



Solaris 10 5/08 インストールガイド (カスタム **JumpStart**/ 上級編)



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

Part No: 820-4540-10
2008年4月

Sun Microsystems, Inc. (以下米国 Sun Microsystems 社とします) は、本書に記述されている製品に含まれる技術に関連する知的財産権を所有します。特に、この知的財産権はひとつかそれ以上の米国における特許、あるいは米国およびその他の国において申請中の特許を含んでいることがあります。それらに限定されるものではありません。

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.

この配布には、第三者によって開発された素材を含んでいることがあります。

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ッ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、Sun のロゴマーク、Solaris のロゴマーク、Java Coffee Cup のロゴマーク、docs.sun.com、Java および Solaris は、米国およびその他の国における米国 Sun Microsystems 社の商標、登録商標もしくは、サービスマークです。

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。

Wnn8 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(C) OMRON Co., Ltd. 1995-2000. All Rights Reserved. Copyright(C) OMRON SOFTWARE Co., Ltd. 1995-2007 All Rights Reserved.

「ATOK for Solaris」は、株式会社ジャストシステムの著作物であり、「ATOK for Solaris」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」および「推測変換」は、株式会社ジャストシステムの登録商標です。

「ATOK for Solaris」に添付す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。

「ATOK for Solaris」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)。

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、OPEN LOOK 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。

本書で言及されている製品や含まれている情報は、米国輸出規制法で規制されるものであり、その他の国の輸出入に関する法律の対象となることがあります。核、ミサイル、化学あるいは生物兵器、原子力の海洋輸送手段への使用は、直接および間接を問わず厳しく禁止されています。米国が禁輸の対象としている国や、限定はされませんが、取引禁止顧客や特別指定国民のリストを含む米国輸出排除リストで指定されているものへの輸出および再輸出は厳しく禁止されています。

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: Solaris 10 5/08 Installation Guide: Custom JumpStart and Advanced Installations

Part No: 820-4042-10

Revision A

目次

はじめに	9
パートI カスタム JumpStart の使用	13
1 Solaris インストールの計画についての参照先	15
計画とシステム要件についての参照先	15
2 カスタム JumpStart (概要)	17
カスタム JumpStart とは	17
カスタム JumpStart の使用例	17
JumpStart プログラムが Solaris ソフトウェアをインストールする際の流れ	19
3 カスタム JumpStart インストールの準備 (作業)	23
作業マップ: カスタム JumpStart インストールの準備	23
ネットワーク上のシステム用のプロファイルサーバーの作成	25
▼ JumpStart ディレクトリをサーバー上に作成する方法	26
すべてのシステムがプロファイルサーバーにアクセスできるようにする	28
スタンドアロンシステム用のプロファイルフロッピーディスクの作成	31
▼ SPARC: プロファイルフロッピーディスクを作成する方法	32
▼ x86: GRUB を使用してプロファイルフロッピーディスクを作成する方法	34
rules ファイルの作成	36
rules ファイルの構文	36
▼ rules ファイルを作成する方法	37
rules ファイルの例	38
プロファイルの作成	40
プロファイルの構文	40
▼ プロファイルを作成する方法	41

プロファイルの例	41
プロファイルのテスト	55
▼ プロファイルをテストするために、一時的な Solaris 環境を作成する方法	55
▼ プロファイルをテストする方法	56
プロファイルテストの例	59
rules ファイルの妥当性を検査する	59
▼ rules ファイルの妥当性を検査する方法	60
4 カスタム JumpStart オプション機能の使用 (作業)	63
開始スクリプトの作成	63
開始スクリプトに関する注意事項	64
開始スクリプトによる派生プロファイルの作成	64
終了スクリプトの作成	65
終了スクリプトに関する注意事項	66
▼ 終了スクリプトでファイルを追加する方法	66
終了スクリプトによるパッケージとパッチの追加	67
終了スクリプトによるルート環境のカスタマイズ	69
終了スクリプトによるシステムのルートパスワードの設定	70
終了スクリプトによる非対話式インストール	72
圧縮された構成ファイルの作成	72
▼ 圧縮された構成ファイルを作成する方法	73
圧縮された構成ファイルの例	73
ディスク構成ファイルの作成	74
▼ SPARC: ディスク構成ファイルを作成する方法	74
SPARC: ディスク構成ファイルの例	75
▼ x86: ディスク構成ファイルを作成する方法	76
x86: ディスク構成ファイルの例	77
サイト固有のインストールプログラムの使用	79
5 カスタムルールおよびプローブキーワードの作成 (作業)	81
プローブキーワード	81
custom_probes ファイルの作成	82
custom_probes ファイルの構文	82
custom_probes 内の関数名の構文	83
▼ custom_probes ファイルを作成する方法	83

custom_probes ファイルとキーワードの例	83
custom_probes ファイルの妥当性検査	85
▼ custom_probes ファイルを妥当性検査する方法	85
6 カスタム JumpStart インストールの実行(作業)	87
JumpStart インストールの制限事項	87
SPARC: 作業マップ: カスタム JumpStart インストールのためのシステムの設定	89
SPARC: カスタム JumpStart インストールの実行	90
▼ カスタム JumpStart インストールを使用して Solaris フラッシュアーカイブをインストールする方法	90
▼ SPARC: カスタム JumpStart プログラムを使用してインストールまたはアップグレードする方法	93
SPARC: boot コマンドのコマンドリファレンス	94
x86: 作業マップ: カスタム JumpStart インストールのためのシステムの設定	96
x86: カスタム JumpStart インストールの実行	97
▼ x86: カスタム JumpStart プログラムおよび GRUB を使用してインストールまたはアップグレードを実行する方法	98
x86: GRUB ブートコマンドの編集によるカスタム JumpStart インストールの実行	100
x86: システムのブートのためのコマンドリファレンス	102
7 カスタム JumpStart によるインストール(例)	105
サイトの設定例	105
インストールサーバーの作成	107
x86: マーケティングシステム用のブートサーバーの作成	108
JumpStart ディレクトリの作成	109
JumpStart ディレクトリの共有	109
SPARC: エンジニアリング部のプロファイルの作成	110
x86: マーケティング部のプロファイルの作成	110
rules ファイルの更新	111
rules ファイルの妥当性を検査する	112
SPARC: ネットワーク上でインストールするためのエンジニアリングシステムの設定	112
x86: ネットワーク上でインストールするためのマーケティングシステムの設定	113
SPARC: エンジニアリングシステムのブートと Solaris ソフトウェアのインストール	114

x86: マーケティングシステムのブートと Solaris ソフトウェアのインストール	114
8 カスタム JumpStart (リファレンス)	115
ルールキーワードと値の説明	115
プロファイルキーワードと値	119
プロファイルキーワードのクイックリファレンス	120
プロファイルキーワードの説明と例	121
カスタム JumpStart の環境変数	164
プローブキーワードと値	166
パート II 付録	169
A 問題発生時の解決方法 (作業)	171
ネットワークインストールの設定に関する問題	171
システムのブートに関する問題	172
メディアからのブート時のエラーメッセージ	172
メディアからのブート時の一般的な問題	173
ネットワークからのブート時のエラーメッセージ	174
ネットワークからのブート時の一般的な問題	178
Solaris OS の初期インストール	178
▼ x86: IDE ディスクの不良ブロックの検査	179
Solaris OS のアップグレード	181
アップグレード時のエラーメッセージ	181
アップグレード時の一般的な問題	183
▼ 問題発生後にアップグレードを継続する方法	184
x86: GRUB を使用する場合の Solaris Live Upgrade に関する問題	184
▼ Veritas VxVm の実行中に Solaris Live Upgrade を使用してアップグレードするとシステムパニックが発生する	187
x86: 既存のサービスパーティションが存在しないシステムでは、デフォルトでサービスパーティションが作成されない	189
▼ ネットワークインストールイメージまたは Solaris Operating System DVD からのソフトウェアのインストール	190
▼ Solaris SOFTWARE - 1 CD またはネットワークインストールイメージからのインストール	190

B	その他のSVR4パッケージ要件(リファレンス)	193
	現在のOSの変更を回避	193
	絶対パスの使用	193
	pkgadd -R コマンドの使用	194
	\$PKG_INSTALL_ROOT と \$BASEDIR の相違点の概要	194
	スクリプト作成のガイドライン	195
	ディスククライアントの互換性維持	196
	パッケージの検証	196
	インストール中およびアップグレード中のユーザー操作の回避	197
	ゾーンのパッケージパラメータの設定	199
	背景情報	202
	用語集	203
	索引	217

はじめに

このマニュアルでは、SPARC® および x86 アーキテクチャベースの、ネットワークに接続されたシステムとネットワークに接続されていないシステムの両方で、Solaris™ オペレーティングシステム (OS) をインストールおよびアップグレードする方法を説明します。このマニュアルでは、カスタム JumpStart インストールの方法と、インストール時に RAID-1 ボリュームを作成する方法についても説明します。

このマニュアルには、システムハードウェアや周辺装置を設定する方法は記載されていません。

注 - このリリースでは、SPARC および x86 系列のプロセッサアーキテクチャ (UltraSPARC®, SPARC64, AMD64, Pentium, Xeon EM64T) を使用するシステムをサポートします。サポートしているシステムは、<http://www.sun.com/bigadmin/hcl> の Solaris 10 Hardware Compatibility List に記載されています。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書の x86 に関連する用語については、以下を参照してください。

- 「x86」は、64 ビットおよび 32 ビットの x86 互換製品系列を指します。
- 「x64」は、AMD64 または EM64T システムに関する 64 ビット特有の情報を指します。
- 「32 ビット x86」は、x86 をベースとするシステムに関する 32 ビット特有の情報を指します。

サポートしているシステムについては、Solaris 10 Hardware Compatibility List を参照してください。

対象読者

このマニュアルは、Solaris OS のインストールを担当するシステム管理者を対象としています。このマニュアルには、次の 2 種類の情報が含まれています。

- ネットワーク環境で複数の Solaris マシンを管理するエンタープライズシステム管理者向けの上級 Solaris インストール情報
- Solaris のインストールやアップグレードをとときどき行うシステム管理者向けの基本 Solaris インストール情報

関連マニュアル

表 P-1 に、システム管理者向けのマニュアルの一覧を示します。

表 P-1 Solaris をインストールするシステム管理者向けのマニュアル

説明	情報
システム要件または計画の概要に関する情報が必要ですか。あるいは、GRUB ベースのブート、Solaris ゾーン区分技術、または RAID-1 ボリュームの作成に関する概要が必要ですか。	『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』
DVD または CD メディアから 1 つのシステムをインストールする必要がありますか。Solaris インストールプログラムは、手順を追ってインストールを案内します。	『Solaris 10 5/08 インストールガイド (基本編)』
停止時間をほとんど設けしないで、システムをアップグレードしたり、パッチを適用したりする必要がありますか。Solaris Live Upgrade を使うことにより、アップグレード時のシステム停止時間を短縮します。	『Solaris 10 5/08 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』
ネットワークやインターネットを介してセキュリティー保護されたインストールを行う必要がありますか。WAN ブートを使用して、リモートクライアントをインストールします。あるいは、ネットワークインストールイメージからネットワークを介してインストールする必要がありますか。Solaris インストールプログラムは、手順を追ってインストールを案内します。	『Solaris 10 5/08 インストールガイド (ネットワークインストール)』
複数のシステムをすばやくインストールしたり、パッチを適用したりする必要がありますか。Solaris フラッシュソフトウェアを使用して Solaris フラッシュアーカイブを作成し、クローンシステム上に OS のコピーをインストールします。	『Solaris 10 5/08 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』
システムのバックアップが必要ですか。	『Solaris のシステム管理 (デバイスとファイルシステム)』の第 23 章「ファイルシステムのバックアップと復元 (概要)」
トラブルシューティングに関する情報、既知の問題の一覧、またはこのリリース用のパッチの一覧が必要ですか。	『Solaris ご使用にあたって』
使用しているシステムが Solaris 上で動作することを確認する必要がありますか。	SPARC: 『Solaris Sun ハードウェアマニュアル』
このリリースで追加されたパッケージ、削除されたパッケージ、または変更されたパッケージを確認する必要がありますか。	『Solaris パッケージリスト』
使用しているシステムやデバイスが Solaris SPARC ベースのシステム、x86 ベースのシステム、およびその他のサードパーティーベンダーで動作するかどうかを確認する必要がありますか。	Solaris Hardware Compatibility List for x86 Platforms

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、次のサービスに関する情報も提供しています。

- 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す。

表 P-2 表記上の規則

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

パート I

カスタム JumpStart の使用

このパートでは、カスタム JumpStart インストールのための準備や実行手順について説明します。

Solaris インストールの計画についての参照先

このマニュアルでは、自動化された JumpStart インストールプログラムを使って Solaris オペレーティングシステムをインストールする方法について説明します。このマニュアルでは、JumpStart プログラムを使用したインストールに関して理解しておく必要のある情報をすべて提供しますが、JumpStart インストールの準備を始める前に、インストールマニュアルコレクション内の計画マニュアルを参照しておくことも役立ちます。次に示す参照先には、システムをインストールする前に役立つ情報が記載されています。

計画とシステム要件についての参照先

『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』には、システム要件と、ファイルシステムを計画するうえでのガイドラインやアップグレードの計画などの計画の概要が記載されています。この節では、このマニュアルの各章の概要を説明します。

計画マニュアルの章の説明	参照先
この章では、Solaris インストールプログラムの新機能について説明します。	『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の第 2 章「Solaris インストールの新機能」
この章では、Solaris OS のインストールやアップグレードを実施する前に決定する必要のある内容について説明します。たとえば、ネットワークインストールイメージや DVD メディアをどのようなときに使用するかを判断するために必要な情報や、すべての Solaris インストールプログラムについての説明を記載しています。	『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の第 3 章「Solaris のインストールおよびアップグレード (ロードマップ)」

計画マニュアルの章の説明	参照先
<p>この章では、Solaris OS のインストールやアップグレードに伴うシステム要件について説明します。また、ディスク容量やデフォルトスワップ空間の割り当ての計画に関する一般的な指針についても説明します。アップグレードの制限についても説明します。</p>	<p>『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の第4章「システム要件、ガイドライン、およびアップグレード (計画)」</p>
<p>この章には、システムのインストールやアップグレードに必要な情報の収集に役立つチェックリストが含まれています。ここで提供される情報は、対話式インストールの実行時などに役立ちます。このチェックリストでは、対話式インストールを行うために必要なすべての情報が得られます。</p>	<p>『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の第5章「インストールやアップグレードの前に収集すべき情報 (計画)」</p>
<p>これらの章では、Solaris OS のインストールまたはアップグレードに関連するいくつかの技術の概要を説明します。これらの技術に関連するガイドラインと要件も含まれています。これらの章には、GRUB ベースのブート、Solaris ゾーン区分技術、およびインストール時に作成できる RAID-1 ボリュームについての情報が含まれています。</p>	<p>『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』のパート II 「GRUB、Solaris ゾーン、および RAID-1 ボリュームに関連するインストールについて」</p>

カスタム JumpStart (概要)

この章では、カスタム JumpStart のインストール処理について紹介し、全体の概要を示します。

- 17 ページの「カスタム JumpStart とは」
- 19 ページの「JumpStart プログラムが Solaris ソフトウェアをインストールする際の流れ」

カスタム JumpStart とは

カスタム JumpStart では、あらかじめ作成したプロファイルを使って、複数のシステムのインストールやアップグレードを自動的にかつ同時に行うことができます。プロファイルには、どのようにソフトウェアをインストールするかを定義します。さらに、インストール前とインストール後に実行する作業を、シェルスクリプトを使用して指定することができます。システムのインストールまたはアップグレードにどのプロファイルとスクリプトを使用するかを選択できます。カスタム JumpStart では、選択したプロファイルとスクリプトに基づいて、システムのインストールまたはアップグレードが行われます。また、`sysidcfg` ファイルを使用して構成情報を指定することにより、カスタム JumpStart インストールを完全に自動化することも可能です。

カスタム JumpStart の使用例

次に、1 例を挙げてカスタム JumpStart の処理を説明します。この事例では、次の条件に従ってシステムを設定するものとします。

- 100 台の新しいシステムに Solaris をインストールします。
- うち 70 台は SPARC システムで、エンジニアリング部が所有し、これらは「開発者」Solaris OS ソフトウェアグループでスタンドアロンシステムとしてインストールします。

- 残りの 30 台は x86 システムで、マーケティング部が所有し、これらは「エンドユーザー」 Solaris OS ソフトウェアグループでスタンドアロンシステムとしてインストールします。

まず、システム管理者はシステムグループごとに `rules` ファイルとプロファイルを作成する必要があります。`rules` ファイルは、Solaris ソフトウェアをインストールするシステムグループごと(または1つのシステム)のルールが入ったテキストファイルです。各ルールは1つ以上のシステム属性にもとづいてシステムグループを識別し、各グループをプロファイルにリンクします。

プロファイルは、グループ内の各システムに Solaris ソフトウェアがどのようにインストールされるかを定義するテキストファイルです。`rules` ファイルとプロファイルは、JumpStart ディレクトリに置かれている必要があります。

この例の場合、システム管理者は2つの異なるルールで `rules` ファイルを作成します。1つはエンジニアリング部用のルールで、もう1つはマーケティング部用のルールです。ルールごとに、エンジニアリング部とマーケティング部を区別するために、システムのネットワーク番号が使用されます。

各ルールには、適切なプロファイルへのリンクも含まれています。たとえば、エンジニアリング部用のルールでは、エンジニアリング部用に作成した `eng_profile` というプロファイルへのリンクを追加します。マーケティング部用のルールでは、マーケティング部用に作成した `market_profile` というプロファイルへのリンクを追加します。

`rules` ファイルとプロファイルは、フロッピーディスクまたはサーバー上に保存できます。

- プロファイルフロッピーディスクは、カスタム JumpStart インストールを、ネットワークに接続されていないスタンドアロンシステムで実行するときに必要です。
- プロファイルサーバーは、カスタム JumpStart インストールを、サーバーにアクセスできるネットワークに接続されたシステムで実行するときに使用します。

`rules` ファイルとプロファイルを作成した後、`check` スクリプトを使用して、これらのファイルの妥当性をチェックする必要があります。`check` スクリプトが正常に動作する場合、`rules.ok` ファイルが作成されます。`rules.ok` は `rules` ファイルの生成バージョンであり、JumpStart プログラムによって Solaris ソフトウェアのインストールに使用されます。

JumpStart プログラムが Solaris ソフトウェアをインストールする際の流れ

rules ファイルとプロファイルの検証が終わったら、カスタム JumpStart インストールを開始します。JumpStart プログラムは、rules.ok ファイルを読み取ります。続いて、Solaris ソフトウェアがインストールされるシステムに一致するシステム属性を持つ最初のルールを検索します。一致するルールが見つかったら、JumpStart プログラムはそのルール内に指定されているプロファイルを使用してシステムに Solaris ソフトウェアをインストールします。

図 2-1 に、ネットワークに接続されていないスタンドアロンシステムでのカスタム JumpStart インストールの流れを示します。この例では、システム管理者は Pete のシステムでカスタム JumpStart インストールを開始しています。JumpStart プログラムは、システムのプロッピーディスクドライブに挿入されているプロッピーディスク上の rules ファイルにアクセスし、ルール 2 をシステムに対応付けます。ルール 2 には、JumpStart プログラムが Pete のプロファイルを使用して Solaris ソフトウェアをインストールするように指定されています。JumpStart プログラムは、Pete のプロファイルを読み取り、このプロファイル内に指定されている指示にもとづいて Solaris ソフトウェアをインストールします。

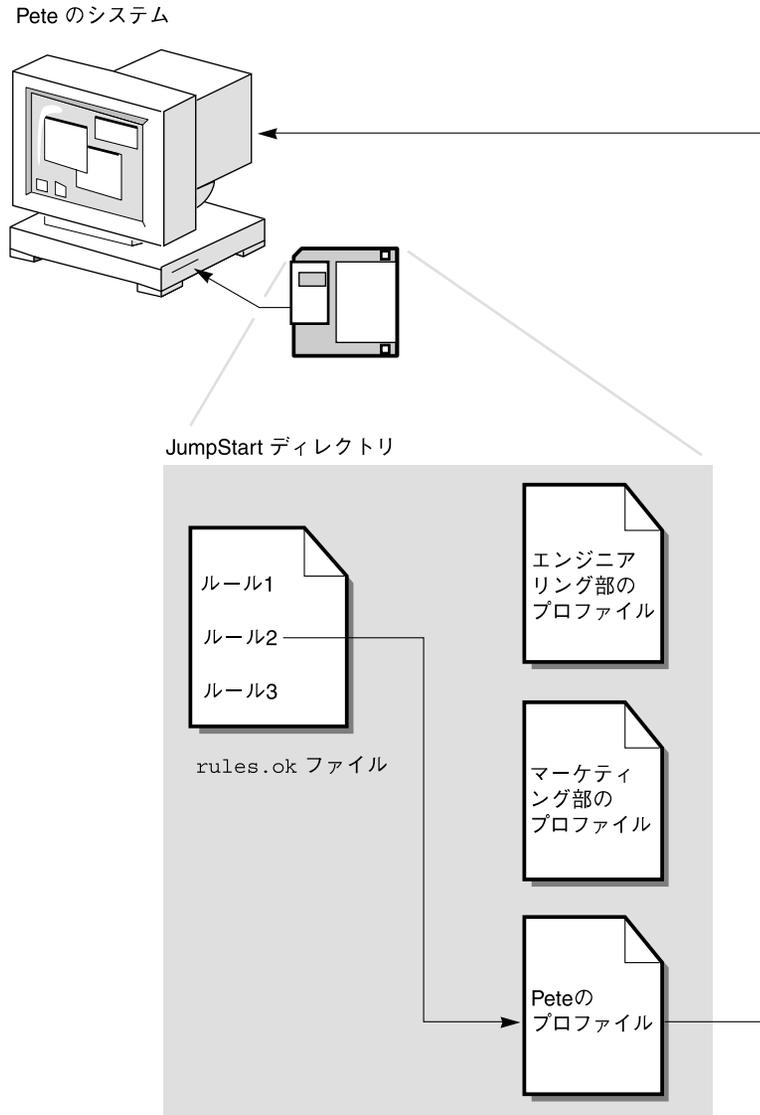


図 2-1 カスタム JumpStart インストールの流れ: ネットワークに接続されていない場合の例

図 2-2 に、ネットワーク上の複数のシステムでのカスタム JumpStart インストールの流れを示します。事前に個別のプロファイルが設定され、1 台のサーバーにまとめて保存されています。システム管理者は、エンジニアリング部のシステムの中の 1 台でカスタム JumpStart インストールを開始します。JumpStart プログラムは、サーバー上の JumpStart/ ディレクトリに存在する rules ファイルにアクセスします。JumpStart プログラムは、エンジニアリング部のシステムをルール 1 に対応付けます。ルール 1

には、JumpStart プログラムがエンジニアリング部のプロファイルを使用して Solaris ソフトウェアをインストールするように指定されています。JumpStart はエンジニアリング部のプロファイルを読み取り、このプロファイル内に指定されている指示にもとづいて Solaris ソフトウェアをインストールします。

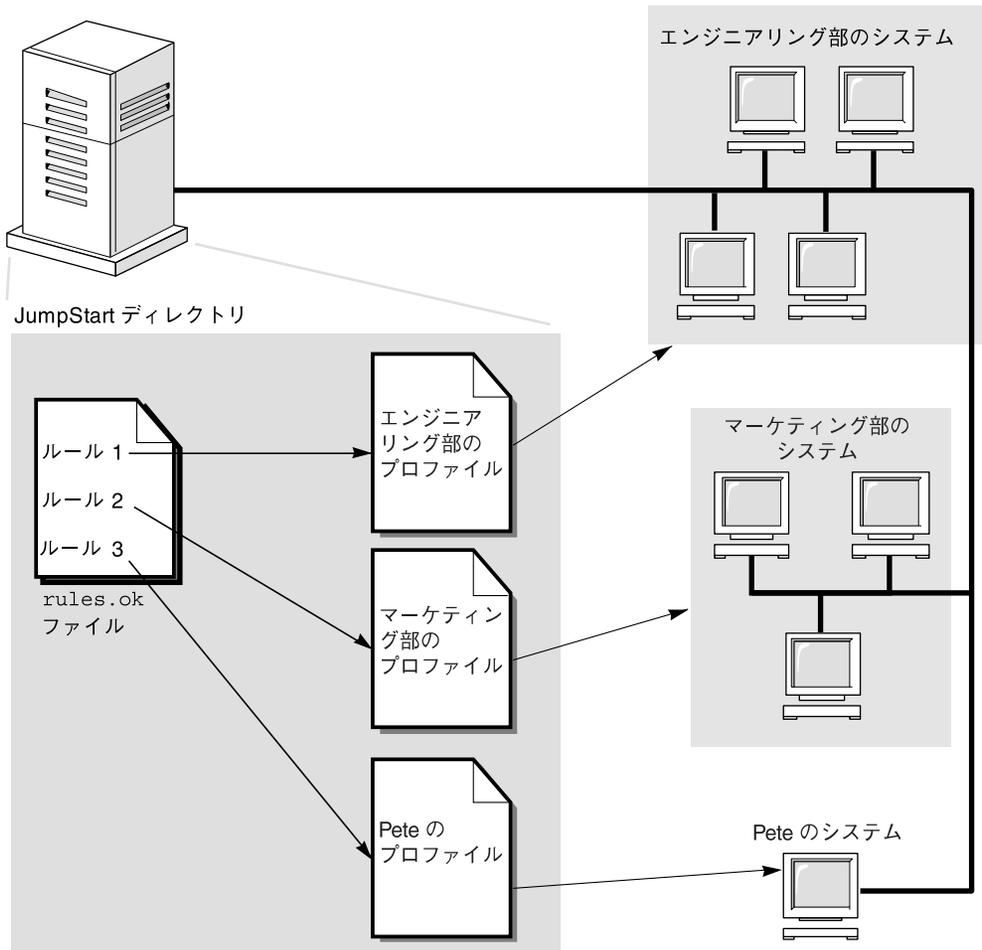


図 2-2 カスタム JumpStart インストールの流れ: ネットワークに接続されている場合の例

図 2-3 に、JumpStart プログラムがカスタム JumpStart ファイルを検索する順序を示します。

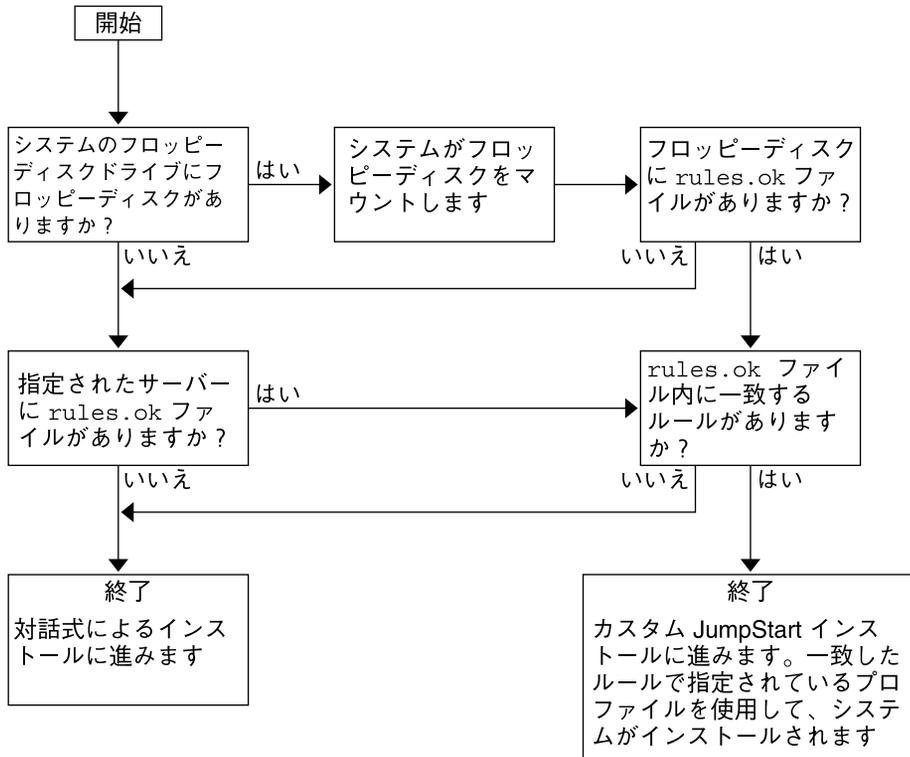


図 2-3 カスタム JumpStart インストール実行時の流れ

カスタム JumpStart インストールの準備 (作業)

この章では、カスタム JumpStart インストールを使用して、各サイトのシステムに Solaris ソフトウェアをインストールするために必要な準備の手順について説明します。

- 23 ページの「作業マップ: カスタム JumpStart インストールの準備」
- 25 ページの「ネットワーク上のシステム用のプロファイルサーバーの作成」
- 31 ページの「スタンドアロンシステム用のプロファイルフロッピーディスクの作成」
- 36 ページの「rules ファイルの作成」
- 40 ページの「プロファイルの作成」
- 55 ページの「プロファイルのテスト」
- 59 ページの「rules ファイルの妥当性を検査する」

作業マップ: カスタム JumpStart インストールの準備

表 3-1 作業マップ: カスタム JumpStart インストールの準備

作業	説明	参照先
旧バージョンの Solaris ソフトウェアがシステム上にインストールされている場合は、システムをアップグレードする方法を決定します。	システムに以前の Solaris リリースがインストールされている場合は、システムをどのようにアップグレードするかを決める必要があります。システムアップグレードの前と後に実施すべき作業を明確にしておいてください。計画を立てることによって、プロファイル、開始スクリプト、終了スクリプトなどの作成が容易になります。	『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の「アップグレード計画」

表 3-1 作業マップ:カスタム JumpStart インストールの準備 (続き)

作業	説明	参照先
JumpStart ディレクトリを作成します。	<p>サーバーの場合</p> <p>ネットワーク接続されたシステムに対してカスタム JumpStart インストールを実行する場合は、プロファイルサーバーを作成する必要があります。プロファイルサーバーには、カスタム JumpStart ファイル用の JumpStart ディレクトリが入っています。</p> <p>フロッピーディスクの場合</p> <p>ネットワークに接続されていないシステムに対してカスタム JumpStart インストールを実行する場合は、プロファイルフロッピーディスクを作成する必要があります。プロファイルフロッピーディスクには、カスタム JumpStart ファイルが入っています。</p>	<p>25 ページの「ネットワーク上のシステム用のプロファイルサーバーの作成」</p> <p>31 ページの「スタンドアロンシステム用のプロファイルフロッピーディスクの作成」</p>
ルールを rules ファイルに追加します。	各システムグループまたは単一のシステムをどのようにインストールするか決定したあとで、インストールする各グループに対してルールを作成します。各ルールは、1つ以上のシステム属性に基づいてグループを識別し、それぞれのグループを1つのプロファイルにリンクします。	36 ページの「rules ファイルの作成」
ルールごとにプロファイルを作成します。	プロファイルは、どのように Solaris ソフトウェアをシステムにインストールするか(たとえば、どのソフトウェアグループをインストールするか)を定義するテキストファイルです。すべてのルールはプロファイルを指定して、ルールが一致したときにシステムがどのようにインストールされるかを定義します。通常は、ルールごとに異なるプロファイルを作成します。しかし、複数のルールで同じプロファイルを使用することも可能です。	40 ページの「プロファイルの作成」
(省略可能) プロファイルをテストします。	プロファイルの作成後、 <code>pfinstall(1M)</code> コマンドを使用すれば、プロファイルを使用してシステムをインストールまたはアップグレードする前に、プロファイルをテストできます。	55 ページの「プロファイルのテスト」

表 3-1 作業マップ: カスタム JumpStart インストールの準備 (続き)

作業	説明	参照先
rules ファイルの妥当性をチェックします。	rules.ok ファイルは、rules ファイルから生成されたファイルで、JumpStart プログラムはこれを使用して、インストールするシステムとプロファイルとを一致させます。rules ファイルの妥当性をチェックするには、check スクリプトを使用する必要があります。	59 ページの「rules ファイルの妥当性を検査する」

ネットワーク上のシステム用のプロファイルサーバーの作成

ネットワーク上のシステム用にカスタム JumpStart インストールを設定するには、サーバー上に JumpStart ディレクトリと呼ばれるディレクトリを作成する必要があります。JumpStart ディレクトリには、重要なカスタム JumpStart ファイル (rules ファイル、rules.ok ファイル、プロファイルなど) がすべて置かれます。JumpStart ディレクトリは、プロファイルサーバーのルート (/) ディレクトリに作成する必要があります。

JumpStart ディレクトリがあるサーバーを「プロファイルサーバー」と呼びます。プロファイルサーバーは、インストールサーバーまたはブートサーバーと同じにすることも、あるいはまったく異なるサーバーとすることもできます。プロファイルサーバーは、異なったプラットフォームのシステムにカスタム JumpStart ファイルを提供できます。たとえば、x86 サーバーは、SPARC ベースのシステムと x86 ベースのシステムの両方に、カスタム JumpStart ファイルを提供できます。

注-プロファイルサーバーの作成後、システムがそのサーバーにアクセスできるように設定する必要があります。詳しい手順については、29 ページの「すべてのシステムがプロファイルサーバーにアクセスできるようにする方法」を参照してください。

▼ JumpStart ディレクトリをサーバー上に作成する方法

注-この手順では、システムでボリュームマネージャーを実行していると仮定しています。ディスク管理にボリュームマネージャーを使用しない場合は、『Solaris のシステム管理 (デバイスとファイルシステム)』で、ボリュームマネージャーを使用しないでリムーバブルメディアを管理する方法を確認してください。

- 1 JumpStart ディレクトリを作成するサーバーを見つけます。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 3 サーバーに JumpStart ディレクトリを作成します。

```
# mkdir -m 755 jumpstart_dir_path
```

`jumpstart_dir_path` は、JumpStart ディレクトリの絶対パスです。
たとえば、次のコマンドは、ルート (/) ディレクトリに `jumpstart` というディレクトリを作成し、アクセス権を 755 に設定します。

```
# mkdir -m 755 /jumpstart
```
- 4 次のエントリを追加して、`/etc/dfs/dfstab` ファイルを編集します。

```
share -F nfs -o ro,anon=0 jumpstart_dir_path
```

たとえば、次のエントリは `/jumpstart` ディレクトリを共有します。

```
share -F nfs -o ro,anon=0 /jumpstart
```
- 5 `shareall` と入力し、**Enter** キーを押します。
- 6 サンプルのカスタム JumpStart ファイルを JumpStart ディレクトリにコピーするかどうかを決定します。
 - コピーしない場合は、[手順 9](#)に進みます。
 - コピーする場合は、次の表を利用して次に行う作業を決定します。

サンプルのコピー元	参照先
プラットフォームに対応した Solaris Operating System DVD または Solaris SOFTWARE - 1 CD	サーバーの CD-ROM ドライブに、Solaris Operating System DVD または Solaris SOFTWARE - 1 CD を挿入します。 ボリュームマネージャーが自動的に CD または DVD をマウントします。
ローカルディスク上にある、プラットフォームに対応した Solaris Operating System DVD または Solaris SOFTWARE - 1 のイメージ	Solaris Operating System DVD または Solaris SOFTWARE - 1 イメージが置かれたディレクトリに移動します。コマンド例を示します。 <code>cd /export/install</code>

- 7 サンプルのカスタム **JumpStart** ファイルをプロファイルサーバー上の **JumpStart** ディレクトリにコピーします。

```
# cp -r media_path/Solaris_10/Misc/jumpstart_sample/* jumpstart_dir_path
```

media_path CD、DVD、またはローカルディスク上のイメージのパス
jumpstart_dir_path カスタム JumpStart ファイルの例を配置する、プロファイルサーバー上のパス

たとえば、次のコマンドは、*jumpstart_sample* ディレクトリをプロファイルサーバー上の */jumpstart* ディレクトリにコピーします。

- SPARC システムの場合:

```
cp -r /cdrom/cdrom0/s0/Solaris_10/Misc/jumpstart_sample/* /jumpstart
```

- x86 システムの場合:

```
cp -r /cdrom/cdrom0/Solaris_10/Misc/jumpstart_sample/* /jumpstart
```

- 8 サンプル **JumpStart** ファイルを更新し、それらのファイルがサイトの環境内で動作するようにします。
- 9 **root** が **JumpStart** ディレクトリを所有していて、そのアクセス権が **755** に設定されていることを確認します。
- 10 ネットワーク上のシステムがプロファイルサーバーにアクセスできるように設定します。
詳しい手順については、29 ページの「すべてのシステムがプロファイルサーバーにアクセスできるようにする方法」を参照してください。

すべてのシステムがプロファイルサーバーにアクセスできるようにする

プロファイルサーバーを作成する際に、システムがカスタム JumpStart インストール中にプロファイルサーバーにアクセスできるようにする必要があります。このためには、次の方法があります。

コマンドまたはファイル	アクセスの提供	参照先
<code>add_install_client</code> コマンド	<p>ネットワークインストールでシステムを追加するたびに、<code>add_install_client</code> コマンドに <code>-c</code> オプションを使用してプロファイルサーバーを指定します。</p> <p>注-NFSを使用していない場合は、別の方法を使用してアクセスを提供する必要があります。</p> <ul style="list-style-type: none"> ■ SPARC システムでは <code>boot</code> コマンドを使用します ■ x86 システムでは GRUB メニューを編集します 	<ul style="list-style-type: none"> ■ DVD メディアの場合は、『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の「DVD イメージを使用してネットワークからインストールするシステムの追加」を参照してください。 ■ CD メディアの場合は、『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の「CD イメージを使用してネットワークからインストールするシステムの追加」を参照してください。

コマンドまたはファイル	アクセスの提供	参照先
システムのブート時に JumpStart ディレクトリの場所を指定します	<ul style="list-style-type: none"> ■ SPARC システムでは、<code>boot</code> コマンドを使用してシステムをブートします。システムのブート時にプロファイルサーバー上の JumpStart ディレクトリの場所を指定します。カスタム JumpStart 構成ファイルは、圧縮して1つのファイルにする必要があります。続いて、圧縮されたその構成ファイルを HTTP サーバーまたは HTTPS サーバーに保存します。 ■ x86 システムでは、システムをブートするときに GRUB メニューのブートエントリを編集して、プロファイルサーバーの JumpStart ディレクトリの場所を指定します。カスタム JumpStart 構成ファイルは、圧縮して1つのファイルにする必要があります。続いて、圧縮されたその構成ファイルを HTTP サーバーまたは HTTPS サーバーに保存します。GRUB メニューのエントリを編集する場合は、圧縮ファイルの場所を指定します。 	<ul style="list-style-type: none"> ■ 72 ページの「圧縮された構成ファイルの作成」 ■ 93 ページの「SPARC: カスタム JumpStart プログラムを使用してインストールまたはアップグレードする方法」の手順 5 ■ 72 ページの「圧縮された構成ファイルの作成」 ■ 100 ページの「x86: GRUB ブートコマンドの編集によるカスタム JumpStart インストールの実行」
<code>/etc/bootparams</code> ファイル	<code>/etc/bootparams</code> ファイルにワイルドカードを追加します。	29 ページの「すべてのシステムがプロファイルサーバーにアクセスできるようにする方法」

▼ すべてのシステムがプロファイルサーバーにアクセスできるようにする方法

次に示す手順は、ネットワークインストールの情報を次の場所に格納している場合にのみ実行してください。

- `/etc/bootparams` ファイル内。
- ネームサービス `bootparams` データベース内。 `bootparams` データベースを更新するには、[手順 3](#) に示されているエントリを追加してください。

次に示す手順は、同じタイプのシステム(たとえばすべて SPARC システム)で実行する必要があります。

次のような場合には、この手順を実行しないでください。

- フロッピーディスクに JumpStart ディレクトリを保存する場合。
- システムをブートするときにプロファイルサーバーの場所を指定する場合。異なるアーキテクチャーのシステムがある場合、システムをブートするときにプロファイルサーバーの場所を指定する必要があります

前述の場合には、SPARC システムでは boot コマンドを使用し、x86 システムでは GRUB メニューを使用してください。

注-DHCP サーバーにネットワークインストールの情報を格納することもできます。

- SPARC システムの場合は、`add_install_client` コマンドに `-d` オプションを指定して、カスタム JumpStart プログラムが DHCP サーバーを使用するように指定します。または `boot` コマンドに `dhcp` オプションを指定して、カスタム JumpStart プログラムが DHCP サーバーを使用するように指定します。このオプションを使用する手順については、94 ページの「SPARC: boot コマンドのコマンドリファレンス」を参照してください。
- x86 システムでは、次のいずれかの方法で `dhcp` を使用します。
 - インストールサーバーを使用する場合は、`add_install_client` コマンドに `-d` オプションを指定して、カスタム JumpStart プログラムが DHCP サーバーおよび PXE を使用するように指定します。
 - GRUB メニューの GRUB エントリを編集して `dhcp` オプションを追加します。GRUB エントリを編集する手順については、100 ページの「x86: GRUB ブートコマンドの編集によるカスタム JumpStart インストールの実行」を参照してください

-
- 1 インストールサーバーまたはブートサーバーにスーパーユーザーとしてログインします。
 - 2 テキストエディタを使用して `/etc/bootparams` を開きます。
 - 3 このエントリを追加します。

```
* install_config=server:jumpstart_dir_path
```

```
*  
      すべてのシステムにアクセスできるように指定するワイルド  
      カード文字
```

```
server      JumpStart ディレクトリがあるプロファイルサーバーのホスト  
            名
```

```
jumpstart_dir_path  JumpStart ディレクトリの絶対パス
```

たとえば、次のエントリはすべてのシステムが、`sherlock` というプロファイルサーバーにある `/jumpstart` ディレクトリにアクセスできるようにします。

```
* install_config=sherlock:/jumpstart
```



注意-この手順を使用した場合、インストールクライアントを起動したときに次のエラーメッセージが表示されることがあります。

```
WARNING:getfile:RPC failed:error 5:(RPC Timed out).
```

このエラーメッセージの詳細は、174ページの「ネットワークからのブート時のエラーメッセージ」を参照してください。

これで、すべてのシステムはプロファイルサーバーにアクセスできるようになりました。

スタンドアロンシステム用のプロファイルフロッピーディスクの作成

JumpStart ディレクトリが収められているフロッピーディスクを「プロファイルフロッピーディスク」と呼びます。ネットワークに接続されていないシステムには、プロファイルサーバーへのアクセス権はありません。このため、システムがネットワークに接続されていない場合はフロッピーディスク上にJumpStart ディレクトリを作成する必要があります。プロファイルフロッピーディスクを作成するシステムには、フロッピーディスクドライブが必要です。

JumpStart ディレクトリには、rulesファイル、rules.okファイル、プロファイルなどの基本的なカスタムJumpStart ファイルがすべて収められています。JumpStart ディレクトリは、プロファイルフロッピーディスクのルート(/)ディレクトリに作成する必要があります。

次のいずれかの手順を参照してください。

- 32 ページの「SPARC: プロファイルフロッピーディスクを作成する方法」
- 34 ページの「x86: GRUB を使用してプロファイルフロッピーディスクを作成する方法」

▼ SPARC: プロファイルフロッピーディスクを作成する方法

注-この手順では、システムでボリュームマネージャーを実行していると仮定しています。フロッピーディスク、CD、およびDVDの管理にボリュームマネージャーを使用しない場合は、『Solarisのシステム管理(デバイスとファイルシステム)』で、ボリュームマネージャーを使用しないでリムーバブルメディアを管理する方法を確認してください。

- 1 フロッピーディスクドライブを持つ SPARC システムを見つけます。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solarisのシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。
- 3 空のフロッピーディスクまたは上書き可能なディスクをフロッピーディスクドライブに挿入します。
- 4 フロッピーディスクをマウントします。
volcheck
- 5 フロッピーディスクに UNIX ファイルシステム (UFS) が含まれているかどうかを確認します。
システム上の /etc/mnttab ファイルに、次のようなエントリがあるかどうかを調べます。

```
/vol/dev/diskette0/scrap /floppy/scrap ufs suid,rw,largefiles,dev=1740008 927147040
```

 - エントリが存在する場合は、[手順 7](#)へ進みます。
 - エントリが存在しない場合は、次の手順へ進みます。
- 6 フロッピーディスク上に UFS を作成します。
newfs /vol/dev/aliases/floppy0
- 7 サンプルのカスタム JumpStart ファイルを JumpStart ディレクトリにコピーするかどうかを決定します。
 - コピーしない場合は、[手順 10](#)へ進みます。
 - コピーする場合は、次の表を利用して次に行う作業を決定します。

サンプルのコピー元	参照先
Solaris Operating System DVD (SPARC 版) または Solaris SOFTWARE - 1 CD (SPARC 版)	<p>サーバーの CD-ROM ドライブに Solaris Operating System DVD (SPARC 版) または Solaris SOFTWARE - 1 CD (SPARC 版) を挿入します。</p> <p>ボリュームマネージャーが自動的に CD または DVD をマウントします。</p>
ローカルディスク上の Solaris Operating System DVD (SPARC 版) または Solaris SOFTWARE - 1 CD (SPARC 版) イメージ	<p>Solaris Operating System DVD (SPARC 版) または Solaris SOFTWARE - 1 CD (SPARC 版) イメージが置かれたディレクトリに移動します。コマンド例を示します。</p> <pre>cd /export/install</pre>

- 8 サンプルのカスタム **JumpStart** ファイルをプロファイルフロッピーディスクの **JumpStart** ディレクトリにコピーします。

```
# cp -r media_path/Solaris_10/Misc/jumpstart_sample/* jumpstart_dir_path
```

media_path CD、DVD、またはローカルディスク上のイメージのパス

jumpstart_dir_path カスタム JumpStart のサンプルファイルを置くプロファイルフロッピーディスクのパス

注-すべてのカスタム JumpStart インストールファイルは、フロッピーディスクのルート (/) ディレクトリに置かれている必要があります。

たとえば、次のコマンドは、Solaris SOFTWARE - 1 CD (SPARC 版) の *jumpstart_sample* の内容を、プロファイルフロッピーディスク *scrap* のルート (/) ディレクトリにコピーします。

```
cp -r /cdrom/cdrom0/s0/Solaris_10/Misc/jumpstart_sample/* /floppy/scrap
```

- 9 プロファイルフロッピーディスク上の **JumpStart** ファイルの例を更新して、サイトの環境内で動作するようにします。
- 10 **root** が **JumpStart** ディレクトリを所有していて、そのアクセス権が **755** に設定されていることを確認します。
- 11 フロッピーディスクを取り出します。

```
# eject floppy
```

これで、プロファイルフロッピーディスクの作成が完了しました。rules ファイルを更新して、プロファイルフロッピーディスクにプロファイルを作成して、カスタム JumpStart インストールを実行できます。続行するには、[36 ページの「rules ファイルの作成」](#)へ進みます。

▼ x86: GRUB を使用してプロファイルフロッピーディスクを作成する方法

GRUB を使用してプロファイルフロッピーディスクを作成するには、この手順を使用します。GRUB メニューを使用すればインストール手順の実行中にブート処理を実行できます。以前のリリースではシステムのブートに Solaris Device Configuration Assistant (デバイス構成用補助) が必要となる場合がありますでしたが、GRUB メニューはそれに代わるものです。

注- この手順では、システムでボリュームマネージャーを実行していると仮定しています。フロッピーディスク、CD、および DVD の管理にボリュームマネージャーを使用しない場合は、『Solaris のシステム管理 (デバイスとファイルシステム)』で、ボリュームマネージャーを使用しないでリムーバブルメディアを管理する方法を確認してください。

- 1 フロッピーディスクドライブを持つ x86 システムを見つけます。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 3 空のフロッピーディスクまたは上書き可能なディスクをフロッピーディスクドライブに挿入します。
- 4 フロッピーディスクをマウントします。
`# volcheck`
- 5 サンプルのカスタム JumpStart ファイルを JumpStart ディレクトリにコピーするかどうかを決定します。
 - コピーしない場合は、[手順 8](#)へ進みます。
 - コピーする場合は、次の表を利用して次に行う作業を決定します。

サンプルのコピー元	参照先
Solaris Operating System DVD (x86 版) または Solaris SOFTWARE - 1 CD (x86 版)	サーバーの CD-ROM ドライブに、Solaris Operating System DVD (x86 版) または Solaris SOFTWARE - 1 CD (x86 版) を挿入します。 ボリュームマネージャーが自動的に CD または DVD をマウントします。

サンプルのコピー元	参照先
ローカルディスク上の Solaris Operating System DVD (x86 版) または Solaris SOFTWARE - 1 CD (x86 版) イメージ	Solaris Operating System DVD (x86 版) または Solaris SOFTWARE - 1 CD (x86 版) イメージが置かれたディレクトリに移動します。入力例を示します。 <code>cd /export/install</code>

- 6 サンプルのカスタム **JumpStart** ファイルをプロファイルフロッピーディスクの **JumpStart** ディレクトリにコピーします。

```
# cp -r media_path/Solaris_10/Misc/jumpstart_sample/* jumpstart_dir_path
```

media_path CD、DVD、またはローカルディスク上のイメージのパス

jumpstart_dir_path カスタム JumpStart のサンプルファイルを置くプロファイルフロッピーディスクのパス

注-すべてのカスタム JumpStart インストールファイルは、プロファイルフロッピーディスクのルート (/) ディレクトリに置かれている必要があります。

たとえば、次のコマンドは、Solaris SOFTWARE - 1 CD (x86 版) の *jumpstart_sample* の内容を、プロファイルフロッピーディスク *scrap* のルート (/) ディレクトリにコピーします。

```
cp -r /cdrom/cdrom0/Solaris_10/Misc/jumpstart_sample/* /floppy/scrap
```

