root_path_base_config

説明: ルートパスで cwd (作業用ディレクトリの変更) を許可するかどうかを指定します (yes/no)。

デフォルト値: no

security_cwd_in_user_path_base_config

説明: ユーザーパスで cwd (作業用ディレクトリの変更) を許可するかどうかを指定します (yes/no)。

デフォルト値: no

security_displaymanager_remote_access_base_config

説明: ディスプレイマネージャーへのリモートアクセスを可能にします。

デフォルト値: no

security_encryption_base_config

説明: ターゲットホストからの通信の暗号化を可能にします。

デフォルト値: md5

security_fail_delay_base_config

説明: 各ログイン試行後の時間 (秒単位)。

デフォルト値: 1

security_faillog_enab_base_config

説明: /var/log/faillog でのエラーの記録を有効にします

デフォルト値: yes

security_gid_max_base_config

説明: グループ ID の最大数

デフォルト値: 60000

security_gid_min_base_config

説明: グループ ID の最小数

デフォルト値: 101

`security_kdm_shutdown_base_config`
説明: KDM シャットダウン
デフォルト値: all

`security_lastlog_enab_base_config`
説明: lastlog 有効
デフォルト値: yes

`security_obscure_checks_enab_base_config`
説明: あいまい検査を有効にします
デフォルト値: no

`security_pass_max_days_base_config`
説明: パスワードの最大有効日数
デフォルト値: 99999

`security_pass_max_len_base_config`
説明: ユーザーパスワードの最大文字数
デフォルト値: 8

`security_pass_min_days_base_config`
説明: パスワードの最小有効日数
デフォルト値: 0

`security_pass_min_len_base_config`
説明: ユーザーパスワードの最小文字数
デフォルト値: 5

`security_pass_warn_age_base_config`
説明: ユーザーパスワードの期限切れ前にユーザーに警告する残り日数
デフォルト値: 7

`security_passwd_use_cracklib_base_config`
説明: Cracklib ユーティリティーを使用してパスワードを暗号化するかどうかを指定します
デフォルト値: no

`security_permission_security_base_config`
説明: アクセス権のセキュリティーレベルを指定します
デフォルト値: easy

`security_run_updatedb_as_base_config`
説明: updatedb ジョブを実行するユーザーを指定します
デフォルト値: root

`security_uid_max_base_config`

説明: ユーザー ID の最大数

デフォルト値: 60000

`security_uid_min_base_config`

説明: ユーザー ID の最小数

デフォルト値: 500

`users_base_config`

説明: root ユーザーを指定します

デフォルト値: root

`user_root_encrypted_base_config`

説明: インストール中に暗号化された root パスワードを渡すかどうかを指定します

デフォルト値: true

`user_root_password_base_config`

説明: 暗号化された root パスワード

デフォルト値: JgkyZIU7eRxPU

`user_root_fullname_base_config`

説明: root ユーザーのフル名

`user_root_gid_base_config`

説明: root ユーザーのグループ ID

`user_root_grouplist_base_config`

説明: root ユーザーが属する追加グループのスペース区切りリスト

`user_root_home_base_config`

説明: root ユーザーのホームディレクトリ

`user_root_shell_base_config`

説明: root ユーザーのデフォルトシェル

`user_root_uid_base_config`

説明: root ユーザーのユーザー ID

`x11_color_depth_base_config`

説明: ディスプレイデバイスの発色数

`x11_configure_x11_base_config`

説明: X11 ウィンドウシステムを構成するかどうかを指定します (true/ false)

デフォルト値: false

`x11_start_x11_base_config`

説明: X11 ウィンドウシステムを起動するかどうかを指定します (true/ false)

デフォルト値: false

- `x11_display_manager_base_config`
説明: ディスプレイマネージャーを指定します
- `x11_enable_3d_base_config`
説明: X11 ウィンドウシステムの 3D を有効にします
- `x11_window_manager_base_config`
説明: ウィンドウマネージャーソフトウェアを指定します
- `x11_resolution_base_config`
説明: ディスプレイデバイスに使用する解像度を指定します
- `x11_monitor_display_frequency_base_config`
説明: ディスプレイデバイスのモニター周波数
- `x11_monitor_display_bandwidth_base_config`
説明: モニターの帯域幅
- `x11_monitor_display_height_base_config`
説明: モニターの高さ
- `x11_monitor_display_max_hsync_base_config`
説明: モニターの最大水平同期
- `x11_monitor_display_max_vsync_base_config`
説明: モニターの最大垂直同期
- `x11_monitor_display_min_hsync_base_config`
説明: モニターの最小水平同期
- `x11_monitor_display_min_vsync_base_config`
説明: モニターの最小垂直同期
- `x11_monitor_display_width_base_config`
説明: モニターの表示幅
- `x11_monitor_device_base_config`
説明: モニターデバイス
- `x11_monitor_vendor_base_config`
説明: モニターのベンダー
- `bootloader_location_base_config`
説明: ブートローダーの位置

デフォルト値: mbr
- `bootloader_repl_mbr_base_config`
説明: マスターブートレコード (MBR) の内容を消去するかどうかを指定します

デフォルト値: true
- `bootloader_initrd_modules_base_config`
説明: initrd ブートローダーモジュールを指定します

`bootloader_kernel_parameters_base_config`

説明: ブートローダーに対するカーネルパラメータを指定します

`bootloader_loader_type_base_config`

説明: ブートローダーの種類を指定します

デフォルト値: `grub`

`bootloader_loader_device_base_config`

説明: ブートローダーデバイスを指定します

`general_keyboard_base_config`

説明: キーボードマップを指定します

デフォルト値: `english-us`

`general_mouse_id_base_config`

説明: マウス ID を指定します

デフォルト値: `none`

`general_mouse_device_base_config`

説明: マウスデバイスを指定します

`general_clock_hwclock_base_config`

説明: ハードウェアクロック (UTC または `localtime`) を指定します

デフォルト値: `localtime`

`general_timezone_base_config`

説明: タイムゾーンを指定します

デフォルト値: `US/Pacific`

`general_language_base_config`

説明: システムにインストールする言語を指定します

デフォルト値: `en_US`

`general_mode_confirm_base_config`

説明: 確認モード (`true/false`)

デフォルト値: `false`

`general_mode_forceboot_base_config`

説明: 強制的なブート (`true/false`)

デフォルト値: `true`

`general_mode_interactive_boot_base_config`

説明: 対話形式でのブート (`true/false`)

デフォルト値: `false`

`general_mode_reboot_base_config`

説明: リブート (true/ false)

デフォルト値: true

`dhclient_additional_options_base_config`

説明: 追加の DHCP クライアントオプションを指定します

`dhclient_client_id_base_config`

説明: DHCP クライアント ID

`dhclient_hostname_option_base_config`

説明: DHCP ホスト名のオプション

`drive_list_base_config`

説明: ターゲットホストのハードディスクドライブのパイプ区切りリスト。この変数値の構文は次のとおりです。

name!init!use!device

次に例を示します。

```
sda!true!all!/dev/sda|sdb!true!linux!/dev/sdb
```

前述の例は、次の変数と値ペアのセットを持つ 2 つのハードディスクドライブを特定します。

```
base_config_drive_list="sda sdb"
```

```
base_config_drive_sda_init=true
```

```
base_config_drive_sda_use=all
```

```
base_config_drive_sda_device=/dev/sda
```

```
base_config_drive_sdb_init=true
```

```
base_config_drive_sdb_use=linux
```

```
base_config_drive_sdb_device=/dev/sdb
```

デフォルト値: sda!true!all!/dev/sda

`partition_1_drive_base_config`

説明: パーティションのドライブ名。この変数の値は、`drive_list_base_config` で宣言したドライブのいずれかである必要があります。

`partition` 変数の値は 1 から始まり、途中省略することなく順に 1 ずつ増えていく必要があります。

デフォルト値: sda

`partition_1_id_base_config`

説明: パーティションのパーティション ID

例:

`partition_1_nr_base_config`

説明:パーティションのパーティション番号

例:

`partition_1_mount_base_config`

説明:パーティションのマウントポイント

デフォルト値:/boot

`partition_1_filesystem_base_config`

説明:パーティションのファイルシステムの種類

デフォルト値:ext3

`partition_1_format_base_config`

説明:パーティションをフォーマットするかどうかを指定します (true/false)

デフォルト値:true

`partition_1_size_base_config`

説明:パーティションのパーティションサイズ (M バイト数、auto、または max)

デフォルト値:100M

`partition_1_raid_device_base_config`

説明:パーティションの RAID デバイス

例:

`partition_1_primary_base_config`

説明:パーティションを強制的にプライマリパーティションにする (true/false)

デフォルト値:true

`partition_2_drive_base_config`

説明:パーティションのドライブ名

デフォルト値:sda

`partition_2_id_base_config`

説明:パーティションのパーティション ID

例:

`partition_2_nr_base_config`

説明:パーティションのパーティション番号

例:

`partition_2_mount_base_config`

説明:パーティションのマウントポイント

デフォルト値:/

partition_2_filesystem_base_config

説明:パーティションのファイルシステムの種類

デフォルト値:ext3

partition_2_format_base_config

説明:パーティションをフォーマットするかどうかを指定します (true/false)

デフォルト値:true

partition_2_size_base_config

説明:パーティションのパーティションサイズ(Mバイト数、auto、または max)

デフォルト値:10GB

partition_2_raid_device_base_config

説明:パーティションの RAID デバイス

例:

partition_2_primary_base_config

説明:パーティションを強制的にプライマリパーティションにする (true/false)

デフォルト値:true

partition_3_drive_base_config

説明:パーティションのドライブ名

デフォルト値:sda

partition_3_id_base_config

説明:パーティションのパーティションID

例:

partition_3_nr_base_config

説明:パーティションのパーティション番号

例:

partition_3_mount_base_config

説明:パーティションのマウントポイント

デフォルト値:swap

partition_3_filesystem_base_config

説明:パーティションのファイルシステムの種類

デフォルト値:swap

partition_3_format_base_config

説明:パーティションをフォーマットするかどうかを指定します (true/false)

デフォルト値:true

`partition_3_size_base_config`

説明:パーティションのパーティションサイズ(Mバイト数、auto、またはmax)

デフォルト値:4096M

`partition_3_raid_device_base_config`

説明:パーティションのRAIDデバイス

例:

`partition_3_primary_base_config`

説明:パーティションを強制的にプライマリパーティションにする(true/false)

デフォルト値:true

`partition_4_drive_base_config`

説明:パーティションのドライブ名

`partition_4_id_base_config`

説明:パーティションのパーティションID

`partition_4_nr_base_config`

説明:パーティションのパーティション番号

`partition_4_mount_base_config`

説明:パーティションのマウントポイント

`partition_4_filesystem_base_config`

説明:パーティションのファイルシステムの種類

デフォルト値:ext3

`partition_4_format_base_config`

説明:パーティションをフォーマットするかどうかを指定します(true/false)

デフォルト値:true

`partition_4_size_base_config`

説明:パーティションのパーティションサイズ(Mバイト数、auto、またはmax)

`partition_4_raid_device_base_config`

説明:パーティションのRAIDデバイス

`partition_4_primary_base_config`

説明:パーティションを強制的にプライマリパーティションにする(true/false)

デフォルト値:true

`partition_5_drive_base_config`

説明:パーティションのドライブ名

デフォルト値:sda

`partition_5_id_base_config`
説明: パーティションのパーティション ID

例:

`partition_5_nr_base_config`
説明: パーティションのパーティション番号

`partition_5_mount_base_config`
説明: パーティションのマウントポイント

`partition_5_filesystem_base_config`
説明: パーティションのファイルシステムの種類

`partition_5_format_base_config`
説明: パーティションをフォーマットするかどうかを指定します (true/false)

`partition_5_size_base_config`
説明: パーティションのパーティションサイズ (M バイト数、auto、または max)

`partition_5_raid_device_base_config`
説明: パーティションの RAID デバイス

`partition_51_primary_base_config`
説明: パーティションを強制的にプライマリパーティションにする (true/false)

`partition_6_drive_base_config`
説明: パーティションのドライブ名

`partition_6_id_base_config`
説明: パーティションのパーティション ID

`partition_6_nr_base_config`
説明: パーティションのパーティション番号

`partition_6_mount_base_config`
説明: パーティションのマウントポイント

`partition_6_filesystem_base_config`
説明: パーティションのファイルシステムの種類

`partition_6_format_base_config`
説明: パーティションをフォーマットするかどうかを指定します (true/false)

`partition_6_size_base_config`
説明: パーティションのパーティションサイズ (M バイト数、auto、または max)

`partition_6_raid_device_base_config`
説明: パーティションの RAID デバイス

`partition_6_primary_base_config`
説明: パーティションを強制的にプライマリパーティションにする (true/false)

`partition_7_drive_base_config`
説明: パーティションのドライブ名

デフォルト値:sda

partition_7_id_base_config

説明:パーティションのパーティションID

partition_7_nr_base_config

説明:パーティションのパーティション番号

partition_7_mount_base_config

説明:パーティションのマウントポイント

partition_7_filesystem_base_config

説明:パーティションのファイルシステムの種類

partition_7_format_base_config

説明:パーティションをフォーマットするかどうかを指定します (true/false)

partition_7_size_base_config

説明:パーティションのパーティションサイズ(Mバイト数、auto、またはmax)

partition_7_raid_device_base_config

説明:パーティションのRAIDデバイス

partition_7_primary_base_config

説明:パーティションを強制的にプライマリパーティションにする (true/false)

partition_8_drive_base_config

説明:パーティションのドライブ名

partition_8_id_base_config

説明:パーティションのパーティションID

partition_8_nr_base_config

説明:パーティションのパーティション番号

partition_8_mount_base_config

説明:パーティションのマウントポイント

partition_8_filesystem_base_config

説明:パーティションのファイルシステムの種類

partition_8_format_base_config

説明:パーティションをフォーマットするかどうかを指定します (true/false)

partition_8_size_base_config

説明:パーティションのパーティションサイズ(Mバイト数、auto、またはmax)

partition_8_raid_device_base_config

説明:パーティションのRAIDデバイス

partition_8_primary_base_config

説明:パーティションを強制的にプライマリパーティションにする (true/false)

raid_1_device_base_config

説明:RAID ボリュームのデバイスを指定します。

例: /dev/sda1

`raid_1_chunk_size_base_config`

説明: RAID ストリップのサイズを指定します (バイト単位)。

`raid_1_filesystem_base_config`

説明: RAID ボリュームに使用するファイルシステム

`raid_1_format_base_config`

説明: RAID ボリュームのフォーマットを指定します。

`raid_1_parity_algorithm_base_config`

説明: RAID 5 で使用するパリティアルゴリズム。値は、`left-asymmetric`、`right-asymmetric`、`left-symmetric`、または `right-symmetric` のいずれかです。

