



Solaris 10 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: 819-0341-12
2005 年 12 月

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

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、JumpStart、Solaris Flash、Sun Java、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。

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

「ATOK」は、株式会社ジャストシステムの登録商標です。

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

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

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

Unicode は、Unicode, Inc. の商標です。

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

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザ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ないものとします。

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

原典: Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations

Part No: 817-5506-12

Revision A



051222@13215



目次

はじめに	11
------	----

パート I	Solaris のインストールおよびアップグレードの計画の概要	15
-------	--	----

1	Solaris インストールの新機能	17
	Solaris 10 1/06 リリースにおける Solaris インストールの新機能	17
	非大域ゾーンをインストールしている場合の Solaris OS のアップグレード	17
	x86: GRUB ベースのブート	18
	Solaris リリースのアップグレードサポートの変更	19
	Solaris 10 3/05 リリースにおける Solaris インストールの新機能	20
	インストール手順の統一を含めた Solaris インストールの変更	20
	カスタム JumpStart インストールのパッケージとパッチの機能拡張	22
	インストール時に複数のネットワークインタフェースを構成する	22
	SPARC: 64 ビットパッケージに関する変更点	22
	カスタム JumpStart インストール方式による新しいブート環境の作成	23
	限定ネットワークソフトウェアグループ	23
	Virtual Table of Contents を使用してディスクパーティションテーブルを変更する	24
	x86: デフォルトブートディスクパーティションレイアウトの変更	24
2	Solaris のインストールおよびアップグレード (ロードマップ)	27
	作業マップ: Solaris ソフトウェアのインストールまたはアップグレード	27
	ネットワークからインストールするか DVD または CD を使ってインストールするか	30
	初期インストールかアップグレードか	31
	初期インストール	31

	アップグレード	32
	Solaris インストール方法の選択	32
	Sun Java System Application Server Platform Edition 8	35
3	Solaris のインストールおよびアップグレード (計画)	37
	システム要件と推奨事項	37
	ディスク容量とスワップ空間の割り当て	39
	ディスク容量に関する一般的な計画と推奨事項	40
	ソフトウェアグループごとの推奨ディスク容量	41
	アップグレード	43
	アップグレードに関する制限事項	44
	アップグレードプログラム	44
	アップグレードではなく Solaris フラッシュアーカイブをインストール	45
	ディスク容量の再配置を伴うアップグレード	46
	アップグレード前のシステムのバックアップ	47
	システムで稼働中の Solaris OS のバージョンを確認する方法	47
	ロケールの値	48
	プラットフォーム名とグループ	48
	ゾーンのインストールと構成	49
	Solaris ゾーン区分技術 (概要)	49
	Solaris ゾーン (計画)	51
	SPARC: 64 ビットパッケージに関する変更点	53
	x86: パーティション分割の推奨事項	54
	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でサービスパーティションを保持	55
4	x86: Solaris インストールのための GRUB ベースのブート	57
	x86: GRUB ベースのブート (概要)	57
	x86: GRUB ベースのブートの動作	58
	x86: GRUB デバイス命名規則	58
	x86: GRUB ベースのインストールについての情報の参照先	59
	x86: GRUB ベースのブート (計画)	60
	x86: ネットワークからの GRUB ベースのインストールの実行	61
	GRUB メインメニューについて	61
	x86: GRUB メニューの menu.lst ファイルの検出 (作業)	65
	▼ GRUB メニューの menu.lst ファイルの検出	65
	▼ アクティブな menu.lst ファイルが別のブート環境にある場合の GRUB メニューの menu.lst ファイルの検出	66

- ▼ Solais Live Upgrade ブート環境がマウントされている場合の GRUB メニューの menu.lst ファイルの検出 67
- ▼ 使用しているシステムに x86 ブートパーティションがある場合の GRUB メニューの menu.lst ファイルの検出 68

パート II カスタム JumpStart の使用 69

- 5 カスタム JumpStart (概要) 71
 - カスタム JumpStart とは 71
 - カスタム JumpStart の使用例 71
 - 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 73
- 6 カスタム JumpStart インストールの準備 (作業) 77
 - 作業マップ: カスタム JumpStart インストールの準備 78
 - 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 79
 - ▼ JumpStart ディレクトリをサーバー上に作成する方法 80
 - 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 82
 - ▼ 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 83
 - 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 85
 - ▼ SPARC: プロファイルフロッピーディスクを作成する方法 85
 - ▼ 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 87
 - ▼ x86: Solaris 10 3/05 リリースの場合: プロファイルフロッピーディスクを作成する方法 89
 - rules ファイルの作成 93
 - rules ファイルの構文 93
 - ▼ rules ファイルを作成する方法 94
 - rules ファイルの例 95
 - プロファイルの作成 96
 - プロファイルの構文 97
 - ▼ プロファイルを作成する方法 97
 - プロファイルの例 98
 - プロファイルのテスト 110
 - ▼ 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 111
 - ▼ プロファイル进行测试する方法 112
 - プロファイルテストの例 114
 - rules ファイルの妥当性を検査する 115
 - ▼ rules ファイルの妥当性を検査する方法 115