- 7 プロファイルフロッピーディスク上の **JumpStart** ファイルの例を更新して、サイトの環境内で動作するようにします。
- 8 **root** が **JumpStart** ディレクトリを所有していて、そのアクセス権が **755** に設定されていることを確認します。
- 9 「ファイル・マネージャ (**File Manager**)」ウィンドウの「取り出し (**Eject Disk**)」をクリックするか、コマンド行に `eject floppy` と入力して、フロッピーディスクを取り出します。
- 10 「リムーバブルメディア・マネージャ (**Removable Media Manager**)」画面で「了解 (**OK**)」をクリックします。
- 11 フロッピーディスクを手動で取り出します。

参照 これでは、プロファイルフロッピーディスクの作成が完了しました。rules ファイルを更新して、プロファイルフロッピーディスクにプロファイルを作成して、カスタム JumpStart インストールを実行できます。続行するには、[36 ページの「rules ファイルの作成」](#)へ進みます。

rules ファイルの作成

rules ファイルは、Solaris OS をインストールする各システムグループのルールが記述されているテキストファイルです。各ルールは1つ以上のシステム属性にもとづいてシステムグループを識別し、各グループをプロファイルにリンクします。プロファイルは、グループ内の各システムに Solaris ソフトウェアがどのようにインストールされるかを定義するテキストファイルです。たとえば、次のルールは JumpStart プログラムが basic_prof プロファイル内の情報を使用し、sun4u プラットフォームグループを持つすべてのシステムに対してインストールを実行することを示します。

```
karch sun4u - basic_prof -
```

rules ファイルを使用して、カスタム JumpStart インストールに必要な rules.ok ファイルを作成します。

注-31 ページの「スタンドアロンシステム用のプロファイルフロッピーディスクの作成」または 25 ページの「ネットワーク上のシステム用のプロファイルサーバーの作成」の手順に従って JumpStart ディレクトリを設定した場合は、rules ファイルのサンプルがすでに JumpStart ディレクトリに置かれています。rules ファイルのサンプルには、説明といくつかのルール例があります。サンプルの rules ファイルを利用する場合は、使用しないルール例は必ずコメントにしておいてください。

rules ファイルの構文

rules ファイルには、次の属性が必要です。

- ファイルに rules という名前が付けられている。
- ファイルに1つ以上のルールが含まれている。

rules ファイルには、次の要素を任意に含めることができます。

- コメント
行中の # 記号の後ろのテキストは、JumpStart によってコメントとして扱われ
ます。行が # 記号で始まる場合は、行全体がコメントとして扱われます。
- 1 行以上の空白行

- 1つまたは複数の複数行ルール
1つのルールを新しい行に折り返すには、Return キーを押す直前にバックslash文字 (\) を入力します。

▼ rules ファイルを作成する方法

- 1 テキストエディタを使って、rules という名前のテキストファイルを作成します。または、作成済みの JumpStart ディレクトリからサンプル rules ファイルを開きます。
- 2 Solaris ソフトウェアをインストールする各システムグループの rules ファイルにルールを追加します。

rules ファイルのキーワードと値については、115 ページの「ルールキーワードと値の説明」を参照してください。

rules ファイル内のルールは、次の構文に従っている必要があります。

```
!rule_keyword rule_value && !rule_keyword rule_value ... begin profile finish
```

! ルールキーワードの前で使用し、否定を示す記号。

rule_keyword ホスト名(hostname)やメモリーサイズ(memsize)などの一般的なシステム属性を記述する、事前定義された字句単位または語。
rule_keyword は、同じ属性を持つシステムを1つのプロファイルに対応させるためにルール値と共に使用されます。ルールキーワードの一覧は、115 ページの「ルールキーワードと値の説明」を参照してください。

rule_value 対応するルールキーワードに特定のシステム属性を付与する値。
ルール値については、115 ページの「ルールキーワードと値の説明」を参照してください。

&& ルールキーワードとルール値のペアを1つのルールに結合するために使用する記号(論理演算子 AND)。カスタム JumpStart インストール時に、システムがルール内のすべてのペアに一致しなければ、ルールの一致は成立しません。

begin インストール開始前に実行される Bourne シェルスクリプト名。開始スクリプトがない場合、このフィールドにマイナス記号(-)を指定する必要があります。開始スクリプトはすべて、JumpStart ディレクトリになければなりません。

開始スクリプトの詳細な作成方法については、63 ページの「開始スクリプトの作成」を参照してください。

profile テキストファイル名。システムがルールに一致したとき Solaris ソフトウェアがシステムにどのようにインストールされるかを定義して

います。プロファイル内の情報は、プロファイルキーワードと、それらに対応するプロファイル値から構成されます。すべてのプロファイルは JumpStart ディレクトリになければなりません。

注- プロファイルフィールドの別の使用方法については、79 ページの「サイト固有のインストールプログラムの使用」と 64 ページの「開始スクリプトによる派生プロファイルの作成」を参照してください。

finish インストール終了後に実行される Bourne シェルスクリプト名。終了スクリプトがない場合、このフィールドにマイナス記号 (-) を指定する必要があります。終了スクリプトはすべて、JumpStart ディレクトリになければなりません。

終了スクリプトの詳しい作成方法については、65 ページの「終了スクリプトの作成」を参照してください。

各ルールには、少なくとも次の要素を含める必要があります。

- キーワード、値、および対応するプロファイル
- *begin* および *finish* フィールド内のマイナス記号 (-) (開始および終了スクリプトが指定されない場合)

- 3 rules ファイルを JumpStart ディレクトリに保存します。
- 4 root が rules ファイルを所有していて、そのアクセス権が 644 に設定されていることを確認します。

rules ファイルの例

次のサンプルファイルは、rules ファイル内のルールをいくつか示しています。各行には、ルールキーワードとそのキーワードに有効な値があります。JumpStart プログラムは、rules ファイルを上から下へ走査します。

ルールキーワードと値を既知のシステムに一致させることができた場合、JumpStart プログラムはプロファイルフィールドに示されているプロファイルによって指定された Solaris ソフトウェアをインストールします。

rules ファイルの制限に関する詳細なリストについては、36 ページの「rules ファイルの構文」を参照してください。

例3-1 ruleファイル

```

# rule keywords and rule values      begin script      profile      finish script
# -----
hostname eng-1                       -                basic_prof   -
network 192.168.255.255 && !model \
'SUNW,Sun-Blade-100'                 -                net_prof     -
model SUNW,SPARCstation-LX           -                lx_prof      complete
network 192.168.2.0 && karch i86pc     setup            x86_prof     done
memsize 64-128 && arch i386           -                prog_prof    -
any -                                  -                generic_prof -

```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

hostname	このルールは、システムのホスト名が <code>eng-1</code> の場合に一致します。 <code>basic_prof</code> プロファイルは、ルールに一致するシステムに Solaris ソフトウェアをインストールするために使用されます。
network	このルールは、システムがサブネット <code>192.168.255.255</code> 上にあり、Sun Blade™ 100 (<code>SUNW,Sun-Blade-100</code>) でない場合に一致します。 <code>net_prof</code> プロファイルは、このルールに一致するシステムに Solaris ソフトウェアをインストールするために使用されます。このルールでは、バックslash文字 (\) を使って、1つのルールを新規行に継続する例も示されます。
model	このルールは、システムが <code>SPARCstation LX</code> である場合に一致します。 <code>lx_prof</code> プロファイルと <code>complete</code> 終了スクリプトは、このルールに一致するシステムに Solaris ソフトウェアをインストールするために使用されます。
network	このルールは、システムがサブネット <code>192.168.2.0</code> 上にあって、 <code>x86</code> ベースの <code>sun4u</code> システムである場合に一致します。 <code>setup</code> 開始スクリプト、 <code>x864u_prof</code> プロファイル、および <code>done</code> 終了スクリプトは、このルールに一致するシステムに Solaris ソフトウェアをインストールするために使用されます。
memsize	このルールは、システムが <code>64 ~ 128M</code> バイトのメモリーを持つ <code>x86</code> システムである場合に一致します。 <code>prog_prof</code> プロファイルは、このルールに一致するシステムに Solaris ソフトウェアをインストールするために使用されます。
any	このルールは、前述のルールに一致しなかったすべてのシステムに一致します。 <code>generic_prof</code> プロファイルは、このルールに一致するシステムに Solaris ソフトウェアをインストールするために使用されます。 <code>any</code> を使用する場合は、この値を常に <code>rules</code> ファイルの最後のファイルとして使用する必要があります。

プロファイルの作成

プロファイルは、システムに Solaris ソフトウェアをインストールする方法を定義するテキストファイルです。プロファイルには、インストール要素(インストールするソフトウェアグループなど)を指定します。各ルールには、システムのインストール方法を示すプロファイルを指定します。プロファイルは、ルールごとに異なるものを作成しても、複数のルールに同じものを使用しても構いません。

プロファイルは、1つ以上のプロファイルキーワードとその値から成ります。各プロファイルキーワードは、JumpStart プログラムがどのようにしてシステムに Solaris ソフトウェアをインストールするかを制御するコマンドです。たとえば、次のプロファイルキーワードと値は、JumpStart プログラムがシステムをサーバーとしてインクルードすることを指定しています。

```
system_type server
```

注- 次のいずれかの手順に従って JumpStart ディレクトリを作成した場合は、サンプルのプロファイルがすでに JumpStart ディレクトリに入っています。

- 25 ページの「ネットワーク上のシステム用のプロファイルサーバーの作成」
 - 31 ページの「スタンドアロンシステム用のプロファイルフロッピーディスクの作成」
-

プロファイルの構文

プロファイルには、次の要素を含める必要があります。

- 最初のエントリが `install_type` プロファイルキーワードであること
- キーワードは 1 行に 1 つだけであること
- アップグレードされるシステムが、アップグレードできるルート (/) ファイルシステムを複数持っている場合は、`root_device` キーワード

プロファイルには次の要素を含めることができます。

- コメント
行中の # 記号の後ろのテキストは、JumpStart プログラムによってコメントとして扱われます。行が # 記号で始まる場合は、行全体がコメントとして扱われます。
- 1 行以上の空白行

▼ プロファイルを作成する方法

- 1 テキストエディタを使用してテキストファイルを作成します。ファイルにわかりやすい名前を付けます。または、作成した **JumpStart** ディレクトリ内のプロファイル例を開きます。

注- プロファイルには、Solaris ソフトウェアをシステムにインストールするときの使用方法を表す名前を付けてください。たとえば、basic_install、eng_profile、user_profile などの名前を付けます。

- 2 プロファイルにプロファイルキーワードと値を追加します。
プロファイルキーワードと値の一覧は、[119 ページの「プロファイルキーワードと値」](#)を参照してください。

注- プロファイルキーワードとプロファイル値には、大文字と小文字の区別があります。

- 3 **JumpStart** ディレクトリにプロファイルを保存します。
- 4 プロファイルの所有者が root で、そのアクセス権が **644** に設定されていることを確認します。
- 5 (省略可能) プロファイルをテストします。
プロファイルのテスト方法については、[55 ページの「プロファイルのテスト」](#)を参照してください。

プロファイルの例

次のプロファイル例は、さまざまなプロファイルキーワードとプロファイル値を使用して、Solaris ソフトウェアをシステムにどのようにインストールするかを指定する方法を示しています。[119 ページの「プロファイルキーワードと値」](#)では、プロファイルキーワードと値を説明しています。

例 3-2 リモートファイルシステムのマウントとパッケージの追加および削除

```
# profile keywords      profile values
# -----
install_type          initial_install
system_type           standalone
partitioning          default
```

例3-2 リモートファイルシステムのマウントとパッケージの追加および削除 (続き)

```

filesys          any 512 swap # specify size of /swap
cluster         SUNWCprog
package         SUNWman delete
cluster         SUNWCacc
    
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

- | | |
|--------------|---|
| install_type | install_type キーワードは、すべてのプロフィールに必要です。 |
| system_type | system_type キーワードは、システムをスタンドアロンシステムとしてインストールするように定義します。 |
| partitioning | ファイルシステムスライスは、インストールするソフトウェアごとに default 値を使用して決定されます。swap は 512M バイトに設定され、値 any に基づいてすべてのディスクにインストールされます。 |
| cluster | 「開発者システムサポートソフトウェアグループ」(SUNWCprog) がシステムにインストールされます。 |
| package | 標準のマニュアルページがネットワークのファイルサーバー (s_ref) からマウントされる場合は、マニュアルページパッケージはシステムにインストールされません。System Accounting ユーティリティーが入ったパッケージはシステムにインストールされません。 |

例3-3 リモートファイルシステムのマウントと Sun 以外のパッケージの追加

```

# profile keywords      profile values
# -----
install_type          initial_install
system_type          standalone
partitioning         default
filesys              any 512 swap # specify size of /swap
cluster             SUNWCprog
cluster             SUNWCacc
package             apache_server \
                   http://package.central/packages/apache timeout 5
    
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

- | | |
|--------------|--------------------------------------|
| install_type | install_type キーワードは、すべてのプロフィールに必要です。 |
|--------------|--------------------------------------|

例 3-3 リモートファイルシステムのマウントと Sun 以外のパッケージの追加 (続き)

system_type	system_type キーワードは、システムをスタンドアロンシステムとしてインストールするように定義します。
partitioning	ファイルシステムスライスは、インストールするソフトウェアごとに default 値を使用して決定されます。swap は 512M バイトに設定され、値 any に基づいてすべてのディスクにインストールされます。
cluster	「開発者システムサポートソフトウェアグループ」(SUNWCprog) がシステムにインストールされます。
package	HTTP サーバー上に置かれた Sun 以外のパッケージがシステムにインストールされます。

例 3-4 ファイルシステムのインストール場所の指定

```
# profile keywords      profile values
# -----
install_type          initial_install
system_type           standalone
partitioning          explicit
filesys               c0t0d0s0 auto /
filesys               c0t3d0s1 auto swap
filesys               any auto usr
cluster               SUNWCall
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

partitioning	partitioning の値が explicit であるため、ファイルシステムスライスは、filesys キーワードによって指定します。ルート (/) のサイズは、値 auto により選択したソフトウェアに基づいて自動的に設定されて c0t0d0s0 にインストールされます。swap も同様に、必要なサイズに設定された上で c0t3d0s1 上にインストールされます。usr は、値 any により選択したソフトウェアに基づきインストールプログラムによってインストール先が決定されます。
cluster	全体ディストリビューションソフトウェアグループ (SUNWCall) がシステムにインストールされます。

例 3-5 パッチのアップグレードとインストール

```
# profile keywords      profile values
# -----
```

例3-5 パッチのアップグレードとインストール (続き)

```

install_type      upgrade
root_device       c0t3d0s2
backup_media      remote_filesystem timber:/export/scratch
package           SUNWbcp delete
package           SUNWxwman add
cluster           SUNWCacc add
patch             patch_list nfs://patch_master/Solaris_10/patches \
                  retry 5
locale            de
    
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

- | | |
|--------------|---|
| install_type | このプロファイルは、ディスク容量を再配置することによってシステムをアップグレードします。この例では、システム上のファイルシステムのいくつかにアップグレード用の容量が十分でないため、ディスク領域を再配置する必要があります。 |
| root_device | c0t3d0s2 のルートファイルシステムがアップグレードされます。 |
| backup_media | リモートシステム timber が、ディスク容量の再配置中のデータのバックアップに使用されます。バックアップメディアのキーワード値については、127 ページの「 backup_media プロファイルキーワード 」を参照してください。 |
| package | バイナリ互換パッケージ (SUNWbcp) は、アップグレード後、システムにインストールされません。 |
| package | このコードは、X Window System のマニュアルページと、System Accounting ユーティリティがまだシステムにインストールされていない場合に、インストールされるようにするものです。すでにシステム上にあるすべてのパッケージが自動的にアップグレードされます。 |
| patch | アップグレードでインストールされるパッチのリストです。パッチリストは、patch_master という名前の NFS サーバーの Solaris_10/patches ディレクトリに格納されています。マウントに失敗した場合、NFS マウントが5回試行されます。 |
| locale | ドイツ語対応パッケージがシステムにインストールされます。 |

例3-6 アップグレード用ディスク容量の再配置

```

# profile keywords      profile values
# -----
install_type           upgrade
    
```

例 3-6 アップグレード用ディスク容量の再配置 (続き)

```

root_device          c0t3d0s2
backup_media         remote_filesystem timber:/export/scratch
layout_constraint    c0t3d0s2 changeable 100
layout_constraint    c0t3d0s4 changeable
layout_constraint    c0t3d0s5 movable
package              SUNWbcp delete
package              SUNWxwman add
cluster              SUNWCacc add
locale               de

```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

install_type	このプロファイルは、ディスク容量を再配置することによってシステムをアップグレードします。この例では、システム上のファイルシステムのいくつかにアップグレード用の容量が十分でないため、ディスク領域を再配置する必要があります。
root_device	c0t3d0s2 のルートファイルシステムがアップグレードされます。
backup_media	リモートシステム timber が、ディスク容量の再配置中のデータのバックアップに使用されます。バックアップメディアのキーワード値については、 127 ページの「backup_media プロファイルキーワード」 を参照してください。
layout_constraint	<p>layout_constraint キーワードは、アップグレードのためにディスク領域の再配置を行う時に自動レイアウトが次のことを実施できることを示します。</p> <ul style="list-style-type: none"> ■ スライス 2 と 4 を変更する。これらのスライスはほかの場所へ移動が可能であり、サイズも変更できます。 ■ スライス 5 を移動する。スライスをほかの場所へ移動することはできますが、そのサイズを変更することはできません。
package	バイナリ互換パッケージ (SUNWbcp) は、アップグレード後、システムにインストールされません。
package	このコードは、X Window System のマニュアルページと、System Accounting ユーティリティーがまだシステムにインストールされていない場合に、インストールされるようになるものです。すでにシステム上にあるすべてのパッケージが自動的にアップグレードされます。

例 3-6 アップグレード用ディスク容量の再配置 (続き)

locale ドイツ語対応パッケージがシステムにインストールされま
す。

例 3-7 HTTP サーバーから Solaris フラッシュアーカイブを取得する

次の例のプロファイルは、カスタム JumpStart プログラムが HTTP サーバーから Solaris フラッシュアーカイブを取得するように指定しています。

```
# profile keywords           profile values
# -----
install_type                flash_install
archive_location            http://192.168.255.255/flasharchive/solarisarchive
partitioning                explicit
filesys                     c0t1d0s0 4000 /
filesys                     c0t1d0s1 512 swap
filesys                     c0t1d0s7 free /export/home
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

install_type このプロファイルにより、Solaris フラッシュアーカイブがク
ローンシステムにインストールされます。すべてのファイル
は、初期インストールにより上書きされます。

archive_location Solaris フラッシュアーカイブは、HTTP サーバーから取得され
ます。

partitioning partitioning の値が explicit であるため、ファイルシステムスラ
イスは、filesys キーワードによって指定します。ルート (/) の
サイズは Solaris フラッシュアーカイブのサイズに基づいて割り
当てています。ルートファイルシステムは c0t1d0s0 にインス
トールされます。swap は、必要なサイズに設定された上で
c0t1d0s1 上にインストールされます。/export/home のサイズは
残りのディスク容量に基づいて決定されます。/export/home は
c0t1d0s7 にインストールされます。

例 3-8 HTTPS サーバーから Solaris フラッシュアーカイブを取得する

次の例のプロファイルは、カスタム JumpStart プログラムがセキュリティー保護され
た HTTP サーバーから Solaris フラッシュアーカイブを取得するように指定していま
す。

例 3-8 HTTPS サーバーから Solaris フラッシュアーカイブを取得する (続き)

```
# profile keywords      profile values
# -----
install_type           flash_install
archive_location       https://192.168.255.255/solarisupdate.flar
partitioning           explicit
filesystems            c0t1d0s0 4000 /
filesystems            c0t1d0s1 512 swap
filesystems            c0t1d0s7 free /export/home
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

<code>install_type</code>	このプロファイルにより、Solaris フラッシュアーカイブがクローンシステムにインストールされます。すべてのファイルは、初期インストールにより上書きされます。
<code>archive_location</code>	圧縮された Solaris フラッシュアーカイブが HTTPS サーバーから取得されます。
<code>partitioning</code>	<code>partitioning</code> の値が <code>explicit</code> であるため、ファイルシステムスライスは、 <code>filesystems</code> キーワードによって指定します。ルート (/) のサイズは Solaris フラッシュアーカイブのサイズに基づいて割り当てています。 <code>swap</code> は、必要なサイズに設定された上で <code>c0t1d0s1</code> 上にインストールされます。 <code>/export/home</code> のサイズは残りのディスク容量に基づいて決定されます。 <code>/export/home</code> は <code>c0t1d0s7</code> にインストールされます。

例 3-9 Solaris フラッシュアーカイブの取得と Sun 以外のパッケージのインストール

次の例のプロファイルは、カスタム JumpStart プログラムが HTTP サーバーから Solaris フラッシュアーカイブを取得するように指定しています。

```
# profile keywords      profile values
# -----
install_type           flash_install
archive_location       http://192.168.255.255/flasharchive/solarisarchive
partitioning           explicit
filesystems            c0t1d0s0 4000 /
filesystems            c0t1d0s1 512 swap
filesystems            c0t1d0s7 free /export/home
package                SUNWnew http://192.168.254.255/Solaris_10 timeout 5
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

例 3-9 Solaris フラッシュアーカイブの取得と Sun 以外のパッケージのインストール (続き)

<code>install_type</code>	このプロファイルにより、Solaris フラッシュアーカイブがクローンシステムにインストールされます。すべてのファイルは、初期インストールにより上書きされます。
<code>archive_location</code>	Solaris フラッシュアーカイブは、HTTP サーバーから取得されます。
<code>partitioning</code>	<code>partitioning</code> の値が <code>explicit</code> であるため、ファイルシステムスライスは、 <code>filesystem</code> キーワードによって指定します。ルート (<code>/</code>) のサイズは Solaris フラッシュアーカイブのサイズに基づいて割り当てています。ルートファイルシステムは <code>c0t1d0s0</code> にインストールされます。 <code>swap</code> は、必要なサイズに設定された上で <code>c0t1d0s1</code> 上にインストールされます。 <code>/export/home</code> のサイズは残りのディスク容量に基づいて決定されます。 <code>/export/home</code> は <code>c0t1d0s7</code> にインストールされます。
<code>package</code>	SUNWnew パッケージは HTTP サーバー 192.168.254.255 の <code>Solaris_10</code> ディレクトリから追加されます。

例 3-10 NFS サーバーから Solaris フラッシュ差分アーカイブを取得する

次の例のプロファイルは、カスタム JumpStart プログラムが NFS サーバーから Solaris フラッシュアーカイブを取得するように指定しています。`flash_update` キーワードは、これが差分アーカイブであることを示しています。差分アーカイブでは、2つのシステムイメージの相違部分のみがインストールされます。

```
# profile keywords      profile values
# -----
install_type           flash_update
archive_location       nfs installserver:/export/solaris/flasharchive \
                        /solarisdiffarchive
no_master_check
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

<code>install_type</code>	このプロファイルは、Solaris フラッシュ差分アーカイブをクローンシステムにインストールします。アーカイブで指定されたファイルだけがインストールされます。
<code>archive_location</code>	Solaris フラッシュアーカイブは、NFS サーバーから取得されます。
<code>no_master_check</code>	クローンシステムは、有効なシステムイメージであるかどうかチェックされません。有効なシステムイメージが、元のマス

例 3-10 NFS サーバーから Solaris フラッシュ差分アーカイブを取得する (続き)

ターシステムを使用して構築されているはずですが。

例 3-11 空のブート環境の作成

次の例のプロファイルは、カスタム JumpStart プログラムが空のブート環境を作成するように指定しています。空のブート環境には、ファイルシステムは含まれません。また、現在のブート環境からコピーされることもありません。あとで Solaris フラッシュアーカイブをこのブート環境に格納し、アクティブにすることができます。

```
# profile keywords      profile values
# -----
install_type           initial_install
system_type            standalone
partitioning           explicit
filesys                c0t0d0s0 auto /
filesys                c0t3d0s1 auto swap
filesys                any auto usr
cluster                SUNWCall
bootenv createbe       bename second_BE \
filesystem /:/dev/dsk/c0t1d0s0:ufs \
filesystem -:/dev/dsk/c0t1d0s0:swap \
filesystem /export:shared:ufs
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

partitioning	partitioning の値が explicit であるため、ファイルシステムスライスは、filesys キーワードによって指定します。ルート (/) のサイズは、値 auto により選択したソフトウェアに基づいて自動的に設定されて c0t0d0s0 にインストールされます。swap も同様に、必要なサイズに設定された上で c0t3d0s1 上にインストールされます。usr は、値 any により選択したソフトウェアに基づきインストールプログラムによってインストール先が決定されます。
cluster	全体ディストリビューションソフトウェアグループ (SUNWCall) がシステムにインストールされます。
bootenv createbe	空の非アクティブブート環境がディスク c0t1d0 上に設定されません。ルート (/)、スワップ、および /export のファイルシステムがそれぞれ作成されますが、空のままです。この 2 番目のブート環境には、あとで Solaris フラッシュアーカイブをインストー

例 3-11 空のブート環境の作成 (続き)

ルできます。その後、この新しいブート環境をアクティブにして、現在のブート環境と交代させることができます。

キーワード値とその基本的な使用方法については、次の関連情報を参照してください。

- キーワード値については、119 ページの「プロファイルキーワードと値」を参照してください。
- Solaris Live Upgrade を使用した非アクティブなブート環境の作成、アップグレード、およびアクティブ化に関する背景情報については、『Solaris 10 5/08 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の第 2 章「Solaris Live Upgrade (概要)」を参照してください。
- Solaris フラッシュアーカイブの使用に関する背景情報については、『Solaris 10 5/08 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』の第 1 章「Solaris フラッシュ (概要)」を参照してください。

例 3-12 Solaris フラッシュアーカイブインストール時の RAID-1 ボリュームの作成

次の例のプロファイルには、カスタム JumpStart プログラムが Solaris ボリュームマネージャーテクノロジーを使用してルート (/)、swap、/usr、および /export/home ファイルシステムの RAID-1 ボリューム (ミラー) を作成することが示されています。Solaris フラッシュアーカイブはブート環境にインストールされます。

```
# profile keywords      profile values
# -----
install_type           flash_install
arhcrive_location      nfs server:/export/home/export/flash.s10.SUNWCall
partitioning           explicit
fileSYS                mirror:d10 c0t0d0s0 c0t1d0s0 4096 /
fileSYS                mirror c0t0d0s1 2048 swap
fileSYS                mirror:d30 c0t0d0s3 c0t1d0s3 4096 /usr
fileSYS                mirror:d40 c0t0d0s4 c0t1d0s4 4096 /usr
fileSYS                mirror:d50 c0t0d0s5 c0t1d0s5 free /export/home
metadb                 c0t1d0s7 size 8192 count 3
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

install_type このプロファイルにより、Solaris フラッシュアーカイブがクローンシステムにインストールされます。すべてのファイルは、初期インストールにより上書きされます。

例 3-12 Solaris フラッシュアーカイブインストール時の RAID-1 ボリュームの作成 (続き)

archive_location	Solaris フラッシュアーカイブは、NFS サーバーから取得されません。
partitioning	partitioning の値が explicit であるため、ファイルシステムスライスは、filesys キーワードによって指定します。
filesys	ルート (/) ファイルシステムが作成され、スライス c0t0d0s0 および c0t1d0s0 にミラー化されます。ルート (/) ファイルシステムのサイズは 4096M バイトに設定されます。c0t0d0s0 および c0t1d0s0 をミラー化する RAID-1 ボリュームに d10 という名前が付けられます。
filesys	swap ファイルシステムが作成され、スライス c0t0d0s1 にミラー化されます。サイズは 2048M バイトです。カスタム JumpStart プログラムにより、ミラーに名前が割り当てられます。
filesys	/usr ファイルシステムが作成され、スライス c0t1d0s3 および c0t0d0s3 にミラー化されます。/usr ファイルシステムのサイズは 4096M バイトに設定されます。RAID-1 ボリュームに d30 という名前が付けられます。
filesys	/usr ファイルシステムが作成され、スライス c0t1d0s4 および c0t0d0s4 にミラー化されます。/usr ファイルシステムのサイズは 4096M バイトに設定されます。RAID-1 ボリュームに d40 という名前が付けられます。
metadb	3つの状態データベースの複製 (metadb) がスライス c0t1d0s7 にインストールされます。サイズは 8192 ブロック (4M バイト) です。

- ミラー化されたファイルシステムをインストール時に作成する方法の概要については、『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の第 8 章「インストール時の RAID-1 ボリューム (ミラー) の作成 (概要)」を参照してください。
- ミラー化されたファイルシステムの作成のガイドラインと要件については、『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の第 9 章「インストール時の RAID-1 ボリューム (ミラー) の作成 (計画)」を参照してください。
- キーワード値については、142 ページの「filesys プロファイルキーワード (RAID-1 ボリュームの作成)」と 148 ページの「metadb プロファイルキーワード (状態データベースの複製の作成)」を参照してください。

例 3-13 ルートファイルシステムをミラー化するための RAID-1 ボリュームの作成

次の例のプロファイルには、カスタム JumpStart プログラムが Solaris ボリュームマネージャテクノロジーを使用してルート (/) ファイルシステムの RAID-1 ボリューム (ミラー) を作成することが示されています。

```
# profile keywords      profile values
# -----
install_type          initial_install
cluster               SUNWCXall
fileys                 mirror:d30 c0t1d0s0 c0t0d0s0 /
fileys                 c0t0d0s3 512 swap
metadb                c0t0d0s4 size 8192 count 4
metadb                c0t1d0s4 size 8192 count 4
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

- cluster 「全体ディストリビューション+OEM」ソフトウェアグループ (SUNWCXall) がシステムにインストールされます。
- fileys ルート (/) ファイルシステムが作成され、スライス c0t1d0s0 および c0t0d0s0 にミラー化されます。c0t1d0s0 および c0t0d0s0 をミラー化する RAID-1 ボリュームに d30 という名前が付けられます。カスタム JumpStart プログラムにより、2つのサブミラーに名前が割り当てられます。
- fileys swap ファイルシステムが作成され、スライス c0t0d0s3 にミラー化されま
す。サイズは 512M バイトです。
- metadb 4つの状態データベースの複製 (metadb) がスライス c0t0d0s4 にインス
トールされます。サイズは 8192 ブロック (4M バイト) です。
- metadb 4つの状態データベースの複製 (metadb) がスライス c0t1d0s4 にインス
トールされます。サイズは 8192 ブロック (4M バイト) です。

- インストール中に RAID-1 ボリュームを作成する方法の概要については、
『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の
第 8 章「インストール時の RAID-1 ボリューム (ミラー) の作成 (概要)」を参照して
ください。
- RAID-1 ボリュームの作成のガイドラインと要件については、『Solaris 10 5/08 イン
ストールガイド (インストールとアップグレードの計画)』の第 9 章「インストー
ル時の RAID-1 ボリューム (ミラー) の作成 (計画)」を参照してください。
- キーワード値については、142 ページの「fileys プロファイルキーワード
(RAID-1 ボリュームの作成)」と 148 ページの「metadb プロファイルキーワード (状
態データベースの複製の作成)」を参照してください。

例 3-14 複数のファイルシステムをミラー化するための RAID-1 ボリュームの作成

次の例のプロファイルには、カスタム JumpStart プログラムが Solaris ボリュームマネージャーテクノロジーを使用して、ルート (/)、swap、および /usr ファイルシステムの RAID-1 ボリューム(ミラー)を作成することが示されています。

```
# profile keywords      profile values
# -----
install_type           initial_install
cluster                SUNWCXall
filesystems            mirror:d100 c0t1d0s0 c0t0d0s0 200 /
filesystems            c0t1d0s5 500 /var
filesystems            c0t0d0s5 500
filesystems            mirror c0t0d0s1 512 swap
metadb                 c0t0d0s3 size 8192 count 5
filesystems            mirror c0t1d0s4 c0t0d0s4 2000 /usr
filesystems            c0t1d0s7 free /export/home
filesystems            c0t0d0s7 free
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

- cluster 「全体ディストリビューション+OEM」ソフトウェアグループ (SUNWCXall) がシステムにインストールされます。
- filesystems ルート (/) ファイルシステムが作成され、スライス c0t1d0s0 および c0t0d0s0 にミラー化されます。ルート (/) ファイルシステムのサイズは 200M バイトです。c0t1d0s0 および c0t0d0s0 をミラー化する RAID-1 ボリュームに d100 という名前が付けられます。
- filesystems /var ファイルシステムがスライス c0t1d0s5 にインストールされます。サイズは 500M バイトです。ルート (/) ファイルシステムが作成され、スライス c0t1d0s0 および c0t0d0s0 にミラー化されます。ルート (/) ファイルシステムのサイズは 200M バイトです。c0t1d0s0 および c0t0d0s0 をミラー化する RAID-1 ボリュームに d100 という名前が付けられます。
- filesystems swap ファイルシステムが作成され、スライス c0t0d0s1 にミラー化されます。サイズは 512M バイトです。カスタム JumpStart プログラムにより、ミラーに名前が割り当てられます。
- metadb 5つの状態データベースの複製 (metadb) がスライス c0t0d0s3 にインストールされます。サイズは 8192 ブロック (4M バイト) です。
- filesystems /usr ファイルシステムが作成され、スライス c0t1d0s4 および c0t0d0s4 にミラー化されます。/usr ファイルシステムのサイズは 2000M バイトになります。カスタム JumpStart プログラムにより、ミラーに名前が割り当てられます。

例 3-14 複数のファイルシステムをミラー化するための RAID-1 ボリュームの作成 (続き)

- ミラー化されたファイルシステムをインストール時に作成する方法の概要については、『Solaris 10 5/08 インストールガイド(インストールとアップグレードの計画)』の第 8 章「インストール時の RAID-1 ボリューム(ミラー)の作成(概要)」を参照してください。
- ミラー化されたファイルシステムの作成のガイドラインと要件については、『Solaris 10 5/08 インストールガイド(インストールとアップグレードの計画)』の第 9 章「インストール時の RAID-1 ボリューム(ミラー)の作成(計画)」を参照してください。
- キーワード値については、142 ページの「[filesys プロファイルキーワード \(RAID-1 ボリュームの作成\)](#)」と 148 ページの「[metadb プロファイルキーワード \(状態データベースの複製の作成\)](#)」を参照してください。

例 3-15 x86: fdisk キーワードの使用方法

```
# profile keywords      profile values
# -----
install_type           initial_install
system_type            standalone

fdisk                  c0t0d0 0x04 delete
fdisk                  c0t0d0 solaris maxfree
cluster                SUNWCall
cluster                SUNWCacc delete
```

この例で使用されているキーワードと値の一部について、次のリストで説明します。

- | | |
|---------|--|
| fdisk | タイプ DOSOS16 (04 16 進) の fdisk パーティションはすべて、c0t0d0 ディスクから削除されます。 |
| fdisk | Solaris fdisk パーティションが、c0t0d0 ディスク上の最大の連続空き領域に作成されます。 |
| cluster | 全体ディストリビューションソフトウェアグループ (SUNWCall) がシステムにインストールされます。 |
| cluster | システムアカウントユーティリティ (SUNWCacc) は、システムにインストールされません。 |

プロファイルのテスト

プロファイルの作成後、`pfinstall(1M)` コマンドを使ってプロファイル进行测试します。プロファイルのテストは、実際にプロファイルを使用してシステムのインストールまたはアップグレードを行う前に行なってください。プロファイルのテストは、ディスク容量を再配置するアップグレードプロファイルを作成する場合、特に有用です。

`pfinstall` から生成されるインストール出力を調べることによって、意図どおりにプロファイルが動作するかを簡単にチェックできます。たとえば、実際にシステムでアップグレードを行う前に、そのシステムが Solaris の新しいリリースにアップグレードするための十分なディスク容量を持っているかどうかをプロファイルで調べることができます。

`pfinstall` を利用すると、次の要素に対してプロファイル进行测试できます。

- `pfinstall` を実行しているシステムのディスク構成。
- その他のディスク構成。ディスクの構造 (ディスクのバイト/セクタ、フラグ、スライスなど) を示すディスク構成ファイルを使用します。ディスク構成ファイルの作成方法については、74 ページの「ディスク構成ファイルの作成」と76 ページの「x86: ディスク構成ファイルを作成する方法」を参照してください。

注- ディスク構成ファイルを使用して、システムのアップグレードに使用するプロファイル进行测试することはできません。代わりに、システムの実際のディスク構成およびシステムに現在インストールされているソフトウェアに対して、プロファイル进行测试する必要があります。

▼ プロファイル进行测试するために、一時的な Solaris 環境を作成する方法

特定の Solaris リリース用のプロファイルを正常かつ正確にテストするには、同じリリースの Solaris 環境内でテストする必要があります。たとえば、Solaris 初期インストールプロファイル进行测试する場合は、Solaris OS を実行しているシステム上で `pfinstall` コマンドを実行します。

次のどちらかの状況でプロファイル进行测试する場合は、一時的なインストール環境を作成する必要があります。

- 以前のバージョンの Solaris ソフトウェアを使用しているシステム上で Solaris 10 5/08 のアップグレードプロファイル进行测试する場合。
- Solaris 10 5/08 の初期インストールプロファイル进行测试するための Solaris 10 5/08 システムをまだインストールしていない場合。

- 1 次に示すメディアのいずれか1つのイメージからシステムをブートします。

SPARC システムの場合:

- Solaris Operating System DVD (SPARC 版)
- Solaris SOFTWARE - 1 CD (SPARC 版)

x86 システムの場合:

- Solaris Operating System DVD (x86 版)
- Solaris SOFTWARE - 1 CD (x86 版)

注-アップグレードプロファイルをテストする場合は、アップグレードするシステムをブートしてください。

- 2 システムを識別するための質問に答えます。
- 3 インストールプログラムを終了するには、!を入力します。次のプロンプトに続いて入力します。

```
The Solaris installation program will assist you in installing software for Solaris.  
<Press ENTER to continue> {"!" exits}
```

- 4 このシェルから `pfinstall` コマンドを実行します。 `pfinstall` コマンドの詳細な使用方法については、[56 ページの「プロファイルをテストする方法」](#)の[手順7](#)を参照してください。

▼ プロファイルをテストする方法

x86 のみ - `locale` キーワードを使用している場合は、`pfinstall -D` コマンドでプロファイルをテストすると失敗します。回避方法については、[181 ページの「Solaris OS のアップグレード」](#) セクションのエラーメッセージ「`could not select locale`」を参照してください。

- 1 プロファイルをテストするためのシステムを決定します。プロファイルの作成時に意図したシステムと同じプラットフォームタイプ (SPARC または x86) のシステムを選んでください。
アップグレードプロファイルをテストする場合は、実際にアップグレードするシステム上でテストしてください。
- 2 次の表に基づいて、次に行うことを決定します。

状態	参照先
初期インストールプロファイル进行测试します (Solaris 10 5/08 ソフトウェアが動作しているシステムがある場合)。	システムのスーパーユーザーになり、 手順5 へ進みます。
アップグレードプロファイル进行测试します (初期インストールプロファイル进行测试するための Solaris 10 5/08 が動作しているシステムがない場合)。	一時的な Solaris 10 5/08 環境を作成し、プロファイル进行测试します。詳細については、 55 ページの「プロファイル进行测试するために、一時的な Solaris 環境を作成する方法」 を参照してください。続いて、 手順3 へ進みます。

3 一時的なマウントポイントを作成します。

```
# mkdir /tmp/mnt
```

4 テストするプロファイル(1つ以上)が入ったディレクトリをマウントします。

マウントするメディア	コマンド
リモート NFS ファイルシステムをマウントします (ネットワーク上のシステムの場合)。	<code>mount -F nfs server_name:path /tmp/mnt</code>
SPARC: UFS フォーマットのフロッピーディスクをマウントします。	<code>mount -F ufs /dev/diskette /tmp/mnt</code>
PCFS フォーマットのフロッピーディスクをマウントします。	<code>mount -F pcfs /dev/diskette /tmp/mnt</code>

5 特定のシステムメモリーサイズでプロファイル进行测试するには、SYS_MEMSIZE にそのメモリーサイズを M バイト単位で指定します。

```
# SYS_MEMSIZE=memory_size
# export SYS_MEMSIZE
```

6 [手順4](#) でディレクトリをマウントしましたか。

- マウントした場合は、/tmp/mnt にディレクトリを変更します。

```
# cd /tmp/mnt
```

- マウントしなかった場合は、プロファイルのある場所にディレクトリを変更します。通常は、JumpStart ディレクトリです。

```
# cd jumpstart_dir_path
```

7 pfinstall(1M) コマンドによってプロファイル进行测试します。

```
# /usr/sbin/install.d/pfinstall -D:-d disk_config_file -c path profile
```



注意 - `-d` または `-D` オプションを指定してください。いずれかを指定しないと `pfinstall` は指定されたプロファイルを使用して Solaris ソフトウェアをインストールします。その結果、システム上のデータがすべて上書きされます。

`-D` `pfinstall` は、現在のシステムのディスク構成を使用してプロファイル进行测试します。アップグレードプロファイル进行测试する場合は、`-D` オプションを使用する必要があります。

`-d disk_config_file` `pfinstall` は、ディスク構成ファイル `disk_config_file` を使用してプロファイル进行测试します。`pfinstall` が動作しているディレクトリに `disk_config_file` が置かれていない場合は、パスを指定する必要があります。

ディスク構成ファイルの作成方法については、[74 ページ](#) の「ディスク構成ファイルの作成」を参照してください。

注 - アップグレードプロファイル `install_type upgrade` には、`-d disk_config_file` オプションを使用できません。常に、システムのディスク構成に対してアップグレードプロファイル进行测试する必要があります(つまり、`-D` オプションを使用する必要があります)。

`-c path` Solaris ソフトウェアイメージのパス。このオプションは、たとえば、システムでボリュームマネージャーを使用して Solaris SOFTWARE - 1 CD を各自のプラットフォームにマウントする場合に使用します。

注 - プラットフォーム対応の Solaris Operating System DVD または Solaris SOFTWARE - 1 CD イメージからブートした場合は、`-c` オプションは不要です。DVD または CD イメージは、ブート処理の一環として `/cdrom` 上にマウントされます。

`profile` テストするプロファイル名。`profile` が `pfinstall` が実行されているディレクトリにない場合は、パスを指定する必要があります。

プロファイルテストの例

次の例は、`pfinstall` を使用して `basic_prof` というプロファイル进行测试する方法を示しています。このプロファイルテストは、Solaris 10 5/08 ソフトウェアがインストールされているシステム上のディスク構成に対して行われます。`basic_prof` プロファイルは、`/jumpstart` ディレクトリにあります。ボリュームマネージャーを使用しているため、Solaris Operating System DVD イメージへのパスが指定されています。

例 3-16 Solaris 10 5/08 システムを使用したプロファイルテスト

```
# cd /jumpstart
# /usr/sbin/install.d/pfinstall -D -c /cdrom/pathname basic_prof
```

次の例は、`pfinstall` を使用して Solaris 10 5/08 システム上で `basic_prof` というプロファイル进行测试する方法を示しています。このテストは、`535_test` ディスク構成ファイルに対して行われます。このテストは、システムメモリーが 64M バイトあるか確認します。この例では、`/export/install` ディレクトリにある Solaris SOFTWARE - 1 CD (SPARC 版) または Solaris SOFTWARE - 1 CD (x86 版) のイメージを使用しています。

例 3-17 ディスク構成ファイルによるプロファイルテスト

```
# SYS_MEMSIZE=64
# export SYS_MEMSIZE
# /usr/sbin/install.d/pfinstall -d 535_test -c /export/install basic_prof
```

rules ファイルの妥当性を検査する

`rules` ファイルとプロファイルは、`check` スクリプトを実行して、これらのファイルが正しく設定されていることを確認してからでないと使用できません。すべてのルールとプロファイルが正しく設定されている場合は、`rules.ok` ファイルが作成されます。このファイルは、カスタム JumpStart インストールソフトウェアがシステムをプロファイルと照合するために必要です。

表 3-2 に、`check` スクリプトの実行内容を示します。

表 3-2 check スクリプトの動作

手順	説明
1	rules ファイルの構文検査を行います。 check は、ルールキーワードが正当かどうか、および各ルールに対して <i>begin</i> フィールド、 <i>class</i> フィールド、および <i>finish</i> フィールドが指定されているかどうかを確認します。 <i>begin</i> フィールドおよび <i>finish</i> フィールドには、ファイル名でなくマイナス記号 (-) も指定可能です。
2	rules ファイルにエラーがなければ、ルールに指定された各プロファイルの構文検査が行われます。
3	エラーが見つからなければ、check は、rules ファイルから rules.ok ファイルを生成します。コメントと空行をすべて削除し、すべてのルールを保持し、さらに、末尾に次のコメント行を追加します。 # version=2 checksum=num

▼ rules ファイルの妥当性を検査する方法

- 1 check スクリプトが **JumpStart** ディレクトリにあることを確認します。

注 - check スクリプトは、Solaris Operating System DVD または Solaris SOFTWARE - 1 CD の Solaris_10/Misc/jumpstart_sample ディレクトリにあります。

- 2 **JumpStart** ディレクトリに移動します。
- 3 check スクリプトを実行して、rules ファイルの妥当性検査を行います。

```
$ ./check -p path -r file_name
```

-p path 使用しているシステムの check スクリプトではなく Solaris ソフトウェアイメージの check スクリプトを使用して、rules を検証します。*path* は、ローカルディスク、マウントされた Solaris Operating System DVD、または Solaris SOFTWARE - 1 CD 上のイメージです。

以前のバージョンの Solaris を実行しているシステムでは、このオプションを使って、check の最新バージョンを実行します。

-r file_name 名前が rules 以外の rules ファイル名を指定します。このオプションを使用すると、rules ファイルに組み込む前にルールの妥当性を検査できます。

check スクリプトを実行すると、rules ファイルと各プロファイルの妥当性検査の結果が報告されます。エラーが発生しなかった場合、次の情報が報告されます。

The custom JumpStart configuration is ok

- 4 rules.ok ファイルの所有者が root になっていて、アクセス権が 644 に設定されていることを確認します。

参照 rules ファイルの妥当性を検査したあとの、カスタム JumpStart オプション機能については、[第4章カスタム JumpStart オプション機能の使用 \(作業\)](#)を参照してください。カスタム JumpStart インストールの詳しい実行方法については、[第6章カスタム JumpStart インストールの実行 \(作業\)](#)を参照してください。

カスタム JumpStart オプション機能の使用(作業)

この章では、カスタム JumpStart インストールのオプション機能について説明します。

- 63 ページの「開始スクリプトの作成」
- 65 ページの「終了スクリプトの作成」
- 72 ページの「圧縮された構成ファイルの作成」
- 74 ページの「ディスク構成ファイルの作成」
- 79 ページの「サイト固有のインストールプログラムの使用」

注 - この章の手順は、カスタム JumpStart ファイルの提供に使用される SPARC サーバーと x86 サーバーの両方(「プロファイルサーバー」と呼ばれる)に利用できません。プロファイルサーバーは、異なる種類のプラットフォームのシステムにカスタム JumpStart ファイルを提供できません。たとえば、SPARC サーバーは、SPARC ベースのシステムと x86 ベースのシステムの両方に、カスタム JumpStart ファイルを提供できません。

開始スクリプトの作成

開始スクリプトは、rules ファイル内に指定する、ユーザー定義の Bourne シェルスクリプトです。Solaris ソフトウェアがシステムにインストールされる前に作業を実行します。開始スクリプトは、カスタム JumpStart インストールで Solaris をインストールするときのみ使用できます。

開始スクリプトは、次のような場合に使用してください。

- 派生プロファイルを作成する
- アップグレード前にファイルのバックアップをとる

開始スクリプトに関する注意事項

- スクリプト内には、初期インストール時やアップグレードインストール時に /a に対して行われるファイルシステムのマウントを妨げるような記述をしないでください。JumpStart プロファイルが /a にファイルシステムをマウントできない場合は、エラーが発生してインストールは失敗します。
- インストール中の開始スクリプトの出力は、/tmp/begin.log へ書き込まれます。インストール完了後に、このログファイルの出力先が /var/sadm/system/logs/begin.log へ変更されます。
- 開始スクリプトの所有者が root になっていて、アクセス権が 644 に設定されていることを確認します。
- 開始スクリプト内では、カスタム JumpStart 環境変数を使用できます。環境変数のリストについては、164 ページの「カスタム JumpStart の環境変数」を参照してください。
- 開始スクリプトは、JumpStart ディレクトリに保存してください。

注 - Solaris 10 リリースでは、JumpStart インストール時にプロンプトが表示されないようにするための、サンプルの JumpStart スクリプト `set_nfs4_domain` が、メディアで提供されていました。このスクリプトは、インストール時に NFSv4 プロンプトが表示されないようにしていました。このスクリプトはもう必要ありません。Solaris 10 8/07 リリース以降では、`sysidcfg` のキーワード `nfs4_domain` を使ってプロンプトが表示されないようにします。`set_nfs4_domain` スクリプトには、プロンプトを抑制する機能がなくなりました。