`raid_1_partition_id_base_config`

説明:

デフォルト値:

`raid_1_persistent_superblock_base_config`

説明: 永続的スーパーブロックを使用するかどうかを指定します (`true/false`)。永続的スーパーブロックとは、各 RAID デバイスの最後に割り当てられる小さなディスク領域で、この領域を確保することにより、カーネルは、ディスクが別の SCSI コントローラに移動された場合でも、安全に RAID デバイスを検出できます。

`raid_1_mount_base_config`

説明: RAID デバイスのマウントポイントを指定します。

`raid_1_format_base_config`

説明: 作成する RAID ボリュームのタイプを指定します (0、1、または 5)

`raid_2_device_base_config`

説明: RAID ボリュームのデバイスを指定します。

例: /dev/sda2

`raid_2_chunk_size_base_config`

説明: RAID ストリップのサイズを指定します (バイト単位)。

`raid_2_filesystem_base_config`

説明: RAID ボリュームに使用するファイルシステム

`raid_2_format_base_config`

説明: RAID ボリュームのフォーマットを指定します。

`raid_2_parity_algorithm_base_config`

説明: RAID 5 で使用するパリティアルゴリズム。値は、`left-asymmetric`、`right-asymmetric`、`left-symmetric`、または `right-symmetric` のいずれかです。

`raid_2_partition_id_base_config`

説明:

デフォルト値:

raid_2_persistent_superblock_base_config

説明: 永続的スーパーブロックを使用するかどうかを指定します (true/false)。永続的スーパーブロックとは、各 RAID デバイスの最後に割り当てられる小さなディスク領域で、この領域を確保することにより、カーネルは、ディスクが別の SCSI コントローラに移動された場合でも、安全に RAID デバイスを検出できます。

raid_2_mount_base_config

説明: RAID デバイスのマウントポイントを指定します。

raid_2_format_base_config

説明: 作成する RAID ボリュームのタイプを指定します (0、1、または5)

software_base_base_config

説明: インストールするソフトウェアベース (Minimal、default、Minimal+X11、Full-Installation.sel) を指定します

デフォルト値: Minimal

software_addons_base_config

説明: インストールするアドオンソフトウェアを指定します

software_kernel_base_config

説明: インストールするカーネル

software_packages_base_config

説明: インストールする追加パッケージのスペース区切りリスト

software_remove_base_config

説明: 削除するパッケージのスペース区切りリスト

bootproto_base_config

説明: ターゲットホストのブートプロトコル。

デフォルト値: dhcp

network_interface_base_config

説明: プロビジョニング中に使用するネットワークインタフェース。

デフォルト値: :[targetableComponent:linux_network_interface]

ip_base_config

説明: ターゲットホストの IP アドレス。

デフォルト値: :[targetableComponent:ethernet_ip_address]

netmask_base_config

説明: プロビジョニング中に使用するネットマスク。

デフォルト値: :[targetableComponent:ethernet_netmask]

gateway_base_config

説明: プロビジョニング中に使用するゲートウェイ。

デフォルト値: :[targetableComponent:default_router]

nameservers_base_config

説明: プロビジョニング中に使用するネームサーバーのスペース区切りリスト。

startmode_base_config

説明: 起動モード

例:

broadcast_base_config

説明: ブロードキャスト

例:

networkifs_base_config

説明: 構成する追加ネットワークインタフェースを指定します。この変数は、ターゲットホスト変数 `networkifs_redhat_base_config` から算出されます

例: `eth1! --ip 1.2.3.4 --netmask 255.255.255.0 --bootproto static`

dns_dhcp_hostname_base_config

説明: DHCP からホスト名を取得するかどうかを指定します (true/false)

デフォルト値: false

dns_dhcp_resolv_base_config

説明: DNS 解決ファイルを変更するかどうかの指定

デフォルト値: false

hostname_base_config

説明: ターゲットホストのホスト名。

デフォルト値: `:[targetableComponent:installPath]`

domain_base_config

説明: DNS ドメイン名

例: `west.example.com`

ip_forward_base_config

説明: IP 転送を有効にするかどうかを指定します (true/false)

例: false

fw_allow_traceroute_base_config

説明: traceroute を許可します

fw_autoprotect_services_base_config

説明: Autoprotect サービス

fw_ipsec_trust_base_config

説明: IPSEC を信頼

fw_log_accept_all_base_config

説明: すべて受け付け

`fw_log_accept_crit_base_config`

説明: `critical` を受け付け

`fw_log_drop_all_base_config`

説明: すべてドロップ

`fw_log_drop_crit_base_config`

説明: `critical` をドロップ

`fw_masquerade_base_config`

説明: マスカレード

`fw_protect_from_internal_base_config`

説明: 内部ネットワークから保護

`fw_route_base_config`

説明: ファイアウォールルート

`fw_start_firewall_base_config`

説明: ファイアウォールを起動するかどうかを指定します

`fw_dev_dmz_base_config`

説明: DMZ のインタフェースを指定します

`fw_dev_ext_base_config`

説明: 外部インタフェースを指定します

`fw_dev_int_base_config`

説明: 内部インタフェースを指定します

`fw_masq_nets_base_config`

説明: マスカレードネットワークを指定します

`fw_services_dmz_ip_base_config`

説明: DMZ の IP アドレス

`fw_services_dmz_tcp_base_config`

説明: DMZ TCP サービスのリスト。名前リスト、ポート番号、または最小および最大範囲を指定します。

`fw_services_dmz_udp_base_config`

説明: DMZ UDP サービスのリスト。名前リスト、ポート番号、または最小および最大範囲を指定します。

`fw_services_ext_ip_base_config`

説明: 外部 IP アドレス

`fw_services_ext_tcp_base_config`

説明: 外部 TCP サービスのリスト。名前リスト、ポート番号、または最小および最大範囲を指定します。

`fw_services_ext_udp_base_config`

説明: 外部 UDP サービスのリスト。名前リスト、ポート番号、または最小および最大範囲を指定します。

- `fw_services_int_ip_base_config`
説明: 内部 IP アドレス
- `fw_services_int_tcp_base_config`
説明: 内部 TCP サービスのリスト。名前リスト、ポート番号、または最小および最大範囲を指定します。
- `fw_services_int_udp_base_config`
説明: 内部 UDP サービスのリスト。名前リスト、ポート番号、または最小および最大範囲を指定します。
- `script_pre_1_base_config`
説明: JET サーバー上のプリインストールスクリプトの位置を指定します。
- `script_pre_1_interpreter_base_config`
説明: プリインストールスクリプトのインタプリタを指定します。
- `script_post_1_base_config`
説明: JET サーバー上のポストインストールスクリプトの位置を指定します。
- `script_post_1_interpreter_base_config`
説明: ポストインストールスクリプトのインタプリタを指定します。
- `script_chroot_1_base_config`
説明: 代替ルートディレクトリの jail 内でポストインストールスクリプトを実行するかどうかを指定します。
- `script_chroot_1_interpreter_base_config`
説明: ポストインストールスクリプトに使用するインタプリタ。
- `script_chroot_1_chrooted_base_config`
説明: 別のルートディレクトリでスクリプトを実行するかどうかを指定します。
- `partitionadd_base_config`
説明: パーティション 9 から始まるパーティションをターゲットホストに追加します。この変数値の構文は次のとおりです。

n!drive!id!nr!mount!filesystem!format!size!raid_device!primary

ここでの各要素の意味は次のとおりです。

n には、パーティション番号を指定します。

disk には、パーティションのデバイスを指定します。

id には、パーティションの ID を指定します。

nr には、パーティション番号を指定します。

mount には、パーティションのマウントポイントを指定します。

filesystem には、パーティションのファイルシステムの種類を指定します。

format は、パーティションをフォーマットするかどうかを指定します。

size には、パーティションのサイズ (M バイト単位) を指定します。

maxsize には、パーティションの最大サイズを指定します。

grow は、パーティションの拡張を許可するかどうかを指定します。

primary は、このパーティションを強制的にプライマリパーティションにするかどうかを指定します。

変数値には、すべての追加パーティションの情報を含めます。たとえば、9 番目と 10 番目のパーティションを作成する場合は、この変数に、次の例のような値を設定します。

```
9!sda!!!/export9!ext3!true!10GB!!!
```

この値によって、次の変数セットが作成されます。

```
base_config_partition_9_drive=sda
base_config_partition_9_id=
base_config_partition_9_nr=
base_config_partition_9_mount=/export9
base_config_partition_9_filesystem=ext3
base_config_partition_9_format=true
base_config_partition_9_size=10GB
base_config_partition_9_raid_device=
base_config_partition_9_primary=
```

`raidadd_base_config`

説明: デバイス 3 から始まる RAID デバイスをターゲットホストに追加します。この変数値の構文は次のとおりです。

```
n!device!chunk_size!filesystem!format!parity_algorithm!partition_id!persistent_superblock!mount!raid_type
```

ここでの各要素の意味は次のとおりです。

n には、RAID ボリューム番号を指定します。

device には、RAID ボリュームのデバイスを指定します。

chunk_size には、ボリュームの RAID レベルを指定します。

filesystem には、RAID ボリュームに使用するファイルシステムを指定します。

format は、RAID ボリュームとして使用するディスクをフォーマットするかどうかを指定します。

parity_algorithm には、RAID ボリュームの整合性検査に使用するアルゴリズムを指定します。

partition_id には、RAID ボリュームのパーティション ID を指定します。

persistent_superblock は、RAID デバイスにスーパーブロックを作成するかどうかを指定します。

mount には、RAID ボリュームのマウントポイントを指定します。

raid_type には、RAID ボリュームの種類を指定します。

変数値には、すべての追加パーティションの情報を含めます。*raidadd_base_config* 変数の基本構文は、*partitionadd_base_config* 変数の構文と同じです。

scriptpreadd_base_config

説明: 番号 2 から始まる追加のプリインストールスクリプトを OS プロビジョニング ファイルに追加します。この変数値の構文は次のとおりです。

n!pre!pre_interpreter

ここでの各要素の意味は次のとおりです。

n には、スクリプトのインデックス番号を指定します。インデックス番号 2 から始めます。

pre には、プリインストールスクリプトの名前を指定します。

pre_interpreter には、プリインストールスクリプトに使用するインタプリタを指定します。

変数値には、使用するすべての追加プリインストールスクリプトの情報を含めます。*scriptpreadd_base_config* 変数の基本構文は、*partitionadd_base_config* 変数の構文と同じです。

scriptpostadd_base_config

説明: 番号 2 から始まる追加のポストインストールスクリプトを OS プロビジョニング ファイルに追加します。この変数値の構文は次のとおりです。

n!post!post_interpreter

ここでの各要素の意味は次のとおりです。

n には、スクリプトのインデックス番号を指定します。インデックス番号 2 から始めます。

post には、ポストインストールスクリプトの名前を指定します。

post_interpreter には、ポストインストールスクリプトに使用するインタプリタを指定します。

変数値には、使用するすべての追加ポストインストールスクリプトの情報を含めます。*scriptpreadd_base_config* 変数の基本構文は、*partitionadd_base_config* 変数の構文と同じです。

scriptchrootadd_base_config

説明: 番号 2 から始まる追加の chroot スクリプトを OS プロビジョニングファイルに追加します。この変数値の構文は次のとおりです。

n!chroot!interpreter!chrooted

ここでの各要素の意味は次のとおりです。

n には、スクリプトのインデックス番号を指定します。インデックス番号 2 から始めます。

chroot には、chroot スクリプトの名前を指定します。

interpreter には、chroot スクリプトに使用するインタプリタを指定します。

chrooted は、別のルートディレクトリでスクリプトを実行するかどうかを指定します。

変数値には、使用するすべての追加ポストインストールスクリプトの情報を含めません。scriptpreadd_base_config 変数の基本構文は、partitionadd_base_config 変数の構文と同じです。