7	カスタム JumpStart オプション機能の使用 (作業)	117
	開始スクリプトの作成	117
	開始スクリプトに関する注意事項	118
	開始スクリプトによる派生プロファイルの作成	118
	終了スクリプトの作成	119
	終了スクリプトに関する注意事項	119
	▼ 終了スクリプトでファイルを追加する方法	120
	終了スクリプトによるパッケージとパッチの追加	120
	終了スクリプトによるルート環境のカスタマイズ	123
	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	123
	終了スクリプトによる非対話式インストール	125
	圧縮された構成ファイルの作成	125
	▼ 圧縮された構成ファイルを作成する方法	126
	圧縮された構成ファイルの例	126
	ディスク構成ファイルの作成	127
	▼ SPARC: ディスク構成ファイルを作成する方法	127
	SPARC: ディスク構成ファイルの例	128
	▼ x86: ディスク構成ファイルを作成する方法	129
	x86: ディスク構成ファイルの例	130
	サイト固有のインストールプログラムの使用	132
8	カスタムルールおよびプローブキーワードの作成 (作業)	133
	プローブキーワード	133
	custom_probes ファイルの作成	134
	custom_probes ファイルの構文	134
	custom_probes 内の関数名の構文	135
	▼ custom_probes ファイルを作成する方法	135
	custom_probes ファイルとキーワードの例	135
	custom_probes ファイルの妥当性検査	137
	▼ custom_probes ファイルを妥当性検査する方法	137
9	カスタム JumpStart インストールの実行 (作業)	139
	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	139
	SPARC: カスタム JumpStart インストールの実行	141
	▼ 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	141
	▼ 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	143

	SPARC: boot コマンドのコマンドリファレンス	144
	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	146
	x86: カスタム JumpStart インストールの実行	147
	▼ 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	148
	x86: GRUB boot コマンドの編集によるカスタム JumpStart インストールの実行	150
	▼ x86: GRUB boot コマンドを変更する方法	150
	▼ x86: 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	152
	x86: b コマンドまたは boot コマンドのコマンドリファレンス	154
10	カスタム JumpStart によるインストール (例)	159
	サイトの設定例	159
	インストールサーバーの作成	161
	x86: マーケティングシステム用のブートサーバーの作成	162
	JumpStart ディレクトリの作成	163
	JumpStart ディレクトリの共有	163
	SPARC: エンジニアリング部のプロファイルの作成	163
	x86: マーケティング部のプロファイルの作成	164
	rules ファイルの更新	165
	rules ファイルの妥当性を検査する	165
	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	166
	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	167
	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	168
	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	168
11	カスタム JumpStart (リファレンス)	171
	ルールキーワードと値の説明	171
	プロファイルキーワードと値	175
	プロファイルキーワードのクイックリファレンス	176
	プロファイルキーワードの説明と例	177
	非大域ゾーンでアップグレードする際のプロファイルキーワードの制限	219
	カスタム JumpStart の環境変数	219
	プローブキーワードと値	221

パート III RAID-1 ボリュームの使用 225

- 12 インストール時の **RAID-1** ボリューム (ミラー) の作成 (概要) 227
 - RAID-1 ボリュームを使用する理由 227
 - RAID-1 ボリュームの機能 228
 - Solaris ボリュームマネージャーコンポーネントの概要 230
 - 状態データベースと状態データベースの複製 230
 - RAID-0 ボリューム (連結) 231
 - RAID-1 ボリューム (ミラー) 232
 - RAID-1 ボリュームのディスク配置の例 233
- 13 インストール時の **RAID-1** ボリューム (ミラー) の作成 (計画) 235
 - システム要件 235
 - 状態データベースの複製のガイドラインと要件 236
 - 状態データベースの複製用のスライスの選択 236
 - 状態データベースの複製の数の選択 237
 - コントローラ間で状態データベースの複製を分散 237
 - RAID-1 ボリュームと RAID-0 ボリュームの要件とガイドライン 237
 - カスタム JumpStart と Solaris Live Upgrade のガイドライン 238
 - ディスクとコントローラの選択のガイドライン 241
 - スライスの選択のガイドライン 241
 - シングルユーザーモードでのブートが RAID-1 ボリュームに与える影響 242