非大域ゾーンがインストールされている場合に新しい `nfs4_domain` キーワードが `sysidcfg` ファイルに含まれていると、非大域ゾーンを最初にブートしたときにドメインが設定されます。それ以外の場合は、Solaris 対話式インストールプログラムが起動して、ブートプロセスが完了する前にドメイン名を入力するように促すプロンプトが表示されます。

『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の「`nfs4_domain` キーワード」を参照してください。

開始スクリプトによる派生プロファイルの作成

カスタム JumpStart インストール時に開始スクリプトが動的に作成するプロファイルを「派生プロファイル」と呼びます。派生プロファイルは、特定のシステムを1つのプロファイルに対応させる `rules` ファイルを設定できない場合に必要となります。たとえば、異なるハードウェア構成要素を持つ同一のシステムモジュール (異なるフレームバッファを持つシステムなど) には、派生プロファイルを使用する必要があります。

派生プロファイルを使用するルールの設定手順は次のとおりです。

- プロファイルフィールドに、プロファイルの代わりに等号(=)を指定します。
- インストール対象のシステムに依存する派生プロファイルを作成する開始スクリプトを `begin` フィールドに指定します。

システムが、プロファイルフィールドに等号(=)があるルールと一致すると、開始スクリプトは、システムに Solaris ソフトウェアをインストールするのに使用する派生プロファイルを作成します。

次に、毎回同じ派生プロファイルを作成する開始スクリプトの例を示します。もちろん、ルールの評価に応じて異なる派生プロファイルを生成する開始スクリプトも作成できます。

例 4-1 派生プロファイルを作成する開始スクリプト

```
#!/bin/sh
echo "install_type      initial_install"    > ${SI_PROFILE}
echo "system_type      standalone"      >> ${SI_PROFILE}
echo "partitioning     default"         >> ${SI_PROFILE}
echo "cluster          SUNWCprog"       >> ${SI_PROFILE}
echo "package          SUNWman    delete" >> ${SI_PROFILE}
echo "package          SUNWolman  delete" >> ${SI_PROFILE}
echo "package          SUNWxwman  delete" >> ${SI_PROFILE}
```

この例の開始スクリプトは、派生プロファイル名に環境変数 `SI_PROFILE` を使用する必要があります。この派生プロファイル名は、デフォルトでは `/tmp/install.input` に設定されています。

注-開始スクリプトを使って派生プロファイルを作成する場合は、スクリプトにエラーがないことを確認してください。派生プロファイルは、開始スクリプトが実行されてから作成されます。このため、`check` スクリプトで派生プロファイルの妥当性は検査されません。

終了スクリプトの作成

終了スクリプトは、`rules` ファイル内に指定する、ユーザー定義の Bourne シェルスクリプトです。Solaris ソフトウェアがシステムにインストールされた後、システムがリポートする前に作業を実行します。終了スクリプトは、カスタム JumpStart インストールで Solaris をインストールするときのみ使用できます。

終了スクリプトは、次のような作業に利用できます。

- ファイルの追加
- パッケージまたはパッチの追加

- ルート環境のカスタマイズ
- システムのルートパスワードの設定
- 追加ソフトウェアのインストール

終了スクリプトに関する注意事項

- Solaris インストールプログラムは、システムのファイルシステムを /a にマウントします。ファイルシステムは、システムがリブートするまで /a にマウントされています。終了スクリプトを使用して、新しくインストールされたファイルシステムの階層 (/a) を変更し、ファイルの追加、変更、または削除ができます。
- インストール中の終了スクリプトの出力は、/tmp/finish.log へ書き込まれます。インストール完了後に、このログファイルの出力先が /var/sadm/system/logs/finish.log へ変更されます。
- 終了スクリプトの所有者が root で、アクセス権が 644 に設定されていることを確認します。
- 終了スクリプト内では、カスタム JumpStart 環境変数を使用できます。環境変数のリストについては、164 ページの「カスタム JumpStart の環境変数」を参照してください。
- 終了スクリプトは、JumpStart ディレクトリに保存してください。

▼ 終了スクリプトでファイルを追加する方法

終了スクリプトにより、JumpStart ディレクトリにあるファイルをインストールされたシステムへ追加できます。ファイルを追加できるのは、SI_CONFIG_DIR 変数で指定されるディレクトリに JumpStart ディレクトリがマウントされるためです。このディレクトリは、デフォルトで /tmp/install_config に設定されます。

注-インストールされたシステムにすでに存在するファイルに、JumpStart ディレクトリからファイルをコピーして、ファイルを置換することもできます。

- 1 インストール済みシステムに追加するすべてのファイルを JumpStart ディレクトリにコピーします。
- 2 新しくインストールされたファイルシステムの階層にコピーするファイルごとに、次の行を終了スクリプトに挿入します。

```
cp ${SI_CONFIG_DIR}/file_name /a/path_name
```

例 4-2 終了スクリプトでファイルを追加する

たとえば、自分のサイトのすべてのユーザーを対象に開発された特別なアプリケーション `site_prog` があると仮定します。`site_prog` のコピーを `JumpStart` ディレクトリに置いた場合、終了スクリプト内の次の行は `JumpStart` ディレクトリからシステムの `/usr/bin` ディレクトリに `site_prog` をコピーします。

```
cp ${SI_CONFIG_DIR}/site_prog /a/usr/bin
```

終了スクリプトによるパッケージとパッチの追加

終了スクリプトを作成すると、Solaris ソフトウェアをシステムにインストールした後に、パッケージやパッチを自動的に追加できます。終了スクリプトを使用してパッケージを追加すると時間を節約できるだけでなく、サイトのさまざまなシステムにパッケージおよびパッチをインストールする上での整合性を維持できます。

終了スクリプト内で `pkgadd(1M)` コマンドや `patchadd(1M)` コマンドを使用するときは、`-R` オプションを使って、`/a` をルートパスとして指定します。

- 例 4-3 に、パッケージを追加する終了スクリプトの例を示します。
- 例 4-4 に、パッチを追加する終了スクリプトの例を示します。

例 4-3 終了スクリプトでパッケージを追加する

```
#!/bin/sh

BASE=/a
MNT=/a/mnt
ADMIN_FILE=/a/tmp/admin

mkdir ${MNT}
mount -f nfs sherlock:/export/package ${MNT}
cat >${ADMIN_FILE} <<DONT_ASK
mail=root
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=ask
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
DONT_ASK
```

例4-3 終了スクリプトでパッケージを追加する (続き)

```
/usr/sbin/pkgadd -a ${ADMIN_FILE} -d ${MNT} -R ${BASE} SUNwxyz
umount ${MNT}
rmdir ${MNT}
```

次に、この例のいくつかのコマンドの説明を示します。

- 次のコマンドは、インストールするパッケージを含むサーバー上にディレクトリをマウントします。

```
mount -f nfs sherlock:/export/package ${MNT}
```

- 次のコマンドは、一時的なパッケージ管理ファイル `admin` を作成し、パッケージのインストール時に `pkgadd(1M)` コマンドがチェックも質問の表示も行わないようにします。この一時的なパッケージ管理ファイルは、パッケージの追加時に自動インストールを実施するために使用してください。

```
cat >${ADMIN_FILE} <<DONT_ASK
```

- 次の `pkgadd` コマンドは、`-a` オプション(パッケージ管理ファイルを指定)と `-R` オプション(ルートパスを指定)を使用してパッケージを追加します。

```
/usr/sbin/pkgadd -a ${ADMIN_FILE} -d ${MNT} -R ${BASE} SUNwxyz
```

例4-4 終了スクリプトでパッチを追加する

```
#!/bin/sh

#####
#
# USER-CONFIGURABLE OPTIONS
#
#####

# The location of the patches to add to the system after it's installed.
# The OS rev (5.x) and the architecture ('mach') will be added to the
# root. For example, /foo on a 8 SPARC would turn into /foo/5.8/sparc
LUPATCHHOST=ins3525-svr
LUPATCHPATHROOT=/export/solaris/patchdb
#####
#
# NO USER-SERVICEABLE PARTS PAST THIS POINT
#
#####

BASEDIR=/a
```

例4-4 終了スクリプトでパッチを追加する (続き)

```
# Figure out the source and target OS versions
echo Determining OS revisions...
SRCREV='uname -r'
echo Source $SRCREV

LUPATCHPATH=$LUPATCHPATHROOT/$SRCREV/'mach'

#
# Add the patches needed
#
echo Adding OS patches
mount $LUPATCHHOST:$LUPATCHPATH /mnt >/dev/null 2>&1
if [ $? = 0 ] ; then
    for patch in `cat /mnt/*Recommended/patch_order` ; do
        (cd /mnt/*Recommended/$patch ; echo yes | patchadd -u -d -R $BASEDIR .)
    done
    cd /tmp
    umount /mnt
else
    echo "No patches found"
fi
```

注-以前は終了スクリプト環境では、`pkgadd` コマンドや `patchadd` コマンドとともに `chroot(1M)` コマンドが使用されていました。まれに、一部のパッケージやパッチで `-R` オプションが正しく動作しないことがあります。`chroot` コマンドを実行する前には、`/a` ルートパスにダミーの `/etc/mnttab` ファイルを作成する必要があります。

ダミーの `/etc/mnttab` ファイルを作成するには、終了スクリプトに次の行を追加します。

```
cp /etc/mnttab /a/etc/mnttab
```

終了スクリプトによるルート環境のカスタマイズ

終了スクリプトでは、すでにシステムにインストールされたファイルをカスタマイズすることもできます。たとえば、[例4-5](#)の終了スクリプトは、ルート(`/`)ディレクトリの `.cshrc` ファイルに情報を追加することによってルート環境をカスタマイズします。

例4-5 終了スクリプトによるルート環境のカスタマイズ

```
#!/bin/sh
#
# Customize root's environment
#
echo "***adding customizations in /.cshrc"
test -f a/.cshrc || {
cat >> a/.cshrc <<EOF
set history=100 savehist=200 filec ignoreeof prompt="\$user@'uname -n'> "
alias cp cp -i
alias mv mv -i
alias rm rm -i
alias ls ls -FC
alias h history
alias c clear
unset autologout
EOF
}
```

終了スクリプトによるシステムのルートパスワードの設定

Solaris ソフトウェアのシステムへのインストールが完了すると、システムは再起動します。ブートプロセス終了前に、システムは root パスワードを入力するように求めてきます。パスワードを入力するまで、システムはブート処理を終了できません。

set_root_pw という終了スクリプトが、auto_install_sample ディレクトリに保存されています。この終了スクリプトは、プロンプトを表示することなく root パスワードを自動的に設定する方法を示します。set_root_pw については、例 4-6 を参照してください。

注-システムの root パスワードを終了スクリプトで設定した場合、ユーザーが、終了スクリプト内にある暗号化されたパスワードからルートのパスワードを発見しようと試みる可能性があります。ユーザーに root パスワードを解読されないよう、対策を講じてください。

例4-6 終了スクリプトによるシステムの root パスワードの設定

```
#!/bin/sh
#
#      @(#)set_root_pw 1.4 93/12/23 SMI
#
```

例 4-6 終了スクリプトによるシステムの root パスワードの設定 (続き)

```

# This is an example Bourne shell script to be run after installation.
# It sets the system's root password to the entry defined in PASSWD.
# The encrypted password is obtained from an existing root password entry
# in /etc/shadow from an installed machine.

echo "setting password for root"

# set the root password
PASSWD=dK05IBkSF42lw
#create a temporary input file
cp /a/etc/shadow /a/etc/shadow.orig

mv /a/etc/shadow /a/etc/shadow.orig
nawk -F: '{
    if ( $1 == "root" )
        printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n",$1,passwd,$3,$4,$5,$6,$7,$8,$9
    else
        printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n",$1,$2,$3,$4,$5,$6,$7,$8,$9
    }' passwd="$PASSWD" /a/etc/shadow.orig > /a/etc/shadow
#remove the temporary file
rm -f /a/etc/shadow.orig
# set the flag so sysidroot won't prompt for the root password
sed -e 's/0 # root/1 # root/' ${SI_SYS_STATE} > /tmp/state.$$
mv /tmp/state.$$ ${SI_SYS_STATE}

```

次に、この例のいくつかのコマンドの説明を示します。

- 次のコマンドは、PASSWD 変数に、システムの /etc/shadow ファイルの既存のエントリから取得した暗号化された root パスワードを設定します。

```
#create a temporary input file
```

- 次のコマンドは、/a/etc/shadow の一時入力ファイルを作成します。

```
cp /a/etc/shadow /a/etc/shadow.orig
```

- 次のコマンドは、\$PASSWD をパスワードフィールドとして使用して、新しくインストールしたシステム用の /etc/shadow ファイルにある root エントリを変更します。

```
if ( $1 == "root" )
```

- 次のコマンドは、一時的な /a/etc/shadow ファイルを削除します。

```
rm -f /a/etc/shadow.orig
```

例 4-6 終了スクリプトによるシステムの root パスワードの設定 (続き)

- 次のコマンドは、状態ファイルのエントリを 0 から 1 へ変更します。これによりユーザーは root パスワードの入力を求められません。この状態ファイルには、SI_SYS_STATE 変数 (現在の値は /a/etc/.sysIDtool.state) を使用してアクセスします。この値の変更によってスクリプトで問題が発生することを防ぐには、必ず \$SI_SYS_STATE を使用してこのファイルを参照してください。ここに示されている sed コマンドでは、0 の後ろと 1 の後ろにタブ文字が入っています。

```
sed -e 's/0 # root/1 # root/' ${SI_SYS_STATE} > /tmp/state.$$
```

終了スクリプトによる非対話式インストール

終了スクリプトを使用すると、Solaris OS のインストールに続いてさらにソフトウェアをインストールできます。Solaris インストールプログラムは、インストール中に情報を入力するプロンプトを表示します。自動インストールを行うには、`-nodisplay` オプションまたは `-noconsole` オプションを指定して Solaris インストールプログラムを実行します。

表 4-1 Solaris インストールオプション

オプション	説明
<code>-nodisplay</code>	GUI を使用せずにインストーラを実行します。 <code>-locales</code> オプションを用いてインストール条件を変更した場合を除き、デフォルトの設定に従って製品がインストールされます。
<code>-noconsole</code>	対話形式のテキストコンソールデバイスを介さずにインストールを実行します。UNIX スクリプトで <code>-nodisplay</code> と共に使用すると便利です。

詳細については、`installer(1M)` のマニュアルページを参照してください。

圧縮された構成ファイルの作成

`add_install_client` コマンドを使用してカスタム JumpStart 構成ファイルの場所を指定する代わりに、システムのブート時に構成ファイルの場所を指定することができます。ただし、指定できるファイル名は 1 つだけです。したがって、すべてのカスタム JumpStart 構成ファイルを 1 つのファイルに圧縮する必要があります。

- **SPARC** システムの場合、`boot` コマンドにファイルの場所を指定します
- **x86** システムの場合、GRUB メニューの GRUB エントリを編集してファイルの場所を指定します

圧縮された構成ファイルの形式は、次のいずれかにしてください。

- tar
- compress で圧縮された tar
- zip
- bzip で圧縮された tar

▼ 圧縮された構成ファイルを作成する方法

- 1 プロファイルサーバー上の **JumpStart** ディレクトリに移動します。
`# cd jumpstart_dir_path`
- 2 圧縮ツールを使って、すべてのカスタム **JumpStart** 構成ファイルを 1 つのファイルに圧縮します。

注 - 圧縮構成ファイルに相対パスを含めないでください。カスタム **JumpStart** 構成ファイルは、圧縮されたファイルと同じディレクトリに置かれている必要があります。

圧縮された構成ファイルには、次のファイルが含まれている必要があります。

- プロファイル
- rules
- rules.ok

圧縮された構成ファイルに、さらに `sysidcfg` ファイルを入れることもできます。

- 3 圧縮された構成ファイルを **NFS** サーバー、**HTTP** サーバー、またはローカルハードディスク上に保存します。

圧縮された構成ファイルの例

次の例は、`tar` コマンドを使って、圧縮された構成ファイルを `config.tar` という名前で作成します。カスタム **JumpStart** 構成ファイルはすべて `/jumpstart` ディレクトリにあります。

例4-7 圧縮された構成ファイルの作成

```
# cd /jumpstart
# tar -cvf config.tar *
a profile 1K
a rules 1K
a rules.ok 1K
a sysidcfg 1K
```

ディスク構成ファイルの作成

この節では、単一ディスク構成ファイルおよび複数ディスク構成ファイルの作成方法を説明します。ディスク構成ファイルを作成すると、単一のシステムから `pfinstall(1M)` を使用して、異なるディスク構成に対してプロファイルをテストできます。

▼ SPARC: ディスク構成ファイルを作成する方法

- 1 テストするディスクのある **SPARC** ベースのシステムを探します。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBACの構成(作業マップ)」を参照してください。

- 3 `prtvtoc(1M)` コマンドの出力をファイルにリダイレクトして、単一ディスク構成ファイルを作成します。

```
# prtvtoc /dev/rdisk/device_name >disk_config_file
```

`/dev/rdisk/device_name` システムのディスクのデバイス名。 `device_name` は、`cwtxdys2` または `cxdys2` の形式で指定する必要があります。

`disk_config_file` ディスク構成ファイルの名前。

- 4 **Solaris** ソフトウェアのインストールのテストを、複数ディスク上で実行するかどうか決定します。
 - 実行しない場合は、ここで終了です。
 - 実行する場合は、各ディスク構成ファイルを結合し、その出力を新しいファイルに保存します。

```
# cat disk_file1 disk_file2 >multi_disk_config
```

新しいファイルは、次に示すように複数ディスク構成ファイルになります。

```
# cat 104_disk2 104_disk3 104_disk5 >multi_disk_test
```

- 5 前の手順で作成した複数ディスク構成ファイル内で、ディスクデバイス名のターゲット番号がそれぞれ固有であるかどうか調べます。
 - 固有である場合は、ここで終了です。

- 固有でない場合は、テキストエディタでファイルを開き、ターゲット番号を固有のものにします。

たとえば、次に示すように、異なるディスクデバイス名について同じターゲット番号(t0)がファイルに含まれているとします。

```
* /dev/rdisk/c0t0d0s2 partition map
...
* /dev/rdisk/c0t0d0s2 partition map
```

この場合は次のように、2番目のターゲット番号をt2に設定します。

```
* /dev/rdisk/c0t0d0s2 partition map
...
* /dev/rdisk/c0t2d0s2 partition map
```

SPARC: ディスク構成ファイルの例

次の例では、104Mバイトのディスクを持つ SPARC 搭載システムで、単一ディスク構成ファイル 104_test を作成します。

例4-8 SPARC: ディスク構成ファイルの作成

prtvtoc コマンドの出力を 104_test という単一ディスク構成ファイルに保存します。

```
# prtvtoc /dev/rdisk/c0t3d0s2 >104_test
```

104_test ファイルは、次のようになります。

```
* /dev/rdisk/c0t3d0s2 partition map
*
* Dimensions:
*   512 bytes/sector
*   72 sectors/track
*   14 tracks/cylinder
*  1008 sectors/cylinder
*  2038 cylinders*   2036 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*
* Partition  Tag  Flags      First   Sector   Last
*            Tag  Flags      Sector  Count    Sector  Mount Directory
*   1         2    00          0    164304   164303   /
*   2         5    00          0    2052288 2052287
*   3         0    00    164304    823536  987839   /disk2/b298
```

例 4-8 SPARC: ディスク構成ファイルの作成 (続き)

```

5      0      00      987840      614880 1602719 /install/298/sparc/work
7      0      00      1602720      449568 2052287 /space

```

これで SPARC システム用のディスク構成ファイルの作成が終了しました。ディスク構成ファイルを使ってプロファイル进行测试する方法については、55 ページの「[プロファイルのテスト](#)」を参照してください。

▼ **x86: ディスク構成ファイルを作成する方法**

- 1 テストするディスクがある **x86** ベースのシステムを探します。
- 2 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理(セキュリティサービス)』の「RBAC の構成(作業マップ)」を参照してください。

- 3 `fdisk(1M)` コマンドの出力をファイルに保存して、単一ディスク構成ファイルを部分的に作成します。

```
# fdisk -R -W disk_config_file -h /dev/rdsk/device_name
```

`disk_config_file` ディスク構成ファイルの名前。

`/dev/rdsk/device_name` ディスク全体の `fdisk` レイアウトのデバイス名。
`device_name` は、`cwtxdys0` または `cxdys0` の形式で指定する必要があります。

- 4 `prtvtoc(1M)` コマンドの出力をディスク構成ファイルの末尾に書き込みます。

```
# prtvtoc /dev/rdsk/device_name >>disk_config
```

`/dev/rdsk/device_name` システムのディスクのデバイス名。`device_name` は、`cwtxdys2` または `cxdys2` の形式で指定する必要があります。

`disk_config` ディスク構成ファイルの名前。

- 5 **Solaris** ソフトウェアのインストールのテストを、複数ディスク上で実行するかどうか決定します。
 - 実行しない場合は、ここで終了です。
 - 実行する場合は、各単一ディスク構成ファイルを結合し、その出力を新しいファイルに保存します。

```
# cat disk_file1 disk_file2 >multi_disk_config
```

新しいファイルは、次に示すように複数ディスク構成ファイルになります。

```
# cat 104_disk2 104_disk3 104_disk5 >multi_disk_test
```

- 6 前の手順で作成した複数ディスク構成ファイル内で、ディスクデバイス名のターゲット番号がそれぞれ固有であるかどうか調べます。

- 固有である場合は、ここで終了です。
- 固有でない場合は、テキストエディタでファイルを開き、ターゲット番号を固有のものにします。

たとえば、次に示すように、異なるディスクデバイス名について同じターゲット番号(t0)がファイルに含まれているとします。

```
* /dev/rdisk/c0t0d0s2 partition map
...
* /dev/rdisk/c0t0d0s2 partition map
```

この場合は次のように、2番目のターゲット番号をt2に設定します。

```
* /dev/rdisk/c0t0d0s2 partition map
...
* /dev/rdisk/c0t2d0s2 partition map
```

x86: ディスク構成ファイルの例

次の例では、500Mバイトのディスクを持つx86システムで、単一ディスク構成ファイル500_testを作成します。

例4-9 x86: ディスク構成ファイルの作成

最初に、fdiskコマンドの出力を500_testという名前のファイルに保存します。

```
# fdisk -R -W 500_test -h /dev/rdisk/c0t0d0p0
```

この500_testファイルの内容は次のとおりです。

```
* /dev/rdisk/c0t0d0p0 default fdisk table
* Dimensions:
*   512 bytes/sector
*   94 sectors/track
*   15 tracks/cylinder
*   1455 cylinders
*
```

例 4-9 x86: ディスク構成ファイルの作成 (続き)

```
* HBA Dimensions:
*   512 bytes/sector
*   94 sectors/track
*   15 tracks/cylinder
*   1455 cylinders
*
* systid:
* 1:   DOSOS12
* 2:   PCIXOS
* 4:   DOSOS16
* 5:   EXTDOS
* 6:   DOSBIG
* 86:  DOSDATA
* 98:  OTHEROS
* 99:  UNIXOS
* 130: SUNIXOS
*
* Id Act Bhead Bsect  Bcyl  Ehead  Esect  Ecyl Rsect  Numsect
130 128 44   3     0    46   30    1001 1410  2050140
```

次に、prtvtoc コマンドの出力を 500_test ファイルに追加します。

```
# prtvtoc /dev/rdisk/c0t0d0s2 >>500_test
```

これで 500_test ファイルは、完全なディスク構成ファイルになりました。

```
* /dev/rdisk/c0t0d0p0 default fdisk table
* Dimensions:
*   512 bytes/sector
*   94 sectors/track
*   15 tracks/cylinder
*   1455 cylinders
*
* HBA Dimensions:
*   512 bytes/sector
*   94 sectors/track
*   15 tracks/cylinder
*   1455 cylinders
*
* systid:
* 1:   DOSOS12
* 2:   PCIXOS
* 4:   DOSOS16
* 5:   EXTDOS
* 6:   DOSBIG
```

例 4-9 x86: ディスク構成ファイルの作成 (続き)

```
* 86:  DOSDATA
* 98:  OTHEROS
* 99:  UNIXOS
* 130: SUNIXOS
*
* Id  Act Bhead Bsect Bcyl  Ehead  Esec  Ecyl Rsect  Numsect
130  128 44   3    0   46   30   1001 1410  2050140
* /dev/rdisk/c0t0d0s2 partition map
*
* Dimensions:
*   512 bytes/sector
*   94 sectors/track
*   15 tracks/cylinder
*  1110 sectors/cylinder
*  1454 cylinders
*  1452 accessible cylinders
*
* Flags:
*  1: unmountable
* 10: read-only
*
* Partition  Tag  Flags      First  Sector  Last
* Partition  Tag  Flags      Sector  Count   Sector  Mount Directory
*   2         5   01        1410  2045910 2047319
*   7         6   00        4230  2043090 2047319  /space
*   8         1   01         0     1410    1409
*   9         9   01        1410    2820   422987
```

これで x86 システム用のディスク構成ファイルの作成が終了しました。ディスク構成ファイルを使ってプロファイルをテストする方法については、[55 ページの「プロファイルのテスト」](#)を参照してください。

サイト固有のインストールプログラムの使用

開始スクリプトと終了スクリプトを使用すると、独自のインストールプログラムにより Solaris ソフトウェアをインストールできます。

プロファイルフィールドにマイナス記号 (-) を指定すると、プロファイルと Solaris インストールプログラムの代わりに、開始スクリプトと終了スクリプトが、Solaris ソフトウェアのインストール方法を制御するようになります。

たとえば次のルールに一致した場合は、開始スクリプト `x_install.beg` と終了スクリプト `x_install.fin` が、`clover` という名前のシステムに Solaris ソフトウェアをインストールします。

```
hostname clover x_install.beg - x_install.fin
```

カスタムルールおよびプローブキーワードの作成(作業)

この章では、独自のカスタムルールとプローブキーワードを作成するための情報と手順を示します。

- 81 ページの「[プローブキーワード](#)」
- 82 ページの「[custom_probes ファイルの作成](#)」
- 85 ページの「[custom_probes ファイルの妥当性検査](#)」

プローブキーワード

プローブキーワードを理解するには、まずルールキーワードが何であるかを思い出してください。ルールキーワードは、ホスト名(hostname)やメモリーサイズ(memsize)などの一般的なシステム属性を記述する事前定義された字句単位または語です。ルールキーワードとそれに関連する値を使用すると、同じ属性を持つシステムをプロファイルに対応付けることができます。システム属性を対応させるこの方法では、グループ内の各システムに Solaris ソフトウェアをどのようにインストールするかが定義されます。

カスタム JumpStart の環境変数は、開始スクリプトおよび終了スクリプトで使用しますが、必要に応じて設定します。たとえば、すでにシステムにインストールされているオペレーティングシステムに関する情報は、installed ルールキーワードを使用した後にのみ、SI_INSTALLED から入手できます。

場合によっては、システムの照合とプロファイルの実行以外の目的で、この同じ情報を開始スクリプトまたは終了スクリプトで抽出する必要があります。プローブキーワードは、この解決方法となります。プローブキーワードは属性情報を抽出するので、照合条件を設定してプロファイルを実行する必要がなくなります。

プローブキーワードと値については、166 ページの「[プローブキーワードと値](#)」を参照してください。

custom_probes ファイルの作成

115 ページの「ルールキーワードと値の説明」と 166 ページの「プローブキーワードと値」で説明されているルールキーワードとプローブキーワードでは、ニーズに合わない場合もあります。custom_probes ファイルを作成して、独自のカスタムルールまたはプローブキーワードを定義できます。

custom_probes ファイルは、2 種類のタイプの関数を含む Bourne シェルスクリプトです。custom_probes ファイルは、rules ファイルを保存したのと同じ JumpStart ディレクトリに保存する必要があります。次に、custom_probes ファイルに定義できる 2 種類の関数を示します。

- **プローブ** - 必要な情報を収集するかあるいは実際の作業を実行し、定義された対応する SI_ 環境変数を設定します。プローブ関数はプローブキーワードになります。
- **比較** - 対応するプローブ関数を呼び出してプローブ関数の出力を比較し、キーワードが一致する場合は 0、キーワードが一致しない場合は 1 を返します。比較関数はルールキーワードになります。

custom_probes ファイルの構文

custom_probes ファイルには、任意の有効な Bourne シェルコマンド、変数、またはアルゴリズムを含めることができます。

注 - custom_probes ファイルで単一の引数を必要とするプローブおよび比較関数を定義できます。rules ファイル内で対応するカスタムプローブキーワードを使用する場合、このキーワードの後に続く引数が (\$1 として) 解釈されます。

rules ファイル内で対応するカスタムルールキーワードを使用する場合、引数は順番に解釈されます。この順番は、キーワードの直後から開始し、次に && または開始スクリプトが現れる直前で終了します。

custom_probes ファイルは、次の要件を満たしている必要があります。

- 名前が custom_probes である
- 所有者が root である
- 実行可能であり、アクセス権が 755 に設定されている
- 少なくとも 1 つのプローブ関数と対応する比較関数を含む

より明確で効果的な構成にするために、最初にすべてのプローブ関数をファイルの一番上に定義してから、比較関数を続けて定義してください。

custom_probes 内の関数名の構文

プローブ関数名は、probe_ で始める必要があります。比較関数名は、cmp_ で始める必要があります。

probe_ で始まる関数は、新しいプローブキーワードを定義します。たとえば、関数 probe_tcx は、新しいプローブキーワード tcx を定義します。cmp_ で始まる関数は、新しいルールキーワードを定義します。たとえば、cmp_tcx は、新しいルールキーワード tcx を定義します。

▼ custom_probes ファイルを作成する方法

- 1 テキストエディタを使用して **Bourne** シェルスクリプトのテキストファイルを作成します。このファイルに custom_probes という名前を付けます。
- 2 custom_probes テキストファイルで、プローブ関数と比較関数を定義します。

注-customer_probes ファイルには、引数を必要とするプローブおよび比較関数を定義できます。rules ファイル内で対応するカスタムプローブキーワードを使用する場合、このキーワードの後に続く引数は順番に (\$1、\$2 などとして) 解釈されます。

rules ファイル内で対応するカスタムルールキーワードを使用する場合、引数は順番に解釈されます。この順番は、キーワードの直後から開始し、次に && または開始スクリプトが現れる直前で終了します。

- 3 custom_probes ファイルを **JumpStart** ディレクトリに保存します (rules ファイルと同じ)。
- 4 root が rules ファイルを所有していて、そのアクセス権が **644** に設定されていることを確認します。

custom_probes ファイルとキーワードの例

プローブ関数と比較関数の例は次のディレクトリにも入っています。

- Solaris ソフトウェアがインストールされたシステムの
/usr/sbin/install.d/chkprobe
- Solaris Operating System DVD または Solaris SOFTWARE - 1 CD 上の
/Solaris_10/Tools/Boot/usr/sbin/install.d/chkprobe

次の custom_probes ファイルには、TCX グラフィックスカードがあるかどうかをテストするプローブおよび比較関数が含まれます。

例5-1 custom_probes ファイル

```
#!/bin/sh
#
# custom_probe script to test for the presence of a TCX graphics card.
#
#
# PROBE FUNCTIONS
#
probe_tcx() {
    SI_TCX='modinfo | grep tcx | nawk '{print $6}''
    export SI_TCX
}

#
# COMPARISON FUNCTIONS
#
cmp_tcx() {
    probe_tcx

    if [ "X${SI_TCX}" = "X${1}" ]; then
        return 0
    else
        return 1
    fi
}
```

次のサンプル rules ファイルは、前の例で定義されているプローブキーワード(tcx) の使用方法を示しています。TCX グラフィックスカードがシステムにインストールされていて検出されると、profile_tcx が実行されます。そうでない場合は、profile が実行されます。

注- プローブキーワードは、rules ファイルの最初、またはその近くに指定してください。これは、そのプローブキーワードに依存する可能性のあるほかのルールキーワードよりも先に読み取られて実行されるようにするためです。

例5-2 rules ファイルで使用されるカスタムプローブキーワード

```
probe tcx
tcx    tcx    -    profile_tcx    -
any    any    -    profile        -
```

custom_probes ファイルの妥当性検査

プロファイル、rules ファイル、および custom_probes ファイルを使用する前に、check スクリプトを実行して、ファイルが正常に設定されていることを検証する必要があります。すべてのプロファイル、ルール、およびプローブ関数と比較関数が正しく設定されている場合は、rules.ok ファイルと custom_probes.ok ファイルが作成されます。表 5-1 に、check スクリプトの実行内容を示します。

表 5-1 check スクリプトの動作

手順	説明
1	check は、custom_probes ファイルを検索します。
2	ファイルが存在する場合、check は custom_probes ファイルから custom_probes.ok ファイルを作成し、すべてのコメントと空行を削除して Bourne シェルコマンド、変数、およびアルゴリズムだけを保持します。次に check は次のコメント行をファイルの最後に追加します。 # version=2 checksum=num

▼ custom_probes ファイルを妥当性検査する方法

- 1 check スクリプトが **JumpStart** ディレクトリにあることを確認します。

注 - check スクリプトは、Solaris Operating System DVD または Solaris SOFTWARE - 1 CD の Solaris_10/Misc/jumpstart_sample ディレクトリにあります。

- 2 **JumpStart** ディレクトリに移動します。
- 3 check スクリプトを実行して、rules ファイルと custom_probes ファイルの妥当性を検査します。

```
$ ./check -p path -r file_name
```

-p path 使用しているシステムの check スクリプトではなく、プラットフォームに対応した Solaris ソフトウェアイメージの check スクリプトを使用して、custom_probes ファイルの妥当性を検査します。path は、ローカルディスク上のイメージか、マウントされた Solaris Operating System DVD または Solaris SOFTWARE - 1 CD 上のイメージです。

以前のバージョンの Solaris を稼働中のシステムでは、このオプションを指定することにより、check の最新バージョンを実行します。

`-r file_name` custom_probes 以外のファイル名を指定します。-r オプションを使用することで、一連の関数を custom_probes ファイル内にまとめる前にそれらの妥当性検査を実施できます。

check スクリプトを実行すると、rules ファイル、custom_probes ファイル、および各プロファイルの妥当性が報告されます。エラーが検出されなければ、スクリプトは次のメッセージを表示します。「The custom JumpStart configuration is ok」と表示され、JumpStart ディレクトリ内に rules.ok ファイルと custom_probes.ok ファイルが作成されます。

4 custom_probes.ok ファイルが実行可能かどうか調べます。

- 実行可能な場合、手順 5 へ進みます。
- 実行可能でない場合は、次のコマンドを入力します。

```
# chmod +x custom_probes
```

5 root が custom_probes.ok ファイルを所有し、そのアクセス権が 755 に設定されていることを確認してください。

カスタム JumpStart インストールの実行 (作業)

この章では、SPARC システムまたは x86 システムでカスタム JumpStart インストールを実行する手順を説明します。これらの手順は、Solaris ソフトウェアをインストールしようとするシステム上で実行する必要があります。

- 93 ページの「SPARC: カスタム JumpStart プログラムを使用してインストールまたはアップグレードする方法」
- 98 ページの「x86: カスタム JumpStart プログラムおよび GRUB を使用してインストールまたはアップグレードを実行する方法」

JumpStart インストールの制限事項

JumpStart インストールの実行時に、いくつかの問題が発生する可能性があります。それぞれの情報については、次の表を確認してください。

表 6-1 JumpStart インストールの制限事項

問題	説明	参照先
サンプルの JumpStart スクリプトを使って NFSv4 プロンプトを抑制する必要がなくなりました	<p>Solaris 10 リリースでは、JumpStart インストール時にプロンプトが表示されないようにするための、サンプルの JumpStart スクリプト <code>set_nfs4_domain</code> が、メディアで提供されていました。このスクリプトは、インストール時に NFSv4 プロンプトが表示されないようにしていました。このスクリプトはもう必要ありません。Solaris 10 8/07 リリース以降では、<code>sysidcfg</code> のキーワード <code>nfs4_domain</code> を使ってプロンプトが表示されないようにします。set_nfs4_domain スクリプトには、プロンプトを抑制する機能がなくなりました。</p> <p>非大域ゾーンがインストールされている場合に新しい <code>nfs4_domain</code> キーワードが <code>sysidcfg</code> ファイルに含まれていると、非大域ゾーンを最初にブートしたときにドメインが設定されます。それ以外の場合は、Solaris 対話式インストールプログラムが起動して、ブートプロセスが完了する前にドメイン名を入力するように促すプロンプトが表示されます。</p>	『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の「 <code>nfs4_domain</code> キーワード」
<code>sysidcfg</code> ファイルでキーボード言語を選択すると、プロンプトが表示されなくなります	キーボードが自己識別型でない場合に、JumpStart インストール時にプロンプトが表示されないようにするには、 <code>sysidcfg</code> ファイルでキーボードの言語を選択します。JumpStart インストールの場合、デフォルトは米国英語 (U.S. English) 用です。別の言語とそれに対応するキー配列を選択するには、 <code>sysidcfg</code> ファイルでキーボードのキーワードを設定します。	<ul style="list-style-type: none"> ■ 『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の「<code>sysidcfg</code> ファイルキーワード」 ■ 次のマニュアルページを参照してください。 <ul style="list-style-type: none"> ■ <code>sysidtool(1M)</code> ■ <code>sysidcfg(4)</code>
非大域ゾーンが含まれている場合は、Solaris Live Upgrade を使ってアップグレードを行います	JumpStart を使っても、非大域ゾーンがインストールされているシステムをアップグレードできますが、アップグレードには Solaris Live Upgrade が推奨されています。JumpStart では、膨大なアップグレード時間が必要となる場合があります。これは、アップグレードの実行に要する時間が、インストールされている非大域ゾーンの数に比例して増加するからです。	『Solaris 10 5/08 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』
アーカイブには非大域ゾーンを含めることはできません	Solaris フラッシュアーカイブを使ってインストールを行う場合は、非大域ゾーンが含まれているアーカイブがシステムに正しくインストールされません。	非大域ゾーンの作成の概要については、『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』を参照してください。

表 6-1 JumpStart インストールの制限事項 (続き)

問題	説明	参照先
SPARC: その他のハードウェア要件	使用中のプラットフォームで JumpStart インストールを実行するためのその他の要件については、ハードウェアのマニュアルを参照してください。	

SPARC: 作業マップ: カスタム JumpStart インストールのためのシステムの設定

表 6-2 作業マップ: カスタム JumpStart インストールのためのシステムの設定

作業	説明	参照先
システムがサポートされているかを確認します。	ハードウェアのマニュアルで、Solaris 環境におけるシステムサポートを確認します。	『Solaris Sun ハードウェアマニユア』 (http://docs.sun.com)
Solaris ソフトウェアをインストールするための十分なディスク容量がシステムにあるかを確認します。	Solaris ソフトウェアのインストールに求められる十分な容量があるかを確認します。	『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の第 4 章「システム要件、ガイドライン、およびアップグレード (計画)」
(省略可能) システムパラメータを設定します。	インストールやアップグレードの際に情報を入力する手間を省くために、システム構成情報を事前に設定しておくことができます。	『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の第 2 章「システム構成情報の事前設定 (作業)」
カスタム JumpStart インストールを実行するようにシステムを準備します。	rules ファイルとプロファイルファイルを作成して検証します。	第 3 章カスタム JumpStart インストールの準備 (作業)
(省略可能) カスタム JumpStart のオプション機能を準備します。	開始スクリプト、終了スクリプトなどのオプション機能を使用する場合は、それらのスクリプトまたはファイルを用意します。	第 4 章カスタム JumpStart オプション機能の使用 (作業) および第 5 章カスタムルールおよびプロブキーワードの作成 (作業)

表 6-2 作業マップ: カスタム JumpStart インストールのためのシステムの設定 (続き)

作業	説明	参照先
(省略可能) Solaris ソフトウェアをネットワークからインストールする準備を行います。	リモートの Solaris Operating System DVD (SPARC 版) または Solaris SOFTWARE (SPARC 版) イメージをインストールする場合は、インストールサーバーまたはブートサーバーからシステムをブートしてインストールするように、システムを設定する必要があります。	『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の第 5 章「DVD メディアを使用したネットワークインストール (作業)」 『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の第 6 章「CD メディアを使用したネットワークインストール (作業)」
(省略可能) Solaris フラッシュアーカイブのインストール準備をします。	Solaris フラッシュアーカイブのインストールに固有の設定を行います。	90 ページの「カスタム JumpStart インストールを使用して Solaris フラッシュアーカイブをインストールする方法」
インストールまたはアップグレードを実行します。	システムをブートしてインストールまたはアップグレードを開始します。	93 ページの「SPARC: カスタム JumpStart プログラムを使用してインストールまたはアップグレードする方法」

SPARC: カスタム JumpStart インストールの実行

カスタム JumpStart インストールの際、JumpStart プログラムは、インストール対象のシステムと `rules.ok` ファイル内のルールを照合します。この際、最初から順にルールが読み取られ、インストール中のシステムがルールに定義されたすべてのシステム属性に一致するかどうか検査されます。システムがルールと一致すると、JumpStart プログラムによる `rules.ok` ファイルの読み取りが停止して、一致したルールのプロファイルに基づいてシステムのインストールが開始されます。

▼ カスタム JumpStart インストールを使用して Solaris フラッシュアーカイブをインストールする方法

初期インストールの場合は完全アーカイブを、アップグレードの場合 (過去にアーカイブをインストールしたことがある場合は差分アーカイブを、それぞれインストールできます。カスタム JumpStart インストールまたは Solaris Live Upgrade を使用して、非アクティブブート環境にアーカイブをインストールできます。ここでは、カスタム JumpStart を使ってアーカイブをインストールする手順を紹介します。

- 完全アーカイブと差分アーカイブの概要については、『Solaris 10 5/08 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』の第1章「Solaris フラッシュ (概要)」を参照してください。
- Solaris Live Upgrade を使って、非アクティブなブート環境にアーカイブをインストールする手順については、『Solaris 10 5/08 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「プロファイルを使用して Solaris フラッシュアーカイブをインストールする」を参照してください。

1 次の制限事項を確認します。

説明	例
<p>注意: <code>archive_location</code> キーワードを使用して Solaris フラッシュアーカイブをインストールする場合は、アーカイブおよびインストールメディアに同一のオペレーティングシステムのバージョンが格納されている必要があります。</p> <p>注意-非大域ゾーンがインストールされていると、Solaris フラッシュアーカイブは正常に作成できません。Solaris フラッシュ機能には Solaris ゾーン区分技術との互換性はありません。Solaris フラッシュアーカイブを作成する場合、そのアーカイブの配備条件が次のいずれかの場合は、作成されたアーカイブは正しくインストールされません。</p> <ul style="list-style-type: none"> ■ アーカイブが非大域ゾーンに作成された場合 ■ アーカイブが、非大域ゾーンがインストールされている大域ゾーンに作成された場合 	<p>たとえば、アーカイブが Solaris 10 5/08 オペレーティングシステムで、DVD メディアを使用している場合は、Solaris 10 5/08 DVD メディアを使用してアーカイブをインストールする必要があります。オペレーティングシステムのバージョンが一致しないと、クローンシステムでのインストールに失敗します。</p>

- 2 インストールサーバーで、カスタム JumpStart の `rules` ファイルを作成します。
カスタム JumpStart ファイルの詳しい作成方法については、[第3章カスタム JumpStart インストールの準備 \(作業\)](#)を参照してください。
- 3 インストールサーバーで、カスタム JumpStart のプロファイルを作成します。
Solaris フラッシュアーカイブプロファイルの例については、[41 ページの「プロファイルの例」](#)を参照してください。

表 8-2 に示された既存のカスタム JumpStart キーワードのうち、Solaris フラッシュアーカイブのインストール時に有効なものは次のキーワードだけです。

キーワード	初期インストール	差分アーカイブ
(必須)archive_location	○	○
fdisk (x86 のみ)	○	○
filesystem	○	
注 - filesystem キーワードに値 auto は設定できません。		
forced_deployment		○
(必須) Install_type	○	○
local_customization	○	○
no_content_check		○
no_master_check		○
package	○	
root_device	○	○

- a. キーワード `install_type` の値を次のいずれかに設定します。
 - 完全アーカイブインストールの場合、値を `flash_install` に設定します。
 - 差分アーカイブインストールの場合、値を `flash_update` に設定します。
- b. `archive_location` プロファイルキーワードを使用して、**Solaris** フラッシュアーカイブへのパスを追加します。
`archive_location` キーワードの詳細については、[121 ページ](#)の「`archive_location` プロファイルキーワード」を参照してください。
- c. ファイルシステム構成を指定します。
 Solaris フラッシュアーカイブの抽出プロセスでは、パーティションの自動配置はサポートされません。
- d. (省略可能) アーカイブのインストール時に追加パッケージもインストールする場合は、`package` キーワードを使用してください。詳細については、[150 ページ](#)の「`package` プロファイルキーワード」を参照してください。
- e. (省略可能) クローンシステムに **Solaris** フラッシュアーカイブを追加インストールする場合は、インストールするアーカイブごとに `archive_location` 行を指定してください。

- 4 インストールサーバーで、**Solaris** フラッシュアーカイブを使ってインストールするクライアントを追加します。
詳細については、次を参照してください。
 - 『Solaris 10 5/08 インストールガイド(ネットワークインストール)』の「DVDイメージを使用してネットワークからインストールするシステムの追加」
 - 『Solaris 10 5/08 インストールガイド(ネットワークインストール)』の「CDイメージを使用してネットワークからインストールするシステムの追加」
- 5 クローンシステムへのカスタム **JumpStart** インストールを実行します。
詳細については、93 ページの「**SPARC: カスタム JumpStart プログラムを使用してインストールまたはアップグレードする方法**」を参照してください。

▼ **SPARC: カスタム JumpStart プログラムを使用してインストールまたはアップグレードする方法**

- 1 システムがネットワークに接続されている場合は、**Ethernet** コネクタまたはそれに類似したネットワークアダプタがシステムに装着されていることを確認します。
- 2 tip(1) ラインで接続されているシステムをインストールする場合は、ウィンドウ表示が横 **80** 桁、縦 **24** 行以上あることを確認します。
現在の tip ウィンドウサイズを確認するには、`stty(1)` コマンドを使用します。
- 3 システムの **DVD-ROM** または **CD-ROM** ドライブを使用して **Solaris** ソフトウェアをインストールする場合は、ドライブに **Solaris Operating System DVD (SPARC 版)** または **Solaris SOFTWARE - 1 CD (SPARC 版)** を挿入します。
- 4 プロファイルフロッピーディスクを使用する場合は、システムのプロッピーディスクドライブにそのフロッピーディスクを挿入します。
- 5 システムをブートします。
 - 新しく購入したばかり (未使用) のシステムの場合は、システムの電源を入れます。
 - 既存システムのインストールまたはアップグレードを行う場合は、システムをシャットダウンします。ok プロンプトで、`boot` コマンドに適切なオプションを入力します。`boot` コマンドの構文は次のとおりです。

```
ok boot [cd-dvd|net] - install [url|ask] options
```

たとえば、次のコマンドを入力すると、JumpStart プロファイルを使用してネットワーク経由で OS がインストールされます。

```
ok boot net - install http://131.141.2.32/jumpstart/config.tar
```

boot コマンドのオプションの詳細は、次の表を参照してください。

SPARCのみ-ハードウェアおよびシステムコンポーネントがシステムによって検査されて、SPARCシステムがブートします。ブートには数分間かかります。

- 6 sysidcfg でシステム情報の事前構成を行わなかった場合は、プロンプトが表示された時点でシステム構成についての質問に答えます。

- 7 画面の指示に従って、ソフトウェアをインストールします。

JumpStart プログラムが Solaris ソフトウェアのインストールを終了すると、システムが自動的にリブートします。

インストールが終了すると、インストールログがファイルに保存されます。インストールログは、次のディレクトリに作成されます。

- /var/sadm/system/logs
- /var/sadm/install/logs

SPARC: boot コマンドのコマンドリファレンス

boot コマンドの構文は次のとおりです。

```
ok boot [cd-dvd|net] - install [url|ask] options
```

次の表に、JumpStart インストールに適した boot コマンドのコマンド行オプションを示します。

オプション	説明
[<i>cd-dvd net</i>]	CD または DVD からブートするか、ネットワーク上のインストールサーバーからブートするかを指定します。 <ul style="list-style-type: none"> ▪ <i>cd-dvd</i> - cdrom を使用して CD または DVD からブートします。 ▪ <i>net</i> - ネットワーク上のインストールサーバーからブートすることを指定します。