追加ネットワークインタフェースの構成

networkifs_base_config 変数は、network_interface_base 変数を使用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、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 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 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 192.168.2.0
```

SUSE Linux 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: 5.2

OS プロビジョニングプラグイン 3.0 によるプロビジョニングに N1 SPS 5.1 リリースを使用する場合は、この変数の値を 5.1 に設定します。

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

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指定します。

注- プロビジョニング後、ターゲットでユーザーアカウントを使用できない場合は、この変数値を y に設定して、プロビジョニング後のレポートでターゲットホスト上の N1 SPS リモートエージェントを起動するようにします。

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: 6GR9g0i1vRVFw 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

setuid_spsra

説明: リモートエージェントのホスト名または IP アドレス。

例: solaris10-target

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_parms]

一般的な SUSE Linux OS プロビジョニングプラグイン変数

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 NI 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]

`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 (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_boot_server_name`

説明: SUSE Linux 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: SUSE Linux 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/SUSE/SLES-9-x86-64/installroot

`osp_profile_boot_ip_address`

説明: SUSE Linux ブートサーバーの IP アドレス。

例: 199.199.8.6

Microsoft Windows Profile のコンポーネント変数

Microsoft Windows Profile コンポーネントには、Windows OS の構成および Windows リモートエージェント用の変数が含まれています。

Windows OS コンポーネントの変数

このリストで説明されている変数によって、Windows OS 用のすべてのインストールおよび構成パラメータが提供されます。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。Windows 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。

`installPath`

説明: Windows RIS イメージサーバーの名前。

例: `risserver-windows`

`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)。

注 - ターゲットホストに Windows OS をプロビジョニングするには、この変数の値が True である必要があります。

`description`

説明: コンポーネントの説明。

例: `Windows 2003 Profile`

`version`

説明: このプロファイルでインストールする Windows バージョン (2000、2003)。

デフォルト: `2003`

`architecture`

説明: このプロファイルでインストールするシステムのプロセッサアーキテクチャー (x86、x64)。

デフォルト: `x86`

`min_disk_size`

説明: 配備するこのプロファイルの最小ディスクサイズ。

デフォルト: `10GB`

`boot_server_name`

説明: Windows ブートサーバーの名前。

例: `biwindows1`

`boot_server_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。

例: `192.168.5.0`

`boot_server_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: `192.168.5.0`

`boot_srv_active_directory_domain`

説明: このサブネットの Windows Active Directory ドメイン。

例: `winprov.n1lab.sun.com`

`windows_product_directory`

説明: ブートサーバー上の Windows 製品のディレクトリ名。

例: `D:\RemoteInstall\Setup\English\Images\Win2003`

`sif_file`

説明: Software Information File (SIF) の位置。

例: `D:\RemoteInstall\Setup\English\Images\Win2003\i386\templates\ristndrd.sif`

`active_directory_userid`

説明: Active Directory ユーザー ID。

例: `nlospadmin`

`boot_server_access_protocol`

説明: ブートサーバーへのアクセスに使用するプロトコル (RSH か SSH)。

注-カスタム Windows プロファイルでは、RSH プロトコルはサポートされません。カスタム Windows プロファイルでは、通信プロトコルとして SSH を使用する必要があります。

例:ssh

boot_server_access_userid

説明:ブートサーバーへのアクセスに使用するユーザー ID。

例:n1ospssh

boot_server_access_password

説明:ブートサーバーへのアクセスに使用するパスワード。この変数の値として、暗号化されたパスワードを入力します。パスワードを暗号化する方法の詳細については、169 ページの「パスワードの暗号化」を参照してください。

例:Clkjgfw==

temporary_sif_directory

説明:RemInst share 内の一時 Software Information File (sif) の位置。

例:D:\RemoteInstall\N1opsif

ciw_directory

説明:CIW および OS Chooser ファイルの位置。

例:D:\RemoteInstall\OSChooser\English

isp_scripts_directory

説明:OS プロビジョニングプラグインスクリプトの位置。

例:C:\N1ospRisUtil\N1ospScripts

computer_name

説明:ターゲットホストのホスト名。

例:mssystem

user_full_name

説明:ターゲットホストのユーザーのフルネーム。

例:Default user

organization_name

説明:ターゲットホストが所属する組織の名前。

例:Default organization

product_key

説明:インストールする Windows OS の製品キー。製品キーは OS のソフトウェア ディストリビューションに記載されており、xxxxx-xxxxx-xxxxx-xxxxx-xxxxx の形式です。

repartition

説明: ターゲットホストの最初のドライブのすべてのパーティションを、NTFS ファイルシステムで再フォーマットするかどうかを指定します。

デフォルト: Yes

administrator_password

説明: ターゲットホストの管理者のテキスト形式のパスワードをクリアします。

デフォルト: default123

timezone

説明: ターゲットホストのタイムゾーン。

デフォルト値: %TIMEZONE%

profile_language

説明: ターゲットホストにインストールする言語。詳細は、[Microsoft Web サイト](#)にある「Supported Unattended RIS Variables」を参照してください。

例: 0409

profile_language_group

説明: システムにインストールする言語グループ。

例: 1

ClientAccessLicense_automode_setting

説明: Windows OS ライセンスが接続クライアント数または同時使用ユーザー数のどちらのライセンス方式であるかを指定します。

例: PerSeat

ClientAccessLicense_autousers_setting

説明: サーバー用に購入されたクライアントライセンス数を指定します。
ClientAccessLicense_automode_setting に PerServer を設定した場合は、この変数の値を設定する必要があります。

例: 100

dns_server_ip_address

説明: このサブネット上の DNS サーバーの IP アドレス。

例: 192.168.5.167

domain_name

説明: ターゲットホストのドメイン名

注 - この値を指定した場合は、workgroup_name 変数の値を指定しないでください。

例: n1lab.west.example.com

domain_administrator_name

説明: ドメイン管理者のユーザーアカウントの名前。

例: domadmin

domain_administrator_password

説明: ドメイン管理者のユーザーアカウントのパスワード。

例: domadmin567

workgroup_name

説明: ターゲットホストが所属しているワークグループの名前。

注- この値を指定した場合は、`domain_name`、`domain_administrator_name`、および `domain_administrator_password` 変数の値を指定しないでください。

例: n1labs

install_terminal_server

説明: ターゲットホストに端末サーバーをインストールするかどうかを指定します。端末サーバーをインストールする場合は On、端末サーバーをインストールしない場合は Off に設定します。

例: On

terminal_server_allow_connections

説明: 端末サーバー経由でのこのターゲットホストへの遠隔接続を許可するかどうかを指定します。

例: 1

terminal_server_licensing_mode

説明: 端末サービスによるクライアントアクセスライセンスの管理方法を指定します。有効な値は PerDevice か PerUser です。

例: PerDevice

terminal_server_permissions_setting

説明: 端末サーバーユーザーのアクセス権を指定します。この変数に 0 を設定すると、Windows 2000、Windows XP、および Windows 2003 のユーザーと同じアクセス権が端末サーバーユーザーに割り当てられます。この変数を 1 に設定すると、端末サーバー 4.0 のユーザーと同じアクセス権が割り当てられます。

例: 0

pnp_drivers_path

説明: distribution share 下の Plug & Play ドライバの位置。複数のパスを含めるには、セミコロン (;) で区切ります。

例: Drivers\Nic;Drivers\Lsi

`mass_storage_drivers_entry1`

説明: Setup Loader で使用する OEM の SCSI ドライバを指定します。

例: LSI Logic PCI SCSI/FC MPI Driver (Server 2003 32-bit) = OEM

`mass_storage_drivers_entryn`

説明: Setup Loader で使用する追加の OEM または市販の SCSI ドライバを指定します。 *n* には、2～5 の範囲の番号を指定します。最大 5 まで、ドライバを 1 つ追加するたびに *n* の値を 1 大きくします (例: `mass_storage_drivers_entry2`)。

`mass_storage_drivers_entry6`

説明: Setup Loader で使用する市販の SCSI ドライバを指定します。

注 - 市販のドライバは、`mass_storage_drivers_entry_` 設定の最後に指定します。

例: IDE CD-ROM (ATAPI 1.2)/PCI IDE Controller = RETAIL

`oem_boot_filen`

説明: `OEM\Textmode` フォルダに存在する、外部記憶装置の SCSI ドライバファイルを指定します。番号 1 から始めて 10 まで、ドライバファイルを 1 つ追加するたびに *n* の値を 1 ずつ大きくします (例: `oem_boot_file1`)。

例: `lsi_sas.sys`

`gui_run_once_cmdn`

説明: GUI モードのセットアップの完了後に実行する個別のコマンドを指定します。番号 1 から始めて 10 まで、コマンドを 1 つ追加するたびに *n* の値を 1 ずつ大きくします (例: `gui_run_once_cmd1`)。

例: `dcpromo /answer:%SYSTEMDRIVE%\N1OSP\dc_answerfile`

`cmd5` まで

`help_page_url`

説明: HTML 形式のヘルプの URL を指定します。

例: `http://n1labs.west.example.com/help.html`

`home_page_url`

説明: ブラウザのホームページの URL を指定します。

例: `http://www.example.com`

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は範囲に関しては、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` やその他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な存在です。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ければ、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

- 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。

高度な構成プロパティ

この付録では、OS プロビジョニング処理の高度な構成プロパティを設定する方法について説明します。この付録では、次の項目について説明します。

- 299 ページの「プラグイン提供の DHCP の無効化」
- 300 ページの「フォアグラウンドでの配備ジョブの実行」
- 300 ページの「新しいターゲットホストの検査」

プラグイン提供の DHCP の無効化

デフォルトでは、OS プロビジョニングプラグインによって ISC DHCP サーバーの作成と起動を行い、ネットワーク上でターゲットホストをブートおよびインストールできるようにします。ただし、独自の DHCP サーバーを使用してネットワーク上の DHCP 要求に応答するようにする場合は、プロビジョニングソフトウェアの DHCP サーバーを無効にできます。

注-プロビジョニングソフトウェアによって提供される ISC DHCP サーバーを無効にする場合は、ネットワーク上で DHCP 要求に応答できるように DHCP サーバーを正しく構成して、ネットワークブート要求に応答する必要があります。

プロビジョニングソフトウェアが提供する ISC DHCP サーバーを無効にするには、`nlosp-dhcp.properties` ファイルにある `com.sun.n1.isp.dhcp.DHCPManager.serverBinding` プロパティを次のように設定します。

```
com.sun.n1.isp.dhcp.DHCPManager.serverBinding=com.sun.n1.isp.dhcp.NoDHCP
```

`nlosp-dhcp.properties` ファイルは、OS プロビジョニングサーバーの `/opt/SUNWnlosp/etc/nlosp-dhcp.properties` にあります。

フォアグラウンドでの配備ジョブの実行

デフォルトでは、プロビジョニングプランはバックグラウンドで実行されます。ただし、次の作業を行うことによって、フォアグラウンドで配備を実行し、その状況を簡単に確認できるようにすることもできます。

1. N1 SPS ソフトウェアで物理ロックを無効にします。



注意-N1 SPS ソフトウェアでは、ホストの物理ロックの無効化はサポートされていません。

config.properties ファイル内の pe.lockPhysicalHost プロパティを次のように設定します。