パート IV 付録 243

- A 問題発生時の解決方法 (作業) 245
 - ネットワークインストールの設定に関する問題 245
 - システムのブートに関する問題 246
 - メディアからのブート時のエラーメッセージ 246
 - メディアからのブート時の一般的な問題 247
 - ネットワークからのブート時のエラーメッセージ 249
 - ネットワークからのブート時の一般的な問題 252
 - Solaris OS の初期インストール 253
 - ▼ x86: IDE ディスクの不良ブロックの検査 254
 - Solaris OS のアップグレード 255
 - アップグレード時のエラーメッセージ 255
 - アップグレード時の一般的な問題 256

▼ 問題発生後にアップグレードを継続する方法	257
x86: GRUB を使用する場合の Solaris Live Upgrade に関する問題	258
▼ 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	260
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	262
▼ 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	262
▼ 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	263
B その他の SVR4 パッケージ要件 (リファレンス)	265
現在の OS の変更を回避	265
絶対パスの使用	265
pkgadd -R コマンドの使用	266
\$PKG_INSTALL_ROOT と \$BASEDIR の相違点の概要	266
スクリプト作成のガイドライン	267
ディスクリスクライアントの互換性維持	268
パッケージの検証	268
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	269
参照先	270
 用語集	 273
 索引	 291

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™ 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。サポートされるシステムについては、Solaris Hardware Compatibility List (<http://www.sun.com/bigadmin/hcl>) を参照してください。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。

サポートされ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、Solaris ソフトウェアをインストールする際に必要な関連情報の一覧を示します。

表 P-1 関連情報

情報	説明
『Solaris 10 インストールガイド (基本編)』	グラフィカルユーザーインターフェース (GUI) による OS の基本的なインストール方法について説明します。
『Solaris 10 インストールガイド (ネットワークインストール)』	ローカルエリアネットワーク経由または広域ネットワーク経由で Solaris のリモートインストールを実行する方法について説明します。
『Solaris 10 インストールガイド (カスタム JumpStart/ 上級編)』	無人カスタム JumpStart™ インストールの実行に必要なファイルとディレクトリの作成方法について説明します。このマニュアルでは、RAID-1 ボリュームの作成方法についても説明します。
『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』	CD や DVD を使って Solaris OS へアップグレードする際の計画情報を提供します。このマニュアルでは、Solaris Live Upgrade を使って新しいブート環境を作成しアップグレードする方法についても説明します。
『Solaris 10 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)』	Solaris フラッシュアーカイブの作成方法と、Solaris フラッシュアーカイブを使って複数のシステムへ Solaris OS をインストールする方法について説明します。
『Solaris のシステム管理 (デバイスとファイルシステム)』	システムファイルのバックアップ方法など、システム管理タスクについて説明します。

表 P-1 関連情報 (続き)

情報	説明
『Solaris ご使用にあたって』	Solaris リリースに関する、バグ、既知の問題、提供されなくなったソフトウェア、パッチなどが解説されています。
SPARC: 『Solaris Sun ハードウェアマニュアル』 (http://docs.sun.com)	サポート対象のハードウェアについて解説されています。
『Solaris パッケージリスト』	Solaris OS に含まれるパッケージの一覧と説明です。
x86: Solaris Hardware Compatibility List	サポート対象ハードウェアの情報とデバイス構成が解説されています。

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:

表 P-2 表記上の規則 (続き)

字体または記号	意味	例
<i>AaBbCc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、 <code>rm filename</code> 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。	<pre>sun% grep `^#define ` XV_VERSION_STRING'</pre>

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

Solaris のインストールおよびアップグレードの計画の概要

このパートでは、任意のインストールプログラムによる Solaris オペレーティングシステムのインストールまたはアップグレードの計画について説明します。

第 1 章

Solaris インストールの新機能

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 OS すべての機能の詳細は、『Solaris 10 の概要』を参照してください。

- 17 ページの「Solaris 10 1/06 リリースにおける Solaris インストールの新機能」
- 20 ページの「Solaris 10 3/05 リリースにおける Solaris インストールの新機能」

Solaris 10 1/06 リリースにおける Solaris インストールの新機能

この節では、Solaris 10 1/06 リリースの次のような新しいインストール機能について説明します。

非大域ゾーンをインストールしている場合の Solaris OS のアップグレード

Solaris 10 1/06 以降のリリースでは、Solaris ゾーン区分技術により、大域ゾーンである Solaris の単一インスタンス内の非大域ゾーンを構成できるようになりました。非大域ゾーンはアプリケーション実行環境の 1 つで、そこではプロセスがほかのすべてのゾーンから隔離されます。非大域ゾーンがインストールされたシステムが稼働している場合は、標準の Solaris アップグレードプログラムを使用して Solaris 10 1/06 リリースにアップグレードできます。アップグレードに使用できるのは、Solaris の対話式インストールプログラムか、カスタム JumpStart です。非大域ゾーンがインストールされている場合のアップグレードには、若干の制限があります。