オプション	説明
[url ask]	<p>カスタム JumpStart ファイルの場所を指定するか、場所の入力を求めるプロンプトを表示します。</p> <ul style="list-style-type: none"> ■ <code>url</code> - ファイルのパスを指定します。HTTP または HTTPS サーバーに存在するファイルの URL を指定できます。 HTTP サーバー <code>http://server_name:IP_address/jumpstart_dir_path/compressed_config_file&proxy_info</code> ■ <code>sysidcfg</code> ファイルを圧縮構成ファイルに含めた場合は、次の例に示すようにそのファイルを含むサーバーの IP アドレスを指定する必要があります。 <code>http://131.141.2.32/jumpstart/config.tar</code> ■ 圧縮構成ファイルをファイアウォールで防御された HTTP サーバー上の圧縮構成ファイルに保存した場合は、ブート時に <code>proxy</code> 指示子を使用する必要があります (ファイルが配置されたサーバーの IP アドレスを指定する必要はない)。プロキシサーバーの IP アドレスは次のように指定します。 <code>http://www.shadow.com/jumpstart/config.tar&proxy=131.141.6.151</code> ■ <code>ask</code> - 圧縮構成ファイルの場所の入力を求めるプロンプトを表示するように、インストールプログラムを指定します。プロンプトは、システムがブートしてネットワークに接続すると表示されます。このオプションを使用すると、完全に自動化された JumpStart インストールを行うことはできません。 Return キーを押してこのプロンプトへの入力を省略すると、Solaris インストールプログラムはネットワークパラメータを対話的に構成します。インストールプログラムは次に、圧縮構成ファイルの場所を尋ねるプロンプトを表示します。
options	<ul style="list-style-type: none"> ■ <code>dhcp</code> - システムのブートに必要なネットワークインストール情報の取得に DHCP サーバーを使用することを指定します。このオプションは JumpStart インストールには必要ありません。<code>dhcp</code> で DHCP サーバーの使用を指定しないと、<code>/etc/bootparams</code> ファイル、またはネームサービスの <code>bootparams</code> データベースが使用されます。たとえば、静的 IP アドレスを保持する場合には、<code>dhcp</code> を指定しません。 ■ <code>nowin</code> オプションおよび <code>text</code> オプションは、JumpStart インストールには適用されません。これらのオプションは、対話式インストールで役立ちます。詳細は、『Solaris 10 5/08 インストールガイド (基本編)』の「Solaris インストールプログラムを使用してインストールまたはアップグレードを行う方法」を参照してください。

x86:作業マップ:カスタム JumpStart インストールのためのシステムの設定

表 6-3 x86:作業マップ:カスタム JumpStart インストールのためのシステムの設定

作業	説明	参照先
既存のオペレーティングシステムとユーザーデータを保存する必要があるかを決定します。	システムにディスク全体を使用している既存のオペレーティングシステムがある場合は、そのシステムと Solaris 10 5/08 ソフトウェアを共存させるために既存のオペレーティングシステムを保存する必要があります。この決定により、システムのプロファイル内の fdisk(1M) キーワードをどのように指定するかが決まります。	135 ページの「x86: fdisk プロファイルキーワード」
システムがサポートされているかを確認します。	ハードウェアのマニュアルで、Solaris 環境におけるシステムサポートを確認します。	ハードウェアのマニュアル
Solaris ソフトウェアをインストールするための十分なディスク容量がシステムにあるかを確認します。	Solaris ソフトウェアのインストールに求められる十分な容量があるかを確認します。	『Solaris 10 5/08 インストールガイド(インストールとアップグレードの計画)』の第 4 章「システム要件、ガイドライン、およびアップグレード(計画)」
(省略可能) システムパラメータを設定します。	インストールやアップグレードの際に情報を入力する手間を省くために、システム構成情報を事前に設定しておくことができます。	『Solaris 10 5/08 インストールガイド(ネットワークインストール)』の第 2 章「システム構成情報の事前設定(作業)」
カスタム JumpStart インストールを実行するようにシステムを準備します。	rules ファイルとプロファイル ファイルを作成して検証します。	第 3 章カスタム JumpStart インストールの準備(作業)
(省略可能) カスタム JumpStart のオプション機能を準備します。	開始スクリプト、終了スクリプトなどのオプション機能を使用する場合は、それらのスクリプトまたはファイルを用意します。	第 4 章カスタム JumpStart オプション機能の使用(作業)および第 5 章カスタムルールおよびプロンプトキーワードの作成(作業)

表 6-3 x86: 作業マップ: カスタム JumpStart インストールのためのシステムの設定 (続き)

作業	説明	参照先
(省略可能) Solaris ソフトウェアをネットワークからインストールする準備を行います。	リモートの Solaris Operating System DVD (x86 版) または Solaris SOFTWARE (x86 版) イメージをインストールする場合は、インストールサーバーまたはブートサーバーからシステムをブートしてインストールするように、システムを設定する必要があります。	『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の第 6 章「CD メディアを使用したネットワークインストール (作業)」
(省略可能) Solaris フラッシュアーカイブのインストール準備をします。	Solaris フラッシュアーカイブのインストールに固有の設定を行います。	90 ページの「カスタム JumpStart インストールを使用して Solaris フラッシュアーカイブをインストールする方法」
インストールまたはアップグレードを実行します。	システムをブートしてインストールまたはアップグレードを開始します。	98 ページの「x86: カスタム JumpStart プログラムおよび GRUB を使用してインストールまたはアップグレードを実行する方法」

x86: カスタム JumpStart インストールの実行

カスタム JumpStart インストールの際、JumpStart プログラムは、インストール対象のシステムと `rules.ok` ファイル内のルールを照合します。この際、最初から順にルールが読み取られ、インストール中のシステムがルールに定義されたすべてのシステム属性に一致するかどうか検査されます。システムがルールと一致すると同時に、JumpStart プログラムによる `rules.ok` ファイルの読み取りが停止して、一致したルールのプロファイルに基づいてシステムのインストールが開始されます。

カスタム JumpStart を使って、Solaris フラッシュアーカイブをインストールできます。手順については、90 ページの「カスタム JumpStart インストールを使用して Solaris フラッシュアーカイブをインストールする方法」を参照してください。

次のいずれかの手順を選択してください。

- 標準的なカスタム JumpStart の手順については、98 ページの「x86: カスタム JumpStart プログラムおよび GRUB を使用してインストールまたはアップグレードを実行する方法」を参照してください。
- GRUB コマンドを編集してカスタム JumpStart を実行する方法については、100 ページの「x86: GRUB ブートコマンドの編集によるカスタム JumpStart インストールの実行」を参照してください。

▼ x86: カスタム JumpStart プログラムおよび GRUB を使用してインストールまたはアップグレードを実行する方法

GRUB メニューを使用して x86 ベースのシステムに Solaris OS をインストールするには、この手順を使用します。

- 1 システムがネットワークに接続されている場合は、**Ethernet** コネクタまたはそれに類似したネットワークアダプタがシステムに装着されていることを確認します。
- 2 `tip(1)` ラインで接続されているシステムをインストールする場合は、ウィンドウ表示が横 **80** 桁、縦 **24** 行以上あることを確認します。
現在の `tip` ウィンドウサイズを確認するには、`stty(1)` コマンドを使用します。
- 3 プロファイルフロッピーディスクを使用するかどうかを決定します。
システムのブートにプロファイルフロッピーディスクは使用されなくなりましたが、JumpStart ディレクトリのみを収録したフロッピーディスクを用意することはできます。このフロッピーディスクは、JumpStart インストールを実行するときや CD-ROM をブートオフするときなどに使用できます。
 - プロファイルフロッピーディスクを使用する場合は、システムのフロッピーディスクドライブにそのフロッピーディスクを挿入します。
 - プロファイルフロッピーディスクを使用しない場合は、[手順 4](#)に進みます。
- 4 システムのブート方法を決定します。
 - Solaris Operating System DVD または Solaris SOFTWARE - 1 CD からブートする場合は、そのディスクを挿入します。この場合、システムの BIOS が DVD または CD からのブートをサポートしている必要があります。
 - ネットワークからブートする場合は、PXE (Preboot Execution Environment) ネットワークブートを使用してください。システムは PXE をサポートするものでなければなりません。システムの BIOS 設定ツールまたはネットワークアダプタの構成設定ツールを使用して、PXE を使用するようシステムを設定します。
- 5 (省略可能) DVD または CD からブートする場合は、システムの BIOS のブート設定を変更し、DVD または CD メディアからブートするように設定します。手順については、ハードウェアのマニュアルを参照してください。

- 6 システムの電源が入っていない場合は、電源を入れます。システムの電源が入っている場合は、システムをリブートします。

GRUB メニューが表示されます。このメニューにはブートエントリの一覧が表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
|Solaris 10 5/08 image_directory                               |
|Solaris Serial Console ttya                                 |
|Solaris Serial Console ttyb (for lx50, v60x and v65x        |
+-----+
Use the ^ and v keys to select which entry is highlighted. Press
enter to boot the selected OS, 'e' to edit the commands before
booting, or 'c' for a command-line.
```

image_directory は、インストールイメージが置かれているディレクトリの名前です。JumpStart ファイルのパスは、`add_install_client` コマンドと `-c` オプションによって定義されています。

注-ここでGRUBエントリからブートせずに、ブートエントリを編集することもできます。GRUBエントリを編集してから、JumpStartインストールを実行します。GRUBエントリとインストールオプションの一覧の編集方法については、[100 ページの「x86: GRUB ブートコマンドの編集によるカスタム JumpStart インストールの実行」](#)を参照してください。

- 7 プロンプトが表示されたら、次の操作のいずれかを実行します。

Select the type of installation you want to perform:

- 1 Solaris Interactive
- 2 Custom JumpStart
- 3 Solaris Interactive Text (Desktop session)
- 4 Solaris Interactive Text (Console session)
- 5. Apply driver updates
- 6. Single User Shell

Enter the number of your choice.

Please make a selection (1-6).

カスタム JumpStart 方式を選択するには、**2** を入力し、Enter キーを押します。

JumpStart インストールが開始されます。

注-

- 30 秒以内に選択しないと、Solaris の対話式インストールプログラムが起動します。コマンド行に任意のキーを入力することによって、このタイマーを止めることができます。
- 1、3、または4を選択すると、対話式インストールになります。対話式インストールについては、『Solaris 10 5/08 インストールガイド (基本編)』を参照してください。
- 5を選択すると、ドライバの更新がインストールされます。
- 6を選択すると、保守作業を実行できます。

8 sysidcfg でシステム情報の事前構成を行わなかった場合は、プロンプトが表示された時点でシステム構成についての質問に答えます。

9 画面の指示に従って、ソフトウェアをインストールします。

JumpStart プログラムが Solaris ソフトウェアのインストールを終了すると、システムが自動的にリブートします。また、GRUB menu.lst ファイルも自動的に更新されます。インストールした Solaris のインスタンスは、GRUB メニューを次に使用するときに表示されます

インストールが終了すると、インストールログがファイルに保存されます。インストールログは、次のディレクトリに作成されます。

- /var/sadm/system/logs
- /var/sadm/install/logs

x86: GRUB ブートコマンドの編集によるカスタム JumpStart インストールの実行

デバッグのためなど、必要に応じて GRUB ブートコマンドを変更できます。次に、カスタム JumpStart インストールを実行する前に GRUB ブートコマンドを編集する手順を示します。

▼ x86: GRUB ブートコマンドを変更する方法

- 1 インストールを開始するには、98 ページの「x86: カスタム JumpStart プログラムおよび GRUB を使用してインストールまたはアップグレードを実行する方法」に示した手順の、手順 1～手順 5 を実行します。

- 2 システムの電源が入っていない場合は、電源を入れます。システムの電源が入っている場合は、システムをリブートします。

GRUB メニューが表示されます。このメニューにはブートエントリの一覧が表示されます。表示されるエントリはインストールする Solaris インスタンスです。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
|Solaris 10 5/08 image_directory |
|Solaris Serial Console ttya |
|Solaris Serial Console ttyb (1x50, v60x and v68) |
+-----+
```

Use the ^ and v keys to select which entry is highlighted. Press enter to boot the selected OS, 'e' to edit the commands before booting, or 'c' for a command-line.

image_directory は、インストールイメージが置かれているディレクトリの名前です。

注 -

- NFS を使用して、`add_install_client` コマンドと `-c` オプションで JumpStart ディレクトリへのパスを設定した場合は、ブートエントリにそのパスを含める必要はありません。
- NFS を使用していない場合は、JumpStart ディレクトリを含んでいる圧縮構成ファイルへのパスを書き留めておく必要があります。

- 3 ブートプロセスを停止してメニューエントリエディタを使用するには、**e** を入力します。

GRUB 編集メニューが表示されます。

```
kernel /I86PC.Solaris_11-8/multiboot kernel/unix -B console=ttyb,\
install_media=131.141.2.32:/export/mary/v11 \
module /I86PC.Solaris_11-8/x86.new
```

- 4 矢印キーを使用してブートエントリを選択します。
- 5 選択したコマンドを編集するには、**e** を入力します。
次の例のようなコマンドが表示されます。

```
grub edit>kernel /I86PC.Solaris_11-8/multiboot kernel/unix -B \
console=ttyb,install_media=131.141.2.32:/export/mary/\
module /I86PC.Solaris_11-8/x86.new
```

- 6 必要なオプションを入力してコマンドを編集します。

JumpStart インストールの構文は次のとおりです。

```
grub edit>kernel /I86PC.Solaris_11-image_directory/multiboot kernel/unix/ \
- install [url|ask] options -B install_media=media_type
```

JumpStart オプションの説明については、102 ページの「x86: システムのブートのためのコマンドリファレンス」を参照してください。

次の例では、カスタム JumpStart プロファイルを使用して、ネットワーク経由で OS をインストールします。

```
kernel /I86PC.Solaris_11-8/multiboot kernel/unix/ - install \
-B install_media=131.141.2.32:/export/mary/v11 \
module /I86PC.Solaris_11-8/x86.new
```

- 7 編集内容を適用するには、**Enter** キーを押します。
変更が保存され、GRUB メインメニューが表示されます。

注- 変更を保存せずに GRUB メインメニューに戻るには、Escape キーを押します。

- 8 インストールを開始するには、**b** を入力します。

x86: システムのブートのためのコマンドリファレンス

次の表では、GRUB メニューのブートコマンドに使用するコマンド行オプションについて説明します。この表に示されているオプションは、JumpStart インストールに適したオプションです。

ブートコマンドの構文は次のとおりです。

```
kernel /I86PC.Solaris_11-image_directory/multiboot kernel/unix/ - install \
[url|ask] options -B install_media=media_type
```

表 6-4 GRUB メニューのブートコマンドリファレンス

オプション	説明
- install	<p>カスタム JumpStart インストールを実行します。</p> <p>次の例では、システムは DVD メディアからブートして、次のようなオプションが使用されています。</p> <ul style="list-style-type: none"> ■ install はカスタム JumpStart を実行する ■ file://jumpstart/config.tar は、ローカルディスクで JumpStart プロファイルを検出する <pre>kernel /I86pc.Solaris_11.8/multiboot - install file://jumpstart/config.tar \ -B install_media=dvdrom module /I86Solaris_11.8/x86.new</pre>

表 6-4 GRUB メニューのブートコマンドリファレンス (続き)

オプション	説明
[url ask]	<p>カスタム JumpStart ファイルの場所を指定するか、場所の入力を求めるプロンプトを表示します。</p>
	<ul style="list-style-type: none"> ■ <code>url</code> - ファイルのパスを指定します。HTTP または HTTPS サーバーに存在するファイルの URL を指定できます。 HTTP サーバーの場合、構文は次のとおりです。
	<pre>http://server_name:IP_address/jumpstart_dir_path/ compressed_config_file&proxy_info</pre>
	<ul style="list-style-type: none"> ■ <code>sysidcfg</code> ファイルを圧縮構成ファイルに含めた場合は、次の例に示すようにそのファイルを含むサーバーの IP アドレスを指定する必要があります。
	<pre>kernel /I86pc.Solaris_11.8/multiboot install \ http://192.168.2.1/jumpstart/config.tar \ -B install_media=192.168.2.1/export/Solaris_11.8/boot \ module /I86PC.Solaris_11.8/x86.new</pre>
	<ul style="list-style-type: none"> ■ 圧縮構成ファイルをファイアウォールで防御された HTTP サーバー上の圧縮構成ファイルに保存した場合は、ブート時に <code>proxy</code> 指示子を使用する必要があります (ファイルが配置されたサーバーの IP アドレスを指定する必要はない)。プロキシサーバーの IP アドレスは次のように指定します。
	<pre>kernel /I86pc.Solaris_11.8/multiboot install \ http://www.shadow.com/jumpstart/config.tar&proxy=131.141.6.151 \ -B install_media=192.168.2.1/export/Solaris_11.8/boot \ module /I86PC.Solaris_11.8/x86.new</pre>
	<ul style="list-style-type: none"> ■ <code>ask</code> - 圧縮構成ファイルの場所の入力を求めるプロンプトを表示するように、インストールプログラムを指定します。システムがブートしてネットワークに接続されると、入力を求めるプロンプトが表示されます。このオプションを使用すると、完全に自動化された JumpStart インストールを行うことはできません。
	<p>Return キーを押してこのプロンプトへの入力を省略すると、Solaris インストールプログラムはネットワークパラメータを対話的に構成します。インストールプログラムは次に、圧縮構成ファイルの場所を尋ねるプロンプトを表示します。</p>
	<p>次の例では、カスタム JumpStart を実行し、DVD メディアからブートします。システムがネットワークに接続されると、構成ファイルの場所の入力を求めるプロンプトが表示されます。</p>
	<pre>kernel /boot/multiboot kernel/unix install ask -B \ install_media=192.168.2.1:export/sol_11_x86/boot module \ /I86PC.Solaris_11.8_</pre>

表 6-4 GRUB メニューのブートコマンドリファレンス (続き)

オプション	説明
<i>options</i>	<ul style="list-style-type: none"><li data-bbox="315 239 1283 390">■ <code>dhcp</code> - システムのブートに必要なネットワークインストール情報の取得に DHCP サーバーを使用することを指定します。このオプションは JumpStart インストールには必要ありません。<code>dhcp</code> で DHCP サーバーの使用を指定しないと、<code>/etc/bootparams</code> ファイル、またはネームサービスの <code>bootparams</code> データベースが使用されます。たとえば、静的 IP アドレスを保持する場合には、<code>dhcp</code> を指定しません。次に例を示します。 <pre>kernel /I86pc.Solaris_11.8/multiboot install \ dhcp -B install_media=192.168.2.1:/export/Solaris_11.8/ \ boot module /I86PC.Solaris_11.8/x86.new</pre><li data-bbox="315 512 1283 638">■ <code>nowin</code> オプションおよび <code>text</code> オプションは、JumpStart インストールには適用されません。これらのオプションは、対話式インストールで役立ちます。詳細は、『Solaris 10 5/08 インストールガイド (基本編)』の「GRUB 付き Solaris インストールプログラムを使用してインストールまたはアップグレードを行う方法」を参照してください。

カスタム JumpStart によるインストール (例)

この章では、カスタム JumpStart インストールを使用して SPARC システムと x86 システムに Solaris ソフトウェアを設定しインストールする例を示します。

- 105 ページの「サイトの設定例」
- 107 ページの「インストールサーバーの作成」
- 108 ページの「x86: マーケティングシステム用のブートサーバーの作成」
- 109 ページの「JumpStart ディレクトリの作成」
- 109 ページの「JumpStart ディレクトリの共有」
- 110 ページの「SPARC: エンジニアリング部のプロファイルの作成」
- 110 ページの「x86: マーケティング部のプロファイルの作成」
- 111 ページの「rules ファイルの更新」
- 112 ページの「rules ファイルの妥当性を検査する」
- 112 ページの「SPARC: ネットワーク上でインストールするためのエンジニアリングシステムの設定」
- 113 ページの「x86: ネットワーク上でインストールするためのマーケティングシステムの設定」
- 114 ページの「SPARC: エンジニアリングシステムのブートと Solaris ソフトウェアのインストール」
- 114 ページの「x86: マーケティングシステムのブートと Solaris ソフトウェアのインストール」

サイトの設定例

図 7-1 は、この例で使用するサイトの設定を示しています。

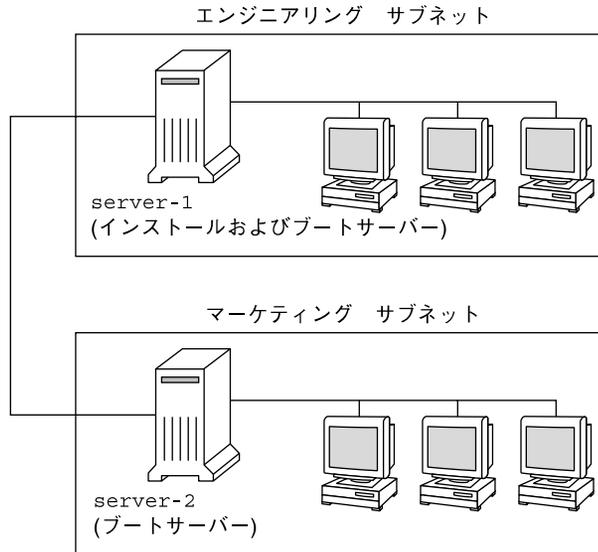


図 7-1 サイトの設定例

この例におけるサイトの状況は次のとおりです。

- SPARC: エンジニアリング部は、グループ固有のサブネット上にあります。このグループでは、ソフトウェアを開発するために SPARCstation™ システムを使用しています。
- x86: マーケティング部も、グループ固有のサブネット上にあります。このグループでは、文書処理、表計算、およびその他のオフィスツールを実行するために x86 システムを使用しています。
- このサイトでは NIS を使用しています。システムの Ethernet アドレス、IP アドレス、およびホスト名は NIS マップに事前設定されています。サイトのサブネットマスク、日付と時刻、および地域の情報も NIS マップに事前設定されています。

注-マーケティングシステムの周辺装置は、`sysidcfg` ファイルに事前設定されています。

- エンジニアリングシステムとマーケティングシステムの両方とも、Solaris 10 5/08 ソフトウェアをネットワーク上でインストールします。

インストールサーバーの作成

両方のグループとも Solaris 10 5/08 ソフトウェアをネットワーク上でインストールする必要があるため、server-1を両方のグループのインストールサーバーにします。server-1 ローカルディスク (/export/install ディレクトリ内)にイメージをコピーするには、`setup_install_server(1M)` コマンドを使用します。次のいずれかのメディアからイメージをコピーします。

- Solaris SOFTWARE CD と Solaris LANGUAGES CD
- Solaris Operating System DVD

イメージは、ディスクから空のディレクトリにコピーする必要があります。次の例では、それぞれのディレクトリ名を `sparc_10` および `x86_10` としています。

例 7-1 SPARC: Solaris 10 5/08 CD のコピー

server-1 の CD-ROM ドライブに Solaris SOFTWARE - 1 CD (SPARC 版) を挿入し、次のコマンドを入力します。

```
server-1# mkdir -p /export/install/sparc_10
server-1# cd /CD_mount_point/Solaris_10/Tools
server-1# ./setup_install_server /export/install/sparc_10
```

server-1 の CD-ROM ドライブに Solaris SOFTWARE - 2 CD (SPARC 版) を挿入し、次のコマンドを入力します。

```
server-1# cd /CD_mount_point/Solaris_10/Tools
server-1# ./add_to_install_server /export/install/sparc_10
```

インストールする Solaris SOFTWARE ごとに、前記のコマンドを実行します。

server-1 の CD-ROM ドライブに Solaris LANGUAGES CD (SPARC 版) を挿入し、次のコマンドを入力します。

```
server-1# cd /CD_mount_point/Solaris_10/Tools
server-1# ./add_to_install_server /export/install/sparc_10
```

例 7-2 x86: Solaris 10 5/08 CD のコピー

server-1 の CD-ROM ドライブに Solaris SOFTWARE - 1 CD (x86 版) を挿入し、次のコマンドを入力します。

```
server-1# mkdir -p /export/install/x86_10
server-1# cd /CD_mount_point/Solaris_10/Tools
server-1# ./setup_install_server /export/install/x86_10
```

server-1 の CD-ROM ドライブに Solaris SOFTWARE - 2 CD (x86 版) を挿入し、次のコマンドを入力します。

例 7-2 x86: Solaris 10 5/08 CD のコピー (続き)

```
server-1# cd /CD_mount_point/Solaris_10/Tools
server-1# ./add_to_install_server /export/install/x86_10
```

インストールする Solaris SOFTWARE ごとに、前記のコマンドを実行します。

server-1 の CD-ROM ドライブに Solaris LANGUAGES CD (x86 版) を挿入し、次のコマンドを入力します。

```
server-1# cd /CD_mount_point/Solaris_10/Tools
server-1# ./add_to_install_server /export/install/x86_10
```

例 7-3 SPARC: Solaris 10 5/08 DVD のコピー

server-1 の DVD-ROM ドライブに Solaris Operating System DVD (SPARC 版) を挿入し、次のコマンドを入力します。

```
server-1# mkdir -p /export/install/sparc_10
server-1# cd /DVD_mount_point/Solaris_10/Tools
server-1# ./setup_install_server /export/install/sparc_10
```

例 7-4 x86: Solaris Operating System DVD (x86 版) のコピー

server-1 の DVD-ROM ドライブに Solaris Operating System DVD (x86 版) を挿入し、次のコマンドを入力します。

```
server-1# mkdir -p /export/install/x86_10
server-1# cd /DVD_mount_point/Solaris_10/Tools
server-1# ./setup_install_server /export/install/x86_10
```

x86: マーケティングシステム用のブートサーバーの作成

異なるサブネット上のインストールサーバーからシステムをブートすることはできません。そこで、server-2 をマーケティング部のサブネット上のブートサーバーにします。setup_install_server(1M) コマンドを使って、Solaris Operating System DVD (x86 版) または Solaris SOFTWARE - 1 CD (x86 版) からブートソフトウェアをコピーします。ブートソフトウェアは、server-2 ローカルディスクの /export/boot ディレクトリにコピーされます。

メディアを選択し、ローカルディスクにブートソフトウェアをインストールします。

- server-2 の CD-ROM ドライブに Solaris SOFTWARE - 1 CD (x86 版) を挿入する場合は、次のコマンドを入力してください。

```
server-2# cd /CD_mount_point/Solaris_10/Tools
server-2# ./setup_install_server -b /export/boot
```

- server-2 の DVD-ROM ドライブに Solaris Operating System DVD (x86 版) を挿入する場合は、次のコマンドを入力してください。

```
server-2# cd /DVD_mount_point/Solaris_10/Tools
server-2# ./setup_install_server -b /export/boot
```

setup_install_server コマンドの -b オプションは、setup_install_server が /export/boot というディレクトリにブート情報をコピーするように指定します。

JumpStart ディレクトリの作成

インストールサーバーとブートサーバーの設定が完了したら、server-1 上に JumpStart ディレクトリを作成します。ネットワーク上のほかの任意のシステム上に作成することもできます。このディレクトリには、Solaris ソフトウェアのカスタム JumpStart インストールに必要なファイルが入っています。このディレクトリを設定するには、/export/install にコピーされている Solaris Operating System DVD イメージまたは Solaris SOFTWARE-1 CD イメージからサンプルディレクトリをコピーします。

```
server-1# mkdir /jumpstart
server-1# cp -r /export/install/sparc_10/Solaris_10/Misc/jumpstart_sample /jumpstart
```

JumpStart ディレクトリの共有

ネットワーク上のシステムから rules ファイルとプロファイルにアクセスできるようにするには、/jumpstart ディレクトリを共有にします。ディレクトリを共有するためには、/etc/dfs/dfstab ファイルに次の行を追加します。

```
share -F nfs -o ro,anon=0 /jumpstart
```

次に、コマンド行に shareall コマンドを入力します。

```
server-1# shareall
```

SPARC: エンジニアリング部のプロファイルの作成

エンジニアリングシステム用として、`/jumpstart` ディレクトリに `eng_prof` という名前のファイルを作成します。`eng_prof` ファイルには次のエントリがあり、エンジニアリング部のシステムにインストールされる Solaris 10 5/08 ソフトウェアを定義しています。

```
install_type  initial_install
system_type   standalone
partitioning  default
cluster       SUNWCprog
filesystems   any 512 swap
```

この例のプロファイルは、次のインストール情報を指定しています。

<code>install_type</code>	アップグレードではなく、初期インストールを行います。
<code>system_type</code>	エンジニアリングシステムはスタンドアロンシステムです。
<code>partitioning</code>	JumpStart ソフトウェアは、デフォルトのディスクパーティションを使用して、エンジニアリングシステムに Solaris ソフトウェアをインストールします。
<code>cluster</code>	開発者システムサポートソフトウェアグループをインストールします。
<code>filesystems</code>	エンジニアリング部の各システムが、512M バイトのスワップ空間を持ちます。

x86: マーケティング部のプロファイルの作成

マーケティングシステム用として、`/jumpstart` ディレクトリに `marketing_prof` という名前のファイルを作成します。`marketing_prof` ファイルには次のエントリがあり、マーケティング部のシステムにインストールされる Solaris 10 5/08 ソフトウェアを定義しています。

```
install_type  initial_install
system_type   standalone
partitioning  default
cluster       SUNWCuser
package       SUNWaudio
```

この例のプロファイルは、次のインストール情報を指定しています。

<code>install_type</code>	アップグレードではなく、初期インストールを行います。
<code>system_type</code>	マーケティングシステムはスタンドアロンシステムです。

partitioning	JumpStart ソフトウェアが、デフォルトのディスクパーティションを使用して、マーケティングシステムに Solaris をインストールします。
cluster	エンドユーザーシステムサポートソフトウェアグループをインストールします。
package	オーディオツールのデモソフトウェアパッケージを各システムに追加します。

rules ファイルの更新

次に、rules ファイルにルールを追加する必要があります。Solaris インストールプログラムは、カスタム JumpStart インストール中に、これらのルールを使用して、各システムに正しいインストール(プロファイル)を選択します。

このサイトでは、各部署は独自のサブネットとネットワークアドレスを持っています。エンジニアリング部はサブネット 255.222.43.0 にあります。マーケティング部はサブネット 255.222.44.0 にあります。この情報を使用すれば、Solaris 10 5/08 によるエンジニアリングシステムとマーケティングシステムのインストール方法を制御できます。/jumpstart ディレクトリ内の rules ファイルを編集して、サンプルのルールをすべて削除し、次の行をファイルに追加します。

```
network 255.222.43.0 - eng_prof -  
network 255.222.44.0 - marketing_prof -
```

これらのルールは、基本的に、255.222.43.0 ネットワーク上のシステムに Solaris 10 5/08 ソフトウェアが eng_prof プロファイルを使用してインストールされることを示しています。255.222.44.0 ネットワーク上のシステムには、Solaris 10 5/08 ソフトウェアが marketing_prof プロファイルを使用してインストールされます。

注 - この例のルールを使用すると、eng_prof と marketing_prof を使って Solaris 10 5/08 ソフトウェアをインストールするシステムを、ネットワークアドレスで識別できます。このほかにも、ホスト名、メモリーサイズ、またはモデルタイプをルールのキーワードとして使用できます。rules ファイルで使用できるキーワードの完全な一覧については、表 8-1 を参照してください。

rules ファイルの妥当性を検査する

rules ファイルとプロファイルの設定が完了したら、check スクリプトを実行して、これらのファイルが正常に設定されていることを検証します。

```
server-1# cd /jumpstart
server-1# ./check
```

check スクリプトによってエラーが検出されない場合は、rules.ok ファイルが作成されます。

SPARC: ネットワーク上でインストールするためのエンジニアリングシステムの設定

/jumpstart ディレクトリとファイルの設定が完了したら、インストールサーバー server-1 上で add_install_client コマンドを実行して、インストールサーバーから Solaris ソフトウェアをインストールするように、エンジニアリングシステムを設定します。server-1 は、エンジニアリング部のサブネットにとってはブートサーバーにも相当します。

```
server-1# cd /export/install/sparc_10/Solaris_10/Tools
server-1# ./add_install_client -c server-1:/jumpstart host-eng1 sun4u
server-1# ./add_install_client -c server-1:/jumpstart host-eng2 sun4u
```

この add_install_client コマンドで使用しているオプションの意味は次のとおりです。

-c サーバー (server-1) と JumpStart ディレクトリへのパス (/jumpstart) を指定します。NFS を使用している場合はこのオプションを使用します。

注-NFS を使用していない場合は、次のコマンドを使用して JumpStart ディレクトリへのパスを指定します。

- **SPARC** システムの場合は、boot コマンドにパスを指定します
 - **x86** システムの場合は、GRUB メニューエントリを編集することによってパスを指定します
-

host-eng1 エンジニアリング部のシステム名です。

host-eng2 エンジニアリング部の別のシステム名です。

sun4u server-1 をインストールサーバーとして使用するシステムのプラットフォームグループを指定します。これは、Ultra 5 システム用のプラット

フォームグループです。

x86: ネットワーク上でインストールするためのマーケティングシステムの設定

次に、ブートサーバー (server-2) で `add_install_client` コマンドを使用します。このコマンドはマーケティングシステムに対して、ブートサーバーからブートし、インストールサーバー (server-1) から Solaris ソフトウェアをインストールするように設定します。

```
server-2# cd /marketing/boot-dir/Solaris_10/Tools
server-2# ./add_install_client -s server-1:/export/install/x86_10 \
-c server-1:/jumpstart host-mkt1 i86pc
server-2# ./add_install_client -s server-1:/export/install/x86_10 \
-c server-1:/jumpstart host-mkt2 i86pc
server-2# ./add_install_client -d -s server-1:/export/install/x86_10 \
-c server-1:/jumpstart SUNW.i86pc i86pc
server-2# ./add_install_client -c server-1:/jumpstart host-mkt1 sun4u
server-2# ./add_install_client -c server-1:/jumpstart host-mkt2 sun4u
```

`add_install_client` コマンドで使用されるオプションの意味は次のとおりです。

- d クライアントが DHCP を使ってネットワークインストールパラメータを取得するように指定します。このオプションは PXE ネットワークブートを使用してネットワークからブートするクライアントには必ず指定します。PXE ネットワークブートを使用しないネットワークブートクライアントには、`-d` は省略できます。
- s インストールサーバー (server-1) と Solaris ソフトウェアへのパス (/export/install/x86_10) を指定します。
- c サーバー (server-1) と JumpStart ディレクトリへのパス (/jumpstart) を指定します。NFS を使用している場合はこのオプションを使用します。

注 - NFS を使用していない場合は、次のコマンドを使用して JumpStart ディレクトリへのパスを指定します。

- SPARC システムの場合は、`boot` コマンドにパスを指定します
 - x86 システムの場合は、GRUB メニューエントリを編集することによってパスを指定します
-

host-mkt1 マーケティング部のシステム名です。

host-mkt2	マーケティング部の別のシステム名です。
sun4u	server-1 をインストールサーバーとして使用するシステムのプラットフォームグループを指定します。これは、Ultra 5 システム用のプラットフォームグループです。
SUNW.i86pc	すべての Solaris x86 クライアントの DHCP クラス名です。1 回のコマンド実行ですべての Solaris x86 DHCP クライアントを構成する場合は、このクラス名を使用してください。
i86pc	このブートサーバーを使用するシステムのプラットフォームグループを指定します。このプラットフォーム名は x86 システムを意味します。

SPARC: エンジニアリングシステムのブートと Solaris ソフトウェアのインストール

サーバーとファイルの設定が完了したら、エンジニアリングシステムをブートします。各システムの ok (PROM) プロンプトに、次の boot コマンドを入力してください。

```
ok boot net - install
```

Solaris OS がエンジニアリング部のシステムに自動的にインストールされます。

x86: マーケティングシステムのブートと Solaris ソフトウェアのインストール

次のいずれかを使って、システムをブートできます。

- Solaris SOFTWARE - 1 CD (x86 版)
- Solaris Operating System DVD (x86 版)
- ネットワーク (PXE ネットワークブートを使用)

Solaris ソフトウェアがマーケティング部のシステムに自動的にインストールされます。

カスタム JumpStart (リファレンス)

この章では、rules ファイル、プロファイル、および開始/終了スクリプトで使用できるキーワードと値のリストを示します。

- 115 ページの「ルールキーワードと値の説明」
- 119 ページの「プロファイルキーワードと値」
- 164 ページの「カスタム JumpStart の環境変数」
- 166 ページの「プローブキーワードと値」

ルールキーワードと値の説明

表 8-1 に、rules ファイルで使用できるキーワードと値の説明を記載します。rules ファイルの詳細な作成方法については、36 ページの「rules ファイルの作成」を参照してください。

表 8-1 ルールキーワードとルール値の説明

キーワード	値	説明
any	マイナス記号 (-)	あらゆるものに一致することを意味します。
arch	<i>processor_type</i> <i>processor_type</i> に有効な値: <ul style="list-style-type: none">■ SPARC: sparc■ x86: i386	システムのプロセッサタイプを照合します。 システムのプロセッサタイプは、uname -p コマンドで調べることができます。

表 8-1 ルールキーワードとルール値の説明 (続き)

キーワード	値	説明
disksize	<p><i>actual_disk_name size_range</i></p> <p><i>actual_disk_name</i> - cxytdz の形式のディスク名 (c0t3d0、c0d0 など)、または特別なワード rootdisk。rootdisk を使用する場合、照合するディスクは次の順番で決定されます。</p> <ul style="list-style-type: none"> ■ SPARC: 事前にインストールされているブートイメージ (つまり、ファクトリ JumpStart がインストールされた新しい SPARC システム) を持つディスク ■ c0t3d0s0 ディスク (存在する場合) ■ 最初に利用可能なディスク (カーネルのプロンプト順で検索される) <p><i>size_range</i> - ディスクのサイズ。M バイト単位の範囲 (x-x) で指定する必要があります。</p> <p>注 - <i>size_range</i> を計算するときは、1M バイトが 1,048,576 バイトであることに注意してください。「535M バイト」ディスクと明記されているディスクでも、ディスク空間が 510M バイトしかない場合があります。535,000,000 / 1,048,576 = 510 であるため、JumpStart プログラムは、「535M バイト」ディスクを 510M バイトのディスクと見なします。この「535M バイト」ディスクは、「530-550」の <i>size_range</i> には一致しません。</p>	<p>システムのディスクの名前とサイズを照合します (M バイト単位)。</p> <p>例:</p> <pre>disksize c0t3d0 250-300</pre> <p>この例では、JumpStart プログラムは c0t3d0 というシステムディスクの照合を試みません。このディスクは 250M から 300M バイトの情報を保持できます。</p> <p>例:</p> <pre>disksize rootdisk 750-1000</pre> <p>この例では、JumpStart プログラムは次の順でディスクを照合します。</p> <ol style="list-style-type: none"> 1. 事前にインストールされているブートイメージが格納されたシステムディスク 2. c0t3d0s0 ディスク (存在する場合) 3. 750M から 1G バイトの情報を格納できる、最初に利用可能なディスク
domainname	<i>actual_domain_name</i>	<p>システムのドメイン名を照合します。ドメイン名でネームサービスが情報を判別する方法を制御します。</p> <p>システムがインストール済みの場合、domainname コマンドによりシステムのドメイン名を表示できます。</p>
hostaddress	<i>actual_IP_address</i>	システムの IP アドレスを照合します。
hostname	<i>actual_host_name</i>	<p>システムのホスト名を照合します。</p> <p>システムがインストール済みの場合、uname -n コマンドによりシステムのホスト名を表示できます。</p>

表 8-1 ルールキーワードとルール値の説明 (続き)

キーワード	値	説明
installed	<p><i>slice version</i></p> <p><i>slice - cwtx dysz</i> の形式のディスクスライス名 (c0t3d0s5 など)、あるいは特別なワード any または rootdisk.any を使用すると、システムに接続されたどのディスクも照合します (カーネルのプロープ順)。rootdisk を使用する場合、照合するディスクは次の順番で決定されます。</p> <ul style="list-style-type: none"> ■ SPARC: 事前にインストールされているブートイメージ (つまり、ファクトリ JumpStart がインストールされた新しい SPARC システム) を持つディスク ■ c0t3d0s0 ディスク (存在する場合) ■ 最初に利用可能なディスク (カーネルのプロープ順で検索される) <p><i>version</i> - バージョン名、あるいは特別なワード any または upgrade.any を使用すると、Solaris または SunOS リリースのどれとでも照合します。upgrade を使用すると、サポートされていてアップグレード可能なすべての Solaris リリースと照合します。</p> <p>Solaris リリースは検出されたが、バージョンを判断できないという場合は、返されるバージョンは SystemV になります。</p>	<p>Solaris ソフトウェアの特定バージョンに対応するルートファイルシステムが存在するディスクを照合します。</p> <p>例:</p> <p>installed c0t3d0s1 Solaris 10</p> <p>この例では、c0t3d0s1 に Solaris のルートファイルシステムを持つシステムを照合しています。</p>
karch	<p><i>actual_platform_group</i></p> <p>有効な値は、sun4u、i86pc、および prep。システムと対応するプラットフォームグループのリストは、『Solaris Sun ハードウェアマニュアル』 (http://docs.sun.com) に記載されています。</p>	<p>システムのプラットフォームグループを照合します。</p> <p>システムがインストール済みの場合は、arch -k コマンドまたは uname -m コマンドにより、システムのプラットフォームグループを表示できます。</p>
memsize	<p><i>physical_mem</i></p> <p>値は M バイト単位の範囲 (x-x) または 1 つの M バイト値で指定します。</p>	<p>システムの物理メモリーサイズを照合します (M バイト単位)。</p> <p>例:</p> <p>memsize 64-128</p> <p>この例では、64M から 128M バイトの物理メモリーサイズを持つシステムと照合しています。</p> <p>システムがインストール済みの場合は、prtconf コマンド (2 行目) によりシステムの物理メモリーサイズを表示できます。</p>

表 8-1 ルールキーワードとルール値の説明 (続き)

キーワード	値	説明
model	<i>actual_platform_name</i>	<p>システムのプラットフォーム名を照合します。有効なプラットフォーム名のリストは、『Solaris Sun ハードウェアマニュアル』(http://docs.sun.com)に記載されています。</p> <p>インストール済みのシステムのプラットフォーム名を見つけるには、<code>uname -i</code> コマンドか <code>prtconf</code> コマンド (5 行目) の出力を使用します。</p> <p>注- <i>actual_platform_name</i> にスペースが含まれている場合は、スペースを下線 (<code>_</code>) で置き換える必要があります。</p> <p>例:</p> <p>SUNW,Sun_4_50</p>
network	<i>network_num</i>	<p>システムのネットワーク番号を照合します。これは JumpStart プログラムが、システムの IP アドレスとサブネットマスクの論理積をとって判別します。</p> <p>例:</p> <p>network 192.168.2.0</p> <p>この例では、IP アドレスが 192.168.2.8 のシステムを照合します (サブネットマスクが 255.255.255.0 の場合)。</p>
osname	Solaris_x	<p>システムにすでにインストールされている Solaris のバージョンを照合します。</p> <p>例:</p> <p>osname Solaris 10</p> <p>この例では、JumpStart プログラムは Solaris 10 5/08 OS がすでにインストールされているシステムを照合します。</p>

表 8-1 ルールキーワードとルール値の説明 (続き)

キーワード	値	説明
probe	<i>probe_keyword</i>	<p>有効なプローブキーワードまたは有効なカスタムプローブキーワード。</p> <p>例:</p> <pre>probe disks</pre> <p>この例は、システムのディスクサイズ (M バイト) をカーネルプローブ順に返します (例: SPARC システムでは <code>c0t3d0s1</code>、<code>c0t4d0s0</code>)。JumpStart プログラムは環境変数 <code>SI_DISKLIST</code>、<code>SI_DISKSIZE</code>、<code>SI_NUMDISKS</code>、および <code>SI_TOTALDISK</code> を設定します。</p> <p>注 - <code>probe</code> キーワードには、属性を照合せず、プロファイルを実行しないという特徴があります。<code>probe</code> キーワードは、値を返すだけです。したがって、<code>probe</code> ルールキーワードで、開始スクリプト、プロファイル、および終了スクリプトは指定できません。</p> <p>プローブキーワードについては、第 5 章カスタムルールおよびプローブキーワードの作成 (作業) を参照してください。</p>
totaldisk	<i>size_range</i>	<p>システムのディスク容量の全体量 (M バイト単位) を照合します。ディスク容量の全体量には、システムに接続されている使用可能なディスクがすべて含まれます。</p> <p>例:</p> <pre>totaldisk 300-500</pre> <p>この例で JumpStart プログラムは、合計 300M から 500M バイトのディスク容量を持つシステムを照合しています。</p> <p>注 - <i>size_range</i> を計算するときは、1M バイトが 1,048,576 バイトであることに注意してください。「535M バイト」ディスクと明記されているディスクでも、ディスク空間が 510M バイトしかない場合があります。535,000,000 / 1,048,576 = 510 であるため、JumpStart プログラムは、「535M バイト」ディスクを 510M バイトのディスクと見なします。この「535M バイト」ディスクは、「530-550」の <i>size_range</i> には一致しません。</p>

プロファイルキーワードと値

次の節では、プロファイルで利用できるプロファイルキーワードとプロファイル値を説明します。プロファイルの詳しい作成方法については、[40 ページの「プロファイルの作成」](#) を参照してください。

プロファイルキーワードのクイックリファレンス

表 8-2 では、インストール方法に基づいて、どのキーワードを使用できるかをすばやく確認できます。プロファイルキーワードの説明で特に注記されていないかぎり、プロファイルキーワードは初期インストールオプションだけで使用できます。

表 8-2 プロファイルキーワードの概要

プロファイルキーワード	インストール方法				
	スタンドアロンシステム(ネットワークに接続されていない)	スタンドアロンシステム(ネットワークに接続されている)またはサーバー	OS サーバー	アップグレード	ディスク容量の再配置を使用するアップグレード
archive_location (Solaris フラッシュアーカイブをインストールする場合)	○	○			
backup_media					○
boot_device	○	○	○		
bootenv createbe	○	○	○		
client_arch			○		
client_root			○		
client_swap			○		
cluster (ソフトウェアグループを追加する場合)	○	○	○		
cluster (クラスタを追加または削除する場合)	○	○	○	○	○
dontuse	○	○	○		
fdisk (x86 のみ)	○	○	○		
filesys (リモートファイルシステムをマウントする場合)		○	○		
filesys (ローカルファイルシステムを作成する場合)	○	○	○		
filesys (ミラー化されたファイルシステムを作成する場合)	○	○	○		
forced_deployment (Solaris フラッシュ差分アーカイブをインストールする場合)	○	○			

表 8-2 プロファイルキーワードの概要 (続き)

プロファイルキーワード	インストール方法				
	スタンドアロンシステム(ネットワークに接続されていない)	スタンドアロンシステム(ネットワークに接続されている)またはサーバー	OS サーバー	アップグレード	ディスク容量の再配置を使用するアップグレード
geo	○	○	○	○	○
install_type	○	○	○	○	○
layout_constraint					○
local_customization (Solaris フラッシュアーカイブをインストールする場合)	○	○			
locale	○	○	○	○	○
metadb (状態データベースの複製を作成する場合)	○	○	○		
no_master_check (Solaris フラッシュ差分アーカイブをインストールする場合)	○	○			
no_content_check (Solaris フラッシュ差分アーカイブをインストールする場合)	○	○			
num_clients			○		
package	○	○	○	○	○
partitioning	○	○	○		
patch	○	○	○	○	○
root_device	○	○	○	○	○
system_type	○	○	○		
usedisk	○	○	○		

プロファイルキーワードの説明と例

archive_location プロファイルキーワード

archive_location *retrieval_type location*

retrieval_type *retrieval_type* と *location* の値は、Solaris フラッシュアーカイブの格納場所によって異なります。*retrieval_type* と *location* に使用できる値や、*archive_location* キーワードの使用例については、次の各項を参照してください。

- 122 ページの「NFS サーバーに格納されたアーカイブ」
- 123 ページの「HTTP または HTTPS サーバーに格納されたアーカイブ」
- 124 ページの「FTP サーバーに格納されたアーカイブ」
- 125 ページの「ローカルテープに格納されたアーカイブ」
- 126 ページの「ローカルデバイスに格納されたアーカイブ」
- 127 ページの「ローカルファイルに格納されたアーカイブ」

location 場所の詳細については、次の各節を参照してください。



注意 - 非大域ゾーンがインストールされていると、Solaris フラッシュアーカイブは正常に作成できません。Solaris フラッシュ機能には Solaris ゾーン区分技術との互換性はありません。Solaris フラッシュアーカイブを作成する場合、そのアーカイブの配備条件が次のいずれかの場合は、作成されたアーカイブは正しくインストールされません。

- アーカイブが非大域ゾーンに作成された場合
- アーカイブが、非大域ゾーンがインストールされている大域ゾーンに作成された場合

NFS サーバーに格納されたアーカイブ

アーカイブがネットワークファイルシステム (NFS) サーバーに格納されている場合は、*archive_location* プロファイルキーワードに次の構文を使用します。

```
archive_location nfs server_name:/path/filename retry n
```

server_name アーカイブを格納するサーバーの名前です。

path 指定したサーバーから取得するアーカイブの場所です。このパスに \$HOST を含めると、Solaris フラッシュインストールユーティリティーは \$HOST をインストール先のクローンシステムの名前に置き換えます。

filename Solaris フラッシュアーカイブファイルの名前です。

retry n 省略可能なキーワードです。*n* は、Solaris フラッシュユーティリティーがアーカイブのマウントを試みる最大回数です。

例 8-1 NFS サーバーに格納されたアーカイブ

```
archive_location nfs golden:/archives/usrarchive
```

```
archive_location nfs://golden/archives/usrarchive
```

HTTP または HTTPS サーバーに格納されたアーカイブ

アーカイブが HTTP サーバーに格納されている場合は、`archive_location` プロファイルキーワードに次の構文を使用します。

```
archive_location http://server_name:port/path/filename optional_keywords
```