```
pe.lockPhysicalHost=false
```

マスターサーバー上の config.properties ファイルの位置は、そのサーバーで動作している OS によって異なります。このファイルの、各 OS での位置は次のとおりです。

- **Solaris OS** –
/opt/SUNWn1sps/N1_Service_Provisioning_System_5.2/server/config/config.properties
- **Linux OS** –
/opt/sun/N1_Service_Provisioning_System_5.2/server/config/config.properties
- **Windows OS** – c:\Program Files\N1 Service Provisioning System
5.2\server\config\config.properties

2. マスターサーバーを再起動します。
3. OS プロビジョニングサーバーを作成します。

OS プロビジョニングサーバーの作成では、run_provisioning_foreground 変数を true に設定します。

新しいターゲットホストの検査

デフォルトでは、OS プロビジョニングサーバーは、ターゲットホストに OS を配備する前にターゲットホストを検査します。ターゲットの検査を無効にするには、OS プロビジョニングサーバーにある /opt/SUNWn1osp/etc/n1osp-service.properties ファイルを編集し、次のエントリの少なくとも1つをファイルに追加します。

- com.sun.n1.isp.osservice.validate.Target.skipValidation=true
ターゲットホスト検査を省略します。このプロパティのデフォルト値は false です。
- com.sun.n1.isp.osservice.validate.Target.skipNMValidation=true
ターゲットホストのすべてのサービスコントローラまたはネットワーク管理コントローラに対するネットワーク管理検査を省略します。このプロパティのデフォルト値は false です。

- `com.sun.n1.isp.osservice.validate.Target.skipNMValidation.hostname=true`

特定のターゲットインスタンス (*hostname*) のすべてのサービスコントローラまたはネットワーク管理コントローラに対するネットワーク管理検査を省略します。このプロパティのデフォルト値は `false` です。
- `com.sun.n1.isp.osservice.validate.Target.skipNM-ICMP-Validation=true`

ネットワーク管理コントローラに接続されているネットワーク管理 IP または 端末サーバー IP の ICMP エコー検査を省略します。このプロパティのデフォルト値は `false` です。
- `com.sun.n1.isp.osservice.validate.Target.skipNM-ICMP-Validation.hostname=true`

特定のターゲットインスタンス (*hostname*) について、ネットワーク管理コントローラに接続されているネットワーク管理 IP または 端末サーバー IP の ICMP エコー検査を省略します。このプロパティのデフォルト値は `false` です。
- `com.sun.n1.isp.osservice.validate.Target.NMValidation.NMtype=ip,TS`

カスタムタイプのネットワーク管理 (*NMtype*) に対するネットワーク管理の検査方法を指定できます。有効な値は `ip`、`ts`、`ip,ts`、または `none` です。ここで、`ip` は IP 検査、`ts` は端末サーバー検査を意味します。

注 - 内蔵タイプのネットワーク管理 (`alom`、`lom`、`rsc`、`ilo` など) のデフォルトの検査方法はオーバーライドできません。

索引

A

- adv_params_spsra 変数, 234, 253, 280
- alom_access_password 変数, ALOM ターゲット用, 193
- alom_access_userid 変数, ALOM ターゲット用, 193
- alom_ip_address 変数, ALOM ターゲット用, 193
- ALOM ターゲットの変数
 - alom_access_password, 193
 - alom_access_userid, 193
 - alom_ip_address, 193
 - architecture, 191
 - default_router, 191
 - description, 191
 - disk size, 191
 - ethernet_ip_address, 191
 - ethernet_mac_address, 191
 - ethernet_netmask, 191
 - installPath, 191
 - ipmp_networkkifs_base_config, 192
 - kernel_arch, 191
 - networkkifs_base_config, 192
 - osp_control_service, 192
 - sps_ra_adv_parms, 193
 - sps_ra_conn, 192
 - sps_ra_host, 192
 - sps_ra_parent, 192
 - sps_ra_parent_type, 192
 - sps_ra_port, 192
 - sps_ra_ssl_auth, 192
 - sysidcfg_default_route, 192
 - sysidcfg_network_interface, 192
 - target_host, 191
 - terminal_server_hal_jar, 193
 - terminal_server_ip_address, 193
 - terminal_server_password, 193
 - terminal_server_port, 193
 - terminal_server_type, 193
 - terminal_server_userid, 193
 - terminal_server_version, 193
- append_kernel_cmdline_base_config 変数, 248, 258
- architecture 変数
 - ALOM ターゲット用, 191
 - HP ILO ターゲット用, 203
 - IPMI ターゲット用, 188
 - LOM ターゲット用, 197
 - Mid-Range ターゲット用, 201
 - RSC ターゲット用, 194
 - そのほかの IPMI ターゲット用, 206
 - そのほかの遠隔管理ターゲット用, 210
 - 汎用ターゲット用, 216
- auth_cache_base_config 変数, 239
- auth_hesiod_enabled_base_config 変数, 240
- auth_hesiod_left_side_base_config 変数, 240
- auth_hesiod_right_side_base_config 変数, 240
- auth_kerberos5_adminserver_base_config 変数, 240
- auth_kerberos5_kdcs_base_config 変数, 239
- auth_kerberos5_realm_base_config 変数, 240
- auth_ldap_basedn_base_config 変数, 239
- auth_ldap_enable_auth_base_config 変数, 239
- auth_ldap_enable_base_config 変数, 239
- auth_ldap_server_base_config 変数, 239
- auth_ldap_tls_base_config 変数, 239
- auth_md5_base_config 変数, 239
- auth_nis_domain_base_config 変数, 239
- auth_nis_enable_base_config 変数, 239
- auth_nis_server_base_config 変数, 239
- auth_shadow_base_config 変数, 239
- auth_smb_servers_base_config 変数, 240

auth_smb_workgroup_base_config 変数, 240

B

base_config_base_config の変数,

security_kdm_shutdown_base_config, 260

base_config の変数

append_kernel_cmdline_base_config, 248,258

auth_cache_base_config, 239

auth_hesiod_enabled_base_config, 240

auth_hesiod_left_side_base_config, 240

auth_hesiod_right_side_base_config, 240

auth_kerberos5_adminserver_base_config, 240

auth_kerberos5_kdcs_base_config, 239

auth_kerberos5_realm_base_config, 240

auth_ldap_basedn_base_config, 239

auth_ldap_enable_auth_base_config, 239

auth_ldap_enable_base_config, 239

auth_ldap_server_base_config, 239

auth_ldap_tls_base_config, 239

auth_md5_base_config, 239

auth_nis_domain_base_config, 239

auth_nis_enable_base_config, 239

auth_nis_server_base_config, 239

auth_shadow_base_config, 239

auth_smb_servers_base_config, 240

auth_smb_workgroup_base_config, 240

bootloader_append_base_config, 241

bootloader_grub_password_base_config, 242

bootloader_grub_password_encrypted_base_config, 241

bootloader_grub_redirect_fix_base_config, 248

bootloader_initrd_modules_base_config, 263

bootloader_kernel_parameters_base_config, 263

bootloader_lilo_lba32_base_config, 241

bootloader_lilo_linear_base_config, 241

bootloader_loader_device_base_config, 264

bootloader_loader_type_base_config, 264

bootloader_location_base_config, 241,263

bootloader_name_base_config, 241

bootloader_repl_mbr_base_config, 263

bootproto_base_config, 242,272

broadcast_base_config, 273

ClientOSName_base_config, 237,257

clusters_base_config, 248

dedicated_dump_device_base_config, 227

dhclient_additional_options_base_config, 265

dhclient_client_id_base_config, 265

dhclient_hostname_option_base_config, 265

dns_dhcp_hostname_base_config, 273

dns_dhcp_resolv_base_config, 273

domain_base_config, 273

drive_list_base_config, 265

enable_services_base_config, 248

firewall_base_config, 241

firewall_incoming_base_config, 241

firewall_trusted_base_config, 241

fw_allow_traceroute_base_config, 273

fw_autoprotect_services_base_config, 273

fw_dev_dmz_base_config, 274

fw_dev_ext_base_config, 274

fw_dev_int_base_config, 274

fw_ipsec_trust_base_config, 273

fw_log_accept_all_base_config, 273

fw_log_accept_crit_base_config, 273

fw_log_drop_all_base_config, 274

fw_log_drop_crit_base_config, 274

fw_masq_nets_base_config, 274

fw_masquerade_base_config, 274

fw_protect_from_internal_base_config, 274

fw_route_base_config, 274

fw_services_dmz_ip_base_config, 274

fw_services_dmz_tcp_base_config, 274

fw_services_dmz_udp_base_config, 274

fw_services_ext_ip_base_config, 274

fw_services_ext_tcp_base_config, 274

fw_services_ext_udp_base_config, 274

fw_services_int_ip_base_config, 274

fw_services_int_tcp_base_config, 275

fw_services_int_udp_base_config, 275

fw_start_firewall_base_config, 274

gateway_base_config, 242,272

general_clock_hwclock_base_config, 264

general_keyboard_base_config, 264

general_language_base_config, 264

general_mode_confirm_base_config, 264

general_mode_forceboot_base_config, 264

general_mode_interactive_boot_base_config, 264

general_mode_reboot_base_config, 264

general_mouse_device_base_config, 264

general_mouse_id_base_config, 264

general_timezone_base_config, 264

grub_append_base_config, 220
hostname_base_config, 242,273
ip_base_config, 242,272
ip_forward_base_config, 273
keyboard_base_conf, 238
kickstart_media_base_config, 238
lang_additional_base_conf, 238
lang_base_config, 238
ldap_basedn_base_config, 259
ldap_server_base_config, 259
ldap_start_base_config, 259
ldap_tls_base_config, 259
ldap_v2_base_config, 259
mouse_base_config, 238
nameservers_base_config, 243,272
netmask_base_config, 242,272
network_interface_base_config, 242,272
nis_broadcast_base_config, 258
nis_broken_server_base_config, 258
nis_domain_base_config, 258
nis_local_only_base_config, 259
nis_options_base_config, 259
nis_other_1_broadcast_base_config, 259
nis_other_1_domain_base_config, 259
nis_other_1_servers_base_config, 259
nis_servers_base_config, 259
nis_start_autofs_base_config, 259
nis_start_nis_base_config, 259
no_autoreboot_base_config, 249
packages_add_base_config, 248
packages_ignoredeps_base_config, 248
packages_ignoremissing_base_config, 248
packages_remove_base_config, 248
packages_resolvedeps_base_config, 248
partition_1_disk_base_config, 243
partition_1_drive_base_config, 265
partition_1_filesystem_base_config, 266
partition_1_format_base_config, 266
partition_1_fstype_base_config, 243
partition_1_grow_base_config, 243
partition_1_id_base_config, 265
partition_1_maxsize_base_config, 243
partition_1_mount_base_config, 266
partition_1_mountpoint_base_config, 243
partition_1_nr_base_config, 265
partition_1_primary_base_config, 243,266
partition_1_raid_device_base_config, 266
partition_1_size_base_config, 243,266
partition_2_disk_base_config, 244
partition_2_drive_base_config, 266
partition_2_filesystem_base_config, 266
partition_2_format_base_config, 267
partition_2_fstype_base_config, 244
partition_2_grow_base_config, 244
partition_2_id_base_config, 266
partition_2_maxsize_base_config, 244
partition_2_mount_base_config, 266
partition_2_mountpoint_base_config, 244
partition_2_nr_base_config, 266
partition_2_primary_base_config, 244,267
partition_2_raid_device_base_config, 267
partition_2_size_base_config, 244,267
partition_3_disk_base_config, 244
partition_3_drive_base_config, 267
partition_3_filesystem_base_config, 267
partition_3_format_base_config, 267
partition_3_fstype_base_config, 244
partition_3_grow_base_config, 245
partition_3_id_base_config, 267
partition_3_maxsize_base_config, 244
partition_3_mount_base_config, 267
partition_3_mountpoint_base_config, 244
partition_3_nr_base_config, 267
partition_3_primary_base_config, 245,268
partition_3_raid_device_base_config, 268
partition_3_size_base_config, 244,267
partition_4_disk_base_config, 245
partition_4_drive_base_config, 268
partition_4_filesystem_base_config, 268
partition_4_format_base_config, 268
partition_4_fstype_base_config, 245
partition_4_grow_base_config, 245
partition_4_id_base_config, 268
partition_4_maxsize_base_config, 245
partition_4_mount_base_config, 268
partition_4_mountpoint_base_config, 245
partition_4_nr_base_config, 268
partition_4_primary_base_config, 245,268
partition_4_raid_device_base_config, 268
partition_4_size_base_config, 245,268
partition_5_disk_base_config, 245
partition_5_drive_base_config, 268

partition_5_filesystem_base_config, 269
partition_5_format_base_config, 269
partition_5_fstype_base_config, 245
partition_5_grow_base_config, 245
partition_5_id_base_config, 268
partition_5_maxsize_base_config, 245
partition_5_mount_base_config, 269
partition_5_mountpoint_base_config, 245
partition_5_nr_base_config, 269
partition_5_primary_base_config, 245,269
partition_5_raid_device_base_config, 269
partition_5_size_base_config, 245,269
partition_6_disk_base_config, 246
partition_6_drive_base_config, 269
partition_6_filesystem_base_config, 269
partition_6_format_base_config, 269
partition_6_fstype_base_config, 246
partition_6_grow_base_config, 246
partition_6_id_base_config, 269
partition_6_maxsize_base_config, 246
partition_6_mount_base_config, 269
partition_6_mountpoint_base_config, 246
partition_6_nr_base_config, 269
partition_6_primary_base_config, 246,269
partition_6_raid_device_base_config, 269
partition_6_size_base_config, 246,269
partition_7_disk_base_config, 246
partition_7_drive_base_config, 269
partition_7_filesystem_base_config, 270
partition_7_format_base_config, 270
partition_7_fstype_base_config, 246
partition_7_grow_base_config, 246
partition_7_id_base_config, 270
partition_7_maxsize_base_config, 246
partition_7_mount_base_config, 270
partition_7_mountpoint_base_config, 246
partition_7_nr_base_config, 270
partition_7_primary_base_config, 246,270
partition_7_raid_device_base_config, 270
partition_7_size_base_config, 246,270
partition_8_disk_base_config, 246
partition_8_drive_base_config, 270
partition_8_filesystem_base_config, 270
partition_8_format_base_config, 270
partition_8_fstype_base_config, 246
partition_8_grow_base_config, 247
partition_8_id_base_config, 270
partition_8_maxsize_base_config, 247
partition_8_mount_base_config, 270
partition_8_mountpoint_base_config, 246
partition_8_nr_base_config, 270
partition_8_primary_base_config, 247,270
partition_8_raid_device_base_config, 270
partition_8_size_base_config, 246,270
partition_clear_base_config, 243
partition_zerombr_base_config, 243
partitionadd_base_config, 249,275
postscript_chroot_base_config, 248
profile_additional_disks_base_config, 224
pxelinux_console_base_config, 238,258
raid_1_chunk_size_base_config, 271
raid_1_device_base_config, 247,270
raid_1_filesystem_base_config, 271
raid_1_format_base_config, 271
raid_1_level_base_config, 247
raid_1_meta_devices_base_config, 247
raid_1_mount_base_config, 271
raid_1_mountpoint_base_config, 247
raid_1_parity_algorithm_base_config, 271
raid_1_partition_id_base_config, 271
raid_2_chunk_size_base_config, 271
raid_2_device_base_config, 247,271
raid_2_filesystem_base_config, 271
raid_2_format_base_config, 271,272
raid_2_level_base_config, 247
raid_2_meta_devices_base_config, 248
raid_2_mount_base_config, 272
raid_2_mountpoint_base_config, 247
raid_2_parity_algorithm_base_config, 271
raid_2_partition_id_base_config, 271
raidadd_base_config, 250,276
root_password_base_config, 240
root_password_encrypted_base_config, 240
runlevel_base_config, 260
script_1_post_interpreter_base_config, 249
script_1_post_nochroot_base_config, 249
script_1_post_pre_base_config, 249
script_1_pre_base_config, 249
script_1_pre_interpreter_base_config, 249
script_chroot_1_chrooted_base_config, 275
script_chroot_1_base_config, 275
script_chroot_1_interpreter_base_config, 275

- script_post_1_post_interpreter_base_config, 275
 - script_post_1_pre_base_config, 275
 - script_pre_1_base_config, 275
 - script_pre_1_interpreter_base_config, 275
 - scriptchrootadd_base_config, 277
 - scriptpostadd_base_config, 251,277
 - scriptpreadd_base_config, 250,277
 - security_console_shutdown_base_config, 260
 - security_cwd_in_root_path_base_config, 260
 - security_cwd_in_user_path_base_config, 260
 - security_displaymanager_remote_access_base_config, 260
 - security_encryption_base_config, 260
 - security_fail_delay_base_config, 260
 - security_faillog_enab_base_config, 260
 - security_gid_max_base_config, 260
 - security_gid_min_base_config, 260
 - security_lastlog_enab_base_config, 261
 - security_obscure_checks_enab_base_config, 261
 - security_pass_max_days_base_config, 261
 - security_pass_max_len_base_config, 261
 - security_pass_min_days_base_config, 261
 - security_pass_min_len_base_config, 261
 - security_pass_warn_age_base_config, 261
 - security_passwd_use_cracklib_base_config, 261
 - security_permission_security_base_config, 261
 - security_run_updatedb_as_base_config, 261
 - security_uid_max_base_config, 261
 - security_uid_min_base_config, 262
 - software_addons_base_config, 272
 - software_bass_base_config, 272
 - software_kernel_base_config, 272
 - software_packages_base_config, 272
 - software_remove_base_config, 272
 - startmode_base_config, 273
 - suse_media_base_config, 258
 - sysidcfg_default_route_base_config, 221
 - sysidcfg_network_interface_base_config, 220
 - sysidcfg_protocol_ipv6_base_config, 221
 - sysidcfg_root_password_base_config, 220
 - sysidcfg_security_policy_base_config, 221
 - sysidcfg_system_locale_base_config, 220
 - sysidcfg_x86_kdmfile_base_config, 222
 - timezone_base_config, 240
 - timezone_utc_base_config, 240
 - user_root_encrypted_base_config, 262
 - user_root_fullname_base_config, 262
 - user_root_gid_base_config, 262
 - user_root_grouplist_base_config, 262
 - user_root_home_base_config, 262
 - user_root_password_base_config, 262
 - user_root_shell_base_config, 262
 - user_root_uid_base_config, 262
 - users_base_config, 262
 - x11_color_depth_base_config, 262
 - x11_configure_x11_base_config, 262
 - x11_display_manager_base_config, 262
 - x11_enable_3d_base_config, 263
 - x11_monitor_device_base_config, 263
 - x11_monitor_display_bandwidth_base_config, 263
 - x11_monitor_display_frequency_base_config, 263
 - x11_monitor_display_height_base_config, 263
 - x11_monitor_display_max_hsync_base_config, 263
 - x11_monitor_display_max_vsync_base_config, 263
 - x11_monitor_display_min_hsync_base_config, 263
 - x11_monitor_display_min_vsync_base_config, 263
 - x11_monitor_display_width_base_config, 263
 - x11_monitor_vendor_base_config, 263
 - x11_resolution_base_config, 263
 - x11_start_x11_base_config, 262
 - x11_window_manager_base_config, 263
 - x86_confflags_base_config, 222
 - x86_disable_kdmconfig_base_config, 222
 - xconfig_card_base_config, 242
 - xconfig_depth_base_config, 242
 - xconfig_desktop_base_config, 242
 - xconfig_monitor_base_config, 242
 - xconfig_resolution_base_config, 242
 - xconfig_videoram_base_config, 242
- base_config 変数
- client_allocation_base_config, 220,238,258
 - ClientArch_base_config, 219,237,257
 - ClientEther_base_config, 219,237,257
 - ClientOS_base_config, 219,238,258
 - defaultrouter_base_config, 225
 - disable_sysid_probe_base_config, 227
 - dns_disableforbuild_base_config, 225
 - dns_domain_base_config, 225
 - dns_nameservers_base_config, 225
 - dns_searchpath_base_config, 225
 - dumpadm_minfree_base_config, 226
 - enable_altbreak_base_config, 227
 - enable_rootftp_base_config, 226

- enable_rootlogin_base_config, 226
 - enable_savecore_base_config, 226
 - ipmp_networkkifs_base_config, 226
 - networkkifs_base_config, 226, 243, 273
 - nfs_mounts_base_config, 225
 - nfsv4_domain_base_config, 227
 - noautosshutdn_base_config, 226
 - nodename_base_config, 225
 - notrouter_base_config, 225
 - ntp_servers_base_config, 225
 - patchdir_base_config, 227, 238, 258
 - poweroff_afterbuild_base_config, 227
 - productdir_base_config, 227, 238, 258
 - products_base_config, 220, 238, 258
 - profile_add_clusters_base_config, 225
 - profile_add_geos_base_config, 224
 - profile_add_locales_base_config, 224
 - profile_add_packages_base_config, 225
 - profile_base_config, 222
 - profile_cluster_base_config, 222
 - profile_del_clusters_base_config, 225
 - profile_del_geos_base_config, 224
 - profile_del_locales_base_config, 224
 - profile_del_packages_base_config, 225
 - profile_dontuse_base_config, 223
 - profile_root_base_config, 223
 - profile_s3_mtpt_base_config, 223
 - profile_s3_size_base_config, 223
 - profile_s4_mtpt_base_config, 223
 - profile_s4_size_base_config, 223
 - profile_s5_mtpt_base_config, 223
 - profile_s5_size_base_config, 223
 - profile_s6_mtpt_base_config, 223
 - profile_s6_size_base_config, 224
 - profile_s7_mtpt_base_config, 224
 - profile_s7_size_base_config, 224
 - profile_swap_base_config, 223
 - profile_usedisk_base_config, 223
 - shutup_sendmail_base_config, 227
 - sysidcfg_ip_address_base_config, 220
 - sysidcfg_nameservice_base_config, 220
 - sysidcfg_netmask_base_config, 220
 - sysidcfg_terminal_base_config, 221
 - sysidcfg_timeserver_base_config, 221
 - sysidcfg_timezone_base_config, 221
 - ufs_logging_filesys_base_config, 224
 - update_terminal_base_config, 226
 - x86_console_base_config, 221
 - x86_disable_acpi_base_config, 221
 - x86_nowin_base_config, 221
 - x86_safetoreboot_base_config, 221
 - basedir_spsra 変数, 232, 252, 279
 - boot_service_true_false_1 変数, そのほかの遠隔管理ターゲット用, 213
 - boot_service_true_false_2 変数, そのほかの遠隔管理ターゲット用, 214
 - boot_service_true_false 変数
 - LOM ターゲット用, 200
 - そのほかの IPMI ターゲット用, 209
 - boot_service 変数, IPMI ターゲット用, 190
 - bootloader_append_base_config 変数, 241
 - bootloader_grub_password_base_config 変数, 242
 - bootloader_grub_password_encrypted_base_config variable, 241
 - bootloader_grub_redirect_fix_base_config variable, 248
 - bootloader_initrd_modules_base_config 変数, 263
 - bootloader_kernel_parameters_base_config 変数, 263
 - bootloader_lilo_lba32_base_config 変数, 241
 - bootloader_lilo_linear_base_config 変数, 241
 - bootloader_loader_device_base_config 変数, 264
 - bootloader_loader_type_base_config 変数, 264
 - bootloader_location_base_config 変数, 241, 263
 - bootloader_name_base_config 変数, 241
 - bootloader_repl_mbr_base_config 変数, 263
 - bootproto_base_config 変数, 242, 272
 - broadcast_base_config 変数, 273
- ## C
- client_allocation_base_config 変数, 220, 238, 258
 - ClientArch_base_config 変数, 219, 237, 257
 - ClientEther_base_config 変数, 219, 237, 257
 - ClientOS_base_config 変数, 219, 238, 258
 - ClientOSName_base_config 変数, 237, 257
 - clusters_base_config 変数, 248
 - conn_spsra 変数, 233, 253, 280
 - console_service_true_false_1 variable, そのほかの遠隔管理ターゲット用, 213

console_service_true_false_2 変数, そのほかの遠隔管理ターゲット用, 214

console_service_true_false 変数

LOM ターゲット用, 200

そのほかの IPMI ターゲット用, 209

console_service 変数, IPMI ターゲット用, 190

D

dedicated_dump_device_base_config 変数, 227

default_router 変数

ALOM ターゲット用, 191

HP ILO ターゲット用, 203

IPMI ターゲット用, 188

LOM ターゲット用, 197

Mid-Range ターゲット用, 201

RSC ターゲット用, 194

そのほかの IPMI ターゲット用, 206

そのほかの遠隔管理ターゲット用, 210

汎用ターゲット用, 216

defaultrouter_base_config 変数, 225

description 変数

ALOM ターゲット用, 191

HP ILO ターゲット用, 203

IPMI ターゲット用, 188

LOM ターゲット用, 197

Mid-Range ターゲット用, 200

RSC ターゲット用, 194

そのほかの IPMI ターゲット用, 206

そのほかの遠隔管理ターゲット用, 210

汎用ターゲット用, 215

device_type 変数, IPMI ターゲット用, 190

dhclient_additional_options_base_config 変数, 265

dhclient_client_id_base_config 変数, 265

dhclient_hostname_option_base_config 変数, 265

disable_sysid_probe_base_config 変数, 227

disk_size 変数

ALOM ターゲット用, 191

HP ILO ターゲット用, 203

IPMI ターゲット用, 188

LOM ターゲット用, 197

Mid-Range ターゲット用, 201

RSC ターゲット用, 194

そのほかの IPMI ターゲット用, 206

そのほかの遠隔管理ターゲット用, 210

汎用ターゲット用, 216

disk_type 変数

LOM ターゲット用, 199

そのほかの IPMI ターゲット用, 208

そのほかの遠隔管理ターゲット用, 212

dns_dhcp_hostname_base_config 変数, 273

dns_dhcp_resolv_base_config 変数, 273

dns_disableforbuild_base_config 変数, 225

dns_domain_base_config 変数, 225

dns_nameservers_base_config 変数, 225

dns_searchpath_base_config 変数, 225

domain_base_config 変数, 273

domain_name 変数, Mid-Range ターゲット用, 202

domainsc_access_password 変数, Mid-Range ターゲット用, 202

domainsc_access_userid 変数, Mid-Range ターゲット用, 202

drive_list_base_config 変数, 265

dumpadm_minfree_base_config 変数, 226

E

enable_altbreak_base_config 変数, 227

enable_rootftp_base_config 変数, 226