アーカイブが HTTPS サーバーに格納されている場合は、`archive_location` プロファイルキーワードに次の構文を使用します。

```
archive_location https://server_name:port/path/filename optional_keywords
```

<code>server_name</code>	アーカイブを格納するサーバーの名前です。
<code>port</code>	オプションのポートです。 <code>port</code> は、ポート番号でも、実行時に決定されるポート番号を持つ TCP サービスの名前でもかまいません。 ポートを指定しない場合、Solaris フラッシュインストールユーティリティはデフォルトの HTTP ポート番号、80 を使用します。
<code>path</code>	指定したサーバーから取得するアーカイブの場所です。このパスに <code>\$HOST</code> を含めると、Solaris フラッシュインストールユーティリティは <code>\$HOST</code> をインストール先のクローンシステムの名前に置き換えます。
<code>filename</code>	Solaris フラッシュアーカイブファイルの名前です。
<code>optional_keywords</code>	Solaris フラッシュアーカイブを HTTP サーバーから取得するときに指定できるオプションのキーワードです。

表 8-3 `archive_location` HTTP で使用可能なオプションのキーワード

キーワード	値の定義
<code>auth basic user_name password</code>	アーカイブがパスワード保護された HTTP サーバーに格納されている場合は、その HTTP サーバーへのアクセスに必要なユーザー名とパスワードをプロファイルに含める必要があります。 注-カスタム JumpStart で使用しようとするプロファイルにおいてこの認証方法を使用することは、リスクを伴います。これは、承認されていないユーザーが、パスワードが入ったプロファイルにアクセスできる可能性があるためです。

表 8-3 archive_location HTTP で使用可能なオプションのキーワード (続き)

キーワード	値の定義
timeout <i>min</i>	<p>timeout キーワードには、HTTP サーバーからのデータ受信を待機する最大の時間を分単位で指定できます。この時間に達すると、接続が切断され、再接続のあと、再開されます。timeout 値として0(ゼロ)を指定すると、再接続は行われません。</p> <ul style="list-style-type: none"> ■ タイムアウトによる再接続が発生すると、Solaris フラッシュインストールユーティリティーはアーカイブを取得した最後の位置からインストールの再開を試みます。この位置での Solaris フラッシュインストールの再開が不可能な場合、アーカイブの初めから再度データの取得が行われ、タイムアウト前に取得されたデータは破棄されます。 ■ パッケージのインストール中にタイムアウトによる再接続が発生すると、パッケージの先頭からインストールし直されます。タイムアウト以前に取得されたデータは破棄されます。
proxy <i>host:port</i>	<p>proxy キーワードを使用して、プロキシホストとプロキシポートを指定できます。プロキシホストを使用すると、ファイアウォール越しに Solaris フラッシュアーカイブを取得できます。proxy キーワードを指定する場合は、プロキシポートを指定する必要があります。</p>

例 8-2 HTTP または HTTPS サーバーに格納されたアーカイブ

```
archive_location http://silver/archives/usrarchive.flar timeout 5
```

auth basic *user_name password* キーワードの例:

```
archive_location http://silver/archives/usrarchive.flar timeout 5 user1 secret
```

FTP サーバーに格納されたアーカイブ

アーカイブが FTP サーバーに格納されている場合は、archive_location プロファイルキーワードに次の構文を使用します。

```
archive_location ftp://user_name:password@server_name:port/path/filename optional_keywords
```

user_name:password プロファイルファイルにおいて、FTP サーバーにアクセスするために必要なユーザー名とパスワードです。

server_name アーカイブを格納するサーバーの名前です。

port オプションのポートです。port は、ポート番号でも、実行時に決定されるポート番号を持つ TCP サービスの名前でもかまいません。

ポートを指定しない場合、Solaris フラッシュインストールユーティリティーはデフォルトの FTP ポート番号 21 を使用します。

<i>path</i>	指定したサーバーから取得するアーカイブの場所です。このパスに \$HOST を含めると、Solaris フラッシュインストールユーティリティーは \$HOST をインストール先のクローンシステムの名前に置き換えます。
<i>filename</i>	Solaris フラッシュアーカイブファイルの名前です。
<i>optional_keywords</i>	Solaris フラッシュアーカイブを FTP サーバーから取得するときに指定できるオプションのキーワードです。

表 8-4 archive_location FTP で使用可能なオプションのキーワード

キーワード	値の定義
<i>timeout min</i>	<p>timeout キーワードには、HTTP サーバーからのデータ受信を待機する最長の時間を分単位で指定できます。この時間に達すると、接続が切断され、再接続のあと、再開されます。timeout 値として 0 (ゼロ) を指定すると、再接続は行われません。</p> <ul style="list-style-type: none"> ■ タイムアウトによる再接続が発生すると、Solaris フラッシュインストールユーティリティーはアーカイブを取得した最後の位置からインストールの再開を試みます。この位置での Solaris フラッシュインストールの再開が不可能な場合、アーカイブの初めから再度データの取得が行われ、タイムアウト前に取得されたデータは破棄されます。 ■ パッケージのインストール中にタイムアウトによる再接続が発生すると、パッケージの先頭からインストールし直されます。タイムアウト以前に取得されたデータは破棄されます。
<i>proxy host:port</i>	<p>proxy キーワードを使用して、プロキシホストとプロキシポートを指定できます。プロキシホストを使用すると、ファイアウォール越しに Solaris フラッシュアーカイブを取得できます。proxy キーワードを指定する場合は、プロキシポートを指定する必要があります。</p>

例 8-3 FTP サーバーに格納されたアーカイブ

```
archive_location ftp://user1:secret@silver/archives/usrarchive.flar timeout 5
```

ローカルテープに格納されたアーカイブ

アーカイブがテープに格納されている場合は、archive_location プロファイルキーワードに次の構文を使用します。

```
archive_location local_tape device position
```

device Solaris フラッシュアーカイブを格納したテープドライブの名前です。デバイス名が正規のパスである場合は、Solaris フラッシュインストールユーティリティーはデバイスノードへのパスからアーカイブを取得しま

す。正規のパスでないデバイス名を指定した場合は、Solaris フラッシュインストールユーティリティーはパスに `/dev/rmt/` を加えます。

position アーカイブを保存したテープドライブ上の位置を示します。位置が指定されないと、Solaris フラッシュインストールユーティリティーはテープドライブの現在の位置からアーカイブを取得します。*position* を指定することにより、テープドライブ上でアーカイブの前に開始スクリプトまたは `sysidcfg` ファイルを配置できます。

例 8-4 ローカルテープに格納されたアーカイブ

```
archive_location local_tape /dev/rmt/0n 5
```

```
archive_location local_tape 0n 5
```

ローカルデバイスに格納されたアーカイブ

ファイルシステム指向のランダムアクセスデバイス (フロッピーディスクや DVD など) に Solaris フラッシュアーカイブを格納した場合は、ローカルデバイスから Solaris フラッシュアーカイブを取得できます。`archive_location` プロファイルキーワードには次の構文を使用します。

注 - ローカルテープ用の構文を使用すると、ストリーム指向のデバイス (テープなど) からアーカイブを取得できます。

```
archive_location local_device device path/filename file_system_type
```

device Solaris フラッシュアーカイブを格納したドライブの名前です。デバイス名が正規のパスである場合は、デバイスは直接マウントされます。正規のパスでないデバイス名を指定すると、Solaris フラッシュインストールユーティリティーはパスに `/dev/dsk/` を加えます。

path Solaris フラッシュアーカイブへのパスです (指定したデバイス上のファイルシステムのルートからの相対的なパス)。このパスに `$HOST` を含めると、Solaris フラッシュインストールユーティリティーは `$HOST` をインストール先のクローンシステムの名前に置き換えます。

filename Solaris フラッシュアーカイブファイルの名前です。

file_system_type デバイス上のファイルシステムのタイプを指定します。ファイルシステムのタイプを指定しない場合、Solaris フラッシュインストールユーティリティーは、UFS ファイルシステムのマウントを試みます。UFS のマウントに失敗すると、Solaris フラッシュイン

ストールユーティリティーは HSFS ファイルシステムのマウントを試みます。

例 8-5 ローカルデバイスに格納されたアーカイブ

UFS ファイルシステムとしてフォーマットされているローカルハードディスクからアーカイブを取得するには、次のコマンドを使用します。

```
archive_location local_device c0t0d0s0 /archives/$HOST
```

HSFS ファイルシステムを持つローカル CD-ROM からアーカイブを取得するには、次のコマンドを使用します。

```
archive_location local_device c0t0d0s0 /archives/usrarchive
```

ローカルファイルに格納されたアーカイブ

クローンシステムをブートしたミニルートに格納したローカルファイルのアーカイブを取得できます。カスタム JumpStart インストールを実施する時に、DVD、CD または NFS ベースのミニルートからシステムをブートします。このミニルートからインストールソフトウェアがロードされ、実行されます。したがって、DVD、CD または NFS ベースのミニルートに格納した Solaris フラッシュアーカイブは、ローカルファイルとしてアクセスできます。archive_location プロファイルキーワードには次の構文を使用します。

```
archive_location local_file path/filename
```

path アーカイブの位置です。このパスは、システムを Solaris SOFTWARE - 1 CD または Solaris Operating System DVD からブートしている間、システムにローカルファイルとしてアクセスできるものでなければなりません。Solaris SOFTWARE - 1 CD または Solaris Operating System DVD からブートしている間、システムは、/net などの自動マウントされたディレクトリへアクセスできません。

filename Solaris フラッシュアーカイブファイルの名前です。

例 8-6 ローカルファイルに格納されたアーカイブ

```
archive_location local_file /archives/usrarchive
```

backup_media プロファイルキーワード

```
backup_media type path
```

backup_media は、ディスク容量の再配置が必要なアップグレードオプションだけで使用できます。

`backup_media` は、ディスク容量不足のためにアップグレード中にディスク容量の再配置が必要なファイルシステムのバックアップをとるために使用するメディアを定義します。バックアップ用に複数のテープまたはフロッピーディスクが必要な場合は、アップグレード中にテープまたはフロッピーディスクの挿入を求めるプロンプトが表示されます。

有効な <i>type</i> 値	有効な <i>path</i> 値	仕様
<code>local_tape</code>	<code>/dev/rmt/<i>n</i></code>	アップグレードされるシステムのローカルテープドライブを指定します。 <i>path</i> は、テープドライブのキャラクタ型 (raw) デバイスパスである必要があります。 <i>n</i> はテープドライブの番号です。
<code>local_diskette</code>	<code>/dev/rdisketten</code>	アップグレードされるシステムのローカルフロッピーディスクドライブを指定します。 <i>path</i> は、フロッピーディスクドライブのキャラクタ型 (raw) デバイスパスである必要があります。 <i>n</i> はフロッピーディスクドライブの番号です。 バックアップに使用するフロッピーディスクはフォーマットされていなければなりません。
<code>local_filesystem</code>	<code>/dev/dsk/cwtxdydz</code> <code>/file_system</code>	アップグレードされるシステムのローカルファイルシステムを指定します。アップグレードで変更されるローカルファイルシステムは指定できません。 <i>path</i> は、ディスクスライスのブロックデバイスパスでもかまいません。たとえば、 <code>/dev/dsk/cwtxdydz</code> 内の <code>tx</code> は必須ではありません。あるいは、 <i>path</i> は <code>/etc/vfstab</code> ファイルによってマウントされたファイルシステムへの絶対パスでもかまいません。
<code>remote_filesystem</code>	<code>host:/file_system</code>	リモートシステムの NFS ファイルシステムを指定します。 <i>path</i> は、リモートシステム (<i>host</i>) の名前または IP アドレスと、NFS ファイルシステム (<i>file_system</i>) への絶対パスを含まなければなりません。NFS ファイルシステムは、読み取り権と書き込み権を持っている必要があります。
<code>remote_system</code>	<code>user@host :/directory</code>	リモートシェル (rsh) で到達できるリモートシステム上のディレクトリを指定します。アップグレードされるシステムは、リモートシステムの <code>.rhosts</code> ファイル経由で、リモートシステムにアクセスできなければなりません。 <i>path</i> は、リモートシステム (<i>host</i>) の名前と、そのディレクトリ (<i>directory</i>) への絶対パスを含まなければなりません。ユーザーログイン ID (<i>user</i>) を指定しないと、デフォルトで <code>root</code> が使用されます。

例 8-7 backup_media プロファイルキーワード

```

backup_media local_tape /dev/rmt/0

backup_media local_diskette /dev/rdiskette1

backup_media local_filesystem /dev/dsk/c0t3d0s4

backup_media local_filesystem /export

backup_media remote_filesystem system1:/export/temp

backup_media remote_system user1@system1:/export/temp

```

boot_device プロファイルキーワード

```
boot_device device eeprom
```

boot_device には、JumpStart がルート (/) ファイルシステムをインストールするデバイスと、システムのブートデバイスを指定します。boot_device は、ルート (/) ファイルシステムを指定する filesys キーワードと root_device キーワードに一致する必要があります。

プロファイルに boot_device キーワードを指定しなかった場合、インストール時にデフォルトで次の boot_device キーワードが指定されます。

```
boot_device any update
```

device 次のいずれかの値を使用します。

SPARC: cwtxdysz または cxdysz	JumpStart プログラムによってルート (/) ファイルシステムが配置されるディスクスライス (たとえば c0t0d0s0) です。
x86: cwtxdy または cxdy	JumpStart プログラムによってルート (/) ファイルシステムが配置されるディスク (たとえば c0d0) です。
existing	JumpStart プログラムは、システムの既存のブートデバイスにルート (/) ファイルシステムを格納します。
any	JumpStart プログラムは、ルート (/) ファイルシステムを配置する場所を選択します。システムの既存のブートデバイスを使用する場合もありますが、必要であれば異なるブートデバイスを選択します。

eeprom システムの EEPROM を変更または保存する場合に選択します。

システムの現在のブートデバイスを変更する場合、*eeprom* の値でもシステムの EEPROM を変更できます。これにより、システムは新しいブートデバイスから自動的にブートできます。

注 - x86: preserve 値を指定する必要があります。

update JumpStart プログラムは、インストールされるシステムが指定のブートデバイスから自動的にブートするように、システムの EEPROM をそのブートデバイスに変更します。

preserve システムの EEPROM 中のブートデバイス値は変更されません。システムの EEPROM を変更しないで新しいブートデバイスを指定した場合は、システムが新しいブートデバイスから自動的にブートするように、システムの EEPROM を手作業で変更する必要があります。

例 8-8 boot_device プロファイルキーワード

```
boot_device c0t0d0s2 update
```

bootenv createbe プロファイルキーワード

```
bootenv createbe bename new_BE_name filesystem mountpoint:device:fs_options
[filesystem...]
```

bootenv createbe キーワードを使用すると、Solaris OS のインストール時に、空の非アクティブブート環境をすばやく作成できます。少なくとも、ルート (/) ファイルシステムを作成する必要があります。スライスは、指定のファイルシステム用に予約されていますが、ファイルシステムはコピーされません。このブート環境は、名前が付けられてはいますが、実際には、Solaris フラッシュアーカイブがインストールされる時にはじめて作成されることとなります。空のブート環境にアーカイブがインストールされると、ファイルシステムは予約されたスライスにインストールされます。*bename* および *filesystem* の値を次に示します。

bename new_BE_name

bename は、新しく作成するブート環境の名前を指定します。*new_BE_name* は、30 文字以内の英数字で指定してください。マルチバイト文字は使用できません。この名前は、システム上で一意となるように指定する必要があります。

filesystem mountpoint: device:fs_options

filesystem は、新しいブート環境に作成するファイルシステムの種類と数を決定します。少なくとも、ルート (/) ファイルシステムを置くスライスを指定する必要があります。複数のファイルシステムを同一のディスクに置くことも、複数のディスクに分散することもできます。

- **mountpoint** には、任意の有効なマウントポイント、またはスワップスライスを示す - (ハイフン) を指定できます。
- **device** には、インストール対象であるオペレーティングシステムが最初に起動したときに利用可能なデバイスを指定してください。このデバイスは、「free」などの JumpStart の特殊な記憶装置とは無関係です。デバイスとして Solaris ポリウムマネージャーのポリウムや Veritas Volume Manager のポリウムを指定することはできません。device はディスクデバイスの名前です。/dev/dsk/cwtxdysz の形式で表されます。
- **fs_options** には、次のいずれかを指定できます。
 - **ufs**: UFS ファイルシステムを示します。
 - **swap**: スワップファイルシステムを示します。スワップマウントポイントはハイフン (-) で表します。

プロファイルの例とこのキーワードの基本的な使用方法については、次の関連情報を参照してください。

プロファイルの例

例 3-11

非アクティブブート環境の作成、アップグレード、およびアクティブ化を行うための Solaris Live Upgrade の基本的な使用方法

『Solaris 10 5/08 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の第 2 章「Solaris Live Upgrade (概要)」

Solaris フラッシュアーカイブの基本的な使用方法

『Solaris 10 5/08 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』の第 1 章「Solaris フラッシュ (概要)」

client_arch プロファイルキーワード

`client_arch karch_value ...`

client_arch は、OS サーバーが、それ自体が使用するものとは異なるプラットフォームグループをサポートすることを定義します。プロファイルに **client_arch** を指定しない場合、OS サーバーを使用するどのディスクレスクライアントも、サーバーと同じプラットフォームグループでなくてはなりません。OS サーバーにサポートさせたいプラットフォームグループをすべて指定する必要があります。

karch_value の有効な値は、`sun4u` と `i86pc` です。プラットフォーム名とさまざまなシステムの詳細については、『Solaris Sun ハードウェアマニュアル』(<http://docs.sun.com>) を参照してください。

注 - `client_arch` は、`system_type` に `server` を指定したときだけ使用できます。

`client_root` プロファイルキーワード

`client_root` *root_size*

`client_root` は、各クライアントに割り当てるルート領域の大きさ (*root_size*) を M バイト単位で定義します。サーバーのプロファイルに `client_root` を指定しないと、インストールソフトウェアは1つのクライアント当たり 15M バイトのルート領域を割り当てます。このクライアント用のルート領域の大きさは、`num_clients` キーワードを組み合わせて、`/export/root` ファイルシステム用に確保する領域の大きさを決定するときに使用されます。

注 - `client_root` は、`system_type` に `server` を指定したときだけ使用できます。

`client_swap` プロファイルキーワード

`client_swap` *swap_size*

`client_swap` は、各ディスクレスクライアントに割り当てるスワップ空間の大きさ (*swap_size*) を M バイト単位で定義します。`client_swap` を指定しない場合、32M バイトのスワップ空間がデフォルトで割り当てられます。

注 - `client_swap` は、`system_type` に `server` を指定したときだけ使用できます。

例 8-9 `client_swap` プロファイルキーワード

次の例は、各ディスクレスクライアントが 64M バイトのスワップ空間を持つことを定義します。

```
client_swap 64
```

スワップサイズを決定する方法

プロファイルが `swap` のサイズを指定していない場合、JumpStart プログラムはシステムの物理メモリーに基づいてスワップ空間のサイズを決定します。表 8-5 に、カスタム JumpStart インストール中に `swap` のサイズがどのように決定されるかを示します。

表 8-5 swap のサイズの決定

物理メモリー(単位:Mバイト)	スワップ空間(単位:Mバイト)
16-64	32
64-128	64
128-512	128
512 を超える場合	256

JumpStart プログラムは swap のサイズが swap が作成されるディスクの 20% を超えないようにします。ただし、ほかのファイルシステムを配置したあとにディスクに空き領域が残っている場合を除きます。空き領域が残っている場合、JumpStart プログラムは空き領域を swap に割り当てて、可能であれば、表 8-5 に示されている量を割り当てます。

注-物理メモリーとスワップ空間の合計は、32Mバイト以上必要です。

cluster プロファイルキーワード(ソフトウェアグループの追加)

`cluster group_name`

`cluster` は、どのソフトウェアグループをシステムに追加するかを指定します。

注-ソフトウェアグループは、クラスタおよびパッケージの集まりを含むメタクラスタです。ソフトウェアグループは、`cluster` キーワードおよび `group_name` 変数を使ってインストールされます。この `cluster` キーワードは、初期インストールのときにのみインストールできます。この `cluster` キーワードは、`clustertoc(4)` ファイル内のメタクラスタを表します。

クラスタは `SUNWname` という名前のパッケージの集まりです。クラスタは、`cluster` キーワードおよび `cluster_name` 変数を使ってインストールされます。クラスタは初期インストールまたはアップグレードのときに、ソフトウェアグループ(メタクラスタ)から追加または削除できます。

各ソフトウェアグループの `group_name` 名は次のとおりです。

ソフトウェアグループ	<code>group_name</code>
限定ネットワークシステムサポート	<code>SUNWCrnet</code>
コアシステムサポート	<code>SUNWCreq</code>

ソフトウェアグループ	<i>group_name</i>
エンドユーザーシステムサポート	SUNWCuser
開発者システムサポート	SUNWCprog
全体ディストリビューション	SUNWCall
全体ディストリビューションと OEM サポート	SUNWCXall

次の制約があります。

- 1つのプロファイルに指定できるソフトウェアグループの数は1つだけです。
- ソフトウェアグループは、ほかの `cluster` エントリおよび `package` エントリよりも前に指定する必要があります。
- プロファイル内の `cluster` でソフトウェアグループを指定しない場合、デフォルトによりエンドユーザーシステムサポートソフトウェアグループ (SUNWCuser) がシステムにインストールされます。

ソフトウェアグループの詳細は、『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の「ソフトウェアグループごとの推奨ディスク容量」を参照してください。

cluster プロファイルキーワード (クラスタの追加または削除)

`cluster cluster_name add_delete_switch`

`cluster` は、システムにインストールされるソフトウェアグループにクラスタを追加するか、削除するかを指定します。

cluster_name クラスタ名は `SUNWCname` の形式で指定します。

add_delete_switch 指定のクラスタを追加するか削除するかを表すオプションのキーワードです。値 `add` または `delete` を使用します。値 `add` も `delete` も指定しなかった場合、デフォルトで `add` が適用されます。

アップグレード時に `cluster` を使用すると、次の条件が適用されます。

- すでにシステムにあるすべてのクラスタが自動的にアップグレードされます。
- `cluster_name add` を指定したが、`cluster_name` がシステムにインストールされていなかった場合、そのクラスタがインストールされます。
- `cluster_name delete` を指定したが、`cluster_name` がシステムにインストールされていた場合、アップグレードが開始される前にそのパッケージは削除されます。

注-ソフトウェアグループは、クラスタおよびパッケージの集まりを含むメタクラスタです。ソフトウェアグループは、`cluster` キーワードおよび `group_name` 変数を使ってインストールされます。この `cluster` キーワードは、初期インストールのときにのみインストールできます。この `cluster` キーワードは、`clustertoc(4)` ファイル内のメタクラスタを表します。

クラスタはパッケージの集まりです。クラスタはグループ化され、ソフトウェアグループ(メタクラスタ)を形成できます。クラスタ名は常に `SUNW<name>` の形式となります。クラスタは、`cluster` キーワードおよび `cluster_name` 変数を使ってインストールされます。クラスタは初期インストールまたはアップグレードのときに、ソフトウェアグループ(メタクラスタ)から追加または削除できます。

dontuse プロファイルキーワード

dontuse *disk_name* ...

`partitioning default` が指定されると、デフォルトではシステム上のすべての使用可能ディスクが使用されます。dontuse は、JumpStart プログラムに使用させないディスク(1つ以上)を指定するために使用します。*disk_name* は、`cxytzyd` または `cydz` の形式(たとえば、`c0t0d0`)で指定する必要があります。

注-1つのプロファイルで、dontuse キーワードと `usedisk` キーワードを同時に指定することはできません。

x86: fdisk プロファイルキーワード

fdisk *disk_name type size*

fdisk は、x86 システムで fdisk パーティションを設定する方法を定義します。fdisk は2回以上指定できます。次に、fdisk によって x86 システムのパーティション設定を行う際の動作を示します。

- fdisk キーワードを使って (*size* に `delete` か `0` を指定して) 削除しないかぎり、ディスク上のすべての fdisk パーティションが保存されます。また、*size* に `all` を指定した場合は、既存のすべての fdisk パーティションが削除されます。
- ルート (/) ファイルシステムを含む Solaris fdisk パーティションは、そのディスク上でアクティブパーティションとして常に指定されます。

注-システムは、デフォルトでアクティブパーティションから起動します。

- プロファイルで `fdisk` キーワードを指定しないと、インストール時にデフォルトで次の `fdisk` キーワードが指定されます。

```
fdisk all solaris maxfree
```

- `fdisk` エントリは、プロファイルに指定されている順序で処理されます。

disk_name 次の値を使用して、`fdisk` パーティションを作成または削除する場所を指定します。

- `cxydz` または `cydz` - 特定のディスク。たとえば、`c0t3d0`。
- `rootdisk` - システムのルートディスク (インストールが行われる場所) の値を含む変数。ルートディスクは、162 ページの「システムのルートディスクを決定する方法」に説明されているように、`JumpStart` プログラムによって決定されます。
- `all` - 選択されたすべてのディスク。

type 次の値を使用し、指定したディスク上で作成または削除する `fdisk` パーティションのタイプを指定します。

- `solaris` - Solaris `fdisk` パーティション (SUNIXOS `fdisk` タイプ)。
- `dosprimary` - 一次 DOS の `fdisk` パーティション (データ DOS 用に拡張または予約された `fdisk` パーティションではない) の別名。 `size` に値 `delete` を指定して `fdisk` パーティションを削除する場合、`dosprimary` は `DOSHUGE`、`DOSOS12`、および `DOSOS16` `fdisk` タイプの別名になります。 `fdisk` パーティションを作成する場合、`dosprimary` は `DOSHUGE` `fdisk` パーティションの別名になります。
- `DDD` - 整数で表す `fdisk` パーティション。 `DDD` は 1 から 255 までの整数です。

注 - この値は `size` に `delete` を指定した場合のみ指定できます。

- `0xHH` - 16 進数で表す `fdisk` パーティション。 `HH` は 01 から FF までの 16 進数です。

注 - この値は `size` に `delete` を指定した場合のみ指定できます。

いくつかの `fdisk` タイプの整数と 16 進数での表し方を次の表に示します。

fdiskタイプ	DDD	HH
DOSOS12	1	01
PCIXOS	2	02
DOSOS16	4	04
EXTDOS	5	05
DOSHUGE	6	06
DOSDATA	86	56
OTHEROS	98	62
UNIXOS	99	63

size

次のいずれかの値を使用します。

- *DDD* - サイズが *DDD* (Mバイト単位) の *fdisk* パーティションを、指定したディスク上に作成します。*DDD* は整数で指定する必要があります。JumpStart プログラムは、この数値を一番近いシリンダの境界に自動的に繰り上げます。値 0 を指定すると、*delete* を指定するのと同じになります。
- *all* - *fdisk* ディスク全体に1つのパーティションを作成します(既存の *fdisk* パーティションはすべて削除される)。

x86のみ - 値 *all* を指定できるのは、*type* が *solaris* の場合だけです。

- *maxfree* - 指定したディスク上の最も大きい連続する空き領域に *fdisk* パーティションを作成します。指定した *type* の *fdisk* パーティションがディスク上にすでに存在する場合、その既存の *fdisk* パーティションが使用されます。新しい *fdisk* パーティションはディスク上に作成されません。

x86のみ - ディスクには、空き領域と未使用の *fdisk* パーティションが1つ以上存在しなければなりません。領域が確保されないとインストールは失敗します。値 *maxfree* を指定できるのは、*type* が *solaris* または *dosprimary* の場合だけです。

- *delete* - 指定した *type* のすべての *fdisk* パーティションを指定したディスク上で削除します。

filesystem プロファイルキーワード (リモートファイルシステムのマウント)

```
filesystem server:path server_address mount_pt_name mount_options
```

これらの値を指定して `filesystem` を使用すると、JumpStart プログラムはインストールされているシステムがそのブート時に自動的にリモートファイルシステムをマウントするように設定します。filesystem は 2 回以上指定できます。

<i>server</i>	リモートファイルシステムが存在するサーバー名 (後ろにコロンを付ける)。
<i>path</i>	リモートファイルシステムのマウントポイント名。(例: <code>/usr</code> 、 <code>/export/home</code> など)
<i>server_address</i>	サーバーの IP アドレス。 <code>server:path</code> で指定します。ネームサービスがネットワーク上で実行されていない場合、この値 <code>server_address</code> を使用して、サーバーのホスト名と IP アドレスを <code>/etc/hosts</code> ファイルに反映できます。サーバーの IP アドレスを指定しない場合は、マイナス記号 (-) を指定する必要があります。たとえば、ネットワーク上で実行中のネームサービスがある場合、サーバーの IP アドレスを指定する必要はありません。
<i>mount_pt_name</i>	リモートファイルシステムをマウントするマウントポイント名。
<i>mount_options</i>	1 つ以上のマウントオプション。 <code>mount(1M)</code> コマンドの <code>-o</code> オプションと同じです。これらのマウントオプションは、指定された <code>mount_pt_name</code> の <code>/etc/vfstab</code> エントリに追加されます。

注 - 複数のマウントオプションを指定する場合は、マウントオプションはスペースではなくコンマで区切ってください。例:
`ro,quota`

例 8-10 filesystem プロファイルキーワード

```
filesystem sherlock:/export/home/user2 - /home
```

filesystem プロファイルキーワード (ローカルファイルシステムの作成)

```
filesystem slice size file_system optional_parameters
```

これらの値を指定して `filesystem` を使用すると、JumpStart プログラムによりインストール時にローカルファイルシステムが作成されます。filesystem は 2 回以上指定できます。

<i>slice</i>	次のいずれかの値を使用します。
<i>any</i>	JumpStart プロファイルは、ファイルシステムを任意のディスクに配置します。 <hr/> 注 - <i>size</i> が <i>existing</i> 、 <i>all</i> 、 <i>free</i> 、 <i>start:size</i> 、または <i>ignore</i> の場合は、 <i>any</i> は指定できません。 <hr/>
<i>cwtxdysz</i> または <i>cxdysz</i>	JumpStart プログラムがファイルシステムを配置するディスクスライス。たとえば、 <i>c0t0d0s0</i> または <i>c0d0s0</i> 。
<i>rootdisk.sn</i>	システムのルートディスクの値を含む変数。ルートディスクは、 162 ページの「システムのルートディスクを決定する方法」 に説明されているように、JumpStart プログラムによって決定されます。拡張子 <i>sn</i> は、ディスク上の特定のスライスを示します。 <hr/> 注 - ルートディスクは、JumpStart プログラムによって決定され、OS がインストールされる場所を指定します。rules ファイルでは、プロンプキーワード「 <i>rootdisk</i> 」を使用しますが、このキーワードは JumpStart プロファイルで使用される「 <i>rootdisk</i> 」キーワードとは使い方が異なります。rules ファイルのプロンプキーワード「 <i>rootdisk</i> 」を使って、インストールの場所を設定することはできません。プロンプキーワード <i>rootdisk</i> は、インストール時のブート元を決定します。表 8-10 を参照してください。 <hr/>
<i>size</i>	次のいずれかの値を使用します。

num ファイルシステムのサイズを *num* (Mバイト単位) で設定します。

existing 既存のファイルシステムの現在のサイズを使用します。

注 - *existing* の値を使用すると、別の *mount_pt_name* として *file_system* を指定することによって、既存のスライス名を変更できます。

auto 選択したソフトウェアに応じて、ファイルシステムのサイズを自動的に決定します。

all 指定した *slice* は、そのファイルシステム用にディスク全体を使用します。この値を指定すると、指定したディスク上にほかのファイルシステムは存在できません。

free ディスク上の残りの未使用領域をファイルシステム用に使用します。

注 - *fileSYS* の値として *free* を使用する場合は、*fileSYS* はプロファイルの最後のエントリにする必要があります。

start:size ファイルシステムを明示的にパーティションに分割します。 *start* はスライスが始まるシリンダであり、 *size* はそのスライスのシリンダ数です。

file_system

file_system 値はオプションで、 *slice* に *any* または *cwtxdysz* を指定しているときに使用できます。この値を指定しないと *unnamed* がデフォルトで設定されますが、この場合 *optional_parameters* 値を指定できません。次のいずれかの値を使用します。

mount_pt_name ファイルシステムのマウントポイント名です (たとえば、 */var*)。

swap 指定した *slice* が *swap* として使用されます。

overlap 指定した *slice* がディスク領域を表すものとして定義されます。VTOC 値は

V_BACKUP です。デフォルトでは、スライス 2 はディスク全体を表すオーバーラップスライスです。

注 - *size* に *existing*、*all*、または *start:size* を指定した場合だけ *overlap* を指定できません。

- `unnamed` 指定した *slice* が raw スライスとして定義されるので、*slice* にはマウントポイント名がありません。 *file_system* を指定しないと、デフォルトで `unnamed` が設定されます。
- `ignore` 指定した *slice* を使用しないか、JumpStart プログラムで認識しません。このオプションは、インストール時にディスク上の特定のファイルシステムを無視させるために使用できます。JumpStart プログラムは、同じディスク上に同じ名前での新しいファイルシステムを作成します。 `ignore` は、`partitioning existing` を指定したときだけ使用できます。

`optional_parameters` 次のいずれかの値を使用します。

- `preserve` 指定した *slice* 上のファイルシステムを保存します。

注 - *size* に *existing*、*slice* に `cwtxdysz` を指定した場合だけ `preserve` を指定できます。

- `mount_options` 1つ以上のマウントオプション。 `mount(1M)` コマンドの `-o` オプションと同じです。これらのマウントオプションは、指定された *mount_pt_name* の `/etc/vfstab` エントリに追加されます。

注 - 複数のマウントオプションを指定する場合は、マウントオプションはスペースではなくコンマで区切ってください。例:
`ro,quota`

filesys プロファイルキーワード (RAID-1 ボリュームの作成)

`filesys mirror[:name]slice [slice] size file_system optional_parameters`

JumpStart プログラムでは、`filesys mirror` キーワードと一覧にある値を使用して、ミラー化されたファイルシステムの作成に必要な RAID-1 および RAID-0 ボリュームを作成できます。複数のファイルシステムの RAID-1 ボリューム (ミラー) を作成する場合は、`filesys mirror` を繰り返し指定できます。

注 - `filesys mirror` キーワードは、初期インストールでのみサポートされます。

<i>name</i>	任意のキーワードです。RAID-1 ボリューム (ミラー) に名前を付けることができます。ミラー名の先頭の文字は必ず「d」で、その後ろに 0-127 の数字が続きます (例: <code>d100</code>)。ミラー名を指定しない場合は、カスタム JumpStart プログラムによって名前が割り当てられます。ミラー名の指定方法については、『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の「カスタム JumpStart と Solaris Live Upgrade を行うときの RAID ボリューム名の要件とガイドライン」を参照してください。
<i>slice</i>	ディスクスライスを指定します。カスタム JumpStart プログラムは、このディスクスライスに、複製するファイルシステムを配置します。スライスの値は、 <code>cwtxdysz</code> の形式になります (例: <code>c0t0d0s0</code> 、 <code>c0t0d0s5</code> など)。カスタム JumpStart プログラムは、スライス上に RAID-0 ボリューム (単一スライス連結) を作成し、この連結をミラー化するための RAID-1 ボリュームを作成します。2つの RAID-0 ボリュームに、最大2つのスライスを指定できます。
<i>size</i>	ファイルシステムのサイズを M バイト単位で指定します。
<i>file_system</i>	複製するファイルシステムを指定します。カスタム JumpStart プログラムは、指定されたスライスから RAID-1 ボリュームを作成し、この RAID-1 ボリュームを指定されたファイルシステムにマウントします。ルート (<code>/</code>)、 <code>/usr</code> 、 <code>/var</code> などのクリティカルファイルシステムに加えて、 <code>swap</code> もファイルシステムとして指定できます。
<i>optional_parameters</i>	1つ以上のマウントオプション。mount (1M) コマンドの <code>-o</code> オプションと同じです。これらのマウントオプションは、指定された <i>file_system</i> の <code>/etc/vfstab</code> エントリに追加されます。複数のマウントオプションを指定する場合は、マウントオプションはスペースを入れずにコマンドで区切ってください (例: <code>ro,quota</code>)。

ミラー化されたファイルシステムをインストール時に作成する方法の詳細は、『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の第8章「インストール時の RAID-1 ボリューム (ミラー) の作成 (概要)」を参照してください。

forced_deployment プロファイルキーワード (Solaris フラッシュ差分アーカイブのインストール)

forced_deployment

forced_deployment は、ソフトウェアで想定されているものとは異なるクローンシステムに、Solaris フラッシュ差分アーカイブを強制的にインストールします。



注意 - forced_deployment を使用すると、クローンシステムを期待される状態にするために、新規ファイルがすべて削除されます。ファイルを削除して良いかどうか判断できない場合には、デフォルトを使用してください。デフォルトでは、新規ファイルが削除されそうになると、インストールが停止します。

geo プロファイルキーワード

geo region

geo は、システムにインストールする地域ロケールか、あるいはシステムのアップグレード時に追加する地域ロケールを指定します。region は、インストールするロケールを含んだ地理的地域を指定します。region に指定できる値を、次の表に一覧表示します。

値	説明
N_Africa	北アフリカ。エジプトを含みます
C_America	中央アメリカ。コスタリカ、エルサルバドル、グアテマラ、メキシコ、ニカラグア、パナマを含みます
N_America	北アメリカ。カナダ、アメリカ合衆国を含みます
S_America	南アメリカ。アルゼンチン、ボリビア、ブラジル、チリ、コロンビア、エクアドル、パラグアイ、ペルー、ウルグアイ、ベネズエラを含みます
Asia	アジア。日本、韓国、中華人民共和国、台湾、タイを含みます
Ausi	オーストラリア。オーストラリア、ニュージーランドを含みます
C_Europe	中央ヨーロッパ。オーストリア、チェコ、ドイツ、ハンガリー、ポーランド、スロヴァキア、スイスを含みます

値	説明
E_Europe	東ヨーロッパ。アルバニア、ボスニア、ブルガリア、クロアチア、エストニア、ラトビア、リトアニア、マケドニア、ルーマニア、ロシア、セルビア、スロヴェニア、トルコを含みます
N_Europe	北ヨーロッパ。デンマーク、フィンランド、アイスランド、ノルウェー、スウェーデンを含みます
S_Europe	南ヨーロッパ。ギリシャ、イタリア、ポルトガル、スペインを含みます
W_Europe	西ヨーロッパ。ベルギー、フランス、イギリス、アイルランド、オランダを含みます
M_East	中近東。イスラエルを含みます

前記の各地域ロケールを構成するコンポーネントロケール値の完全なリストは、『国際化対応言語環境の利用ガイド』に記載されています。

注- システムに追加する必要がある各ロケールごとに、`geo` キーワードを指定します。

install_type プロファイルキーワード

`install_type initial_upgrade_flash_switch`

`install_type` は、システムに対し、新しい Solaris OS を上書きインストールするか、既存の Solaris OS をアップグレードするか、あるいは、Solaris フラッシュアーカイブをインストールするかを定義します。

注- `install_type` は必須であり、各プロファイル内で最初のプロファイルキーワードとして指定する必要があります。

`initial_upgrade_flash_switch` には、次のオプションのうちの1つを使用する必要があります。

<code>initial_install</code>	Solaris OS の初期インストールの実行を指定します。
<code>upgrade</code>	Solaris OS のアップグレードの実行を指定します。
<code>flash_install</code>	すべてのファイルを上書きする Solaris フラッシュアーカイブをインストールすることを指定します。
<code>flash_update</code>	指定したファイルだけを上書きする Solaris フラッシュ差分アーカイブをインストールすることを指定します。

注- プロファイルキーワードの中には、`initial_install` オプションでしか使用できないものがあります。`upgrade` オプションでしか使用できないものもあります。また、`flash_install` オプションでしか使用できないものがあります。

layout_constraint プロファイルキーワード

`layout_constraint slice constraint minimum_size`

`layout_constraint` は、ファイルシステムがディスク容量不足のためにアップグレード中にディスク容量を再配置する必要がある場合に、制約付き自動配置がファイルシステムで行われることを示します。

制約	説明
このキーワードはアップグレードオプションとの併用でのみ使用されます。	<code>layout_constraint</code> は、ディスク容量の再配置が必要なアップグレードオプションだけで使用できます。
<code>layout_constraint</code> キーワードを指定しない場合	JumpStart プログラムはディスクを次のように配置します。 <ul style="list-style-type: none"> ■ アップグレード用により多くの容量を必要とするファイルシステムは、<code>changeable</code> とマークされます。 ■ より多くの容量を必要とするファイルシステムと同じディスク上にあるファイルシステム (<code>/etc/vfstab</code> ファイルでマウントされる) も <code>changeable</code> とマークされます。 ■ 残りのファイルシステムは <code>fixed</code> とマークされます。これは、自動配置がこれらのファイルシステムを変更できないためです。
1つ以上の <code>layout_constraint</code> キーワードを指定する場合	JumpStart プログラムはディスクを次のように配置します。 <ul style="list-style-type: none"> ■ アップグレード用により多くの容量を必要とするファイルシステムは、<code>changeable</code> とマークされます。 ■ <code>layout_constraint</code> キーワードを指定したファイルシステムは、指定した制約がマークされます。 ■ 残りのファイルシステムは、<code>fixed</code> とマークされます。
ファイルシステムが <code>changeable</code> とマークされていない場合	アップグレードにより多くの容量を必要とするファイルシステムの制約は変更できませんが (<code>changeable</code> とマークされなければならない)、このようなファイルシステムに <code>layout_constraint</code> キーワードを使用すれば、その <code>minimum_size</code> 値を変更できます。

制約	説明
ファイルシステムがアップグレード用により多くの容量を必要とする場合	自動配置機能がディスク容量の再配置を行う際には、より多くのファイルシステム、特にアップグレード用により多くの容量を必要とするファイルシステムと同じディスク上にあるファイルシステムを、 <code>changeable</code> または <code>movable</code> であると選択します。
<i>slice</i>	制約を指定するファイルシステムのディスクスライスを指定します。システムのディスクスライスは、 <code>cwtxdysz</code> または <code>cxdsz</code> の形式で指定する必要があります。
<i>constraint</i>	指定したファイルシステムに対して、次のいずれか 1 つの制約を選択します。
<code>changeable</code>	<p>自動配置機能はファイルシステムをほかの場所に移動して、そのサイズを変更できます。<code>changeable</code> 制約は、<code>/etc/vfstab</code> ファイルによってマウントされたファイルシステムにしか指定できません。<code>minimum_size</code> 値を指定すれば、ファイルシステムのサイズを変更できます。</p> <p>ファイルシステムを <code>changeable</code> とマークして、<code>minimum_size</code> 値を指定しないと、そのファイルシステムの最小サイズは、必要な最小サイズより 10% 大きな値に設定されます。たとえば、ファイルシステムの最小サイズが 100M バイトの場合、変更されるサイズは 110M バイトになります。<code>minimum_size</code> を指定した場合、残りの空き領域(元のサイズから最小サイズを引いたもの)はほかのファイルシステム用に使用されます。</p>
<code>movable</code>	自動配置機能はファイルシステムを(同じディスクまたは異なるディスク上の)ほかのスライスに移動できますが、サイズは変更しません。
<code>available</code>	自動配置機能は、ファイルシステムのすべての領域を使用して領域を割り当て直します。ファイルシステムのすべてのデータは失われます。 <code>available</code> 制約は、 <code>/etc/vfstab</code> ファイルでマウントされないファイルシステムだけに指定できます。
<code>collapse</code>	自動配置機能は、指定したファイルシステムをその親ファイルシステムに移動して閉じこめます。このオプションは、アップグレードの一部としてシステム上のファイルシステム数を減らすために使用できます。たとえば、システムにファイルシステム <code>/usr</code>

と `/usr/share` が存在する場合、`/usr/share` ファイルシステムを閉じ込めると、このファイルシステムは `/usr` (その親) に移動します。collapse 制約は、`/etc/vfstab` ファイルでマウントされるファイルシステムにしか指定できません。

minimum_size この値は、自動配置機能がディスク容量を再配置するときに、ファイルシステムに割り当てる最小サイズを指定します(基本的にファイルシステムのサイズを変更する)。まだ割り当てられていない領域が追加される場合、ファイルシステムのサイズは最終的にこの指定した値より大きくなる可能性があります。しかし、指定される値よりサイズが小さくなることはありません。*minimum_size* 値は省略可能です。このオプション値を使用できるのは、ファイルシステムを `changeable` とマークした場合だけです。最小サイズは、ファイルシステムの既存の内容に必要なサイズより小さい値には設定できません。

例 8-11 `layout_constraint` プロファイルキーワード

```
layout_constraint c0t3d0s1 changeable 200
```

```
layout_constraint c0t3d0s4 movable
```

```
layout_constraint c0t3d1s3 available
```

```
layout_constraint c0t2d0s1 collapse
```

local_customization プロファイルキーワード (Solaris フラッシュアーカイブのインストール)

```
local_customization local_directory
```

クローンシステムに Solaris フラッシュアーカイブをインストールする前に、カスタムスクリプトを作成して、クローンシステム上のローカル構成を保存できます。`local_customization` キーワードは、これらのスクリプトの格納先ディレクトリを示します。`local_directory` は、クローンシステム上のスクリプトのパスです。

配置前および配置後スクリプトについては、『Solaris 10 5/08 インストールガイド (Solaris フラッシュアーカイブの作成とインストール)』の「カスタムスクリプトの作成」を参照してください。

locale プロファイルキーワード

```
locale locale_name
```


no_content_check プロファイルキーワード (Solaris フラッシュアーカイブのインストール)

no_content_check

Solaris フラッシュ差分アーカイブを使用してクローンシステムをインストールする場合、no_content_check キーワードを使用してファイルごとの検証を省略できます。ファイルごとの検証により、クローンシステムがマスターシステムの複製であることが保証されます。クローンシステムが元のマスターシステムの複製であることが確実である場合を除き、このキーワードの使用は避けてください。



注意-no_content_checkを使用すると、クローンシステムを期待される状態にするために、新規ファイルがすべて削除されます。ファイルを削除して良いかどうか判断できない場合には、デフォルトを使用してください。デフォルトでは、新規ファイルが削除されそうになると、インストールが停止します。

Solaris フラッシュ差分アーカイブのインストール方法については、90 ページの「カスタム JumpStart インストールを使用して Solaris フラッシュアーカイブをインストールする方法」を参照してください。

no_master_check プロファイルキーワード (Solaris フラッシュアーカイブのインストール)

no_master_check

Solaris フラッシュ差分アーカイブを使用してクローンシステムをインストールする場合、no_master_check キーワードを使用して、クローンシステムが元のマスターシステムから構築されたものかどうかのチェックを省略できます。クローンシステムが元のマスターシステムの複製であることが確実である場合を除き、このキーワードの使用は避けてください。

Solaris フラッシュ差分アーカイブのインストール方法については、90 ページの「カスタム JumpStart インストールを使用して Solaris フラッシュアーカイブをインストールする方法」を参照してください。

num_clients プロファイルキーワード

num_clients *client_num*

サーバーがインストールされているときには、各ディスクレスクライアントのルート (/) とスワップファイルシステムに領域が割り当てられます。num_clients は、サーバーがサポートするディスクレスクライアント数 (*client_num*) を定義します。プロファイルで num_clients を指定しないと、デフォルトで5つのディスクレスクライアントが割り当てられます。

注 - num_clients は、system_type が server として指定されているときだけ使用できません。

package プロファイルキーワード

package *package_name* [add [*retrieval_type* *location*]| delete]

package は、初期インストールとアップグレードオプションの両方で使用できます。package キーワードを指定すると、次の処理が可能になります。

- インストールする Solaris ディストリビューションからソフトウェアグループにパッケージを追加します。
- インストールするディストリビューション以外からソフトウェアグループにパッケージを追加します。
- インストールまたはアップグレードするソフトウェアグループからパッケージを除外(削除)します。
- Solaris フラッシュアーカイブのインストール時に、インストールするディストリビューション以外からパッケージを追加します。

package_name

パッケージ名は SUNWname の形式で指定します。パッケージの詳細とその名前を表示するには、インストール済みシステムで pkginfo -l コマンドを使用します。

add | delete

指定されたパッケージを追加または削除します。値 add も delete も指定しなかった場合、デフォルトで add が適用されます。

注 - プロファイルに別のパッケージエントリを追加し、場所を省略することで、複数のパッケージを追加できます。場所を指定しない場合、直前のパッケージの場所が、後続のすべてのパッケージに適用されます。

[*retrieval_type* *location*]

インストールする Solaris ディストリビューションに含まれないパッケージ(複数可)を追加します。retrieval_type と location の値は、パッケージの格納場所によって異なります。以降の節では、retrieval_type と location に指定可能な値と、package_name キーワードの使用例を示します。

NFS サーバーに格納されたパッケージ

パッケージが NFS サーバーに格納されている場合、次のいずれかの構文で package キーワードを使用します。

```
package package_name add nfs server_name:/path [retry n]
package package_name add nfs://server_name:/path [retry n]
```

<i>package_name</i>	パッケージ名は SUNWname の形式で指定します。パッケージの詳細とその名前を表示するには、インストール済みシステムで <code>pkginfo -l</code> コマンドを使用します。
<i>server_name</i>	パッケージを格納したサーバーの名前を指定します。
<i>path</i>	指定されたサーバー上のパッケージディレクトリの場所を指定します。パスに <code>\$HOST</code> が含まれる場合、 <code>\$HOST</code> はインストールするホストシステムの名前で置き換えられます。
<i>retry n</i>	オプションのキーワードです。 <i>n</i> は、インストールプロセスがディレクトリのマウントを試みる回数の最大値です。