enable_rootlogin_base_config 変数, 226

enable_savecore_base_config 変数, 226

enable_services_base_config 変数, 248

ethernet_ip_address 変数

ALOM ターゲット用, 191

HP ILO ターゲット用, 203

IPMI ターゲット用, 188

LOM ターゲット用, 197

Mid-Range ターゲット用, 200

RSC ターゲット用, 194

そのほかの IPMI ターゲット用, 206

そのほかの遠隔管理ターゲット用, 210

汎用ターゲット用, 216

ethernet_mac_address 変数

ALOM ターゲット用, 191

HP ILO ターゲット用, 203

IPMI ターゲット用, 188

LOM ターゲット用, 197

Mid-Range ターゲット用, 200

RSC ターゲット用, 194

そのほかの IPMI ターゲット用, 206

そのほかの遠隔管理ターゲット用, 210
汎用ターゲット用, 215

ethernet_netmask 変数
ALOM ターゲット用, 191
HP ILO ターゲット用, 203
IPMI ターゲット用, 188
LOM ターゲット用, 197
Mid-Range ターゲット用, 201
RSC ターゲット用, 194
そのほかの IPMI ターゲット用, 206
そのほかの遠隔管理ターゲット用, 210
汎用ターゲット用, 216

F

firewall_base_config 変数, 241
firewall_incoming_base_config 変数, 241
firewall_trusted_base_config 変数, 241
fw_allow_traceroute_base_config 変数, 273
fw_autoprotect_services_base_config 変数, 273
fw_dev_dmz_base_config 変数, 274
fw_dev_ext_base_config 変数, 274
fw_dev_int_base_config 変数, 274
fw_ipsec_trust_base_config 変数, 273
fw_log_accept_all_base_config 変数, 273
fw_log_accept_crit_base_config 変数, 273
fw_log_drop_all_base_config 変数, 274
fw_log_drop_crit_base_config 変数, 274
fw_masq_nets_base_config 変数, 274
fw_masquerade_base_config 変数, 274
fw_protect_from_internal_base_config 変数, 274
fw_route_base_config 変数, 274
fw_services_dmz_ip_base_config 変数, 274
fw_services_dmz_tcp_base_config 変数, 274
fw_services_dmz_udp_base_config 変数, 274
fw_services_ext_ip_base_config 変数, 274
fw_services_ext_tcp_base_config 変数, 274
fw_services_ext_udp_base_config 変数, 274
fw_services_int_ip_base_config 変数, 274
fw_services_int_tcp_base_config 変数, 275
fw_services_int_udp_base_config 変数, 275
fw_start_firewall_base_config 変数, 274

G

gateway_base_config 変数, 242, 272
general_clock_hwclock_base_config 変数, 264
general_keyboard_base_config 変数, 264
general_language_base_config 変数, 264
general_mode_confirm_base_config 変数, 264
general_mode_forceboot_base_config 変数, 264
general_mode_interactive_boot_base_config 変数, 264
general_mode_reboot_base_config 変数, 264
general_mouse_device_base_config 変数, 264
general_mouse_id_base_config 変数, 264
general_timezone_base_config 変数, 264
groupid_spsra 変数, 233, 252, 280
groupname_spsra 変数, 232, 252, 279
grub_append_base_config 変数, 220
guid 変数
HP ILO ターゲット用, 205
IPMI ターゲット用, 190
そのほかの IPMI ターゲット用, 208
そのほかの遠隔管理ターゲット用, 212
guid 変数, 汎用ターゲット用, 218

H

homedir_spsra 変数, 233, 252, 280
host_spsra 変数, 233, 253, 280
hostname_base_config 変数, 242, 273
HP ILO ターゲットの変数
architecture, 203
default_router, 203
description, 203
disk_size, 203
ethernet_ip_address, 203
ethernet_mac_address, 203
ethernet_netmask, 203
guid, 205
ilo_access_password, 205
ilo_access_userid, 205
ilo_ip_address, 205
ilo_version, 205
installPath, 203
ipmp_networkkifs_base_config, 204
kernel_arch, 203
linux_network_interface, 205

networkifs_base_config, 204
networkifs_redhat_base_config, 205
networkifs_suse_base_config, 205
osp_control_service, 204
sps_ra_adv_parms, 205
sps_ra_conn, 204
sps_ra_host, 204
sps_ra_parent, 204
sps_ra_parent_type, 204
sps_ra_port, 204
sps_ra_ssl_auth, 204
sysidcfg_default_route, 204
sysidcfg_network_interface, 204
target_host, 203

I

ilo_access_password 変数, HP ILO ターゲット用, 205
ilo_access_userid 変数, HP ILO ターゲット用, 205
ilo_ip_address 変数, HP ILO ターゲット用, 205
ilo_version 変数, HP ILO ターゲット用, 205
install_ra_from_snapshot_spsra 変数, 232
install_spsra 変数, 232, 251, 279
installPath 変数
 ALOM ターゲット用, 191
 HP ILO ターゲット用, 203
 IPMI ターゲット用, 187
 LOM ターゲット用, 197
 Mid-Range ターゲット用, 200
 RedHat Linux のプロビジョニング用, 254
 RSC ターゲット用, 194
 Solaris OS のプロビジョニング用, 234
 SUSE Linux のプロビジョニング用, 281
 そのほかの IPMI ターゲット用, 206
 そのほかの遠隔管理ターゲット用, 209
 汎用ターゲット用, 215
ip_base_config 変数, 242, 272
ip_forward_base_config 変数, 273
ipmi_access_password 変数
 IPMI ターゲット用, 190
 そのほかの IPMI ターゲット用, 208
ipmi_access_userid 変数
 IPMI ターゲット用, 190
 そのほかの IPMI ターゲット用, 208
ipmi_hal_jar 変数, そのほかの IPMI ターゲット用, 208
ipmi_ip_address 変数
 IPMI ターゲット用, 190
 そのほかの IPMI ターゲット用, 208
ipmi_version 変数, そのほかの IPMI ターゲット用, 209
IPMI ターゲットの変数
 architecture, 188
 boot_service, 190
 console_service, 190
 default_router, 188
 description, 188
 device_type, 190
 disk_size, 188
 ethernet_ip_address, 188
 ethernet_mac_address, 188
 ethernet_netmask, 188
 guid, 190
 installPath, 187
 ipmi_access_password, 190
 ipmi_access_userid, 190
 ipmi_ip_address, 190
 ipmp_networkifs_base_config, 189
 kernel_arch, 188
 linux_network_interface, 189
 networkifs_base_config, 188
 networkifs_redhat_base_config, 189
 networkifs_suse_base_config, 190
 osp_control_service, 189
 sps_ra_adv_parms, 189
 sps_ra_conn, 189
 sps_ra_host, 189
 sps_ra_parent, 189
 sps_ra_parent_type, 189
 sps_ra_port, 189
 sps_ra_ssl_auth, 189
 ssh_access_password, 190
 ssh_access_userid, 190
 sysidcfg_default_route, 188
 sysidcfg_network_interface, 188
 target_host, 187
ipmp_networkifs_base_config 変数, 226
 ALOM ターゲット用, 192
 HP ILO ターゲット用, 204
 IPMI ターゲット用, 189

LOM ターゲット用, 198
Mid-Range ターゲット用, 201
RSC ターゲット用, 195
そのほかの IPMI ターゲット用, 207
そのほかの遠隔管理ターゲット用, 211
汎用ターゲット用, 216

J

JAR ファイル, 47-49
Java アーカイブ, 「JAR ファイル」を参照
JET
 既存のユーザー, 61
 モジュール, 59-60
jre_path_spsra 変数, 233, 252, 280
JumpStart, 「JET」を参照
JumpStart Enterprise Toolkit, 「JET」を参照

K

kernel_arch 変数
 ALOM ターゲット用, 191
 HP ILO ターゲット用, 203
 IPMI ターゲット用, 188
 LOM ターゲット用, 197
 Mid-Range ターゲット用, 201
 RSC ターゲット用, 194
 そのほかの IPMI ターゲット用, 206
 そのほかの遠隔管理ターゲット用, 210
 汎用ターゲット用, 216
keyboard_base_conf 変数, 238
kickstart_media_base_config 変数, 238

L

lang_additional_base_conf 変数, 238
lang_base_config 変数, 238
ldap_basedn_base_config 変数, 259
ldap_server_base_config 変数, 259
ldap_start_base_config 変数, 259
ldap_tls_base_config 変数, 259
ldap_v2_base_config 変数, 259

linux_network_interface 変数
 HP ILO ターゲット用, 205
 IPMI ターゲット用, 189
 そのほかの IPMI ターゲット用, 208
 そのほかの遠隔管理ターゲット用, 211
 汎用ターゲット用, 217
lom_access_password 変数, LOM ターゲット用, 199
lom_access_userid 変数, LOM ターゲット用, 199
lom_hal_jar 変数, LOM ターゲット用, 199
lom_version 変数, LOM ターゲット用, 199
LOM ターゲットの変数
 architecture, 197
 boot_service_true_false, 200
 console_service_true_false, 200
 default_router, 197
 description, 197
 disk_size, 197
 disk_type, 199
 ethernet_ip_address, 197
 ethernet_mac_address, 197
 ethernet_netmask, 197
 installPath, 197
 ipmp_networkkifs_base_config, 198
 kernel_arch, 197
 lom_access_password, 199
 lom_access_userid, 199
 lom_hal_jar, 199
 lom_version, 199
 networkkifs_base_config, 198
 osp_control_service, 198
 power_service_true_false, 200
 sps_ra_adv_parms, 199
 sps_ra_conn, 198
 sps_ra_host, 198
 sps_ra_parent, 198
 sps_ra_parent_type, 198
 sps_ra_port, 198
 sps_ra_ssl_auth, 198
 sysidcfg_default_route, 198
 sysidcfg_network_interface, 198
 target_host, 197
 terminal_server_hal_jar, 199
 terminal_server_ip_address, 199
 terminal_server_password, 199
 terminal_server_port, 199
 terminal_server_type, 199

terminal_server_userid, 199
terminal_server_version, 199

M

Mid-Range ターゲットの変数

default_router, 201
ipmp_networkkifs_base_config, 201
networkkifs_base_config, 201
sysidcfg_default_route, 201
sysidcfg_network_interface, 201

Mid-Range ターゲットの変数

architecture, 201
description, 200
disk_size, 201
domain_name, 202
domainsc_access_password, 202
domainsc_access_userid, 202
ethernet_ip_address, 200
ethernet_mac_address, 200
ethernet_netmask, 201
installPath, 200
kernel_arch, 201
osp_control_service, 201
platformsc_access_password, 202
platformsc_access_userid, 202
platformsc_ip_address, 202
sps_ra_adv_parms, 202
sps_ra_conn, 202
sps_ra_host, 202
sps_ra_parent, 202
sps_ra_parent_type, 202
sps_ra_port, 202
sps_ra_ssl_auth, 202
target_host, 200

mouse_base_config 変数, 238

N

nameservers_base_config 変数, 243, 272
netmask_base_config 変数, 242, 272
netmgmt_hal_jar_1 変数, そのほかの遠隔管理ターゲット用, 212

netmgmt_hal_jar_2 変数, そのほかの遠隔管理ターゲット用, 214
netmgmt_ip_address_1 変数, そのほかの遠隔管理ターゲット用, 212
netmgmt_ip_address_2 変数, そのほかの遠隔管理ターゲット用, 214
netmgmt_password_1 変数, そのほかの遠隔管理ターゲット用, 213
netmgmt_password_2 変数, そのほかの遠隔管理ターゲット用, 214
netmgmt_port_1 変数, そのほかの遠隔管理ターゲット用, 212
netmgmt_port_2 変数, そのほかの遠隔管理ターゲット用, 214
netmgmt_protocol_1 変数, そのほかの遠隔管理ターゲット用, 212
netmgmt_protocol_2 変数, そのほかの遠隔管理ターゲット用, 214
netmgmt_type_1 変数, そのほかの遠隔管理ターゲット用, 212
netmgmt_type_2 変数, そのほかの遠隔管理ターゲット用, 214
netmgmt_userid_1 変数, そのほかの遠隔管理ターゲット用, 213
netmgmt_userid_2 変数, そのほかの遠隔管理ターゲット用, 214
netmgmt_version_1 変数, そのほかの遠隔管理ターゲット用, 212
netmgmt_version_2 変数, そのほかの遠隔管理ターゲット用, 214
network_interface_base_config 変数, 242, 272
networkkifs_base_config 変数, 226, 243, 273
ALOM ターゲット用, 192
HP ILO ターゲット用, 204
IPMI ターゲット用, 188
LOM ターゲット用, 198
Mid-Range ターゲット用, 201
RSC ターゲット用, 195
そのほかの IPMI ターゲット用, 207
そのほかの遠隔管理ターゲット用, 210
汎用ターゲット用, 216
networkkifs_redhat_base_config 変数
HP ILO ターゲット用, 205
IPMI ターゲット用, 189
そのほかの IPMI ターゲット用, 208
そのほかの遠隔管理ターゲット用, 212

汎用ターゲット用, 217
networkifs_suse_base_config 変数
 HP ILO ターゲット用, 205
 IPMI ターゲット用, 190
 そのほかの IPMI ターゲット用, 208
 そのほかの遠隔管理ターゲット用, 212
 汎用ターゲット用, 217
nfs_mounts_base_config 変数, 225
nfsv4_domain_base_config 変数, 227
nis_broadcast_base_config 変数, 258
nis_broken_server_base_config 変数, 258
nis_domain_base_config 変数, 258
nis_local_only_base_config 変数, 259
nis_options_base_config 変数, 259
nis_other_1_broadcast_base_config 変数, 259
nis_other_1_domain_base_config 変数, 259
nis_other_1_servers_base_config 変数, 259
nis_servers_base_config 変数, 259
nis_start_autofs_base_config 変数, 259
nis_start_nis_base_config 変数, 259
no_autoreboot_base_config 変数, 249
noautosshutdown_base_config 変数, 226
nodename_base_config 変数, 225
notrouter_base_config 変数, 225
ntp_servers_base_config 変数, 225

O

osp_control_service 変数
 ALOM ターゲット用, 192
 HP ILO ターゲット用, 204
 IPMI ターゲット用, 189
 LOM ターゲット用, 198
 Mid-Range ターゲット用, 201
 RSC ターゲット用, 195
 そのほかの IPMI ターゲット用, 207
 そのほかの遠隔管理ターゲット用, 211
 汎用ターゲット用, 216
osp_jet_server 変数, 234, 254, 281
osp_profile_architecture 変数, 235, 254, 281
osp_profile_boot_directory 変数, 235
osp_profile_boot_file_name 変数, 236
osp_profile_boot_ip_address 変数, 235, 255, 282
osp_profile_boot_server_name 変数, 235, 254, 281
osp_profile_BS_subnet_address 変数, 235, 254, 281

osp_profile_BS_subnet_ip_address 変数, 235, 254, 281
osp_profile_control_service 変数, 234, 254, 281
osp_profile_description 変数, 235, 254, 281
osp_profile_jumpstart_directory 変数, 236
osp_profile_jumpstart_ip_address 変数, 236
osp_profile_min_disk_size 変数, 235, 254, 281
osp_profile_nbp_boot_file_name 変数, 236
osp_profile_product_directory 変数, 235, 254, 282
osp_profile_product_ip_address 変数, 235, 254, 282
osp_profile_sysidcfg_directory 変数, 235
osp_profile_sysidcfg_ip_address 変数, 235
osp_target_host 変数, 254, 281
 Solaris OS のプロビジョニング用, 234
OS プロビジョニングプラグイン, インポート, 49-50

P

packages_add_base_config 変数, 248
packages_ignoredeps_base_config 変数, 248
packages_ignoremissing_base_config 変数, 248
packages_remove_base_config 変数, 248
packages_resolvedeps_base_config 変数, 248
partition_1_disk_base_config 変数, 243
partition_1_drive_base_config 変数, 265
partition_1_filesystem_base_config 変数, 266
partition_1_format_base_config 変数, 266
partition_1_fstype_base_config 変数, 243
partition_1_grow_base_config 変数, 243
partition_1_id_base_config 変数, 265
partition_1_maxsize_base_config 変数, 243
partition_1_mount_base_config 変数, 266
partition_1_mountpoint_base_config 変数, 243
partition_1_nr_base_config 変数, 265
partition_1_primary_base_config 変数, 243, 266
partition_1_raid_device_base_config 変数, 266
partition_1_size_base_config 変数, 243, 266
partition_2_disk_base_config 変数, 244
partition_2_drive_base_config 変数, 266
partition_2_filesystem_base_config 変数, 266
partition_2_format_base_config 変数, 267
partition_2_fstype_base_config 変数, 244
partition_2_grow_base_config 変数, 244
partition_2_id_base_config 変数, 266
partition_2_maxsize_base_config 変数, 244

- partition_2_mount_base_config 变数, 266
- partition_2_mountpoint_base_config 变数, 244
- partition_2_nr_base_config 变数, 266
- partition_2_primary_base_config 变数, 244,267
- partition_2_raid_device_base_config 变数, 267
- partition_2_size_base_config 变数, 244,267
- partition_3_disk_base_config 变数, 244
- partition_3_drive_base_config 变数, 267
- partition_3_filesystem_base_config 变数, 267
- partition_3_format_base_config 变数, 267
- partition_3_fstype_base_config 变数, 244
- partition_3_grow_base_config 变数, 245
- partition_3_id_base_config 变数, 267
- partition_3_maxsize_base_config 变数, 244
- partition_3_mount_base_config 变数, 267
- partition_3_mountpoint_base_config 变数, 244
- partition_3_nr_base_config 变数, 267
- partition_3_primary_base_config 变数, 245,268
- partition_3_raid_device_base_config 变数, 268
- partition_3_size_base_config 变数, 244,267
- partition_4_disk_base_config 变数, 245
- partition_4_drive_base_config 变数, 268
- partition_4_filesystem_base_config 变数, 268
- partition_4_format_base_config 变数, 268
- partition_4_fstype_base_config 变数, 245
- partition_4_grow_base_config 变数, 245
- partition_4_id_base_config 变数, 268
- partition_4_maxsize_base_config 变数, 245
- partition_4_mount_base_config 变数, 268
- partition_4_mountpoint_base_config 变数, 245
- partition_4_nr_base_config 变数, 268
- partition_4_primary_base_config 变数, 245,268
- partition_4_raid_device_base_config 变数, 268
- partition_4_size_base_config 变数, 245,268
- partition_5_disk_base_config 变数, 245
- partition_5_drive_base_config 变数, 268
- partition_5_filesystem_base_config 变数, 269
- partition_5_format_base_config 变数, 269
- partition_5_fstype_base_config 变数, 245
- partition_5_grow_base_config 变数, 245
- partition_5_id_base_config 变数, 268
- partition_5_maxsize_base_config 变数, 245
- partition_5_mount_base_config 变数, 269
- partition_5_mountpoint_base_config 变数, 245
- partition_5_nr_base_config 变数, 269
- partition_5_primary_base_config 变数, 245,269
- partition_5_raid_device_base_config 变数, 269
- partition_5_size_base_config 变数, 245,269
- partition_6_disk_base_config 变数, 246
- partition_6_drive_base_config 变数, 269
- partition_6_filesystem_base_config 变数, 269
- partition_6_format_base_config 变数, 269
- partition_6_fstype_base_config 变数, 246
- partition_6_grow_base_config 变数, 246
- partition_6_id_base_config 变数, 269
- partition_6_maxsize_base_config 变数, 246
- partition_6_mount_base_config 变数, 269
- partition_6_mountpoint_base_config 变数, 246
- partition_6_nr_base_config 变数, 269
- partition_6_primary_base_config 变数, 246,269
- partition_6_raid_device_base_config 变数, 269
- partition_6_size_base_config 变数, 246,269
- partition_7_disk_base_config 变数, 246
- partition_7_drive_base_config 变数, 269
- partition_7_filesystem_base_config 变数, 270
- partition_7_format_base_config 变数, 270
- partition_7_fstype_base_config 变数, 246
- partition_7_grow_base_config 变数, 246
- partition_7_id_base_config 变数, 270
- partition_7_maxsize_base_config 变数, 246
- partition_7_mount_base_config 变数, 270
- partition_7_mountpoint_base_config 变数, 246
- partition_7_nr_base_config 变数, 270
- partition_7_primary_base_config 变数, 246,270
- partition_7_raid_device_base_config 变数, 270
- partition_7_size_base_config 变数, 246,270
- partition_8_disk_base_config 变数, 246
- partition_8_drive_base_config 变数, 270
- partition_8_filesystem_base_config 变数, 270
- partition_8_format_base_config 变数, 270
- partition_8_fstype_base_config 变数, 246
- partition_8_grow_base_config 变数, 247
- partition_8_id_base_config 变数, 270
- partition_8_maxsize_base_config 变数, 247
- partition_8_mount_base_config 变数, 270
- partition_8_mountpoint_base_config 变数, 246
- partition_8_nr_base_config 变数, 270
- partition_8_primary_base_config 变数, 247,270
- partition_8_raid_device_base_config 变数, 270
- partition_8_size_base_config 变数, 246,270
- partition_clear_base_config 变数, 243
- partition_zerombr_base_config 变数, 243

partitionadd_base_config 変数, 249, 275
patchdir_base_config 変数, 227, 238, 258
platformsc_access_password 変数, Mid-Range ター
ゲット用, 202
platformsc_access_userid 変数, Mid-Range ター
ゲット用, 202
platformsc_ip_address 変数, Mid-Range ターゲット
用, 202
port_spsra 変数, 233, 253, 280
postscript_chroot_base_config 変数, 248
power_service_true_false_1 変数, そのほかの遠隔管
理ターゲット用, 212
power_service_true_false_2 変数, そのほかの遠隔管
理ターゲット用, 214
power_service_true_false 変数
LOM ターゲット用, 200
そのほかの IPMI ターゲット用, 209
poweroff_afterbuild_base_config 変数, 227
product_version_spsra 変数, 232, 251, 279
productdir_base_config 変数, 227, 238, 258
products_base_config 変数, 220, 238, 258
profile_add_clusters_base_config 変数, 225
profile_add_geos_base_config 変数, 224
profile_add_locales_base_config 変数, 224
profile_add_packages_base_config 変数, 225
profile_additional_disks_base_config 変数, 224
profile_base_config 変数, 222
profile_cluster_base_config 変数, 222
profile_del_clusters_base_config 変数, 225
profile_del_geos_base_config 変数, 224
profile_del_locales_base_config 変数, 224
profile_del_packages_base_config 変数, 225
profile_dontuse_base_config 変数, 223
profile_root_base_config 変数, 223
profile_s3_mtpt_base_config 変数, 223
profile_s3_size_base_config 変数, 223
profile_s4_mtpt_base_config 変数, 223
profile_s4_size_base_config 変数, 223
profile_s5_mtpt_base_config 変数, 223
profile_s5_size_base_config 変数, 223
profile_s6_mtpt_base_config 変数, 223
profile_s6_size_base_config 変数, 224
profile_s7_mtpt_base_config 変数, 224
profile_s7_size_base_config 変数, 224
profile_swap_base_config 変数, 223
profile_usedisk_base_config 変数, 223

pxelinux_console_base_config 変数, 238, 258

R

raid_1_chunk_size_base_config 変数, 271
raid_1_device_base_config 変数, 247, 270
raid_1_filesystem_base_config 変数, 271
raid_1_format_base_config 変数, 271
raid_1_level_base_config 変数, 247
raid_1_meta_devices_base_config 変数, 247
raid_1_mount_base_config 変数, 271
raid_1_mountpoint_base_config 変数, 247
raid_1_parity_algorithm_base_config 変数, 271
raid_1_partition_id_base_config 変数, 271
raid_1_persistent_superblock_base_config 変
数, 271
raid_1_raid_type_base_config 変数, 271
raid_2_chunk_size_base_config 変数, 271
raid_2_device_base_config 変数, 247, 271
raid_2_filesystem_base_config 変数, 271
raid_2_format_base_config 変数, 271
raid_2_level_base_config 変数, 247
raid_2_meta_devices_base_config 変数, 248
raid_2_mount_base_config 変数, 272
raid_2_mountpoint_base_config 変数, 247
raid_2_parity_algorithm_base_config 変数, 271
raid_2_partition_id_base_config 変数, 271
raid_2_persistent_superblock_base_config 変
数, 271
raid_2_raid_type_base_config 変数, 272
raidadd_base_config 変数, 250, 276
RedHat Linux プロビジョニングの変数
installPath, 254
osp_jet_server, 254
osp_profile_architecture, 254
osp_profile_boot_ip_address, 255
osp_profile_boot_server_name, 254
osp_profile_BS_subnet_address, 254
osp_profile_BS_subnet_ip_address, 254
osp_profile_control_service, 254
osp_profile_description, 254
osp_profile_min_disk_size, 254
osp_profile_product_directory, 254
osp_profile_product_ip_address, 254
osp_target_host, 254

- root_password_base_config 変数, 240
- root_password_encrypted_base_config 変数, 240
- rsc_access_password 変数, RSC ターゲット用, 196
- rsc_access_userid 変数, RSC ターゲット用, 196
- rsc_ip_address 変数, RSC ターゲット用, 196
- RSC ターゲットの変数
 - architecture, 194
 - default_router, 194
 - description, 194
 - disk_size, 194
 - ethernet_ip_address, 194
 - ethernet_mac_address, 194
 - ethernet_netmask, 194
 - installPath, 194
 - ipmp_networkifs_base_config, 195
 - kernel_arch, 194
 - networkifs_base_config, 195
 - osp_control_service, 195
 - rsc_access_password, 196
 - rsc_access_userid, 196
 - rsc_ip_address, 196
 - sps_ra_adv_parms, 196
 - sps_ra_conn, 195
 - sps_ra_host, 195
 - sps_ra_parent, 195
 - sps_ra_parent_type, 195
 - sps_ra_port, 195
 - sps_ra_ssl_auth, 195
 - sysidcfg_default_route, 195
 - sysidcfg_network_interface, 195
 - target_host, 194
 - terminal_server_hal_jar, 196
 - terminal_server_ip_address, 196
 - terminal_server_password, 196
 - terminal_server_port, 196
 - terminal_server_protocol, 193, 196, 199
 - terminal_server_type, 196
 - terminal_server_userid, 196
 - terminal_server_version, 196
- runlevel_base_config 変数, 260
- script_1_post_interpreter_pre_base_config variable, 249
- script_1_post_nochroot_base_config 変数, 249
- script_1_pre_base_config 変数, 249
- script_1_pre_interpreter_base_config 変数, 249
- script_chroot_1_base_config 変数, 275
- script_chroot_1_chrooted_base_config 変数, 275
- script_chroot_1_interpreter_base_config 変数, 275
- script_post_1_base_config 変数, 275
- script_post_1_interpreter_base_config 変数, 275
- script_pre_1_base_config 変数, 275
- script_pre_1_interpreter_base_config 変数, 275
- scriptchrootadd_base_config 変数, 277
- scriptpostadd_base_config 変数, 251, 277
- scriptpreadd_base_config 変数, 250, 277
- security_console_shutdown_base_config 変数, 260
- security_cwd_in_root_path_base_config 変数, 260
- security_cwd_in_user_path_base_config 変数, 260
- security_displaymanager_remote_access_base_config 変数, 260
- security_encryption_base_config 変数, 260
- security_fail_delay_base_config 変数, 260
- security_faillog_enab_base_config 変数, 260
- security_gid_max_base_config 変数, 260
- security_gid_min_base_config 変数, 260
- security_kdm_shutdown_base_config 変数, 260
- security_lastlog_enab_base_config 変数, 261
- security_obscure_checks_enab_base_config 変数, 261
- security_pass_max_days_base_config 変数, 261
- security_pass_max_len_base_config 変数, 261
- security_pass_min_days_base_config 変数, 261
- security_pass_min_len_base_config 変数, 261
- security_pass_warn_age_base_config 変数, 261
- security_passwd_use_cracklib_base_config variable, 261
- security_permission_security_base_config 変数, 261
- security_run_updatedb_as_base_config 変数, 261
- security_uid_max_base_config 変数, 261
- security_uid_min_base_config 変数, 262
- setuid_spsra 変数, 233, 253, 280
- setup_user_acct_spsra 変数, 232, 252, 279
- shutup_sendmail_base_config 変数, 227
- software_addons_base_config 変数, 272
- software_base_base_config 変数, 272

S

script_1_post_base_config 変数, 249

- software_kernel_base_config 変数, 272
- software_packages_base_config 変数, 272
- software_remove_base_config 変数, 272
- Solaris OS プロビジョニングの変数
 - installPath, 234
 - osp_target_host, 234
- Solaris OS プロビジョニング変数
 - osp_jet_server, 234
 - osp_profile_architecture, 235
 - osp_profile_boot_directory, 235
 - osp_profile_boot_file_name, 236
 - osp_profile_boot_ip_address, 235
 - osp_profile_boot_server_name, 235
 - osp_profile_BS_subnet_address, 235
 - osp_profile_BS_subnet_ip_address, 235
 - osp_profile_control_service, 234
 - osp_profile_description, 235
 - osp_profile_jumpstart_directory, 236
 - osp_profile_jumpstart_ip_address, 236
 - osp_profile_min_disk_size, 235
 - osp_profile_nbp_boot_file_name, 236