例 8-12 NFS によるパッケージの追加

この例の `package` プロファイルキーワードは、NFS の場所 `nfs://golden/packages/Solaris_10/` から `SUNWnew` パッケージを追加します。マウントに失敗した場合、NFS マウントが 5 回試行されます。

```
package SUNWnew add nfs golden:/packages/Solaris_10 retry 5
```

HTTP サーバーに格納されたパッケージ

パッケージが HTTP サーバーに格納されている場合、次のいずれかの構文で `package` キーワードを使用します。

```
package package_name add http://server_name[:port] path optional_keywords
package package_name add http server_name[:port] path optional_keywords
```

<i>package_name</i>	パッケージ名は SUNWname の形式で指定します。パッケージの詳細とその名前を表示するには、インストール済みシステムで <code>pkginfo -l</code> コマンドを使用します。
<i>server_name</i>	パッケージを格納したサーバーの名前を指定します。
<i>port</i>	オプションポートを指定します。 <i>port</i> は、ポート番号でも、実行時に決定されるポート番号を持つ TCP サービスの名前でもかまいません。 ポートを指定しなかった場合、デフォルトの HTTP ポート番号 80 が使用されます。
<i>path</i>	指定されたサーバーから取得するパッケージの場所を指定します。HTTP サーバーを使用する場合、データストリーム形式のパッケージを使用する必要があります。

optional_keywords HTTP サーバーからパッケージを取得するとき使用するオプションのキーワードを指定します。

表 8-6 HTTP で使用するオプションの package キーワード

キーワード	値の定義
<i>timeout min</i>	<p><i>timeout</i> キーワードには、HTTP サーバーからのデータ受信を待機する最長の時間を分単位で指定できます。この時間に達すると、接続が切断され、再接続のあと、再開されます。<i>timeout</i> 値として0(ゼロ)を指定すると、再接続は行われません。</p> <p>タイムアウトによる再接続が発生すると、パッケージの先頭からインストールし直されます。タイムアウト以前に取得されたデータは破棄されます。</p>
<i>proxy host:port</i>	<p><i>proxy</i> キーワードを使用して、プロキシホストとプロキシポートを指定できます。プロキシホストを使用すると、ファイアウォール越しに Solaris パッケージを取得できます。<i>proxy</i> キーワードを指定する場合は、プロキシポートを指定する必要があります。</p>

例 8-13 HTTP によるパッケージの追加

この例の package プロファイルキーワードは、HTTP の場所 `http://package.central/10` から、Solaris 10 ディレクトリ内のすべてのパッケージを追加します。データが取得されないまま5分以上経過すると、パッケージデータが再度取得されます。以前のパッケージデータは破棄されます。次のいずれかの形式を使用できます。

```
package SUNWnew add http package.central/Solaris_10 timeout 5
```

```
package SUNWnew add http://package.central/Solaris_10 timeout 5
```

例 8-14 プロキシポートを使った HTTP によるパッケージの追加

この例の package プロファイルキーワードは、HTTP の場所 `http://package.central/10` から、Solaris_10 ディレクトリ内のすべてのパッケージを追加します。*proxy* キーワードを使用すると、ファイアウォール越しにパッケージを取得できます。

```
package SUNWnew add http://package.central/Solaris_10 proxy webcache.east:8080
```

ローカルデバイスに格納されたパッケージ

パッケージをファイルシステム指向のランダムアクセスデバイス(フロッピーディスク、DVD-ROM など)に格納している場合は、ローカルデバイスから Solaris パッケージを取得できます。package キーワードでは次の構文を使用します。

```
package package_name add local_device device path file_system_type
```

<i>package_name</i>	パッケージ名は <code>SUNWname</code> の形式で指定します。パッケージの詳細とその名前を表示するには、インストール済みシステムで <code>pkginfo -l</code> コマンドを使用します。
<i>device</i>	Solaris パッケージが格納されているドライブの名前を指定します。デバイス名が正規のパスである場合は、デバイスは直接マウントされます。正規のパスでないデバイス名を指定すると、インストールユーティリティーはパスに <code>/dev/dsk/</code> を加えます。
<i>path</i>	Solaris パッケージのパスを指定します。指定したデバイス上のルート (<code>/</code>) ファイルシステムからの相対パスで指定する必要があります。
<i>file_system_type</i>	デバイス上のファイルシステムのタイプを指定します。ファイルシステムのタイプを指定しない場合、インストールユーティリティーは、UFS ファイルシステムのマウントを試みます。UFS のマウントに失敗すると、インストールユーティリティーは HSFS ファイルシステムのマウントを試みます。

例 8-15 UFS ファイルシステムのローカルデバイスによるパッケージの追加

この例の `package` プロファイルキーワードは、ローカルデバイス `c0t6d0s0` の `/Solaris_10/Product` ディレクトリから `SUNWnew` パッケージを追加します。これは UFS ファイルシステムです。

```
package SUNWnew add local_device c0t6d0s0 /Solaris_10/Product ufs
```

例 8-16 HSFS ファイルシステムのローカルデバイスによるパッケージの追加

この例の `package` プロファイルキーワードは、ローカルデバイス `c0t6d0s0` の `/Solaris_10/Product` ディレクトリから `SUNWnew` パッケージを追加します。これは HSFS ファイルシステムです。

```
package SUNWnew add local_device c0t6d0s0 /Solaris_10/Product hsfs
```

ローカルファイルに格納されたパッケージ

システムをブートしたミニルートからパッケージをインストールできます。カスタム `JumpStart` インストールを実施する時に、DVD、CD または NFS ベースのミニルートからシステムをブートします。このミニルートからインストールソフトウェアがロードされ、実行されます。したがって、DVD、CD または NFS ベースのミニルートに格納したパッケージは、ローカルファイルとしてアクセスできます。`package` キーワードでは次の構文を使用します。

```
package package_name add local_file path
```

package_name パッケージ名は *SUNWname* の形式で指定します。パッケージの詳細とその名前を表示するには、インストール済みシステムで `pkginfo -l` コマンドを使用します。

path パッケージの位置を指定します。このパスは、システムを Solaris SOFTWARE - 1 CD または Solaris Operating System DVD からブートしている間、システムにローカルファイルとしてアクセスできるものでなければなりません。Solaris SOFTWARE - 1 CD または Solaris Operating System DVD からブートしている間は、システムは `/net` にアクセスできません。

例 8-17 ローカルファイルによるパッケージの追加

この例の `package` プロファイルキーワードは、`/Solaris_10/Product` ディレクトリから `SUNWnew` パッケージを追加します。

```
package SUNWnew add local_file /Solaris_10/Product
```

package キーワードを使用する際の制約

`package` キーワードを使用する際には、次の制約があります。

- いくつかのパッケージは必須であり、削除できないものもあります。
- 各国語対応パッケージを個々に追加または削除するとき、`package` プロファイルキーワードは使用できません。各国語対応パッケージを追加するには、`locale` プロファイルキーワードを使用します。
- FTP サーバーやローカルバックアップ(テープなど)からパッケージを取得することはできません。
- インストールする Solaris ディストリビューション内のパッケージを、別の場所から追加することはできません。Solaris ディストリビューション内のパッケージを指定した場合、これに続けて別の場所を指定することはできません。これは、インストール後のシステムとの整合性を確保するためです。
- ユーザーの介入なしでインストールするには、`pkgadd` コマンドを使ってインストール可能なパッケージでなければなりません。ソフトウェアグループのパッケージと、別の場所にあるパッケージとは、同じ `admin` ファイルを使ってインストールする必要があります。
 - `retrieval_type` が HTTP の場合、パッケージはストリーム形式でなければなりません。
 - `retrieval_type` が NFS サーバー、ローカルデバイス、またはローカルファイルの場合、パッケージは標準パッケージフォーマットに従い、インストールするパッケージと同じ名前のディレクトリ名を指定する必要があります。

- 別の場所からパッケージを追加する場合で、パッケージが現在インストールされていない別のパッケージに依存している場合、このパッケージはインストールされません。インストールログファイルまたはアップグレードログファイルにエラーメッセージが記録されます。
- Solaris フラッシュアーカイブを使ってパッケージをインストールする場合は、次のガイドラインに従います。
 - インストールされたすべてのパッケージとアーカイブに互換性がなければなりません。
 - パッケージがアーカイブ内にある場合、JumpStart は既存のパッケージを上書きします。

package キーワードを使用したアップグレード処理

アップグレードに `package` を使用すると、JumpStart プログラムが次の処理を行います。

- すでにシステム上にあるすべてのパッケージが自動的にアップグレードされます。
- `package_name add` を指定したが、`package_name` がシステムにインストールされていない場合は、そのパッケージがインストールされます。
- `package_name delete` を指定したが、`package_name` がシステムにインストールされていた場合、アップグレードが開始される前にそのパッケージは削除されます。
- `package_name delete` を指定したが、`package_name` がシステムにインストールされていない場合、インストールするように指定したクラスタの一部にそのパッケージが含まれていても、パッケージはインストールされません。

partitioning プロファイルキーワード

`partitioning type`

`partitioning` は、インストール時にファイルシステム用にディスクをスライスに分割する方法を定義します。

プロファイルで `partitioning` を指定しないと、デフォルトで `default` タイプのパーティションが使用されます。

`type` 次のいずれかの値を使用します。

<code>default</code>	JumpStart プログラムはディスクを選択して、指定したソフトウェアをインストールするファイルシステムを作成します。ただし、 <code>filesys</code> キーワードで指定したファイルシステムを除きます。 <code>rootdisk</code> が最初に選択され、指定したソフトウェアが <code>rootdisk</code> に収まらない場合は、さらに別のディスクが使用されます。
----------------------	---

existing JumpStart プログラムは、システムのディスク上にある既存のファイルシステムを使用します。/, /usr、/usr/openwin、/opt、/var を除く、すべてのファイルシステムが保存されます。JumpStart プログラムは、ファイルシステムのスーパーブロックにある最後のマウントポイントフィールドを使用して、スライスがどのファイルシステムのマウントポイントを表しているかを判断します。

注 - filesys プロファイルキーワードと partitioning existing を組み合わせる場合、size は existing である必要があります。

explicit JumpStart プログラムは、ディスクを使用し、fileysys キーワードで指定されるファイルシステムを作成します。fileysys キーワードでルート (/) ファイルシステムだけを指定した場合、すべての Solaris ソフトウェアがルートファイルシステムにインストールされます。

注 - explicit プロファイル値を使用するときには、fileysys プロファイルキーワードを使用して、使用するディスクと作成するファイルシステムを指定してください。

patch プロファイルキーワード

`patch patch_id_list | patch_file patch_location optional_keywords]`

patch_id_list インストールするパッチ ID 番号を指定します。このリストには、Solaris パッチ ID をコンマで区切って指定する必要があります。パッチは、リスト内に指定された順にインストールされません。コンマのあとに空白文字を入力しないでください。たとえば、112467-01,112765-02 のように指定します。

patch_file `patch_location` 内のパッチのリストが格納されたファイルです。パッチは、ファイル内に指定された順にインストールされません。

patch_location パッチのある場所を指定します。次の場所を選択できます。

- ネットワークファイルシステム (NFS) サーバー
- HTTP サーバー
- ローカルデバイス
- ローカルファイル

optional_keywords オプションのキーワードです。パッチが格納されている場所によって異なります。次の節では、指定可能な場所とオプション

のキーワードについて説明します。

NFS サーバーに格納されたパッチ

パッチが NFS サーバーに格納されている場合、次のいずれかの構文で `patch` キーワードを使用します。

```
patch patch_id_list | patch_file nfs server_name:/patch_directory [retry n]
patch patch_id_list | patch_file nfs://server_name/patch_director [retry n]
```

<i>patch_id_list</i>	インストールするパッチ ID 番号を指定します。このリストには、Solaris パッチ ID をコンマで区切って指定する必要があります。パッチは、リスト内に指定された順にインストールされます。
<i>patch_file</i>	<i>patch_location</i> 内のパッチのリストが格納されたファイルです。パッチは、ファイル内に指定された順にインストールされます。
<i>server_name</i>	パッチを格納したサーバーの名前を指定します。
<i>patch_directory</i>	指定されたサーバー上のパッチディレクトリの場所を指定します。標準形式のパッチを使用する必要があります。
<i>retry n</i>	オプションのキーワードです。 <i>n</i> は、インストールユーティリティがディレクトリのマウントを試みる回数の最大値です。

例 8-18 NFS を使った順序付きリストによるパッチの追加

この例の `patch` プロファイルキーワードは、`patch` ファイルに記載されたすべてのパッチを NFS パッチディレクトリ `nfs://patch_master/Solaris/v10/patches` から追加します。パッチは、`patch` 内の順番でインストールされます。マウントに失敗した場合、NFS マウントが 5 回試行されます。

```
patch patch_file nfs://patch_master/Solaris/v10/patches retry 5
```

例 8-19 NFS によるパッチの追加

この例の `patch` プロファイルキーワードは、サーバー `patch_master` のパッチディレクトリ `/Solaris/v10/patches` からパッチ `112467-01` と `112765-02` を追加します。

```
patch 112467-01,112765-02 nfs patch_master:/Solaris/v10/patches
```

HTTP サーバーに格納されたパッチ

パッチが HTTP サーバーに格納されている場合、次のいずれかの構文で `patch` キーワードを使用します。

`patch patch_id_list | patch_file http://server_name [:port] patch_directory optional_http_keywords`

<code>patch patch_id_list patch_file http server_name [:port] patch_directory optional_http_keywords</code>	
<code>patch_id_list</code>	インストールするパッチ ID 番号を指定します。このリストには、Solaris パッチ ID をコンマで区切って指定する必要があります。パッチは、リスト内に指定された順にインストールされません。コンマのあとに空白文字を入力しないでください。たとえば、112467-01,112765-02 のように指定します。
<code>patch_file</code>	<code>patch_location</code> 内のパッチのリストが格納されたファイルです。パッチは、ファイル内に指定された順にインストールされません。
<code>server_name</code>	パッチを格納したサーバーの名前を指定します。
<code>port</code>	オプションポートを指定します。 <code>port</code> は、ポート番号でも、実行時に決定されるポート番号を持つ TCP サービスの名前でもかまいません。 ポートを指定しなかった場合、デフォルトの HTTP ポート番号 80 が使用されます。
<code>patch_directory</code>	指定されたサーバーから取得するパッチディレクトリの場所を指定します。HTTP サーバーを使用する場合、パッチは JAR 形式でなければなりません。
<code>optional_keywords</code>	HTTP サーバーからパッチを取得するとき使用するオプションのキーワードを指定します。

表 8-7 HTTP で使用するオプションの patch キーワード

キーワード	値の定義
<code>timeout min</code>	<code>timeout</code> キーワードには、HTTP サーバーからのデータ受信を待機する最長の時間を分単位で指定できます。この時間に達すると、接続が切断され、再接続のあと、再開されます。 <code>timeout</code> 値として 0 (ゼロ) を指定すると、再接続は行われません。 タイムアウトによる再接続が発生すると、パッケージの先頭からインストールし直されます。タイムアウト以前に取得されたデータは破棄されます。
<code>proxy host:port</code>	<code>proxy</code> キーワードを使用して、プロキシホストとプロキシポートを指定できます。プロキシホストを使用すると、ファイアウォール越しに Solaris パッケージを取得できます。 <code>proxy</code> キーワードを指定する場合は、プロキシポートを指定する必要があります。

例 8-20 HTTP を使った順序付きリストによるパッチの追加

この例の patch プロファイルキーワードは、patch_file ファイルに記載されたすべてのパッチを HTTP の場所 `http://patch.central/Solaris/v10/patches` から追加します。パッチは、patch ファイルに指定された順でインストールされます。データが取得されないまま 5 分以上経過すると、パッチデータが再度取得されます。以前のパッチデータは破棄されます。

```
patch patch_file http://patch.central/Solaris/v10/patches timeout 5
```

例 8-21 HTTP によるパッチの追加

この例の patch プロファイルキーワードエントリは、パッチの場所 `http://patch_master/Solaris/v10/patches` からパッチ 112467-01 と 112765-02 を追加します。

```
patch 112467-01,112765-02 http://patch.central/Solaris/v10/patches
```

ローカルデバイスに格納されたパッチ

パッケージをファイルシステム指向のランダムアクセスデバイス (フロッピーディスク、DVD-ROM など) に格納している場合は、ローカルデバイスから Solaris パッケージを取得できます。patch キーワードでは次の構文を使用します。

```
patch patch_id_list | patch_file local_device \  
device path file_system_type
```

patch_id_list インストールするパッチ ID 番号を指定します。このリストには、Solaris パッチ ID をコンマで区切って指定する必要があります。パッチは、リスト内に指定された順にインストールされます。コンマのあとに空白文字を入力しないでください。たとえば、112467-01,112765-02 のように指定します。

patch_file *patch_location* 内のパッチのリストが格納されたファイルです。パッチは、ファイル内に指定された順にインストールされます。

device Solaris パッケージが格納されているドライブの名前を指定します。デバイス名が正規のパスである場合は、デバイスは直接マウントされます。正規のパスでないデバイス名を指定すると、インストールユーティリティーはパスに `/dev/dsk/` を加えます。

path Solaris パッチのパスを指定します。指定したデバイス上のルート (/) ファイルシステムからの相対パスで指定する必要があります。

file_system_type デバイス上のファイルシステムのタイプを指定します。ファイルシステムのタイプを指定しない場合、インストールユーティリティーは、UFS ファイルシステムのマウントを試みます。UFS の

マウントに失敗すると、インストールユーティリティーはHSFSファイルシステムのマウントを試みます。

例 8-22 ローカルデバイスを使った順序付きリストによるパッチの追加

この例の `patch` プロファイルキーワードは、`patch_file` ファイルに記載されているすべてのパッチをローカルデバイス `c0t6d0s0` の `/Solaris_10/patches` ディレクトリから追加します。パッチのインストールの順番は、`patch` ファイルによって決定されます。

```
patch patch_file c0t6d0s0 /Solaris_10/patches
```

例 8-23 ローカルデバイスによるパッチの追加

この例の `patch` プロファイルキーワードは、ローカルデバイス `c0t6d0s0` のパッチディレクトリ `/Solaris_10/patches` からパッチ `112467-01` と `112765-02` を追加します。

```
patch 112467-01,112765-02 local_device c0t6d0s0 /Solaris_10/patches
```

ローカルファイルに格納されたパッチ

システムをブートしたミニルートからパッチをインストールできます。カスタム `JumpStart` インストールを実施する時に、DVD、CD または NFS ベースのミニルートからシステムをブートします。このミニルートからインストールソフトウェアがロードされ、実行されます。したがって、DVD、CD または NFS ベースのミニルートに格納したパッチは、ローカルファイルとしてアクセスできます。`patch` キーワードでは次の構文を使用します。

```
patch patch_id_list | patch_file local_file patch_directory
```

patch_id_list インストールするパッチ ID 番号を指定します。このリストには、Solaris パッチ ID をコンマで区切って指定する必要があります。パッチは、リスト内に指定された順にインストールされます。コンマのあとに空白文字を入力しないでください。たとえば、`112467-01,112765-02` のように指定します。

patch_file *patch_location* 内のパッチのリストが格納されたファイルです。パッチは、ファイル内に指定された順にインストールされます。

patch_directory パッチディレクトリの場所を指定します。システムを Solaris SOFTWARE - 1 CD または Solaris Operating System DVD からブートしている間、システムからローカルファイルとしてアクセスできるパッチディレクトリでなければなりません。Solaris SOFTWARE - 1 CD または Solaris Operating System DVD からブートしている間

は、システムは `/net` にアクセスできません。

例 8-24 ローカルファイルを使った順序付きリストによるパッチの追加

この例の `patch` プロファイルキーワードは、`patch_file` ファイルに記載されているすべてのパッチを `/Solaris_10/patches` ディレクトリから追加します。パッチのインストールの順番は、`patch` ファイルによって決定されます。

```
patch patch_cal_file /Solaris_10/patches
```

例 8-25 ローカルファイルによるパッチの追加

この例の `patch` プロファイルキーワードは、パッチディレクトリ `/Solaris_10/patches` からパッチ `112467-01` と `112765-02` を追加します。

```
patch 112467-01,112765-02 local_file /Solaris_10/patches
```

patch キーワードを使用する際の制約

`patch` キーワードを使用する際には、次の制約があります。

- FTP の場所やローカルバックアップ (テープなど) からパッチを取得することはできません。
- 署名付きパッチは追加できません。
- `patchadd` コマンドでインストール可能なパッチでなければなりません。
- 現在インストールされていないパッチに依存しているパッチは、インストールされません。インストールログファイルまたはアップグレードログファイルにエラーメッセージが記録されます。
- パッチを正常にインストールするためには、正しい順番でインストールする必要があります。

root_device プロファイルキーワード

```
root_device slice
```

`root_device` は、システムのルートディスクを指定します。[162 ページの「システムのルートディスクを決定する方法」](#) に追加情報があります。

注-ルートディスクは、JumpStart プログラムによって決定され、OSがインストールされる場所を指定します。rules ファイルでは、プローブキーワード「rootdisk」を使用しますが、このキーワードはJumpStart プロファイルで使用される「rootdisk」キーワードとは使い方が異なります。rules ファイルのプローブキーワード「rootdisk」を使って、インストールの場所を設定することはできません。プローブキーワード rootdisk は、インストール時のブート元を決定します。表 8-10 を参照してください。

システムをアップグレードする場合、root_device は指定されるルート (/) ファイルシステムおよびその /etc/vfstab ファイルでマウントされるファイルシステムがアップグレードされることを示します。システム上で複数のルート (/) ファイルシステムがアップグレードできる場合は、root_device を指定する必要があります。slice は、cwtxdysz または cxdysz の形式で指定してください。

root_device キーワードを使用する場合は、次の点を考慮してください。

- 1つだけのディスクを持つシステムで root_device を指定する場合、root_device とディスクが一致する必要があります。また、ルート (/) ファイルシステムを指定する任意の filesys キーワードは、root_device と一致する必要があります。
- RAID-1 ボリューム (ミラー) をアップグレードする場合、root_device に指定する値はミラーの一方である必要があります。他方のミラーは自動的にアップグレードされます。

例 8-26 root_device プロファイルキーワード

```
root_device c0t0d0s2
```

システムのルートディスクを決定する方法

システムのルートディスクは、ルート (/) ファイルシステムを含むシステム上のディスクです。プロファイル内では、JumpStart プログラムがシステムのルートディスクを設定するディスク名の代わりに、この rootdisk 変数を使用できます。表 8-8 では、JumpStart プログラムがインストール用にシステムのルートディスクを決定する方法を説明します。

注-システムのルートディスクサイズが確認されるのは、初期インストール時だけです。アップグレードの場合、システムのルートディスクは変更できません。

表 8-8 JumpStart がシステムのルートディスクを決定する方法 (初期インストールのみ)

手順	操作
1	プロファイル内で <code>root_device</code> キーワードが指定されている場合、JumpStart プログラムは <code>rootdisk</code> をルートデバイスに設定します。
2	プロファイル内で、 <code>rootdisk</code> が設定されていなくて、 <code>boot_device</code> キーワードが指定されている場合、JumpStart プログラムは <code>rootdisk</code> をブートデバイスに設定します。
3	プロファイル内で <code>rootdisk</code> が設定されていなくて、 <code>filesys cwtxdysz size /</code> エントリが指定されている場合、JumpStart プログラムは <code>rootdisk</code> をエントリで指定されたディスクに設定します。
4	プロファイル内で <code>rootdisk</code> が設定されておらず、 <code>rootdisk.sn</code> エントリが指定されている場合、JumpStart プログラムは、システムのディスクから、カーネルのプロープの順番で、指定されたスライス上の既存のルートファイルシステムを検索します。ディスクが見つかった場合、JumpStart プログラムは見つかったディスクに <code>rootdisk</code> を設定します。
5	プロファイル内で、 <code>rootdisk</code> が設定されていなくて、 <code>partitioning existing</code> が指定されている場合、JumpStart プログラムはシステムのディスクで、(カーネルのプロープ順で) 既存のルートファイルシステムを検索します。ルートファイルシステムが見つからなかった場合、あるいは複数のルートファイルシステムが見つかった場合は、エラーが発生します。ルートファイルシステムが見つかった場合、JumpStart プログラムは見つかったディスクに <code>rootdisk</code> を設定します。
6	プロファイル内で <code>rootdisk</code> が設定されていない場合、JumpStart プログラムは、ルート (/) ファイルシステムがインストールされるディスクに <code>rootdisk</code> を設定します。

system_type プロファイルキーワード

`system_type type_switch`

`system_type` は、Solaris OS のインストール先のシステムのタイプを定義します。

`type_switch` は、オプション `standalone` または `server` を表します。このオプションは、Solaris ソフトウェアをインストールするシステムのタイプを指定するために使用します。`system_type` をプロファイルに指定しないと、デフォルトによって `standalone` が使用されます。

usedisk プロファイルキーワード

`usedisk disk_name ...`

`partitioning default` を指定すると、デフォルトではシステム上のすべての使用可能ディスクが使用されます。`usedisk` プロファイルキーワードには、JumpStart プログラムに使用させる 1 つ以上のディスクを指定します。`disk_name` は、`cxydz` または `cydz` 形式 (たとえば `c0t0d0` または `c0d0s0`) で指定する必要があります。

プロファイル内に `usedisk` を指定した場合、JumpStart プログラムは、`usedisk` キーワードの後ろに指定されたディスクだけを使用します。

注- 同じプロファイルに `usedisk` キーワードと `dontuse` キーワードを同時に指定することはできません。

カスタム JumpStart の環境変数

開始および終了スクリプトには、環境変数を使用できます。たとえば、開始スクリプトは、ディスクサイズ (`SI_DISKSIZE`) を抽出し、実際のディスクサイズにもとづいてシステムに特定のパッケージをインストールするかどうかを決定できます。

システムについて収集された情報は、これらの環境変数に格納されます。これらの変数は、通常、`rules` ファイルに使用するルールキーワードと値によって設定するかどうかが決まります。

たとえば、すでにシステムにインストールされているオペレーティングシステムに関する情報は、`installed` キーワードを使用した後にのみ、`SI_INSTALLED` から入手できます。

表 8-9 では、これらの変数とその値について説明します。

表 8-9 インストール環境変数

環境変数	値
<code>SI_ARCH</code>	インストールクライアントのハードウェアアーキテクチャー。 <code>SI_ARCH</code> 変数は、 <code>rules</code> ファイルで <code>arch</code> キーワードを使用する場合に設定されます。
<code>SI_BEGIN</code>	開始スクリプトを使用している場合はその名前。
<code>SI_CLASS</code>	インストールクライアントをインストールするために使用されるプロファイルの名前。
<code>SI_DISKLIST</code>	コマンドで区切られた、インストールクライアント上のディスク名のリスト。 <code>SI_DISKLIST</code> 変数は、 <code>rules</code> ファイルで <code>disksize</code> キーワードを使用して照合する場合に設定されます。 <code>SI_DISKLIST</code> および <code>SI_NUMDISKS</code> 変数は、 <code>rootdisk</code> に使用する物理ディスクを決定するために使用します。 <code>rootdisk</code> については、 162 ページ の「システムのルートディスクを決定する方法」を参照してください。
<code>SI_DISKSIZE</code>	コマンドで区切られた、インストールクライアント上のディスクサイズのリスト。 <code>SI_DISKSIZE</code> 変数は、 <code>rules</code> ファイルで <code>disksize</code> キーワードを使用して照合する場合に設定されます。
<code>SI_DOMAINNAME</code>	ドメイン名。 <code>SI_DOMAINNAME</code> 変数は、 <code>rules</code> ファイルで <code>domainname</code> キーワードを使用して照合する場合に設定されます。
<code>SI_FINISH</code>	終了スクリプトを使用する場合はその名前。
<code>SI_HOSTADDRESS</code>	インストールクライアントの IP アドレス。

表 8-9 インストール環境変数 (続き)

環境変数	値
SI_HOSTNAME	インストールクライアントのホスト名。SI_HOSTNAME 変数は、rules ファイルで hostname キーワードを使用して照合する場合に設定されます。
SI_INSTALLED	特定のオペレーティングシステムが入っているディスクのデバイス名 (Solaris、SunOS、System V など)。SI_INSTALLED 変数は、rules ファイルで installed キーワードを使用して照合する場合に設定されます。SI_INST_OS と SI_INST_VER は、SI_INSTALLED の値を決定するために使用します。
SI_INST_OS	オペレーティングシステムの名前。SI_INST_OS と SI_INST_VER は、SI_INSTALLED の値を決定するために使用します。
SI_INST_VER	オペレーティングシステムのバージョン。SI_INST_OS と SI_INST_VER は、SI_INSTALLED の値を決定するために使用します。
SI_KARCH	インストールクライアントのカーネルアーキテクチャー。SI_KARCH 変数は、rules ファイルで karch キーワードを使用して照合する場合に設定されます。
SI_MEMSIZE	インストールクライアントの物理メモリーの量。SI_MEMSIZE 変数は、rules ファイルで memsize キーワードを使用して照合する場合に設定されます。
SI_MODEL	インストールクライアントのモデル名。SI_MODEL 変数は、rules ファイルで model キーワードを使用して照合する場合に設定されます。
SI_NETWORK	インストールクライアントのネットワーク番号。SI_NETWORK 変数は、rules ファイルで network キーワードを使用して照合する場合に設定されます。
SI_NUMDISKS	インストールクライアントのディスク数。SI_NUMDISKS 変数は、rules ファイルで disksize キーワードを使用して照合する場合に設定されます。SI_NUMDISKS および SI_DISKLIST 変数は、rootdisk に使用する物理ディスクを決定するために使用します。rootdisk については、162 ページの「システムのルートディスクを決定する方法」を参照してください。
SI_OSNAME	Solaris ソフトウェアイメージのオペレーティングシステムリリース。SI_OSNAME 変数は、たとえば、Solaris Operating System DVD または Solaris SOFTWARE - 1 CD イメージ上のオペレーティングシステムのバージョンに基づいたシステムに Solaris ソフトウェアをインストールする場合にスクリプト内で指定できます。
SI_ROOTDISK	論理名 rootdisk によって表されるディスクのデバイス名。SI_ROOTDISK 変数は、rules ファイルで disksize または installed キーワードを rootdisk に設定した場合に設定されます。SI_ROOTDISK 変数は、インストール時のブートデバイスを設定します。 注 - rules ファイルのプロープキーワード「rootdisk」を使って、インストールの場所を設定することはできません。JumpStart プロファイルで設定される「rootdisk」変数については、162 ページの「システムのルートディスクを決定する方法」を参照してください。
SI_ROOTDISKSIZE	論理名 rootdisk によって表されるディスクのサイズ。SI_ROOTDISKSIZE 変数は、rules ファイルで disksize または installed キーワードを rootdisk に設定した場合に設定されます。

表 8-9 インストール環境変数 (続き)

環境変数	値
SI_TOTALDISK	インストールクライアント上のディスク容量の合計。SI_TOTALDISK 変数は、rules ファイルで totaldisk キーワードを使用して照合する場合に設定されます。

プローブキーワードと値

表 8-10 では、ルールキーワードと対応するプローブキーワードについて説明します。

注- プローブキーワードは、rules ファイルの最初、またはその近くに指定してください。

表 8-10 プローブキーワードの説明

ルールキーワード	対応するプローブキーワード	プローブキーワードの説明
any	なし	
arch	arch	カーネルアーキテクチャー (i386 または SPARC) を判断して SI_ARCH を設定します。
disksize	disks	システムのディスクサイズ (M バイト) をカーネルプローブ順 (c0t3d0s0、c0t3d0s1、c0t4d0s0) で返します。disksize は、SI_DISKLIST、SI_DISKSIZES、SI_NUMDISKS、および SI_TOTALDISK を設定します。
domainname	domainname	システムの NIS または NIS+ ドメイン名、あるいは空を返して、SI_DOMAINNAME を設定します。domainname キーワードは、domainname(1M) の出力を返します。
hostaddress	hostaddress	システムの IP アドレス (lo0 ではない ifconfig(1M) -a の出力にリストされた最初のアドレス) を返して、SI_HOSTADDRESS を設定します。
hostname	hostname	システムのホスト名 (uname(1) -n からの出力) を返して、SI_HOSTNAME を設定します。
installed	installed	システムにインストールされた Solaris OS のバージョン名を返して、SI_ROOTDISK と SI_INSTALLED を設定します。 Solaris リリースは検出されたが、バージョンを判断できないという場合は、返されるバージョンは SystemV になります。
karch	karch	システムのプラットフォームグループ (i86pc または sun4u) を返し、SI_KARCH を設定します。プラットフォーム名のリストは、『Solaris Sun ハードウェアマニユアル』(http://docs.sun.com) に記載されています。
memsize	memsize	システム上の物理メモリーのサイズ (M バイト) を返して、SI_MEMSIZE を設定します。

表 8-10 プローブキーワードの説明 (続き)

ルールキーワード	対応するプローブキーワード	プローブキーワードの説明
model	model	システムのプラットフォーム名を返して、SI_MODEL を設定します。プラットフォーム名のリストは、『Solaris Sun ハードウェアマニュアル』(http://docs.sun.com)に記載されています。
network	network	システムのネットワーク番号を返します。これは JumpStart プログラムがシステムの IP アドレスとサブネットマスクの論理和をとって判断します。システムの IP アドレスとサブネットマスクは、lo0 ではない ifconfig(1M) -a 出力にリストされた最初のアドレスから抽出されます。network キーワードは SI_NETWORK も設定します。
osname	osname	CD で検出された Solaris OS のバージョンおよびオペレーティングシステム名を返して、SI_OSNAME を設定します。 Solaris リリースは検出されたが、バージョンを判断できないという場合は、返されるバージョンは SystemV になります。
	rootdisk	システムのルートディスクの名前とサイズ (M バイト) を返して、SI_ROOTDISK を設定します。
totaldisk	totaldisk	システム上のディスク領域の合計 (M バイト) を返して、SI_TOTALDISK を設定します。ディスク容量の合計には、システムに接続された操作可能なディスクすべてが含まれます。

パート II

付録

このパートでは、問題への対処方法と参考情報を提供します。

問題発生時の解決方法 (作業)

この章では、Solaris 10 5/08 ソフトウェアのインストール時に発生する可能性のあるエラーメッセージと一般的な問題の一覧を示します。この章では、それぞれの問題の解決方法も示します。まず、次のリストを使用して、インストールプロセスのどこで問題が発生したか確認してください。

- 171 ページの「ネットワークインストールの設定に関する問題」
- 172 ページの「システムのブートに関する問題」
- 178 ページの「Solaris OS の初期インストール」
- 181 ページの「Solaris OS のアップグレード」

注- この付録で「ブート可能なメディア」と記載されている場合、これはインストールプログラムおよび JumpStart インストールを意味します。

ネットワークインストールの設定に関する問題

Unknown client "*host_name*"

原因: `add_install_client` コマンドの *host_name* 引数は、ネームサービス内のホストではありません。

対処方法: ホスト *host_name* をネームサービスに追加し、`add_install_client` コマンドを実行し直してください。

Error: <system name> does not exist in the NIS ethers map

Add it, and rerun the `add_install_client` command

説明: `add_install_client` コマンドを実行すると、上記のエラーが出てコマンドが失敗します。

原因: インストールサーバーに追加しようとしているクライアントが、サーバーの `/etc/ethers` ファイルに存在しません。

対処方法: 必要な情報をインストールサーバー上の `/etc/ethers` ファイルに追加し、`add_install_client` コマンドを実行し直してください。

1. スーパーユーザーになるか、同等の役割になります。
2. このクライアント上で `ethers` アドレスを見つけます。

```
# ifconfig -a grep ethers
ether 8:0:20:b3:39:1d
```

3. インストールサーバー上で、エディタを使って `/etc/ethers` ファイルを開きます。そのアドレスをリストに追加します。
4. この例のように、クライアント上で `add_install_client` をもう一度実行します。

```
# ./add_install_client bluegill sun4u
```

システムのブートに関する問題

メディアからのブート時のエラーメッセージ

le0: No carrier - transceiver cable problem

原因: システムがネットワークに接続されていません。

対処方法: ネットワークに接続せずに使用しているシステムの場合は、このメッセージは無視してください。ネットワークに接続されているシステムの場合は、Ethernet が正しく接続されているかどうか確認してください。

The file just loaded does not appear to be executable

原因: ブート用の適切なメディアが見つかりません。

対処方法: インストールサーバーからネットワークを介して Solaris 10 5/08 ソフトウェアをインストールするように正しく設定されているか確認します。次に、確認事項の例を示します。

- Solaris Operating System DVD または Solaris SOFTWARE CD のイメージをインストールサーバーにコピーした場合、システムの設定時に正しいプラットフォームグループを指定したことを確認します。
- DVD または CD メディアを使用する場合、Solaris Operating System DVD または Solaris SOFTWARE - 1 CD がインストールサーバーにマウントされ、アクセス可能であることを確認します。

boot: cannot open <filename> (SPARC ベースのシステムのみ)

原因: このエラーは、`boot- file` の値を明示的に上書きしたときに発生します。

注 `-filename` は、問題のファイルの名前を示す変数です。

対処方法: 次の手順に従ってください。

- PROM の `boot-file` の値を (無指定) に設定変更します。
- `diag-switch` が `off` と `true` に設定されているか確認します。

Can't boot from file/device

原因: インストールメディアがブート可能なメディアを見つけることができません。

対処方法: 次の条件が満たされているか確認します。

- DVD-ROM または CD-ROM ドライブがシステムに適切に取り付けられ、電源が入っている。
- Solaris Operating System DVD または Solaris SOFTWARE - 1 CD がドライブに挿入されている。
- ディスクに傷や埃が付いていない。

WARNING: clock gained xxx days -- CHECK AND RESET DATE! (SPARC システムのみ)

説明: これは参考情報です。

対処方法: メッセージは無視して、インストールを継続してください。

Not a UFS file system (x86 システムのみ)

原因: Solaris 10 5/08 ソフトウェアをインストールしたとき (Solaris インストールプログラムまたはカスタム JumpStart を使って)、ブートドライブを選択しませんでした。BIOS を編集してシステムをブートする必要があります。

対処方法: BIOS を選択してブートします。詳細は、BIOS のマニュアルを参照してください。

メディアからのブート時の一般的な問題

システムがブートしない。

説明: 初めてカスタム JumpStart サーバーを設定する場合、エラーメッセージを返さないブート問題が発生することがあります。システムについての情報およびシステムがどのようにブートするかを調べるには、`-v` オプションを指定してブートコマンドを実行してください。`-v` オプションを使用すると、ブートコマンドは画面に関する詳しいデバッグ情報を表示します。

注-このフラグを指定しなくてもメッセージは出力されますが、システムログファイルが出力先となります。詳細については、`syslogd(1M)` を参照してください。

対処方法:SPARC システムの場合、`ok` プロンプトに続いて次のコマンドを入力します。

```
ok boot net -v - install
```

Toshiba SD-M 1401 DVD-ROM が搭載されたシステムで DVD メディアからのブートが失敗する

説明: 使用しているシステムにファームウェアバージョン 1007 の Toshiba SD-M1401 DVD-ROM が搭載されている場合、システムは Solaris Operating System DVD からブートできません。

対処方法:111649-03 以降のパッチを適用して Toshiba SD-M1401 DVD-ROM ドライブのファームウェアを更新します。パッチ 111649-03 は、sunsolve.sun.com から入手可能です。

メモリー増設用以外の PC カードを挿入すると、システムがハングまたはパニックを起こす。(x86 システムのみ)

原因:メモリー増設用以外の PC カードは、ほかのデバイスが使用するのと同じメモリーリソースを使用できません。

対処方法:この問題を解決するには、PC カードのマニュアルを参照してアドレス範囲を確認してください。

システムがプロンプトを出す前にハングする。(x86 システムのみ)

対処方法:サポートされていないハードウェアです。ハードウェアのマニュアルを参照してください。

ネットワークからのブート時のエラーメッセージ

WARNING: getfile: RPC failed: error 5 (RPC Timed out).

説明:インストールクライアントのブート要求に対して、ネットワーク上の複数のサーバーが応答したときに発生するエラーです。インストールクライアントの接続先のブートサーバーが間違っているため、インストールは停止します。次の原因が考えられます。

原因:1 このインストールクライアントが登録された `/etc/bootparams` ファイルが複数のサーバーに存在する可能性があります。

対処方法:1 ネットワーク上の複数のサーバーの /etc/bootparams エントリにインストールクライアントが登録されていないか調べます。複数のサーバーに登録されている場合は、インストールに使用するインストールサーバー(またはブートサーバー)以外のサーバーの /etc/bootparams ファイルから、クライアントの登録を削除します。

原因:2 複数の /tftpboot または /rplboot ディレクトリにこのインストールクライアントが登録されている可能性があります。

対処方法:2 ネットワーク上の複数のサーバーの /tftpboot または /rplboot ディレクトリにインストールクライアントが登録されていないか調べます。複数のサーバーに登録されている場合は、インストールに使用するインストールサーバー(またはブートサーバー)以外のサーバーの /tftpboot または /rplboot ディレクトリから、クライアントの登録を削除します。

原因:3 あるサーバーの /etc/bootparams ファイルにこのインストールクライアントが登録されており、別のサーバーの /etc/bootparams ファイルで、すべてのシステムがプロファイルサーバーにアクセスできるように記述されている可能性があります。次のような記述があります。

```
* install_config=profile_server:path
```

このエラーは、NIS または NIS+ の bootparams テーブルにこのような行が存在していても発生します。

対処方法:3 ワイルドカードエントリがネームサービスの bootparams マップまたはテーブルにある場合 (* install_config= など) は、そのエントリを削除し、ブートサーバーの /etc/bootparams ファイルに追加します。

No network boot server. Unable to install the system. See installation instructions. (SPARC システムのみ)

原因: このエラーは、ネットワークからインストールしようとしているシステムで発生します。このシステムは、適切に設定されていません。

対処方法: ネットワークを介してインストールするようにシステムが適切に設定されているか確認します。『Solaris 10 5/08 インストールガイド(ネットワークインストール)』の「CD イメージを使用してネットワークからインストールするシステムの追加」を参照してください。

prom_panic: Could not mount file system (SPARC システムのみ)

原因: このエラーはネットワークから Solaris をインストールしようとしてブートソフトウェアが次のものを見つけられない場合に発生します。

- Solaris Operating System DVD またはインストールサーバー上の Solaris Operating System DVD イメージコピー
- Solaris SOFTWARE - 1 CD またはインストールサーバー上の Solaris SOFTWARE - 1 CD イメージコピー

対処方法: インストール用のソフトウェアがマウントされ共有されるように設定してあることを確認します。

- インストールサーバーの DVD-ROM または CD-ROM ドライブから Solaris をインストールする場合は、Solaris Operating System DVD または Solaris SOFTWARE - 1 CD が CD-ROM ドライブに挿入されてマウントされていること、および `/etc/dfs/dfstab` ファイルで共有されていることを確認します。
- インストールサーバーのディスク上にある Solaris Operating System DVD イメージまたは Solaris SOFTWARE - 1 CD イメージのコピーからインストールする場合は、そのコピーのディレクトリパスが `/etc/dfs/dfstab` ファイル内で共有されていることを確認します。

Timeout waiting for ARP/RARP packet... (SPARC システムのみ)

原因: 1 クライアントはネットワークを介してブートしようとしていますが、認識してくれるシステムを見つけることができません。

対処方法: 1 システムのホスト名が NIS または NIS+ のネームサービスに登録されていることを確認します。また、ブートサーバーの `/etc/nsswitch.conf` ファイル内の `bootparams` の検索順序を確認します。

たとえば、`/etc/nsswitch.conf` ファイル内にある次の行は、JumpStart または Solaris インストールプログラムが最初に NIS マップから `bootparams` 情報を探すことを示しています。ここで情報が見つからない場合、インストーラはブートサーバーの `/etc/bootparams` ファイルを調べます。

```
bootparams: nis files
```

原因: 2 クライアントの Ethernet アドレスが不正です。

対処方法: 2 インストールサーバーの `/etc/ethers` ファイルにあるクライアントの Ethernet アドレスが正しいことを確認します。

原因: 3 カスタム JumpStart インストールでは、特定のサーバーをインストールサーバーとして使用するようプラットフォームグループを `add_install_client` コマンドによって指定します。`add_install_client` を使用する際に不正な構成値を使用すると、この問題が発生します。たとえば、インストールするマシンが `sun4u` であるのに誤って `i86pc` と指定した場合などが考えられます。

対処方法: 3 正しいアーキテクチャー値を使用して `add_install_client` を実行し直します。

```
ip: joining multicasts failed on tr0 - will use link layer broadcasts for  
multicast (x86 システムのみ)
```

原因: このエラーメッセージは、トークンリングカードを使ってシステムをブートしたときに表示されます。Ethernet のマルチキャストとトークンリングのマルチキャストの動作は異なります。ドライバはこのエラーメッセージを返して、マルチキャストアドレスが無効なことを知らせます。

対処方法: このエラーメッセージは無視してください。マルチキャストがうまく動作しなければ、IP は代わりにレイヤーブロードキャストを使用し、インストールは失敗しません。

Requesting Internet address for *Ethernet_Address* (x86 システムのみ)

原因: クライアントはネットワークを介してブートしようとしていますが、認識してくれるシステムを見つけることができません。

対処方法: システムのホスト名がネームサービスに登録されていることを確認します。システムのホスト名が NIS または NIS+ のネームサービスに登録されているのに、システムがこのエラーメッセージを表示し続ける場合は、リブートしてください。

RPC: Timed out No bootparams (whoami) server responding; still trying... (x86 システムのみ)

原因: クライアントはネットワークからブートしようとしていますが、インストールサーバー上の `/etc/bootparams` ファイルにエントリを持つシステムを見つけることができません。

対処方法: インストールサーバー上で `add_install_client` を使用します。これにより `/etc/bootparams` ファイルに適切なエントリが追加され、クライアントがネットワークからブートできるようになります。

Still trying to find a RPL server... (x86 システムのみ)

原因: システムはネットワークからブートしようとしていますが、サーバーではこのシステムをブートするように設定されていません。

対処方法: インストールサーバー上で、インストールするシステム用に `add_install_client` を実行します。 `add_install_client` コマンドは、必要なネットワークブートプログラムを含む `/rplboot` ディレクトリを設定します。

CLIENT MAC ADDR: FF FF FF FF FF FF (DHCP によるネットワークインストールのみ)

原因: DHCP サーバーが正しく構成されていません。このエラーは、DHCP マネージャ内でオプションやマクロが正しく定義されていない場合に発生する可能性があります。

対処方法: DHCP マネージャで、オプションおよびマクロが正しく定義されていることを確認します。ルーターオプションが定義されており、その値がネットワークインストールで使用するサブネットを正しく表していることを確認します。

ネットワークからのブート時の一般的な問題

システムはネットワークを介してブートされるが、指定したインストールサーバー以外のシステムからブートされる。

原因: このクライアントが登録された `/etc/bootparams` エントリと `/etc/ethers` エントリが別のシステム上に存在します。

対処方法: ネームサーバー上で、インストールするシステムの `/etc/bootparams` エントリを更新します。このエントリは、次の構文に従う必要があります。

```
install_system root=boot_server:path install=install_server:path
```

また、サブネット内で複数のサーバーの `bootparams` ファイルにインストールクライアントが登録されていないか確認します。

システムがネットワークからブートしない (DHCP によるネットワークインストールのみ)。

原因: DHCP サーバーが正しく構成されていません。このエラーは、システムが DHCP サーバーのインストールクライアントとして構成されていない場合に発生することがあります。

対処方法: DHCP マネージャーソフトウェアで、クライアントシステムのインストールオプションとマクロが定義されていることを確認します。詳細は、『Solaris 10 5/08 インストールガイド (ネットワークインストール)』の「DHCP サービスによるシステム構成情報の事前設定 (作業)」を参照してください。

Solaris OS の初期インストール

初期インストールが失敗する。

対処方法: Solaris のインストールが失敗する場合、インストールを再実行する必要があります。インストールを再実行するには、Solaris Operating System DVD、Solaris SOFTWARE - 1 CD、またはネットワークを利用してシステムをブートします。

Solaris ソフトウェアが部分的にインストールされたあとでは、このソフトウェアのインストールを解除することはできません。バックアップからシステムを復元するか、Solaris インストールの処理をもう一度行う必要があります。

```
/cdrom/cdrom0/SUNW xxxx/reloc.cpio: Broken pipe
```

説明: このエラーメッセージは参考情報であり、インストールには影響を及ぼしません。読み取りプロセスを持たないパイプに書き込みを行うと発生します。

対処方法: メッセージは無視して、インストールを継続してください。

WARNING: CHANGE DEFAULT BOOT DEVICE (x86 システムのみ)

原因: これは参考情報です。システムの BIOS に設定されているデフォルトブートデバイスが、ブート時に Solaris Device Configuration Assistant (デバイス構成用補助)を必要とするように設定されている可能性があります。

対処方法: インストールを続行します。Solaris Device Configuration Assistant (デバイス構成用補助)を必要としないデバイスに Solaris ソフトウェアをインストールし終わったら、必要に応じて、BIOS に指定されたシステムのデフォルトのブートデバイスを変更します。

x86 のみ - locale キーワードを使用して初期インストール用のカスタム JumpStart プロファイル进行测试する場合は、`pfinstall -D` コマンドでプロファイル进行测试すると失敗します。回避方法については、181 ページの「Solaris OS のアップグレード」セクションのエラーメッセージ「could not select locale」を参照してください。

▼ x86: IDE ディスクの不良ブロックの検査

IDE ディスクドライブは、Solaris ソフトウェアがサポートするほかのドライブのように、不良ブロックを自動的に無効にしません。IDE ディスク上に Solaris をインストールする前に、ディスクの検査を実行してください。IDE ディスクの検査を行うには、次の手順に従います。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 インストールメディアに応じた方法でブートします。
- 3 インストールタイプの選択を求めるプロンプトが表示されたら、オプション 6 「Single user shell」を選択します。
- 4 `format (1M)` プログラムを起動します。
`format`
- 5 ディスク面の検査をする IDE ディスクドライブを指定します。
`cx dy`
`cx` コントローラ番号
`dy` デバイス番号
- 6 `fdisk` パーティションが存在するかどうかを確認します。

- Solaris fdisk パーティションが存在する場合、[手順 7](#)へ進みます。
- Solaris fdisk パーティションが存在しない場合、fdisk コマンドを使用してディスク上に Solaris パーティションを作成します。

```
format> fdisk
```

- 7 次のように入力して、表面解析を開始します。

```
format> analyze
```

- 8 次のように入力して、現在の設定を確認します。

```
analyze> config
```

- 9 (省略可能) 次のように入力して、設定を変更します。

```
analyze> setup
```

- 10 次のように入力して、不正ブロックを探します。

```
analyze> type_of_surface_analysis
```

type_of_surface_analysis read、write、または compare を指定します。

format が不良ブロックを発見すると、それらの再マッピングを実行します。

- 11 次のように入力して、解析を終了します。

```
analyze> quit
```

- 12 再マッピングするブロックを指定するかどうか決定します。

- 指定しない場合は、[手順 13](#)へ進みます。
- 指定する場合は、次のコマンドを入力します。

```
format> repair
```

- 13 次のように入力して、format プログラムを終了します。

```
quit
```

- 14 次のコマンドを入力して、マルチユーザーモードでメディアを再起動します。

```
# exit
```

Solaris OS のアップグレード

アップグレード時のエラーメッセージ

No upgradable disks

原因: /etc/vfstab ファイルのスワップエントリが原因でアップグレードに失敗しました。

対処方法: /etc/vfstab ファイルの次の行をコメントにしてください。

- アップグレードしないディスク上のスワップファイルとスライスを指定している行
- 存在しないスワップファイルを指定している行
- 使用していないスワップスライスを指定している行

usr/bin/bzcat not found

原因: パッチクラスタが必要なために Solaris Live Upgrade が失敗しています。

対処方法: Solaris Live Upgrade のインストールでパッチを使用する必要が生じました。 <http://sunsolve.sun.com> で最新のパッチリストを確認してください。 SunSolve Web サイトで infodoc 72099 を検索してください。

Upgradeable Solaris root devices were found, however, no suitable partitions to hold the Solaris install software were found. Upgrading using the Solaris Installer is not possible. It might be possible to upgrade using the Solaris Software 1 CDRom. (x86 システムのみ)

原因: 十分な容量がないため、Solaris SOFTWARE - 1 CD でアップグレードできません。

対処方法: アップグレードするには、512Mバイト以上のスワップスライスを作成するか、別のアップグレード方法 (Solaris Operating System DVD の Solaris インストールプログラム、ネットインストールイメージ、JumpStart など) を選択します。

ERROR: Could not select locale (x86 システムのみ)

原因: pfinstall -D コマンドを使用して JumpStart プロファイルをドライランテストするとき、次のような条件下ではテストが失敗します。

- プロファイルに locale キーワードが含まれている。
- GRUB ソフトウェアが含まれているリリースをテストしている。Solaris 10 1/06 以降のリリースでは、GRUB ブートローダーにより、システムにインストールされているさまざまなオペレーティングシステムを GRUB メニューで簡単にブートできます。

GRUB ソフトウェアの導入に伴い、ミニルートは圧縮されています。ソフトウェアでは、圧縮されたミニルートからロケールのリストを見つけることができなくなりました。ミニルートは最小の Solaris ルート (/) ファイルシステムであり、Solaris インストールメディアに収録されています。

対処方法: 次の手順を実行します。次の値を使用してください。

- MEDIA_DIR は /cdrom/cdrom0/
- MINIRoot_DIR は \$MEDIA_DIR /Solaris_10/Tools/Boot
- MINIRoot_ARCHIVE は \$MEDIA_DIR/boot/x86.miniroot
- TEMP_FILE_NAME は /tmp/test

1. スーパーユーザーになるか、同等の役割になります。

役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。

2. ミニルートアーカイブを圧縮解除します。

```
# /usr/bin/gzcat $MINIRoot_ARCHIVE > $TEMP_FILE_NAME
```

3. lofiadm コマンドを使用して、ミニルートデバイスを作成します。

```
# LOFI_DEVICE=/usr/sbin/lofiadm -a $TEMP_FILE_NAME
# echo $LOFI_DEVICE
/dev/lofi/1
```

4. lofi コマンドを使用して、ミニルートディレクトリの下にミニルートをマウントします。

```
# /usr/sbin/mount -F ufs $LOFI_DEVICE $MINIRoot_DIR
```

5. プロファイルをテストします。

```
# /usr/sbin/install.d/pfinstall -D -c $MEDIA_DIR $path-to-jumpstart_profile
```

6. テストが完了したら、lofi デバイスのマウントを解除します。

```
# umount $LOFI_DEVICE
```

7. lofi デバイスを削除します。

```
# lofiadm -d $TEMP_FILE_NAME
```

アップグレード時の一般的な問題

システム上にアップグレード可能なバージョンの Solaris ソフトウェアが存在するにもかかわらず、アップグレードオプションが提供されない。

原因: `/var/sadm` ディレクトリがシンボリックリンクであるか、別のファイルシステムからマウントされたディレクトリです。

対処方法: `/var/sadm` ディレクトリをルート (`/`) または `/var` ファイルシステムに移動します。

原因: `/var/sadm/softinfo/INST_RELEASE` ファイルが存在しません。

対処方法: 2次の形式で新しく `INST_RELEASE` ファイルを作成します。

```
OS=Solaris
VERSION=x
REV=0
```

`x` システム上の Solaris ソフトウェアのバージョン

原因: `/var/sadm/softinfo` に `SUNWusr` が存在しません。

対処方法: 3初期インストールを行う必要があります。この Solaris ソフトウェアはアップグレードできません。

md ドライバの停止または初期化に失敗する。

対処方法: 次の手順に従ってください。

- ファイルシステムが RAID-1 ボリュームでない場合は、`vsftab` ファイル内でコメントにします。
- ファイルシステムが RAID-1 ボリュームである場合は、ミラーを解除し、再インストールします。ミラーを解除する方法については、『Solaris ボリュームマネージャの管理』の「RAID-1 ボリュームの削除 (ミラー化の解除)」を参照してください。

Solaris インストールプログラムがファイルシステムをマウントできないため、アップグレードに失敗する。

原因: アップグレード時に、スクリプトは、アップグレード対象のルート (`/`) ファイルシステム上に、システムの `/etc/vfstab` ファイルに記載されているすべてのファイルシステムをマウントしようとします。インストールプログラムがファイルシステムをマウントできない場合、失敗して終了します。

対処方法: システムの `/etc/vfstab` ファイル内のすべてのファイルシステムがマウントできることを確認します。`/etc/vfstab` ファイル内のマウントできない、あるいは問題の原因になっている可能性があるファイルシステムは、すべてコメン

トにします。Solaris インストールプログラムはアップグレード中、コメントにしたファイルシステムをマウントしません。アップグレードされるソフトウェアを含むシステムベースのファイルシステム(たとえば /usr)は、コメントにできません。

アップグレードが失敗する。

説明: システムにアップグレードに対応できるだけの十分なディスク容量がありません。

原因: 『Solaris 10 5/08 インストールガイド (インストールとアップグレードの計画)』の「ディスク容量の再配置を伴うアップグレード」を参照してディスク容量に問題がないかを確認し、自動配置機能による領域の再配置を行わずに解決できるかどうかを調べます。

RAID-1 ボリュームのルート (/) ファイルシステムのアップグレードに関連する問題
対処方法: ルート (/) ファイルシステムが Solaris ボリュームマネージャーの RAID-1 ボリュームである場合に、アップグレードの問題が発生するときは、『Solaris ボリュームマネージャの管理』の第 25 章「Solaris ボリュームマネージャのトラブルシューティング (作業)」を参照してください。

▼ 問題発生後にアップグレードを継続する方法

アップグレードに失敗し、システムをブートできない場合があります。このような状況は、電源の故障やネットワーク接続の障害などが発生した場合に起こる可能性があります、制御できない場合に発生します。

- 1 **Solaris Operating System DVD、Solaris SOFTWARE-1 CD、またはネットワークからシステムをリブートします。**
- 2 インストール用のアップグレードオプションを選択します。
Solaris インストールプログラムは、システムが部分的にアップグレードされているか判断し、アップグレードを継続します。

x86: GRUB を使用する場合の Solaris Live Upgrade に関する問題

x86 システムで Solaris Live Upgrade と GRUB ブートローダーを使用すると次のようなエラーが発生する可能性があります。

ERROR: The media product tools installation directory *path-to-installation-directory* does not exist.

ERROR: The media *dirctory* does not contain an operating system upgrade image.

説明: これらのエラーメッセージは、新しいブート環境をアップグレードするために `luupgrade` コマンドを使用するときに発生します。

原因: 古いバージョンの Solaris Live Upgrade が使用されています。システムにインストールした Solaris Live Upgrade パッケージは、メディアおよびメディアに記録されているリリースと互換性がありません。

対処方法: Solaris Live Upgrade パッケージは、常にアップグレードするリリースのものを使用してください。

例: 次の例のエラーメッセージは、システムの Solaris Live Upgrade パッケージのバージョンがメディアのパッケージのバージョンと異なることを示しています。

```
# luupgrade -u -n s10u1 -s /mnt
Validating the contents of the media </mnt>.
The media is a standard Solaris media.
ERROR: The media product tools installation directory
</mnt/Solaris_10/Tools/Boot/usr/sbin/install.d/install_config> does
not exist.
ERROR: The media </mnt> does not contain an operating system upgrade
image.
```

ERROR: Cannot find or is not executable: </sbin/biosdev>.

ERROR: One or more patches required by Solaris Live Upgrade has not been installed.

原因: Solaris Live Upgrade で必要とされる 1 つ以上のパッチが、システムにインストールされていません。このエラーメッセージでは、欠落しているすべてのパッチを認識しているわけではありません。

対処方法: Solaris Live Upgrade を使用する前に、必要なパッチすべてを必ずインストールしてください。 <http://sunsolve.sun.com> を参照して、最新のパッチリストを使用しているかどうか確認してください。SunSolve Web サイトで `infodoc 72099` を検索してください。

ERROR: Device mapping command </sbin/biosdev> failed. Please reboot and try again.

原因: 1 Solaris Live Upgrade が、以前の管理作業が原因でデバイスをマップできません。

対処方法: 1 システムをリブートして、もう一度 Solaris Live Upgrade を実行します

原因:2システムをリブートしても同じエラーメッセージが表示される場合は、2つ以上の同一ディスクがあります。デバイスのマッピングコマンドがそれらのディスクを区別できません。

対処方法:2ディスクの一方に、新しいダミーの `fdisk` パーティションを作成します。`fdisk(1M)` のマニュアルページを参照してください。そのあとで、システムをリブートします。

Cannot delete the boot environment that contains the GRUB menu

原因:Solaris Live Upgrade には、ブート環境に GRUB メニューが含まれる場合はブート環境を削除できないという制限があります。

対処方法:`lumake(1M)` コマンドまたは `luupgrade(1M)` コマンドを使用してブート環境を再使用します。

The file system containing the GRUB menu was accidentally remade. However, the disk has the same slices as before. For example, the disk was not re-sliced.

原因:GRUB メニューを含むファイルシステムは、システムをブート可能な状態に維持するために不可欠です。Solaris Live Upgrade コマンドは、GRUB メニューを破棄しません。ただし、Solaris Live Upgrade コマンド以外のコマンドを使用して GRUB メニューのあるファイルシステムを誤って再作成または破棄すると、回復ソフトウェアは GRUB メニューの再インストールを試みます。回復ソフトウェアは、次のリブート時に GRUB メニューを同じファイルシステムに戻します。たとえば、ファイルシステムで `newfs` または `mkfs` コマンドを使用し、誤って GRUB メニューを破棄してしまったとします。GRUB メニューを正しく復元するには、スライスが次の条件を満たす必要があります。

- マウント可能なファイルシステムが含まれている
- スライスが以前に存在していた Solaris Live Upgrade ブート環境の一部である

システムをリブートする前に、必要であればスライスを修正します。

対処方法:システムをリブートします。GRUB メニューのバックアップコピーが自動的にインストールされます。

The GRUB menu's `menu.lst` file was accidentally deleted.

対処方法:システムをリブートします。GRUB メニューのバックアップコピーが自動的にインストールされます。

▼ Veritas VxVm の実行中に Solaris Live Upgrade を使用してアップグレードするとシステムパニックが発生する

実行中の Veritas VxVM で Solaris Live Upgrade を用いてアップグレードを行う場合、次の手順でアップグレードを行わないと、リブート時にシステムパニックが発生します。この問題は、パッケージが Solaris の最新のパッケージガイドラインに従っていない場合に発生します。

- 1 スーパーユーザーになるか、同等の役割になります。
役割には、認証と特権コマンドが含まれます。役割の詳細については、『Solaris のシステム管理 (セキュリティサービス)』の「RBAC の構成 (作業マップ)」を参照してください。
- 2 非アクティブブート環境を作成します。『Solaris 10 5/08 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「新しいブート環境の作成」を参照してください。
- 3 非アクティブブート環境をアップグレードする前に、非アクティブブート環境上の既存の Veritas ソフトウェアを無効にする必要があります。

- a. 非アクティブブート環境をマウントします。

```
# lumount inactive_boot_environment_name mount_point
```

次に例を示します。

```
# lumount solaris8 /mnt
```

- b. 次の例のように、`vfstab` が含まれているディレクトリに移動します。

```
# cd /mnt/etc
```

- c. 次の例のように、非アクティブブート環境の `vfstab` ファイルをコピーします。

```
# cp vfstab vfstab.501
```

- d. 次の例のように、コピーされた `vfstab` 内のすべての Veritas ファイルシステムエントリをコメントにします。

```
# sed '/vx\|dsk/s/^\#/g' < vfstab > vfstab.novxfs
```

各行の最初の文字が # に変わり、その行がコメント行になります。このコメント行は、システムファイルのコメント行とは異なります。

- e. 次の例のように、変更した `vfstab` ファイルをコピーします。

```
# cp vfstab.novxfs vfstab
```

- f. 次の例のように、非アクティブブート環境のシステムファイルがあるディレクトリに移動します。

```
# cd /mnt/etc
```

- g. 次の例のように、非アクティブブート環境のシステムファイルをコピーします。

```
# cp system system.501
```

- h. `drv/vx` を含むすべての `forceload:` エントリをコメントにします。

```
# sed '/forceload:  drv\/vx\/s\/^\/*/' <system> system.novxfs
```

各行の最初の文字が `*` に変わり、その行がコメント行になります。このコメント行は、`vfstab` ファイルのコメント行とは異なります。

- i. 次の例のように、Veritas `install-db` ファイルを作成します。

```
# touch vx/reconfig.d/state.d/install-db
```

- j. 非アクティブブート環境のマウントを解除します。

```
# luumount inactive_boot_environment_name
```

- 4 非アクティブブート環境をアップグレードします。『Solaris 10 5/08 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の第 5 章「Solaris Live Upgrade によるアップグレード (作業)」を参照してください。

- 5 非アクティブブート環境をアクティブにします。『Solaris 10 5/08 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「ブート環境のアクティブ化」を参照してください。

- 6 システムをシャットダウンします。

```
# init 0
```

- 7 非アクティブブート環境をシングルユーザーモードでブートします。

```
OK boot -s
```

`vxvm` または `VXVM` を含むメッセージとエラーメッセージがいくつか表示されますが、これらは無視してかまいません。非アクティブブート環境がアクティブになります。

- 8 Veritas をアップグレードします。
 - a. 次の例のように、システムから Veritas VRTSvmsa パッケージを削除します。

```
# pkgrm VRTSvmsa
```
 - b. Veritas パッケージがあるディレクトリに移動します。

```
# cd /location_of_Veritas_software
```
 - c. システムに最新の Veritas パッケージを追加します。

```
# pkgadd -d 'pwd' VRTSvxvm VRTSvmsa VRTSvmdoc VRTSvman VRTSvmdv
```
- 9 元の vfstab とシステムファイルを復元します。

```
# cp /etc/vfstab.original /etc/vfstab
# cp /etc/system.original /etc/system
```
- 10 システムをリブートします。

```
# init 6
```

x86: 既存のサービスパーティションが存在しないシステムでは、デフォルトでサービスパーティションが作成されない

診断・サービスパーティションの存在しないシステム上に Solaris 10 5/08 OS をインストールすると、インストールプログラムがデフォルトでサービスパーティションを作成しない場合があります。Solaris パーティションと同じディスクにサービスパーティションを作成する場合、Solaris 10 5/08 OS をインストールする前にサービスパーティションを作り直す必要があります。

サービスパーティションが存在するシステム上に Solaris 8 2/02 OS をインストールすると、インストールプログラムがサービスパーティションを保持しない場合があります。サービスパーティションを保持するように fdisk ブートパーティションレイアウトを手動で編集しなかった場合、インストールプログラムはインストール時にサービスパーティションを削除します。

注 - Solaris 8 2/02 OS のインストール時にサービスパーティションの保持を明示的に指定しなかった場合、サービスパーティションを作り直して Solaris 10 5/08 OS にアップグレードすることができなくなる可能性があります。

Solaris パーティションを含むディスクにサービスパーティションを含めたい場合、次のいずれかの方法を選択してください。

▼ ネットワークインストールイメージまたは **Solaris Operating System DVD** からのソフトウェアのインストール

ネットワークインストールイメージからソフトウェアをインストールする場合、またはネットワーク経由で Solaris Operating System DVD からソフトウェアをインストールする場合は、次の手順を実行します。

- 1 ディスクの内容を削除します。
- 2 インストール前に、システムの診断 CD を使ってサービスパーティションを作成します。
サービスパーティションの作成方法については、ハードウェアのマニュアルを参照してください。
- 3 ネットワークからシステムをブートします。
「fdisk パーティションのカスタマイズ」画面が表示されます。
- 4 「デフォルト」をクリックし、デフォルトのブートディスクパーティションレイアウトを読み込みます。
インストールプログラムにより、サービスパーティションが保持され、Solaris パーティションが作成されます。

▼ **Solaris SOFTWARE - 1 CD** またはネットワークインストールイメージからのインストール

Solaris インストールプログラムを使用して、Solaris SOFTWARE - 1 CD またはブートサーバー上のネットワークインストールイメージからインストールを実行するには、次の手順を実行します。

- 1 ディスクの内容を削除します。
- 2 インストール前に、システムの診断 CD を使ってサービスパーティションを作成します。
サービスパーティションの作成方法については、ハードウェアのマニュアルを参照してください。

- 3 インストールプログラムにより、**Solaris** パーティションの作成方法を選択するよう求められます。
- 4 システムをブートします。
- 5 「残りのディスクを使用して Solaris パーティションを配置します」を選択します。
インストールプログラムにより、サービスパーティションが保持され、Solaris パーティションが作成されます。
- 6 インストールが完了します。

その他の SVR4 パッケージ要件 (リファレンス)

この付録は、パッケージ (特に Sun 以外のパッケージ) のインストールや削除を行うシステム管理者を対象としています。これらのパッケージ要件に準拠することによって、次のことが可能になります。

- 現在稼働中のシステムの変更を避け、Solaris Live Upgrade を使ってアップグレードを実行し、非大域ゾーンとディスククライアントを作成および管理します
- カスタム JumpStart などのインストールプログラムの使用時に、インストールを自動化するため、対話的なパッケージ操作を無効にします

この章の内容は次のとおりです。

- 193 ページの「現在の OS の変更を回避」
- 197 ページの「インストール中およびアップグレード中のユーザー操作の回避」
- 199 ページの「ゾーンのパッケージパラメータの設定」

現在の OS の変更を回避

この節の要件に従って、現在稼働中の OS の変更を回避します。

絶対パスの使用

オペレーティングシステムを正常にインストールするには、パッケージが代替ルート (/) ファイルシステム (Solaris Live Upgrade の非アクティブブート環境など) を認識し、正常に対話する必要があります。

パッケージの `pkgmap` ファイル (パッケージマップ) には、絶対パスを指定できます。これらのパスが存在する場合、そのファイルは、`pkgadd` コマンドの `-R` オプションとの相対パスに書き込まれます。絶対パスと相対 (再配置可能) パスの両方を含むパッケージは、代替ルート (/) ファイルシステムにもインストールできます。絶対

ファイルであれ再配置可能ファイルであれ、その前には `$PKG_INSTALL_ROOT` が付加されるため、`pkgadd` によるインストールでは、すべてのパスが正しく解釈されます。

pkgadd -R コマンドの使用

`pkgadd -R` オプションを使ってパッケージをインストールしたり、`pkgrm -R` オプションを使ってパッケージを削除する場合には、パッケージが、動作中のシステムを変更してはいけません。この機能は、カスタム JumpStart、Solaris Live Upgrade、非大域ゾーン、およびディスクレスクライアントによって使用されます。

さらに、`pkgadd -R` オプションを使ってパッケージをインストールしたり、`pkgrm -R` オプションを使ってパッケージを削除する場合には、パッケージに同梱のスクリプトで、動作中のシステムを変更してはいけません。インストールスクリプトを作成する場合には、参照するディレクトリやファイルの前に `$PKG_INSTALL_ROOT` 変数を付加する必要があります。パッケージでは、書き込むすべてのディレクトリやファイルの前に `$PKG_INSTALL_ROOT` を付加する必要があります。さらに、パッケージでは、`$PKG_INSTALL_ROOT` 接頭辞を付加せずにディレクトリを削除すべきではありません。

表 B-1 に、スクリプト構文の例を示します。

表 B-1 インストールスクリプト構文の例

スクリプトタイプ	正しい構文	正しくない構文
Bourne シェル「if」ステートメントの一部	<pre>if [-f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf] ; then</pre>	<pre>if [-f /etc/myproduct.conf] ; \ then</pre>
ファイルの削除	<pre>/bin/rm -f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</pre>	<pre>/bin/rm -f /etc/myproduct.conf</pre>
ファイルの変更	<pre>echo "test=no" > \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</pre>	<pre>echo "test=no" > \ /etc/myproduct.conf</pre>

\$PKG_INSTALL_ROOT と \$BASEDIR の相違点の概要

`$PKG_INSTALL_ROOT` は、パッケージを追加しようとするマシンのルート (/) ファイルシステムの場合です。この値は、`pkgadd` コマンドの `-R` 引数の後にセットされます。たとえば、次のコマンドを実行すると、パッケージのインストール時に `$PKG_INSTALL_ROOT` の値は `/a` になります。

```
# pkgadd -R /a SUNWvxvm
```

\$BASEDIR は、再配置可能なパッケージオブジェクトがインストールされる「再配置可能」なベースディレクトリを指しています。ここにインストールされるのは、再配置可能オブジェクトだけです。再配置可能でないオブジェクト (pkgmap ファイルに「絶対」パスが指定されているオブジェクト) は、必ず、非アクティブブート環境からの相対位置にインストールされます。\$BASEDIR からの相対位置ではありません。再配置可能なオブジェクトがないパッケージは、絶対パッケージ (再配置不可) と呼ばれます。その場合、\$BASEDIR は未定義であるため、これをパッケージに添付されているスクリプトで使用することはできません。

たとえば、パッケージの pkgmap ファイルに次のエントリがあるとします。

```
1 f none sbin/ls 0555 root sys 3541 12322 1002918510
1 f none /sbin/ls2 0555 root sys 3541 12322 2342423332
```

さらに、pkginfo ファイルには、\$BASEDIR が次のように指定されているとします。

```
BASEDIR=/opt
```

このパッケージを次のコマンドでインストールすると、ls は、/a/opt/sbin/ls としてインストールされますが、ls2 は、/a/sbin/ls2 としてインストールされます。

```
# pkgadd -R /a SUNWttest
```

スクリプト作成のガイドライン

パッケージプロシージャースクリプトは、現在稼働中の OS から独立し、OS を変更しないようにする必要があります。スクリプトには、パッケージのインストールや削除の実行中に行うアクションを定義します。事前に決められたプロシージャー名で作成できるスクリプトが4つあります。preinstall、postinstall、preremove、および postremove です。

表 B-2 スクリプトの作成のガイドライン

ガイドライン	Solaris Live Upgrade への影響	非大域ゾーンへの影響
スクリプトは Bourne シェル (/bin/sh) で作成する必要があります。pkgadd コマンドは、スクリプトの実行時にインタプリタとして Bourne シェルを使用します。	○	○
スクリプトはプロセスの開始や停止を行ったり、ps や truss などのコマンドの出力に依存したりしてはいけません。ps や truss はオペレーティングシステムに依存し、稼働中のシステムに関する情報を報告します。	○	○
スクリプトでは、expr、cp、ls などの標準的な UNIX コマンドや、シェルスクリプトの作成を容易にするそのほかのコマンドを自由に使用できます。	○	○

表 B-2 スクリプトの作成のガイドライン (続き)

ガイドライン	Solaris Live Upgrade への影響	非大域ゾーンへの影響
<p>パッケージはサポートされているすべてのリリースで動作する必要があるため、スクリプトで呼び出すコマンドはこれらすべてのリリースで利用可能なものでなければなりません。したがって、Solaris 8 リリースのあとで追加または削除されたコマンドは使用できません。</p>	○	
<p>特定のコマンドまたはオプションが Solaris 8、9、または 10 リリースでサポートされているかどうかを調べるには、http://docs.sun.com で該当するバージョンの Solaris Reference Manual Collection を参照してください。</p>		

ディスクレスクライアントの互換性維持

パッケージでは、パッケージ自体が提供しているコマンドを実行してはいけません。これは、ディスクレスクライアントの互換性を維持するためであると同時に、まだインストールされていない共有ライブラリを必要とするコマンドの実行を避けるためです。

パッケージの検証

すべてのパッケージは `pkgchk` の検証にパスしなければなりません。パッケージを作成したらインストールする前に、次のコマンドでパッケージをチェックする必要があります。

```
# pkgchk -d dir_name pkg_name
```

`dir_name` パッケージがあるディレクトリの名前を指定します。

`pkg_name` パッケージの名前を指定します。

例 B-1 パッケージのテスト

パッケージを作成したら、`pkgadd` コマンドに `-R dir_name` オプションを指定して、これを代替ルート (*/*) ファイルシステムにインストールしてテストする必要があります。パッケージをインストールしたら、次のように、パッケージが正しいかどうかを `pkgchk` コマンドでチェックします。

```
# pkgadd -d . -R /a SUNWvxxvm
```

```
# pkgchk -R /a SUNWvxxvm
```

エラーが表示されないことを確認します。

例 B-2 /export/SUNWvxxm 上のパッケージのテスト

パッケージが /export/SUNWvxxm にある場合は、コマンドを次のように指定します。

```
# pkgchk -d /export SUNWvxxm
```

エラーが表示されないことを確認します。

ファイルの作成、変更、および削除時に、その他のコマンドでパッケージをチェックすることもできます。次にコマンド例を示します。

- たとえば、パッケージが正しく動作するかどうかを `dircmp` や `fssnap` コマンドを使って検証できます。
- さらに、`ps` コマンドでは、パッケージによりデーモンの開始や停止が行われていないことを確認することによってデーモンに対する要件準拠を確認できます。
- `truss`、`pkgadd -v`、および `pkgrm` コマンドで、パッケージインストールの実行要件に準拠しているかどうかを確認できます。ただし、これが常に機能するとはかぎりません。次の例では、`truss` コマンドは、読み取り専用ディレクトリおよび `$TMPDIR` 以外のディレクトリへのアクセス情報をすべて除外し、指定された非アクティブブート環境以外のディレクトリへの読み取り専用でないアクセス情報のみを表示します。

```
# TMPDIR=/a; export TMPDIR
# truss -t open /usr/sbin/pkgadd -R ${TMPDIR} SUNWvxxm \
2>&1 > /dev/null | grep -v O_RDONLY | grep -v \
'open("'"${TMPDIR}
```

インストール中およびアップグレード中のユーザー操作の回避

次の標準 Solaris ユーティリティを使用する際、ユーザーの情報入力なしに、パッケージの追加や削除が行われる必要があります。

- カスタム JumpStart プログラム
- Solaris Live Upgrade
- Solaris インストールプログラム プログラム
- Solaris ゾーン

パッケージをテストして、ユーザー操作なしでインストールされるようにするには、`pkgadd -a` コマンドで新しい管理ファイルを設定します。`-a` オプションは、デフォルトの管理ファイルの代わりにユーザー定義の管理ファイルを使用することを意味します。デフォルトのファイルを使用すると、情報の入力が必要になることが

あります。管理ファイルを作成すれば、`pkgadd` でこのようなチェックを省略し、ユーザーの確認なしでパッケージをインストールすることができます。詳細については、`admin(4)` または `pkgadd(1M)` のマニュアルページを参照してください。

次に、`pkgadd` コマンドによる管理ファイルの使用例を示します。

- 管理ファイルを指定しないと、`pkgadd` は `/var/sadm/install/admin/default` を使用します。このファイルを使用すると、ユーザーの入力が必要になることがあります。

```
# pkgadd
```

- コマンド行に相対的な管理ファイルを指定すると、`pkgadd` は `/var/sadm/install/admin` からこのファイル名を探して使用します。この例では相対的な管理ファイルの名前が `nocheck` であるため、`pkgadd` は `/var/sadm/install/admin/nocheck` を探します。

```
# pkgadd -a nocheck
```

- 絶対ファイル名を指定した場合、`pkgadd` はそれを使用します。この例では、`pkgadd` は `/tmp` 内で `nocheck` 管理ファイルを検索します。

```
# pkgadd -a /tmp/nocheck
```

例 B-3 インストール管理ファイル

次に示すのは、`pkgadd` ユーティリティーでユーザーの入力をほとんど必要としないインストール管理ファイルの例です。パッケージがシステムで利用可能な容量を超えた容量を必要としない限り、`pkgadd` ユーティリティーはこのファイルを使用して、ユーザーに情報の入力を求めることなくインストールを実行します。