 - osp_profile_product_directory, 235
 - osp_profile_product_ip_address, 235
 - osp_profile_sysidcfg_directory, 235
 - osp_profile_sysidcfg_ip_address, 235
 - sps_ra_parent_host, 234
 - sps_ra_parent_host_type, 234
- sps_ra_adv_parms 変数
 - ALOM ターゲット用, 193
 - HP ILO ターゲット用, 205
 - IPMI ターゲット用, 189
 - LOM ターゲット用, 199
 - Mid-Range ターゲット用, 202
 - RSC ターゲット用, 196
 - そのほかの IPMI ターゲット用, 208
 - そのほかの遠隔管理ターゲット用, 211
 - 汎用ターゲット用, 217
- sps_ra_conn 変数
 - ALOM ターゲット用, 192
 - HP ILO ターゲット用, 204
 - IPMI ターゲット用, 189
 - LOM ターゲット用, 198
 - Mid-Range ターゲット用, 202
 - RSC ターゲット用, 195
 - そのほかの IPMI ターゲット用, 207
 - そのほかの遠隔管理ターゲット用, 211
- 汎用ターゲット用, 217
- sps_ra_host 変数
 - ALOM ターゲット用, 192
 - HP ILO ターゲット用, 204
 - IPMI ターゲット用, 189
 - LOM ターゲット用, 198
 - Mid-Range ターゲット用, 202
 - RSC ターゲット用, 195
 - そのほかの IPMI ターゲット用, 207
 - そのほかの遠隔管理ターゲット用, 211
 - 汎用ターゲット用, 217
- sps_ra_parent_host_type 変数, 234
- sps_ra_parent_host 変数, 234
- sps_ra_parent_type 変数
 - ALOM ターゲット用, 192
 - HP ILO ターゲット用, 204
 - IPMI ターゲット用, 189
 - LOM ターゲット用, 198
 - Mid-Range ターゲット用, 202
 - RSC ターゲット用, 195
 - そのほかの IPMI ターゲット用, 207
 - そのほかの遠隔管理ターゲット用, 211
 - 汎用ターゲット用, 217
- sps_ra_parent 変数
 - ALOM ターゲット用, 192
 - HP ILO ターゲット用, 204
 - IPMI ターゲット用, 189
 - LOM ターゲット用, 198
 - Mid-Range ターゲット用, 202
 - RSC ターゲット用, 195
 - そのほかの IPMI ターゲット用, 207
 - そのほかの遠隔管理ターゲット用, 211
 - 汎用ターゲット用, 217
- sps_ra_port 変数
 - ALOM ターゲット用, 192
 - HP ILO ターゲット用, 204
 - IPMI ターゲット用, 189
 - LOM ターゲット用, 198
 - Mid-Range ターゲット用, 202
 - RSC ターゲット用, 195
 - そのほかの IPMI ターゲット用, 207
 - そのほかの遠隔管理ターゲット用, 211
 - 汎用ターゲット用, 217
- sps_ra_ssl_auth 変数
 - ALOM ターゲット用, 192
 - HP ILO ターゲット用, 204

- IPMI ターゲット用, 189
 - LOM ターゲット用, 198
 - Mid-Range ターゲット用, 202
 - RSC ターゲット用, 195
 - その他の IPMI ターゲット用, 207
 - その他の遠隔管理ターゲット用, 211
 - 汎用ターゲット用, 217
 - spsra の変数
 - host_spsra, 233, 253, 280
 - spsra 変数
 - adv_params_spsra, 234, 253, 280
 - basedir_spsra, 232, 252, 279
 - conn_spsra, 233, 253, 280
 - groupid_spsra, 233, 252, 280
 - groupname_spsra, 232, 252, 279
 - homedir_spsra, 233, 252, 280
 - install_ra_from_snapshot_spsra, 232
 - install_spsra, 232, 251, 279
 - jre_path_spsra, 233, 252, 280
 - port_spsra, 233, 253, 280
 - product_version_spsra, 232, 251, 279
 - setuid_spsra, 233, 253, 280
 - setup_user_acct_spsra, 232, 252, 279
 - ssl_auth_spsra, 233, 253, 280
 - user_passwd_spsra, 233, 252, 279
 - userid_spsra, 233, 252, 279
 - username_spsra, 232, 252, 279
 - ssh_access_password 変数, IPMI ターゲット用, 190
 - ssh_access_userid 変数, IPMI ターゲット用, 190
 - ssl_auth_spsra 変数, 233, 253, 280
 - startmode_base_config 変数, 273
 - Sun Data Center Reference Architecture, 42
 - suse_media_base_config 変数, 258
 - SUSE Linux プロビジョニングの変数
 - installPath, 281
 - osp_jet_server, 281
 - osp_profile_architecture, 281
 - osp_profile_boot_ip_address, 282
 - osp_profile_boot_server_name, 281
 - osp_profile_BS_subnet_address, 281
 - osp_profile_BS_subnet_ip_address, 281
 - osp_profile_control_service, 281
 - osp_profile_description, 281
 - osp_profile_min_disk_size, 281
 - osp_profile_product_directory, 282
 - osp_profile_product_ip_address, 282
 - osp_target_host, 281
 - sysidcfg_default_route_base_config 変数, 221
 - sysidcfg_default_route 変数
 - ALOM ターゲット用, 192
 - HP ILO ターゲット用, 204
 - IPMI ターゲット用, 188
 - LOM ターゲット用, 198
 - Mid-Range ターゲット用, 201
 - RSC ターゲット用, 195
 - その他の IPMI ターゲット用, 207
 - その他の遠隔管理ターゲット用 target, 210
 - 汎用ターゲット用, 216
 - sysidcfg_ip_address_base_config 変数, 220
 - sysidcfg_nameservice_base_config 変数, 220
 - sysidcfg_netmask_base_config 変数, 220
 - sysidcfg_network_interface_base_config 変数, 220
 - sysidcfg_network_interface 変数
 - ALOM ターゲット用, 192
 - HP ILO ターゲット用, 204
 - IPMI ターゲット用, 188
 - LOM ターゲット用, 198
 - Mid-Range ターゲット用, 201
 - RSC ターゲット用, 195
 - その他の IPMI ターゲット用, 207
 - その他の遠隔管理ターゲット用, 210
 - 汎用ターゲット用, 216
 - sysidcfg_protocol_ipv6_base_config 変数, 221
 - sysidcfg_root_password_base_config 変数, 220
 - sysidcfg_security_policy_base_config 変数, 221
 - sysidcfg_system_locale_base_config 変数, 220
 - sysidcfg_terminal_base_config 変数, 221
 - sysidcfg_timeserver_base_config 変数, 221
 - sysidcfg_timezone_base_config 変数, 221
 - sysidcfg_x86_kdmfile_base_config 変数, 222
- T**
- target_host 変数
 - ALOM ターゲット用, 191
 - HP ILO ターゲット用, 203
 - IPMI ターゲット用, 187
 - LOM ターゲット用, 197
 - Mid-Range ターゲット用, 200
 - RSC ターゲット用, 194
 - その他の IPMI ターゲット用, 206

- そのほかの遠隔管理ターゲット用, 209
汎用ターゲット用, 215
- terminal_server_hal_jar_1 変数, そのほかの遠隔管理ターゲット用, 213
- terminal_server_hal_jar_2 変数, そのほかの遠隔管理ターゲット用, 215
- terminal_server_hal_jar 変数
ALOM ターゲット用, 193
LOM ターゲット用, 199
RSC ターゲット用, 196
- terminal_server_ip_address_1 variable, そのほかの遠隔管理ターゲット用, 213
- terminal_server_ip_address_2 変数, そのほかの遠隔管理ターゲット用, 215
- terminal_server_ip_address 変数
ALOM ターゲット用, 193
LOM ターゲット用, 199
RSC ターゲット用, 196
- terminal_server_password_1 変数, そのほかの遠隔管理ターゲット用, 213
- terminal_server_password_2 変数, そのほかの遠隔管理ターゲット用, 215
- terminal_server_password 変数
ALOM ターゲット用, 193
LOM ターゲット用, 199
RSC ターゲット用, 196
- terminal_server_port_1 変数, そのほかの遠隔管理ターゲット用, 213
- terminal_server_port_2 変数, そのほかの遠隔管理ターゲット用, 215
- terminal_server_port 変数
ALOM ターゲット用, 193
LOM ターゲット用, 199
RSC ターゲット用, 196
- terminal_server_protocol_1 変数, そのほかの遠隔管理ターゲット用, 213
- terminal_server_protocol_2 変数, そのほかの遠隔管理ターゲット用, 215
- terminal_server_protocol 変数
RSC ターゲット用, 193, 196, 199
- terminal_server_type_1 変数, そのほかの遠隔管理ターゲット用, 213
- terminal_server_type_2 変数, そのほかの遠隔管理ターゲット用, 214
- terminal_server_type 変数
ALOM ターゲット用, 193
LOM ターゲット用, 199
RSC ターゲット用, 196
- terminal_server_userid_1 変数, そのほかの遠隔管理ターゲット用, 213
- terminal_server_userid_2 変数, そのほかの遠隔管理ターゲット用, 215
- terminal_server_userid 変数
ALOM ターゲット用, 193
LOM ターゲット用, 199
RSC ターゲット用, 196
- terminal_server_version_1 変数, そのほかの遠隔管理ターゲット用, 213
- terminal_server_version_2 変数, そのほかの遠隔管理ターゲット用, 215
- terminal_server_version 変数
ALOM ターゲット用, 193
LOM ターゲット用, 199
RSC ターゲット用, 196
- timezone_base_config 変数, 240
- timezone_utc_base_config 変数, 240

U

- ufs_logging_filesys_base_config 変数, 224
- update_terminal_base_config 変数, 226
- user_passwd_spsra 変数, 233, 252, 279
- user_root_encrypted_base_config 変数, 262
- user_root_fullname_base_config 変数, 262
- user_root_gid_base_config 変数, 262
- user_root_grouplist_base_config 変数, 262
- user_root_home_base_config 変数, 262
- user_root_password_base_config 変数, 262
- user_root_shell_base_config 変数, 262
- user_root_uid_base_config 変数, 262
- userid_spsra 変数, 233, 252, 279
- username_spsra 変数, 232, 252, 279
- users_base_config 変数, 262

X

- x11_color_depth_base_config 変数, 262
- x11_configure_x11_base_config 変数, 262
- x11_display_manager_base_config 変数, 262
- x11_enable_3d_base_config 変数, 263

- x11_monitor_device_base_config 変数, 263
 - x11_monitor_display_bandwidth_base_config 変数, 263
 - x11_monitor_display_frequency_base_config 変数, 263
 - x11_monitor_display_height_base_config 変数, 263
 - x11_monitor_display_max_hsync_base_config 変数, 263
 - x11_monitor_display_max_vsync_base_config 変数, 263
 - x11_monitor_display_min_hsync_base_config 変数, 263
 - x11_monitor_display_min_vsync_base_config 変数, 263
 - x11_monitor_display_width_base_config 変数, 263
 - x11_monitor_vendor_base_config 変数, 263
 - x11_resolution_base_config 変数, 263
 - x11_start_x11_base_config 変数, 262
 - x11_window_manager_base_config 変数, 263
 - x86_confflags_base_config 変数, 222
 - x86_console_base_config 変数, 221
 - x86_disable_acpi_base_config 変数, 221
 - x86_disable_kdmconfig_base_config 変数, 222
 - x86_nowin_base_config 変数, 221
 - x86_safetoreboot_base_config 変数, 221
 - xconfig_card_base_config 変数, 242
 - xconfig_depth_base_config 変数, 242
 - xconfig_desktop_base_config 変数, 242
 - xconfig_monitor_base_config 変数, 242
 - xconfig_resolution_base_config 変数, 242
 - xconfig_videoram_base_config 変数, 242
- こ
- 構成のベストプラクティス, 42
- そ
- そのほかの IPMI ターゲットの変数
- architecture, 206
 - boot_service_true_false, 209
 - console_service_true_false, 209
 - default_router, 206
 - description, 206
 - disk_size, 206
 - disk_type, 208
 - ethernet_ip_address, 206
 - ethernet_mac_address, 206
 - ethernet_netmask, 206
 - guid, 208
 - installPath, 206
 - ipmi_access_password, 208
 - ipmi_access_userid, 208
 - ipmi_hal_jar, 208
 - ipmi_ip_address, 208
 - ipmi_version, 209
 - ipmp_networkkifs_base_config, 207
 - kernel_arch, 206
 - linux_network_interface, 208
 - networkkifs_base_config, 207
 - networkkifs_redhat_base_config, 208
 - networkkifs_suse_base_config, 208
 - osp_control_service, 207
 - power_service_true_false, 209
 - sps_ra_adv_parms, 208
 - sps_ra_conn, 207
 - sps_ra_host, 207
 - sps_ra_parent, 207
 - sps_ra_parent_type, 207
 - sps_ra_port, 207
 - sps_ra_ssl_auth, 207
 - sysidcfg_default_route, 207
 - sysidcfg_network_interface, 207
 - target_host, 206
- そのほかの遠隔管理ターゲットの変数
- architecture, 210
 - console_service_true_false_1, 213
 - default_router, 210
 - description, 210
 - disk_size, 210
 - ethernet_ip_address, 210
 - ethernet_mac_address, 210
 - ethernet_netmask, 210
 - installPath, 209
 - ipmp_networkkifs_base_config, 211
 - kernel_arch, 210
 - linux_network_interface, 211
 - networkkifs_base_config, 210
 - networkkifs_redhat_base_config, 212
 - networkkifs_suse_base_config, 212

sysidcfg_default_route, 210
sysidcfg_network_interface, 210
target_host, 209
terminal_server_ip_address_1, 213
そのほかの遠隔管理ターゲット変数
boot_service_true_false_1, 213
boot_service_true_false_2, 214
console_service_true_false_2, 214
disk_type, 212
guid, 212
netmgmt_hal_jar_1, 212
netmgmt_hal_jar_2, 214
netmgmt_ip_address_1, 212
netmgmt_ip_address_2, 214
netmgmt_password_1, 213
netmgmt_password_2, 214
netmgmt_port_1, 212
netmgmt_port_2, 214
netmgmt_protocol_1, 212
netmgmt_protocol_2, 214
netmgmt_type_1, 212
netmgmt_type_2, 214
netmgmt_userid_1, 213
netmgmt_userid_2, 214
netmgmt_version_1, 212
netmgmt_version_2, 214
osp_control_service, 211
power_service_true_false_1, 212
power_service_true_false_2, 214
sps_ra_adv_parms, 211
sps_ra_conn, 211
sps_ra_host, 211
sps_ra_parent, 211
sps_ra_parent_type, 211
sps_ra_port, 211
sps_ra_ssl_auth, 211
terminal_server_hal_jar_1, 213
terminal_server_hal_jar_2, 215
terminal_server_ip_address_2, 215
terminal_server_password_1, 213
terminal_server_password_2, 215
terminal_server_port_1, 213
terminal_server_port_2, 215
terminal_server_protocol_1, 213
terminal_server_protocol_2, 215
terminal_server_type_1, 213

terminal_server_type_2, 214
terminal_server_userid_1, 213
terminal_server_userid_2, 215
terminal_server_version_1, 213
terminal_server_version_2, 215

て

データセンターの構成, 42

は

汎用ターゲットの変数

default_router, 216
guid, 218
ipmp_networkkifs_base_config, 216
linux_network_interface, 217
networkkifs_base_config, 216
networkkifs_redhat_base_config, 217
networkkifs_suse_base_config, 217
osp_control_service, 216
sps_ra_adv_parms, 217
sps_ra_conn, 217
sps_ra_host, 217
sps_ra_parent, 217
sps_ra_parent_type, 217
sps_ra_port, 217
sps_ra_ssl_auth, 217
sysidcfg_default_route, 216
sysidcfg_network_interface, 216

汎用ターゲット変数

architecture, 216
description, 215
disk_size, 216
ethernet_ip_address, 216
ethernet_mac_address, 215
ethernet_netmask, 216
installPath, 215
kernel_arch, 216
target_host, 215

ふ

プラグインのインポート, 49-50

プラグインファイル

インポート, 49-50

場所, 47-49