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
space=ask
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
```

ゾーンのパッケージパラメータの設定

パッケージが保持するパラメータにより、非大域ゾーンのインストールされたシステムで内容を配布および可視にする方法が制御されます。SUNW_PKG_ALLZONES、SUNW_PKG_HOLLOW、およびSUNW_PKG_THISZONE パッケージパラメータは、ゾーンがインストールされているパッケージの特性を定義します。非大域ゾーンのインストールされたシステムでパッケージを管理できるようにするには、これらのパラメータを設定する必要があります。

次の表に、パッケージパラメータの設定に使用できる4つの有効な組み合わせを示します。次の表に示されていない設定の組み合わせは無効であり、そのような設定を選択するとパッケージのインストールは失敗します。

注-3つのパッケージパラメータをすべて設定したことを確認してください。3つのパッケージパラメータをすべて空のままにしてもかまいません。ゾーンのパッケージパラメータが見つからない場合、パッケージツールではfalseの設定として解釈されますが、パラメータの設定を省略しないように強くお勧めします。3つのパッケージパラメータをすべて設定することにより、パッケージをインストールまたは削除するときのパッケージツールの動作を正確に指定します。

表 B-3 ゾーンのパッケージパラメータの有効な設定

SUNW_PKG_ALLZONES の設定	SUNW_PKG_HOLLOW の設定	SUNW_PKG_THISZONE の設定	パッケージの説明
false	false	false	<p>ゾーンのパッケージパラメータのいずれにも値を指定しないパッケージの場合、これがデフォルト設定として使用されます。</p> <p>この設定を持つパッケージは、大域ゾーンまたは非大域ゾーンにインストールできます。</p> <ul style="list-style-type: none"> 大域ゾーン内で pkgadd コマンドを実行すると、パッケージは大域ゾーンおよびすべての非大域ゾーンにインストールされます。 非大域ゾーン内で pkgadd コマンドを実行すると、パッケージはその非大域ゾーンだけにインストールされます。 <p>どちらの場合も、パッケージがインストールされたすべてのゾーンで、パッケージの内容全体が可視になります。</p>

表 B-3 ゾーンのパッケージパラメータの有効な設定 (続き)

SUNW_PKG_ALLZONES の設定	SUNW_PKG_HOLLOW の設定	SUNW_PKG_THISZONE の設定	パッケージの説明
false	false	true	<p>この設定を持つパッケージは、大域ゾーンまたは非大域ゾーンにインストールできます。インストール後に新しい非大域ゾーンを作成した場合、パッケージはこれらの新しい非大域ゾーンには伝達されません。</p> <ul style="list-style-type: none"> ■ 大域ゾーン内で <code>pkgadd</code> コマンドを実行すると、パッケージは大域ゾーンだけにインストールされます。 ■ 非大域ゾーン内で <code>pkgadd</code> コマンドを実行すると、パッケージはその非大域ゾーンだけにインストールされます。 <p>どちらの場合も、パッケージがインストールされたゾーンで、パッケージの内容全体が可視になります。</p>
true	false	false	<p>この設定を持つパッケージは、大域ゾーンだけにインストールできます。 <code>pkgadd</code> コマンドを実行すると、パッケージは大域ゾーンおよびすべての非大域ゾーンにインストールされます。すべてのゾーンで、パッケージの内容全体が可視になります。</p> <p>注-パッケージを非大域ゾーンにインストールしようとすると失敗します。</p>

表 B-3 ゾーンのパッケージパラメータの有効な設定 (続き)

SUNW_PKG_ALLZONES の設定	SUNW_PKG_HOLLOW の設定	SUNW_PKG_THISZONE の設定	パッケージの説明
true	true	false	<p>この設定を持つパッケージは、大域管理者が大域ゾーンだけにインストールできます。pkgadd コマンドを実行すると、パッケージの内容が大域ゾーンに完全にインストールされます。パッケージパラメータの値がこのように設定されている場合、パッケージの内容自体はどの非大域ゾーンにも提供されません。パッケージをインストール済みとして表示するために必要なパッケージインストール情報だけが、すべての非大域ゾーンにインストールされます。これにより、このパッケージに依存するほかのパッケージをインストールできるようになります。「hollow」パッケージの詳細については、『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 24 章「ゾーンがインストールされている Solaris システムでのパッケージとパッチについて (概要)」を参照してください。</p> <p>パッケージの依存関係を検査できるように、パッケージはすべてのゾーンでインストール済みとして表示されます。</p> <ul style="list-style-type: none"> ■ 大域ゾーンでは、パッケージの内容全体が可視になります。 ■ 完全ルート非大域ゾーンでは、パッケージの内容全体が不可視になります。 ■ 非大域ゾーンが大域ゾーンからファイルシステムを継承する場合、このファイルシステムにインストールされているパッケージは非大域ゾーンで可視になります。パッケージで提供されるほかのすべてのファイルは、非大域ゾーン内では不可視になります。 <p>たとえば、疎ルート非大域ゾーンは、特定のディレクトリを大域ゾーンと共有します。これらのディレクトリは読み取り専用です。疎ルート非大域ゾーンは、/platform などのファイルシステムを共有します。もう 1 つの例は、起動するハードウェアだけに関連するファイルがパッケージで提供されている場合です。</p> <p>注-パッケージを非大域ゾーンにインストールしようとすると失敗します。</p>

説明	参照先
パッケージとゾーンの詳細	『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 24 章「ゾーンがインストールされている Solaris システムでのパッケージとパッチについて (概要)」
疎ルートゾーンと完全ルートゾーンの概要	『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 16 章「Solaris ゾーンの紹介」
パッケージの特性およびパラメータの詳細	pkginfo(4)
パッケージパラメータ値の表示の詳細	pkgparam(1)

背景情報

次の参照先から、パッケージ要件の背景情報と特定のコマンド構文を確認できます。

パッケージ要件の詳細と用語の定義	『Application Packaging Developer's Guide』の第 6 章「Advanced Techniques for Creating Packages」
パッケージの追加と削除、ならびにインストール管理ファイルに関する基本情報	『Solaris のシステム管理 (基本編)』の第 16 章「ソフトウェアの管理 (概要)」
この付録で紹介した特定のコマンドに関する詳細情報は、これらのマニュアルページを参照してください	dircmp(1)、fssnap(1M)、ps(1)、または truss(1)、pkgadd(1M)、pkgchk(1M)、または pkgrm(1M)
Solaris Live Upgrade の概要	『Solaris 10 5/08 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の第 2 章「Solaris Live Upgrade (概要)」
カスタム JumpStart の概要	第 2 章カスタム JumpStart (概要)
Solaris ゾーンの概要	『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の第 16 章「Solaris ゾーンの紹介」

用語集

3DES	Triple-Data Encryption Standard (Triple DES) の略。168 ビットの鍵を提供する対称鍵暗号化方法。
AES	Advanced Encryption Standard の略。対称型 128 ビットブロックデータ暗号化技術。米国政府は、2000 年の 10 月に暗号化標準としてこのアルゴリズムの Rijndael 方式を採用しました。DES 暗号化に代わる米国政府の標準として、AES が採用されています。
bootlog-cgi プログラム	WAN ブートインストール時に、リモートクライアントのブートおよびインストールのコンソールメッセージを Web サーバーで収集し保存できるようにする CGI プログラム。
certstore ファイル	特定のクライアントシステムに関するデジタル証明書を格納しているファイル。SSL ネゴシエーションの際、クライアントは証明書ファイルをサーバーに提供するように要求されることがあります。サーバーはこのファイルを使ってクライアントの識別情報を確認します。
CGI	Common Gateway Interface の略。外部プログラムが HTTP サーバーと通信するためのインタフェース。CGI を使用するプログラムは、CGI プログラムまたは CGI スクリプトと呼ばれます。通常サーバーでは処理されないフォームや解析されない出力を、CGI プログラムが処理したり解析したりします。
DES	Data Encryption Standard の略。対称鍵暗号化方法の 1 つ。1975 年に開発され、ANSI により 1981 年に ANSI X.3.92 として標準化されました。DES では 56 ビットの鍵を使用します。
DHCP	Dynamic Host Configuration Protocol の略。アプリケーションレイヤープロトコルの 1 つ。このプロトコルを使えば、TCP/IP ネットワーク上の個々のコンピュータまたはクライアントは、一元的に管理された特定の DHCP サーバー (1 つまたは複数) から IP アドレスなどのネットワーク構成情報を取得できます。この機能は、大規模な IP ネットワークの保持、管理による間接費を削減します。
/etc/netboot ディレクトリ	WAN ブートインストールに必要なクライアント構成情報とセキュリティーデータが格納されている、WAN ブートサーバー上のディレクトリ。
/etc ディレクトリ	重要なシステム構成ファイルや保守コマンドが収められているディレクトリ。

- /export ファイルシステム** OS サーバー上のファイルシステムで、ネットワーク上のほかのシステムと共有されます。たとえば、/export ファイルシステムには、ディスクレスクライアント用のルート (/) ファイルシステムとスワップ空間、それにネットワーク上のユーザーのホームディレクトリを収めることができます。ディスクレスクライアントは、起動と実行の際に OS サーバー上の /export ファイルシステムに依存します。
- fdisk パーティション** x86 ベースのシステム上にある特定のオペレーティングシステム専用のディスクドライブの論理パーティション。x86 システム上に Solaris ソフトウェアをインストールするには、1 つ以上の Solaris fdisk パーティションを設定する必要があります。x86 ベースのシステムでは、1 台のディスクに最大 4 つの fdisk パーティションを作成できます。これらのパーティションは、個別のオペレーティングシステムをインストールして使用できます。各オペレーティングシステムは、独自の fdisk パーティション上に存在しなければなりません。個々のシステムの Solaris fdisk パーティションの数は、1 台のディスクにつき 1 つに限られます。
- GRUB** **x86 のみ:** GRUB (GNU GRand Unified Bootloader) は、簡単なメニューインタフェースを備えたオープンソースのブートローダーです。メニューには、システムにインストールされているオペレーティングシステムのリストが表示されます。GRUB を使用すると、Solaris OS、Linux、または Microsoft Windows などのさまざまなオペレーティングシステムを、簡単にブートすることができます。
- GRUB 編集メニュー** **x86 のみ:** GRUB メインメニューのサブメニューであるブートメニュー。このメニューには、GRUB コマンドが表示されます。これらのコマンドを編集して、ブート動作を変更できます。
- GRUB メインメニュー** **x86 のみ:** システムにインストールされているオペレーティングシステムがリストされたブートメニュー。このメニューから、BIOS または fdisk パーティションの設定を変更することなく、簡単にオペレーティングシステムをブートできます。
- HMAC** メッセージ認証を行うためのキー付きハッシュ方法。HMAC は秘密共有鍵と併用して、MD5、SHA-1 などの繰り返し暗号化のハッシュ関数で使われます。HMAC の暗号の強さは、基になるハッシュ関数のプロパティによって異なります。
- HTTP** Hypertext Transfer Protocol の略。リモートホストからハイパーテキストオブジェクトをフェッチするインターネットプロトコル。このプロトコルは TCP/IP にもとづいています。
- HTTPS** HTTP のセキュリティー保護されたバージョン。SSL (Secure Sockets Layer) を使って実装されます。
- IPv6** IPv6 は、現在のバージョン IPv4 (バージョン 4) から拡張されたインターネットプロトコル (IP) のバージョン (バージョン 6) です。定められた移行方法を使用して IPv6 を採用すると、現在の運用を中断する必要はありません。また、IPv6 には、新しいインターネット機能用のプラットフォームも用意されています。
- JumpStart インストール** インストール方法の 1 つ。出荷時にインストールされている JumpStart ソフトウェアを使用することによって、Solaris ソフトウェアをシステムに自動インストールできます。

JumpStart ディレクトリ	カスタム JumpStart インストールの実行に必要なファイルが含まれているディレクトリ。プロファイルフロッピーディスクを使用してインストールする場合は、フロッピーディスク上のルートディレクトリが JumpStart ディレクトリとなります。カスタム JumpStart インストール用にプロファイルサーバーを使用する場合、必要なカスタム JumpStart ファイルをすべて格納するサーバー上のディレクトリが JumpStart ディレクトリとなります。
Kerberos	強力な秘密鍵暗号方式を使用して、クライアントとサーバーが、セキュリティー保護されていないネットワーク接続で相互を認識できるようにするネットワーク認証プロトコル。
keystore ファイル	クライアントとサーバーとで共有される鍵を格納しているファイル。WAN ブートインストール時に、クライアントシステムは鍵を使って、サーバーから送信されるデータやファイルの完全性の確認と復号化を行います。
LAN	local area network の略。接続用のハードウェアとソフトウェアを介して通信できる、近接したコンピュータシステムの集まり。
LDAP	Lightweight Directory Access Protocol の略。LDAP ネームサービスクライアントとサーバー間の通信に使用される標準の拡張可能なディレクトリアクセスプロトコル。
MD5	Message Digest 5 の略。デジタル署名などのメッセージ認証に使用する繰り返し暗号化のハッシュ関数。1991 年に Rivest 氏によって開発されました。
menu.lst ファイル	x86 のみ: システムにインストールされているすべてのオペレーティングシステムがリストされたファイル。このファイルの内容が、GRUB メニューに表示されるオペレーティングシステムのリストを決定します。GRUB のメニューから、BIOS または fdisk パーティションの設定を変更することなく、簡単にオペレーティングシステムをブートできます。
NIS	SunOS 4.0 (以上) のネットワーク情報サービス。ネットワーク上のシステムとユーザーに関する重要な情報が収められている分散型ネットワークデータベース。NIS データベースは、マスターサーバーとすべてのスレーブサーバーに格納されています。
NIS+	SunOS 5.0 (以上) のネットワーク情報サービス。NIS+ は、SunOS 4.0 (以上) のネットワーク情報サービスである NIS に代わるものです。
/opt ファイルシステム	Sun 以外のソフトウェア製品や別製品のソフトウェア用のマウントポイントが収められているファイルシステム。
OS サーバー	ネットワーク上のシステムにサービスを提供するシステム。ディスクレスクライアントにサービスを提供するには、OS サーバーは、ディスクレスクライアントごとに、ルート (/) ファイルシステムとスワップ領域 (/export/root、/export/swap) 用のディスク容量が必要です。
RAID-0 ボリューム	ストライプ方式または連結方式のボリューム。これらはサブミラーとも呼ばれます。ストライプや連結は、ミラーを構築する基本構成ブロックです。
RAID-1 ボリューム	同じデータのコピーを複数保持しているボリューム。RAID-1 ボリュームは、「サブミラー」と呼ばれる 1 つまたは複数の RAID-0 ボリュームから構成されます。RAID-1 ボリュームは「ミラー」と呼ばれることもあります。

rules.ok ファイル	rules ファイルから生成されたファイル。カスタム JumpStart インストールソフトウェアは、rules.ok ファイルを使ってシステムとプロファイルを照合します。rules.ok ファイルは、check スクリプトを使用して作成してください。
rules ファイル	自動的にインストールするシステムの各グループまたは単一のシステムのルールを含んでいるテキストファイル。各ルールは1つ以上のシステム属性に基づいてシステムグループを識別します。rules ファイルは、各グループをプロファイル (Solaris ソフトウェアをどのようにしてグループ内の個々のシステムにインストールするかを定めたテキストファイル) にリンクします。rules ファイルは、カスタム JumpStart インストールで使用されます。「プロファイル」も参照してください。
Secure Sockets Layer	(SSL) クライアントとサーバーの間にセキュリティー保護された接続を確立するソフトウェアライブラリ。HTTP のセキュリティー保護されたバージョンである HTTPS を実装するために使用されます。
SHA1	Secure Hashing Algorithm の略。2 ⁶⁴ 未満の入力長を操作し、メッセージダイジェストを生成するアルゴリズム。
Solaris Live Upgrade	アクティブブート環境が稼働している間に複製ブート環境のアップグレードを行うことにより、稼働中の環境のダウンタイムをなくすことを可能にするアップグレード方法。
Solaris インストールプログラム	グラフィカルユーザーインターフェイス (GUI) とコマンド行インターフェイス (CLI) を備えたインストールプログラム。ウィザードパネルに、Solaris ソフトウェアやサードパーティソフトウェアをインストールする手順が示されます。
Solaris ゾーン	オペレーティングシステムサービスの仮想化に使用されるソフトウェア区分技術。アプリケーションを実行するための独立した安全な環境を提供します。非大域ゾーンを作成すると、アプリケーション実行環境が生成されます。このアプリケーション実行環境内のプロセスは、その他のゾーンから隔離されます。この隔離により、ゾーン内で実行中のプロセスが別のゾーンで実行中のプロセスから監視されたり操作されたりすることを防ぐことができます。「大域ゾーン」と「非大域ゾーン」も参照してください。
Solaris の DVD イメージまたは CD イメージ	システムにインストールされる Solaris ソフトウェア。Solaris DVD や CD から、または Solaris の DVD イメージや CD イメージをコピーしたインストールサーバーのハードディスク上から利用できます。
Solaris フラッシュ	「マスターシステム」と呼ばれるシステムからファイルのアーカイブを作成する Solaris インストール機能。このアーカイブを使ってほかのシステムのインストールを行うと、そのシステムの構成はマスターシステムと同じになります。「アーカイブ」も参照してください。
sysidcfg ファイル	システムを事前設定する特殊な一連のシステム構成キーワードを指定するファイル。
truststore ファイル	1つ以上のデジタル証明書を格納しているファイル。WAN ブートインストール時に、クライアントシステムは truststore ファイル内のデータを参照して、インストールを実行しようとしているサーバーの識別情報を確認します。
URL	Uniform Resource Locator の略。サーバーやクライアントがドキュメントを要求するために使用するアドレス方式。URL はロケーションとも呼ばれます。URL の形式は <i>protocol://machine:port/document</i> です。

たとえば、<http://www.example.com/index.html> は URL の一例です。

/usr ファイルシステム	スタンドアロンシステムまたはサーバー上のファイルシステム。標準 UNIX プログラムの多くが格納されています。ローカルコピーを保持する代わりに、大きな /usr ファイルシステムをサーバーと共有することにより、システム上で Solaris ソフトウェアをインストールおよび実行するために必要なディスク容量を最小限に抑えることができます。
/var ファイルシステム	システムの存続期間にわたって変更または増大が予想されるシステムファイルが格納されている (スタンドアロンシステム上の) ファイルシステムまたはディレクトリ。これらのファイルには、システムログ、vi ファイル、メールファイル、UUCP ファイルなどがあります。
WAN	Wide Area Network の略。複数のローカルエリアネットワーク (LAN) または地理的に異なる場所にあるシステムを、電話、光ファイバ、衛星などの回線を使って接続するネットワーク。
wanboot-cgi プログラム	WAN ブートインストールで使用されるデータとファイルの取得と転送を行う CGI プログラム。
wanboot.conf ファイル	WAN ブートインストールに必要な構成情報とセキュリティ設定値を指定するテキストファイル。
wanboot プログラム	WAN ブートインストールの実行に必要な、WAN ブートミニルート、クライアント構成ファイル、およびインストールファイルを読み込む、二次レベルのブートプログラム。WAN ブートインストールでは、wanboot バイナリが、ufsboot または inetboot 二次ブートプログラムと同様の処理を実行します。
WAN ブートインストール	HTTP または HTTPS を使って広域ネットワーク (WAN) を介してソフトウェアをブートしインストールできるインストール方式。WAN ブートインストールでは、暗号化された Solaris フラッシュアーカイブをパブリックネットワークを介して転送し、リモートクライアントに対してカスタム JumpStart インストールを実行できます。
WAN ブートサーバー	WAN ブートインストールで使用される構成ファイルとセキュリティファイルを提供する Web サーバー。
WAN ブートミニルート	WAN ブートインストールを実行するために変更されたミニルート。WAN ブートミニルートには、Solaris ミニルートにあるソフトウェアのサブセットが格納されます。「ミニルート」も参照してください。
アーカイブ	マスターシステムからコピーされたファイルの集合体。このファイルには、アーカイブの名前や作成した日付など、アーカイブの識別情報が含まれています。アーカイブをシステムにインストールすると、システムはマスターシステムとまったく同じ構成になります。 更新前のマスターイメージと更新されたマスターイメージの相違部分のみを含む Solaris フラッシュアーカイブを、差分アーカイブとして使用することも可能です。差分アーカイブには、クローンシステムで保持、変更、または削除するファイルが含まれます。差分更新により、指定されたファイルだけが更新されます。また、差分更新を使用可能なシステムは、更新前のマスターイメージとの整合性を保持するソフトウェアを含むシステムのみに限定されます。

アップグレード	ファイルを既存のファイルとマージし、可能な場合には変更を保存するインストール。 Solaris OS をアップグレードすると、新しいバージョンの Solaris OS とシステムのディスク上の既存のファイルが結合されます。アップグレードでは、既存の Solaris OS に対して行なった変更は最大限に保存されます。
アップグレードオプション	Solaris インストールプログラムが提供するオプション。アップグレード時には、新しいバージョンの Solaris とディスク上の既存のファイルが結合されます。前回 Solaris をインストールしてから加えられたローカルの変更内容は、できる限り残されます。
暗号化	認められたユーザー以外は情報を使用できないように、情報を判読不可能にして保護する処理。暗号化は鍵と呼ばれるコードに基づいて行われ、この鍵は情報の復号化に使用されます。「 復号化 」も参照してください。
一次ブートアーカイブ	システムで Solaris OS をブートするために使用されるブートアーカイブ。このブートアーカイブは、一次ブートアーカイブと呼ばれることもあります。「ブートアーカイブ」を参照してください。
インストールサーバー	インストール用に、Solaris DVD または CD のイメージをネットワーク上のほかのシステムに提供するサーバー(「メディアサーバー」とも呼ばれる)。Solaris の DVD イメージまたは CD イメージをサーバーのハードディスクにコピーすることによってインストールサーバーを作成できます。
エンドユーザーシステムサポート	コアシステムサポートソフトウェアグループのほかに、エンドユーザーに推奨するソフトウェアが収められているソフトウェアグループ。これには共通デスクトップ環境 (CDE) や DeskSet ソフトウェアが含まれます。
開始スクリプト	ユーザーが定義する Bourne シェルスクリプト。rules ファイル内で指定され、Solaris ソフトウェアがシステムにインストールされる前に作業を実行します。このスクリプトは、カスタム JumpStart インストールでのみ使用できます。
開発者システムサポート	エンドユーザーシステムサポートソフトウェアグループのほかに、ソフトウェア開発用ライブラリ、インクルードファイル、マニュアルページ、およびプログラミングツールが収められているソフトウェアグループ。
鍵	データの暗号化および復号化に使用されるコード。「 暗号化 」も参照してください。
カスタム JumpStart	ユーザーが定義するプロファイルに基づいて、Solaris ソフトウェアをシステムに自動的にインストールする方法。ユーザーやシステムの種類ごとに、カスタマイズされたプロファイルを作成できます。カスタム JumpStart インストールは、ユーザーが作成する JumpStart インストールです。
カスタムプロープファイル	rules ファイルと同じ JumpStart ディレクトリに存在しなければならないファイルで、次の2つのタイプの関数を含む Bourne シェルスクリプト。含む2つのタイプは、プロープと比較です。プロープ関数は、必要な情報を収集し、実際の作業を行なって、定義に対応した <code>SI_</code> 環境変数を設定します。プロープ関数はプロープキーワードになります。比較関数は、対応するプロープ関数を呼び出してプロープ関数の出力を比較し、キーワードが一致する場合は0、キーワードが一致しない場合は1を返します。比較関数はルールキーワードになります。「rules ファイル」も参照してください。

共有可能ファイルシステム	/export/home や /swap のようなユーザー定義のファイルシステム。Solaris Live Upgrade の使用時に、アクティブブート環境と非アクティブブート環境によって共有されます。共有可能ファイルシステムは、アクティブブート環境と非アクティブブート環境の両方の <code>vfstab</code> ファイル内に同じマウントポイントを持ちます。このため、アクティブブート環境内の共有ファイルを更新すると、非アクティブブート環境のデータも更新されます。共有可能ファイルシステムはデフォルトで共有されますが、ユーザーが宛先スライスを指定することもできます。この場合、そのファイルシステムがコピーされます。
クライアント	通信用のクライアントサーバーモデルでは、計算機能や大容量のメモリーといったサーバーの資源にリモートアクセスするプロセスがクライアントに相当します。
クラスタ	パッケージ(ソフトウェアモジュール)を論理的に集めたもの。Solaris ソフトウェアは複数の「ソフトウェアグループ」に分割され、それぞれがクラスタと「パッケージ」から構成されています。
クリティカルファイルシステム	Solaris OS が必要とするファイルシステム。Solaris Live Upgrade を使用するとき、これらのファイルシステムは、アクティブブート環境と非アクティブブート環境それぞれの <code>vfstab</code> ファイルでは独立したマウントポイントになります。root (/)、/usr、/var、/opt などがクリティカルファイルシステムの例です。これらのファイルシステムは、必ずソースブート環境から非アクティブブート環境にコピーされます。
クローンシステム	Solaris フラッシュアーカイブを使用してインストールするシステム。クローンシステムは、マスターシステムと同一のインストール構成になります。
限定ネットワークシステムサポートソフトウェアグループ	Solaris システム(ネットワークサービスのサポートは一部のみ)をブートし、実行するために必要な最小限のコードを含むソフトウェアグループ。限定ネットワークシステムサポートソフトウェアグループは、テキストベースのマルチユーザーコンソールとシステム管理ユーティリティを提供します。このソフトウェアグループは、システムによるネットワークインタフェースの認識を可能にしますが、ネットワークサービスをアクティブにはしません。
コアシステムサポートソフトウェアグループ	システムで Solaris OS を起動して実行するのに必要な最小限のソフトウェアが収められているソフトウェアグループ。コアには共通デスクトップ環境(CDE)を実行するために必要ないくつかのネットワーク用ソフトウェアとドライバが含まれます。CDE ソフトウェアは、コアには含まれません。
公開鍵	公開鍵暗号方式で使用される暗号化鍵。
公開鍵暗号方式	2つの鍵を使用する暗号方式。その1つは、全員が知っている公開鍵、もう1つは、メッセージの受取人だけが知っている非公開鍵です。
更新	システムにインストールを実行して同じタイプのソフトウェアを変更することまたはそのインストール自体。アップグレードとは異なり、更新によりシステムがダウングレードされる場合があります。初期インストールとは異なり、更新を実行するには同じタイプのソフトウェアがあらかじめインストールされていなければなりません。
コマンド行	コマンドで始まる文字列。多くの場合、コマンドの後には引数(オプション、ファイル名、式などの文字列)が続き、行末(EOL)文字で終わります。
サーバー	資源を管理し、クライアントにサービスを提供するネットワークデバイス。

サブネット	経路指定を簡単にするため、1つの論理ネットワークを小さな物理ネットワークに分割する方式。
サブネットマスク	サブネットアドレス指定のため、インターネットアドレスからビットを選択するために使用されるビットマスク。マスクは32ビットです。インターネットアドレスのネットワーク部分と、ローカル部分の1個以上のビットを選択します。
サブミラー	「RAID-0 ボリューム」を参照してください。
差分アーカイブ	更新前のマスターイメージと更新されたマスターイメージの相違部分のみを含む Solaris フラッシュアーカイブ。差分アーカイブには、クローンシステムで保持、変更、または削除するファイルが含まれます。差分更新は、指定されたファイルのみを変更します。また、変更されていないマスターイメージと一致するソフトウェアを含むシステムのみに適用されます。
時間帯	グリニッジ標準時間を基準に地球の表面を24の地域に経度分割したものの。
システム構成ファイル	(system.conf) WAN ブートインストールで使用する sysidcfg ファイルおよびカスタム JumpStart ファイルの場所を指定するテキストファイル。
終了スクリプト	ユーザーが定義する Bourne シェルスクリプト。rules ファイル内で指定され、Solaris ソフトウェアがシステムにインストールされてから、システムがリブートされるまでの間に作業を実行します。このスクリプトは、カスタム JumpStart インストールで使用します。
状態データベース	Solaris ボリュームマネージャー構成の状態に関する情報を保存するデータベース。状態データベースは、複製された複数のデータベースコピーの集まりです。各コピーは「状態データベースの複製」と呼ばれます。状態データベースは、既知の状態データベースの複製の格納場所と状態をすべて記録しています。
状態データベースの複製	状態データベースのコピー。複製により、データベース内のデータの有効性が保証されます。
初期インストール	現在実行中のソフトウェアを上書きするか、空のディスクを初期化するインストール。 Solaris OS の初期インストール時には、システムのディスク (複数可) が新しいバージョンの Solaris OS で上書きされます。システム上で Solaris OS が稼働していない場合は、初期インストールを行う必要があります。システム上でアップグレード可能なバージョンの Solaris OS が稼働している場合、初期インストールを行うとディスクが上書きされ、この更新可能なバージョンの OS やローカルでの変更内容が失われます。
ジョブ	コンピュータシステムで実行されるユーザー定義の処理。
スーパーユーザー	システム上ですべての管理タスクを実行する特権を持つ、特殊なユーザー。スーパーユーザーは全ファイルの読み取り権とアクセス権、全プログラムの実行権を持ち、任意のプロセスに終了シグナルを送ることができます。
スタンドアロン	ほかのマシンからのサポートを一切必要としないコンピュータ。
スライス	ソフトウェアごとに分割される、ディスク領域の区分。

スワップ空間	メモリーに再ロードできる状態になるまで、メモリー領域の内容を一時的に保持するスライスまたはファイル。/swap または swap ファイルシステムとも呼ばれます。
全体ディストリビューション	Solaris リリース全体が含まれているソフトウェアグループ。
全体ディストリビューションと OEM サポート	Solaris リリース全体と、OEM のための追加ハードウェアサポートを含むソフトウェアグループ。Solaris を SPARC 搭載サーバーシステムにインストールする場合は、このソフトウェアグループを推奨します。
ゾーン	「非大域ゾーン」を参照してください。
ソフトウェアグループ	Solaris ソフトウェアの論理グループ(クラスとパッケージ)。Solaris のインストール時には、次のいずれかのソフトウェアグループをインストールできます。コアシステムサポート、エンドユーザーシステムサポート、開発者システムサポート、または全体ディストリビューションです。また、SPARC システムのみ、全体ディストリビューションと OEM サポートもインストールできます。
大域ゾーン	Solaris ゾーンでは、大域ゾーンは、システムのデフォルトゾーンであり、システム全体の管理制御用ゾーンでもあります。非大域ゾーンの構成、インストール、管理、およびアンインストールは、大域ゾーンからのみ行うことができます。物理デバイス、ルーティング、動的再構成 (DR) といったシステムインフラストラクチャーの管理は、大域ゾーンでのみ行うことができます。大域ゾーン内で実行されるプロセスは、適切な権限が付与されていれば、ほかのゾーンに関連付けられたオブジェクトにアクセスできます。「Solaris ゾーン」と「非大域ゾーン」も参照してください。
チェックサム	一連のデータ項目を合計した結果。一連のデータ項目を検査するために使用されます。データ項目は、数値でも、文字列でもよく、文字列の場合はチェックサム計算時に数値として扱われます。チェックサムの値から、2つのデバイス間の情報交換が正しく行われたかを確認できます。
ディスク (disc)	磁気ディスク (disk) ではなく光学式ディスク (disc) で、CD (コンパクトディスク) 市場で使用される共通の綴り。たとえば、CD-ROM や DVD-ROM は光学式ディスク (disc) です。
ディスク (disk)	1枚以上の磁性体の円盤から成るメディアであり、ファイルなどのデータを格納する同心トラックとセクターで構成されます。「ディスク (disc)」も参照してください。
ディスク構成ファイル	ディスクの構造(たとえば、バイト/セクター、フラグ、スライス)を表現するファイル。ディスク構成ファイルにより、単一システムから pfinstall コマンドを使用して、サイズの異なるディスク上のプロファイルをテストできます。
ディスクレスクライアント	ディスク記憶装置を持たないためサーバーに依存するクライアント。
デジタル証明書	移転や偽造の不可能なデジタルファイルで、通信する両者によって信頼済みの第三者機関から発行されたもの。

電源管理システム	<p>30分間アイドル状態が続くとシステムの状態を自動的に保存し、電源を切断するソフトウェア。米国環境保護庁の省電力 (Energy Star) ガイドライン第2版に準拠したシステムに Solaris ソフトウェアをインストールすると、デフォルトで電源管理システムソフトウェアがインストールされます。デフォルトで電源管理がインストールされているシステムの例として、sun4u SPARC ベースのシステムがあります。リポート後、電源管理ソフトウェアを有効にするかどうかを確認するメッセージが表示されます。</p> <p>Energy Star ガイドラインでは、システムまたはモニターを使用していない場合は、自動的に「休眠状態」(30ワット以下の消費)に入ることを要求されます。</p>
ドキュメントルート ディレクトリ	<p>Webサーバーにアクセスするユーザーに公開されるファイル、画像、およびデータが格納されている、Webサーバーマシン上の階層のルート。</p>
ドメイン	<p>インターネットのネーミング階層の一部。ドメインは管理ファイルを共有する、ローカルネットワーク上のシステムグループを表します。</p>
ドメイン名	<p>ローカルネットワーク上のシステムグループに割り当てられた名前であり、管理ファイルを共有します。ネットワーク情報サービス (NIS) のデータベースが正常に動作するためにはドメイン名が必要です。ドメイン名は、ピリオドで区切られた一連の構成要素名から構成されます (たとえば、tundra.mpk.ca.us)。ピリオドで区切られた各構成要素名は右側に行くにしたがって、全体的な (通常、遠方の) 管理権限領域を表します。</p>
認証局 (CA)	<p>CA は、Certificate Authority の略。デジタル署名および公開鍵と非公開鍵のペアの作成に使用するデジタル証明書を発行する、公証された第三者機関または企業。CA は、一意の証明書を付与された個人が当該の人物であることを保証します。</p>
ネームサーバー	<p>ネットワーク上のシステムに対してネームサービスを提供するサーバー。</p>
ネームサービス	<p>ネットワーク上の全システムに関する重要なシステム情報が収められている分散型ネットワークデータベース。ネットワーク上のシステムは、これを利用して相互通信を行います。ネームサービスを使用することによって、ネットワーク全域にわたるシステム情報を保守、管理、または取得できます。ネームサービスを使用しないと、各システムはローカルの /etc ファイルにシステム情報のコピーを保持しなければなりません。Sun は次のネームサービスをサポートしています。LDAP、NIS、および NIS+ です。</p>
ネットワークイン ストール	<p>CD-ROM または DVD-ROM ドライブがあるシステムから CD-ROM または DVD-ROM ドライブがないシステムにネットワークを介してソフトウェアをインストールする方法。ネットワークインストールを行うには、「ネームサーバー」と「インストールサーバー」が必要です。</p>
ネットワークに接続 されていないシステム	<p>ネットワークに接続されていない、またはほかのシステムに依存しないシステム。</p>
ネットワークに接続 されているシステム	<p>ハードウェアやソフトウェアを介して接続されているシステムのグループ (ホスト)。通信や情報の共有が可能です。ローカルエリアネットワーク (LAN) とも呼ばれます。システムをネットワークに接続するには、通常、1台以上のサーバーが必要です。</p>
ハードリンク	<p>ディスク上のファイルを参照するディレクトリエントリ。複数のハードリンクから同じ物理ファイルを参照することができます。</p>

派生プロファイル	カスタム JumpStart インストール時に、開始スクリプトによって動的に作成されるプロファイル。
パッケージ	モジュール形式でのインストールを可能にするソフトウェアの集まり。Solaris ソフトウェアは複数の「ソフトウェアグループ」に分割され、それぞれがクラスターと「パッケージ」から構成されています。
ハッシュ	入力よりもかなり短い数値を生成する処理によって得られる数値。同じ入力に対しては、常に同じ値が出力されます。ハッシュ関数は、テーブル検索アルゴリズム、エラー検出、改ざん検出などに使用できます。改ざん検出に使用する場合は、同じ結果を生成する別の入力を見つけにくいようなハッシュ関数を選択します。1方向のハッシュ関数の一例としては、MD5 および SHA-1 があります。たとえば、メッセージダイジェストはディスクファイルなどの可変長入力を受け取り、小さい値に変換します。
ハッシュ化	文字列を変換して、この元の文字列を表す値(キー)を得る処理。
パッチアナライザ	手動で、または Solaris インストールプログラムの一部として実行可能なスクリプト。パッチアナライザは、システムを解析し、Solaris Update へのアップグレードを行うことで削除されるパッチがどれであるかを判断します。
パネル	ウィンドウ、ダイアログボックス、アプレットなどの内容をまとめるコンテナ。パネルでは、ユーザーの入力をまとめて受け取り確認することができます。ウィザードでパネルを使用することで、正しい順序で操作を行い、目的の作業を完了することができます。
非公開鍵	公開鍵暗号方式で使用される復号化鍵。
非大域ゾーン	Solaris オペレーティングシステムの単一のインスタンス内に作成された仮想オペレーティングシステム環境。非大域ゾーンでは、1つ以上のアプリケーションをシステムのその他の部分と切り離して実行できます。非大域ゾーンを単純に「ゾーン」と呼ぶこともあります。「Solaris ゾーン」と「大域ゾーン」も参照してください。
ファイルサーバー	ネットワーク上のシステムに対して、ソフトウェアやファイルの記憶領域を提供するサーバー。
ファイルシステム	SunOS™ オペレーティングシステムにおいて、ユーザーがアクセスできるファイルおよびディレクトリから成るツリー構造のネットワークのこと。
ファンクションキー	F1、F2、F3 などの名前が付いた 10 個以上のキーボードキー。これらのキーにはそれぞれ特定の機能が割り当てられています。
ブート	メモリーにシステムソフトウェアを読み込んで起動すること。

ブートアーカイブ	<p>x86のみ:ブートアーカイブは、Solaris OSのブートに使用されるクリティカルなファイルの集まりです。これらのファイルは、ルート(/)ファイルシステムがマウントされる前、システムの起動中に必要です。システムは、2つのブートアーカイブを維持管理しています。</p> <ul style="list-style-type: none">■ システムでSolaris OSをブートするために使用されるブートアーカイブ。このブートアーカイブは、一次ブートアーカイブと呼ばれることもあります。■ 一次ブートアーカイブが損傷を受けたとき、回復のために使用されるブートアーカイブ。このブートアーカイブは、ルート(/)ファイルシステムをマウントすることなくシステムを起動します。GRUBメニューでは、このブートアーカイブはフェイルセーフと呼ばれます。アーカイブの重要な目的は一次ブートアーカイブを再生成することであり、通常、一次ブートアーカイブがシステムのブートに使用されます。
ブート環境	<p>Solaris OSを操作する上で重要な必須ファイルシステム(ディスクスライスおよびマウントポイント)の集まり。ディスクスライスは、同じ1つのディスク上に存在することも、分散された複数のディスク上に存在することもあります。</p> <p>アクティブなブート環境とは、現在ブートしている環境を指します。単一のアクティブなブート環境からだけブートできます。アクティブでないブート環境とは、現在ブートしていないが、次のリブート時にアクティブ化できる状態にある環境のことを指します。</p>
ブートサーバー	<p>同じネットワークのサブネット上のクライアントシステムに、起動に必要なプログラムと情報を提供するサーバーシステム。インストールサーバーの存在するサブネットが、Solarisソフトウェアをインストールする必要があるシステムと異なる場合、ネットワークを介してインストールするにはブートサーバーが必要です。</p>
ブートローダー	<p>x86のみ:ブートローダーは、システムの電源を入れた後に最初に実行されるソフトウェアプログラムです。このプログラムがブートプロセスを開始します。</p>
フェイルセーフブートアーカイブ	<p>x86のみ:一次ブートアーカイブが損傷を受けたとき、回復のために使用されるブートアーカイブ。このブートアーカイブは、ルート(/)ファイルシステムをマウントすることなくシステムを起動します。このブートアーカイブは、GRUBメニューではフェイルセーフと呼ばれます。アーカイブの重要な目的は一次ブートアーカイブを再生成することであり、通常、一次ブートアーカイブがシステムのブートに使用されます。「ブートアーカイブ」を参照してください。</p>
フォーマット	<p>データを一定の構造にしたり、データを保存できるようにディスクをセクターに分割したりすること。</p>
フォールバック	<p>以前に動作していた環境に戻すこと。ブート環境のアクティブ化の処理中や、ブート対象として指定されたブート環境に問題または望ましくない動作が発生する場合にはフォールバックを行います。</p>
復号化	<p>符号化されたデータをプレーンテキストに変換する処理。「暗号化」も参照してください。</p>
プラットフォームグループ	<p>特定のソフトウェア用にベンダーが定義するハードウェアプラットフォームのグループ。たとえばi86pcやsun4cなどです。</p>

プラットフォーム名	uname -i コマンドによって出力される情報。たとえば Ultra 60 のプラットフォーム名は、SUNW,Ultra-60 です。
プローブキーワード	インストールにカスタム JumpStart を使用する場合、システムに関する属性情報を抽出する構文要素。プローブキーワードでは、ルールに必要な一致条件の設定およびプロファイルの実行は必要ありません。「ルール」も参照してください。
プロファイル	カスタム JumpStart を使用する場合に、Solaris ソフトウェアのインストール方法を定義するテキストファイル。たとえば、プロファイルでインストールするソフトウェアグループを定義します。各ルールは、そのルールが一致したときにシステムがインストールされる方法を定義してあるプロファイルを指定します。通常は、ルールごとに異なるプロファイルを作成します。しかし、複数のルールで同じプロファイルを使用することも可能です。「rules ファイル」も参照してください。
プロファイルサーバー	すべての重要なカスタム JumpStart ファイルを JumpStart ディレクトリに持つサーバー。
プロファイルフロッピーディスク	すべての重要なカスタム JumpStart ファイルを、そのルートディレクトリ (JumpStart ディレクトリ) に持つフロッピーディスク。
ホスト名	システムがネットワーク上のほかのシステムから識別される名前。この名前は、特定のドメイン (通常、これは 1 つの組織内にあることを意味する) 内にある全システム間で固有でなければなりません。ホスト名は、文字、数字、マイナス符号 (-) を任意に組み合わせることで作成できますが、先頭と末尾にマイナス符号は使用できません。
ボリューム	システムで単一の論理デバイスとして扱われる、物理スライスやボリュームの集まり。ボリュームは、アプリケーションやファイルシステムにとって物理ディスクと同じように機能します。 一部のコマンド行ユーティリティでは、ボリュームはメタデバイスと呼ばれます。一般的な UNIX 用語では、ボリュームは、「擬似デバイス」または「仮想デバイス」とも呼ばれます。
マウント	マウント要求を行うマシンのディスクまたはネットワーク上のリモートディスクから、ディレクトリにアクセスするプロセス。ファイルシステムをマウントするには、ローカルシステム上のマウントポイントと、マウントするファイルシステム名 (たとえば /usr) が必要です。
マウント解除	マシンに接続されたディスクまたはネットワーク上のリモートディスク上のディレクトリへのアクセスを解除するプロセス。
マウントポイント	リモートマシン上に存在するファイルシステムのマウント先となる、ワークステーション上のディレクトリ。
マスターシステム	Solaris フラッシュアーカイブを作成するシステム。このシステム構成がアーカイブに保存されます。
ミニルート	Solaris インストールメディアに収録されている、起動可能な最小のルート (/) ファイルシステム。ミニルートには、システムをインストールおよびアップグレードするために必要な Solaris ソフトウェアが含まれます。x86 システムでは、ミニルートはシステムにコピーされ、フェイルセーフブートアーカイブとして使用されます。「フェイルセーフブートアーカイブ」を参照してください。

ミラー	「RAID-1 ボリューム」を参照してください。
メタデバイス	「ボリューム」を参照してください。
メディアサーバー	「インストールサーバー」を参照してください。
目録セクション	クローンシステムの検証に使用される Solaris フラッシュアーカイブのセクション。目録セクションには、クローンシステムに保持、追加または削除されるシステム上のファイルの一覧が含まれます。このセクションは、情報提供専用です。このセクションは、ファイルを内部形式でリストするため、スクリプトの記述には使用できません。
矢印キー	数値キーパッド上にある方向を示す4つのキーの1つ。
ユーティリティ	コンピュータを購入すると通常無料で提供される、標準プログラム。
ルート	複数の項目から成る階層構造の最上位。ルートは、ほかのすべての項目を子孫として持つ唯一の項目です。「ルートディレクトリ」または「ルート (/) ファイルシステム」を参照してください。
ルート (/) ファイルシステム	ほかのすべてのファイルシステムの元となる最上位ファイルシステム。ルート (/) ファイルシステムはほかのすべてのファイルシステムがマウントされる元となり、マウント解除されることはありません。ルート (/) ファイルシステムには、カーネル、デバイスドライバ、システムの起動(ブート)に使用されるプログラムなど、システムの稼働に不可欠なディレクトリやファイルが含まれています。
ルートディレクトリ	ほかのすべてのディレクトリの元となる最上位ディレクトリ。
ルール	1つ以上のシステム属性をプロファイルに割り当てる一連の値。ルールは、カスタム JumpStart インストールで使用されます。
連結	RAID-0 ボリューム。複数のスライスが連結された方式では、利用可能な最初のスライスがいっぱいになるまでそのスライスにデータが書き込まれます。そのスライスがいっぱいになると次のスライスに連続してデータが書き込まれます。ミラーに含まれている場合を除き、連結にはデータの冗長性はありません。「RAID-0 ボリューム」も参照してください。
ロケール	同一の言語、風俗、慣習、文化などを共有する地理上または政治上の地域圏(コミュニティ)。たとえば、米国英語のロケールは en_US、英国英語のロケールは en_UK です。
論理デバイス	システムで単一のデバイスとして扱われる、1つまたは複数のディスク上にある物理スライスの集まり。論理デバイスは、Solaris ボリュームマネージャーではボリュームと呼ばれます。ボリュームは、アプリケーションやファイルシステムにとって物理ディスクと同じように機能します。
ボリュームマネージャー	DVD-ROM、CD-ROM、およびフロッピーディスク上のデータへのアクセスを管理および実行するための手段を提供するプログラム。

索引

数字・記号

#

rules ファイル内の, 36
プロファイル内の, 40

(/) ファイルシステム

JumpStart で設定される値, 162

A

add_install_client コマンド, JumpStart ディレクトリ
のアクセス, 28

add_install_client コマンドの -s オプション, 113
&& (アンバサンド) ルールフィールド, 37

AND ルールフィールド, 37

any

プローブキーワード、説明と値, 167
ルールキーワード、説明と値, 115, 166

archive_location キーワード, 121-127

arch プローブキーワード, 166

arch ルールキーワード, 115, 166

auto_install_sample ディレクトリ

check スクリプト, 60, 85

JumpStart ディレクトリへファイルをコピー
, 27, 33, 35

set_root_pw 終了スクリプト, 70

B

backup_media キーワード, 127-129

begin.log ファイル, 64

begin ルールフィールド、説明, 37

boot: cannot open /kernel/unix メッセージ, 172

boot_device キーワード, 129

bootenv createbe キーワード, 130

bootparams ファイル

JumpStart ディレクトリのアクセスの有効
化, 30

更新, 178

C

-c オプション

add_install_client コマンド, 112, 113

pfinstall コマンド, 58

Can't boot from file/device メッセージ, 172

CHANGE DEFAULT BOOT DEVICE メッセージ,
179

check スクリプト

custom_probes.ok ファイルの作成, 85

custom_probes ファイルの妥当性検査, 85, 86

rules.ok ファイルの作成, 60

rules ファイルの妥当性検査, 59, 61, 86

派生プロファイルと, 65

ルールのテスト, 60, 85

check スクリプトの -p オプション, 60, 85

check スクリプトの -r オプション, 60, 85

client_arch キーワード, 131

client_root プロファイルキーワード, 132

CLIENT MAC ADDR エラーメッセージ, 177

clock gained xxx days メッセージ, 172

cluster プロファイルキーワード

説明と値, 133-134, 134

例, 41

CPU (プロセッサ)

プローブキーワード, 166

ルールキーワード, 115, 166

.cshrc ファイル, 69

custom_probes.ok ファイル

作成, 85

説明, 85

custom_probes ファイル

check による妥当性検査, 85, 86

custom_probes のテスト, 85

名前の指定, 82

要件, 82

D

dfstab ファイル, 26, 109

disksize ルールキーワード, 説明と値, 116, 166

disks プロブキーワード, 説明と値, 166

domainname プロブキーワード, 166

domainname ルールキーワード, 116, 166

dontuse プロファイルキーワード, 135, 163

E

eng_profile の例, 110

/etc/bootparams ファイル

JumpStart ディレクトリアクセスの有効化, 178

JumpStart ディレクトリのアクセスの有効化, 30

/etc/dfs/dfstab ファイル, 26, 109

/etc/mnttab ファイル, 32

F

fdisk コマンド, 76

fdisk プロファイルキーワード

説明と値, 135-137

例, 41

filesys キーワード, 138-142, 142-143

filesys プロファイルキーワード

説明と値, 138

例, 41

finish.log ファイル, 66

finish ルールフィールド, 説明, 38

G

geo キーワード, 143

getfile: RPC failed: error 5: RPC Timed out メッセージ, 31

GRUB ベースのブート

インストール, 98, 100

コマンドリファレンス, 102

プロファイルフロッピーディスクの作成, 34

H

hostaddress プロブキーワード, 166

hostaddress ルールキーワード, 116, 166

hostname プロブキーワード, 説明と値, 166

hostname ルールキーワード

説明と値, 116, 166

例, 115-119

I

install_config コマンド, 30, 31

install_type キーワード, 144

install_type プロファイルキーワード

プロファイルのテスト, 59

要件, 40, 41

例, 41

installed プロブキーワード, 説明と値, 166

installed ルールキーワード, 説明と値, 117, 166

IP アドレス

プローブキーワード, 166

ルールキーワード, 116, 166

J

- JumpStart ディレクトリ
 - rules ファイルの例, 36
 - アクセス権, 25, 31
 - 共有, 25, 109
 - 作成
 - SPARC ベースのシステム用フロッピーディスク, 32
 - x86 ベースのシステム用フロッピーディスク, 31, 34
 - サーバー, 25
 - 例, 109
 - 終了スクリプトでファイルを追加, 67
 - ファイルのコピー
 - インストールファイル, 27, 33, 35
 - 終了スクリプトの使用, 66
- JumpStart ディレクトリの共有, 26, 109

K

- karch プロンプトキーワード, 166
- karch ルールキーワード, 117, 166

L

- layout_constraint キーワード, 145-147
- le0: No carrier - transceiver cable problem メッセージ, 172
- locale キーワード, 148

M

- marketing_profile の例, 110
- memsize プロンプトキーワード, 説明と値, 166
- memsize ルールキーワード, 説明と値, 117, 166
- metadb プロファイルキーワード, 148
- mnttab ファイル, 32
- model プロンプトキーワード, 説明と値, 167
- model ルールキーワード, 説明と値, 118, 167

N

- network プロンプトキーワード, 説明と値, 167
- network ルールキーワード, 説明と値, 118, 167
- no_master_check キーワード, 149
- No carrier - transceiver cable problem メッセージ, 172
- noneuclidean プロファイルキーワード, 149
- Not a UFS filesystem メッセージ, 172

O

- osname プロンプトキーワード, 167
- osname ルールキーワード, 118, 167

P

- partitioning
 - fdisk パーティション, 41
 - 例, 41
- partitioning キーワード, 155
- pfinstall コマンド, 55
- probe ルールキーワード, 説明と値, 119
- prtvtoc コマンド
 - SPARC: ディスク構成ファイルの作成, 74
 - x86: ディスク構成ファイルの作成, 76

R

- root_device キーワード, 161
- rootdisk
 - filesystem のスライス値, 139
 - JumpStart で設定される値, 162
 - 定義, 162
- root パスワード, 終了スクリプトによる設定, 70
- RPC failed: error 5: RPC Timed out メッセージ, 31
- RPC Timed out メッセージ, 31, 177
- rule_keyword ルールフィールド, 37
- rule_value ルールフィールド, 37
- rules
 - 構文, 37
 - 妥当性検査, 60
 - 派生プロファイル, 64, 65

rules (続き)
 フィールドの説明, 37, 38
 複数行のルール, 37
 例, 38

rules.ok ファイル
 作成, 59, 60
 説明, 59

rules.ok ファイル, ルールの順序を一致させる, 38

rules.ok ファイル
 ルールの順序を一致させる, 90, 97

rules ファイル
 check による妥当性検査, 61
 カスタム JumpStart の例, 112
 派生プロファイルと, 65
 カスタム JumpStart の例, 111
 構文, 37
 コメント, 36
 作成, 36
 説明, 36
 名前の指定, 36, 37
 複数行のルール, 37
 ルールの追加, 37
 ルールのテスト, 60
 例, 36

rules ファイル内のバックスラッシュ, 37

rules ファイルの行を折り返す, 37

S

set_root_pw 終了スクリプト, 70

setup_install_server コマンドの -b オプション, 109

shareall コマンド, 26, 109

share コマンド
 JumpStart ディレクトリの共有, 26, 109

SI_PROFILE 環境変数, 65

Solaris ソフトウェア
 installed ルールキーワード, 117
 グループ, 133-134
 アップグレード, 134
 プロファイルの例, 41
 リリースまたはバージョン
 installed プロブキーワード, 166
 installed ルールキーワード, 117, 166

Solaris ソフトウェア, リリースまたはバージョン (続き)

 osname プロブキーワード, 167
 osname ルールキーワード, 118, 167

Solaris ソフトウェアのバージョン
 installed プロブキーワード, 166
 installed ルールキーワード, 166
 osname プロブキーワード, 167
 osname ルールキーワード, 118, 167

Solaris ソフトウェアのリリース
 installed プロブキーワード, 166
 installed ルールキーワード, 117, 166
 osname プロブキーワード, 167
 osname ルールキーワード, 118, 167

stty コマンド, 93, 98

SUNWCall グループ, 133-134

SUNWCprog グループ, 133-134

SUNWCreq グループ, 133-134

SUNWCrnet グループ, 133-134

SUNWCuser グループ, 133-134

SUNWCXall グループ, 133-134

system_type プロファイルキーワード
 説明と値, 163
 例, 41

T

timed out RPC エラー, 177

tip ライン接続ディスプレイの要件, 98

tip ライン接続の要件, 93

totaldisk プロブキーワード, 167

totaldisk ルールキーワード, 119, 167

transceiver cable problem メッセージ, 172

U

UFS, 32

Unknown client error メッセージ, 171

usedisk プロファイルキーワード, 説明と値, 163

V

/var/sadm/system/logs/begin.log ファイル, 64
 /var/sadm/system/logs/finish.log ファイル, 66
 volcheck コマンド, 32,34

W

WARNING: CHANGE DEFAULT BOOT
 DEVICE, 179
 WARNING: clock gained xxx days メッセージ, 172

あ

アーカイブ
 JumpStart プロファイルの例, 46, 47, 48, 49
 キーワード、カスタム JumpStart, 121-127
 アクセス権
 JumpStart ディレクトリ, 25, 31
 開始スクリプト, 64
 終了スクリプト, 66
 アップグレード
 アップグレードの失敗, 183
 カスタム JumpStart インストール, 87
 プロファイルキーワード, 134, 144, 155
 アップグレードの失敗, リブートの問題, 183
 アンパサンド (&&) ルールフィールド, 37

い

一致
 rootdisk 値, 162
 一致させる
 ルールの順序, 38, 90, 97
 インストールの準備、カスタム JumpStart, 21, 61

え

エンドユーザーシステムサポート, 133-134

か

開始、check スクリプト, 60, 61
 開始スクリプト
 アクセス権, 64
 概要, 63
 サイト固有のインストールプログラム, 79
 派生プロファイルの作成, 64, 65
 ルールフィールド, 37
 開発者システムサポート, 133-134
 開発者システムサポートソフトウェアグループ、
 プロファイルの例, 41
 カスタム JumpStart インストール, 87
 tip ライン接続の要件, 93, 98
 オプション機能, 63
 開始スクリプト, 63, 65
 概要, 63
 サイト固有のインストールプログラム, 79
 終了スクリプト, 65, 70
 概要, 21
 起動とインストール, 87
 準備, 21, 61
 説明, 21
 プロファイルキーワード, 120
 例, 105, 114
 check スクリプト, 112
 eng_profile の作成, 110
 Solaris フラッシュプロファイル, 46, 47, 48, 49
 JumpStart ディレクトリ, 109
 marketing_profile の作成, 110
 RAID-1 ボリュームプロファイル, 50, 52
 rules ファイルの編集, 111
 WAN ブートインストールプロファイル, 46
 エンジニアリングシステムの設定, 112
 サイトの設定, 105, 106
 スタンドアロンシステム, 19
 ネットワークに接続された, 20
 ネットワークに接続されていない, 19
 ブートとインストール, 114
 マーケティングシステムの設定, 108, 113
 !(感嘆符) ルールフィールド, 37
 感嘆符 (!) ルールフィールド, 37

き

キーワード

- Solaris フラッシュアーカイブ、カスタム JumpStart, 121-127
- プローブ, 81

け

- 限定ネットワークシステムサポート, 133-134

こ

- コアシステムサポート, 133-134
- 構成, ディスク構成ファイルの作成, 74
- コピー
 - JumpStart インストールファイル, 27, 33, 35
 - JumpStart ディレクトリファイル, 66
- コメント
 - rules ファイル内の, 36
 - プロファイル内の, 40

さ

サーバー

- JumpStart ディレクトリの作成, 25
- ルート領域, 132

サイズ

- tip ライン接続ディスプレイのサイズ, 93, 98
- スワップ空間
 - 最大サイズ, 132
 - ディスクレスクライアント, 132
 - プロファイルの例, 21
- ハードディスク
 - プローブキーワード, 166, 167
 - ルート領域, 132
 - ルールキーワード, 116, 119, 166, 167
- メモリー, 117, 166

- サイト固有のインストールプログラム, 79

- 削除, アップグレード時のクラスタ, 134

作成

- custom_probes.ok ファイル, 85
- JumpStart ディレクトリ, サーバー上, 25

作成 (続き)

- RAID-1 ボリューム, 142-143
- rules.ok ファイル, 59, 60, 85
- rules ファイル, 36
- UFS, 32
- ディスク構成ファイル, 74
- プロファイル
 - 説明, 40
 - 派生, 64
- ローカルファイルシステム, 138-142

し

終了スクリプト

- システムの root パスワードの設定, 70
- パッケージとパッチの追加, 67
- ルート環境のカスタマイズ, 69
- ルールフィールド, 38

出力ファイル

- 開始スクリプトログ, 64
- 終了スクリプトログ, 66
- 照合, 派生プロファイル, 65

す

スクリプト

- 開始スクリプト, 63, 65, 79
- 終了スクリプト, 65, 70, 79
- ルールフィールド内の Bourne シェルスクリプト, 37

スタンドアロンシステム

- カスタム JumpStart インストールの例, 19
- プロファイルの例, 41

スライス

- プローブキーワード, 166
- プロファイルの例, 41
- ルールキーワード, 117, 166

スワップファイルシステム

- サイズの決定, 132
- ディスクレスクライアントのスワップ空間, 132
- プロファイルの例, 21
- メモリーサイズと, 132

せ

- セキュリティ、root パスワード、70
- 全体ディストリビューション、133-134
- 全体ディストリビューションと OEM サポート、133-134

そ

- ソフトウェアグループ
 - アップグレード、134
 - プロファイルの例、41
 - プロファイル用、133-134

た

- 代替インストールプログラム、79
- 妥当性

- custom_probes ファイル
 - テスト、85

妥当性検査

- custom_probes ファイル
 - check の使用、86
 - rules ファイル
 - check の使用、59, 61, 86
 - カスタム JumpStart の例、112
 - 派生プロファイルと、65
 - ルールのテスト、60

つ

追加

- rules ファイルヘルールを、37
 - アップグレード時のクラスタ、134
 - 終了スクリプトでパッケージとパッチを、67
 - ソフトウェアグループからパッケージを、150

て

- ディスク構成ファイル
 - 作成
 - SPARC ベースのシステム、74

ディスク構成ファイル, 作成 (続き)

- x86 ベースのシステム、76

説明、55, 74

ディスクレスクライアント

- スワップ空間、132
- プラットフォーム、131

ディスプレイ

- tip ライン接続の要件、93, 98

ディレクトリ

JumpStart

- rules ファイルの例、36
- アクセス権、25, 31
- インストールファイルのコピー、27, 33, 35
- システム用に作成、31
- ディレクトリの共有、26, 109
- ディレクトリの作成、109
- ファイルのコピー、66
- ファイルの追加、67

移動

- JumpStart ディレクトリへ、60, 85
- ローカルディスク上の Solaris SPARC ソフトウェアのイメージ、33
- ローカルディスク上の Solaris ソフトウェアのイメージ、27
- ローカルディスク上の x86 ベースの Solaris イメージ、35

ディレクトリの移動

- JumpStart ディレクトリへ、60, 85
- ローカルディスク上の Solaris SPARC ソフトウェアのイメージ、33
- ローカルディスク上の Solaris ソフトウェアのイメージ、27
- ローカルディスク上の x86 ベースの Solaris イメージ、35

テスト

- custom_probes ファイルの妥当性検査
 - check の使用、85
 - custom_probes のテスト、85
- rules ファイルの妥当性検査
 - check の使用、59, 61, 86
 - カスタム JumpStart の例、112
 - 派生プロファイルと、65
 - ルールのテスト、60
- プロファイル、55, 59

デフォルト

- インストールするソフトウェアグループ, 134
- パーティション分割
 - ディスクの指定, 163
 - ディスクの除外, 135
- 派生プロファイル名, 65

と

- トークンリングカード、ブートエラー, 176
- ドメイン
 - プローブキーワード, 166
 - ルールキーワード, 116, 166

な

名前/名前の指定

- custom_probes ファイル, 82
- rules ファイル, 36, 37
- システムモデル名, 118, 167
- 派生プロファイル名, 65
- ホスト名, 116, 166

ね

- ネットワークインストール, カスタム JumpStart インストール, 例, 20
- ネットワーク番号, 118, 167

は

- パーティション分割
 - fdisk パーティション, 135-137
 - ディスクの除外, 135
 - プロファイルキーワード, 155, 163
- ハードディスク
 - partitioning
 - プロファイルキーワード, 155
 - 例, 41
 - rootdisk 値, 162

ハードディスク (続き)

- サイズ
 - プローブキーワード, 166, 167
 - ルート領域, 132
 - ルールキーワード, 116, 119, 166, 167
 - スワップ空間
 - 最大サイズ, 132
 - ディスクレスクライアント, 132
 - プロファイルの例, 21, 41
 - パーティション分割
 - partitioning default 用に指定, 163
 - partitioning default 用に除外, 135
 - マウント, 138
 - パス、check スクリプト, 60, 85
 - パスワード、root, 70
 - 派生プロファイル, 64, 65
 - パッケージ
 - Solaris Live Upgrade
 - 要件, 193
 - カスタム JumpStart を使用する要件, 193
 - 管理ファイル, 63
 - 追加
 - chroot による, 69
 - 終了スクリプトで, 67
 - パッチ
 - 追加
 - chroot による, 69
 - 終了スクリプトで, 67
- ふ
- ファイルとファイルシステム
 - UFS の作成, 32
 - 開始スクリプトの出力, 64
 - コピー
 - JumpStart インストールファイル, 27, 33, 35
 - 終了スクリプトによる JumpStart ディレクトリファイルの, 66
 - 作成
 - RAID-1 ボリューム, 142-143
 - ローカルファイルシステム, 138-142
 - 終了スクリプトの出力, 66
 - リモートファイルシステムのマウント, 138

ブート

- GRUBによるインストール, 98, 100
- GRUBの使用、コマンドリファレンス, 102
- プロファイルフロッピーディスクの作成, 34

複数行ある rules ファイル, 37

プラットフォーム

- システム属性とプロファイルを一致させる, 38, 90, 97
- システムモデル名, 118, 167
- ディスクレスクライアント, 131
- プローブキーワード, 166
- ルールキーワード, 117, 166

プローブキーワード

- arch, 166
- disks, 166
- domainname, 166
- hostaddress, 166
- hostname, 166
- installed, 166
- karch, 166
- memsize, 166
- model, 167
- network, 167
- osname, 167
- rootdisk, 167
- totaldisk, 167

プロセッサ

- プローブキーワード, 166
- ルールキーワード, 115, 166

フロッピーディスク

- JumpStart ディレクトリのアクセス, 29
- x86: JumpStart ディレクトリ, 31

プロファイル

- コメント, 40
- 作成, 40
- システムを一致させる, 38, 90, 97
- 説明, 40
- テスト, 59
- 名前の指定, 41
- 派生プロファイル, 64, 65
- 要件, 36, 40
- ルールフィールド, 37
- 例, 41

- eng_profile, 110

プロファイル, 例 (続き)

- marketing_profile, 110
- Solaris フラッシュ, 46, 47, 48, 49
- WANブートインストール, 46

プロファイルキーワード, 120, 163

- archive_location, 121-127
- backup_media, 127-129
- boot_device, 129
- bootenv createbe, 130
- client_arch, 131
- client_root, 132
- client_swap, 132
- cluster
 - 説明と値, 133-134, 134
 - 例, 41

dontuse

- usedisk と, 163
- 説明と値, 135

fdisk

- 説明と値, 135-137
- 例, 41

filesystem

- RAID-1 ボリューム, 142-143
- 説明と値, 138
- リモートファイルシステム, 138
- 例, 41
- ローカルファイルシステム, 138-142

forced_deployment、説明と値, 143

geo

- 説明と値, 143

install_type

- 説明と値, 144
- 要件, 40, 41
- 例, 41

layout_constraint、説明と値, 145-147

local_customization、説明と値, 147

locale、説明と値, 148

metadb

- 説明と値, 148
- 例, 41

no_master_check、説明と値, 149

noneuclidean, 149

partitioning

- 説明と値, 155

プロファイルキーワード, partitioning (続き)
例, 41
root_device, 161
system_type
説明と値, 163
例, 41
usedisk, 説明と値, 163
大文字と小文字の区別, 120
状態データベースの複製 (meatball) の作成, 148
パーティション分割
ディスクの指定, 163
ディスクの除外, 135
プロファイルフィールドの等号 (=), 65

へ

変数
SI_PROFILE, 65
SYS_MEMSIZE, 57

ま

マイクロプロセッサ
プローブキーワード, 166
ルールキーワード, 115, 166
マウント
Solaris インストールによる, 66
開始スクリプトに関する注意事項, 64
リモートファイルシステム, 138

め

メモリー
スワップ空間のサイズと, 132
プローブキーワード, 166
ルールキーワード, 117, 166

も

問題発生時の解決方法
DHCP によるネットワークからのブート, 177

問題発生時の解決方法 (続き)
インストールの一般的な問題
DHCP によるネットワークからのブート, 177
システムのブート, 178
間違ったサーバーからのブート, 178

よ

要件
custom_probes ファイル, 82
プロファイル, 36, 40

り

リモートファイルシステム, マウント, 138

る

ルート (/) ファイルシステム, プロファイルの例, 21
ルート環境, 終了スクリプトによるカスタマイズ, 69
ルートファイルシステム, 非アクティブブート環境のパッケージ要件, 193
ルール
rootdisk の一致規則, 162
順序を一致させる, 38, 90, 97
妥当性検査, 85
ルールキーワード, 115
any, 説明と値, 115, 166
arch, 115, 166
disksize, 説明と値, 116, 166
domainname, 116, 166
hostaddress, 116, 166
hostname, 115-119, 166
installed, 説明と値, 117, 166
karch, 117, 166
memsize, 117, 166
model, 118, 167
network, 118, 167
osname, 118, 167

ルールキーワード (続き)

- probe, 119

- totaldisk, 119, 167

ルールフィールド内の Bourne シェルスクリプト, 37

ろ

ログファイル

- 開始スクリプトの出力, 64

- 終了スクリプトの出力, 66