- サポートされるカスタム JumpStart キーワードの数には制限があります。サポートされるカスタム JumpStart キーワードの一覧については、本書を参照してください。

- CD-ROM は配布されませんが、DVD-ROM またはネットワークインストールイメージによってアップグレードできます。
- 非大域ゾーンがインストールされているシステムでは、Solaris Live Upgrade を使用してシステムをアップグレードしないでください。ブート環境は `lucreate` コマンドを使用して作成できますが、`luupgrade` コマンドを使用して非大域ゾーンがインストールされているブート環境をアップグレードすることはできません。その場合は、アップグレードに失敗し、エラーメッセージが表示されます。

Solaris 対話式インストールプログラムの用法については、『Solaris 10 インストールガイド (基本編)』を参照してください。

x86: GRUB ベースのブート

Solaris 10 1/06 以降のリリースでは、オープンソースの GNU GRand Unified BootLoader (GRUB) が x86 システムの Solaris OS に採用されています。GRUB の役割は、ブートアーカイブをシステムのメモリーにロードすることです。ブートアーカイブは、`root (/)` ファイルシステムがマウントされる前のシステム起動時に必要とされる重要なファイルの集合です。ブートアーカイブを使用して Solaris OS をブートします。

もっとも注目すべき変更点は、Solaris Device Configuration Assistant (デバイス構成用補助) が GRUB メニューに置き換えられたことです。GRUB メニューにより、システムにインストールされている別のオペレーティングシステムのブートが容易になります。GRUB メニューは x86 システムのブート時に表示されます。矢印キーを使用して、インストールする OS インスタンスを GRUB メニューから選択できます。OS インスタンスを選択しないと、デフォルトの OS インスタンスがブートします。

GRUB ベースのブート機能により、次の点が改善されます。

- ブート動作の高速化
- USB CD ドライブまたは DVD ドライブからのインストール
- USB ストレージデバイスからブートする機能
- PXE ブート用の簡略化された DHCP セットアップ (ベンダー固有のオプションなし)
- すべての実行時モードドライバの削除
- Solaris Live Upgrade および GRUB メニューを使用して、ブート環境をすばやくアクティブ化およびフォールバックする機能。

GRUB の詳細については、次の節を参照してください。

作業	GRUB の作業	参照先
インストール	GRUB ベースのブートの概要	57 ページの「x86: GRUB ベースのブート (概要)」
	GRUB ベースのブートのインストール計画	60 ページの「x86: GRUB ベースのブート (計画)」
	GRUB メニューを使用したネットワーク経由のブートおよびインストール方法	『Solaris 10 インストールガイド (ネットワークインストール)』の「DVD イメージを使用した、ネットワークからのブートとインストール」
	GRUB メニューとカスタム JumpStart インストール方式によるブートおよびインストール方法	147 ページの「x86: カスタム JumpStart インストールの実行」
	GRUB メニューと Solaris Live Upgrade を使用して、ブート環境をアクティブ化およびフォールバックする方法	■ 『Solaris 10 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」 ■ 『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「障害回復: 元のブート環境へのフォールバック (コマンド行インタフェース)」
システム管理	GRUB メニューの menu.lst ファイルの検出	65 ページの「x86: GRUB メニューの menu.lst ファイルの検出 (作業)」
	GRUB メニューによるシステム管理作業の実行方法	■ 『Solaris のシステム管理 (基本編)』 ■ 『Solaris のシステム管理 (デバイスとファイルシステム)』 ■ bootadm (1M) ■ installgrub (1M)

注 - GNU は、「GNU's Not UNIX」の再帰的頭字語です。詳細については、<http://www.gnu.org> を参照してください。

Solaris リリースのアップグレードサポートの変更

Solaris 10 1/06 以降のリリースは、Solaris 8、9、または 10 リリースからアップグレード可能です。Solaris 7 リリースからのアップグレードはサポートされていません。

Solaris 10 3/05 リリースにおける Solaris インストールの新機能

この節では、Solaris 10 3/05 リリースの次のような新しいインストール機能について説明します。

インストール手順の統一を含めた Solaris インストールの変更

Solaris 10 3/05 以降のリリースでは、Solaris OS のインストールにいくつかの変更が加えられ、より簡単に統一された方法でインストールできます。

変更内容は次のとおりです。

- 今回のリリースには、1 枚のインストール DVD と数枚のインストール CD が付いています。Solaris Operating System DVD には、すべてのインストール CD の内容が含まれています。
 - **Solaris Software 1** – ブート可能な CD は、この CD だけです。この CD から、Solaris インストール用グラフィカルユーザーインターフェース (GUI) またはコンソールベースのインストールを利用できます。また、どちらのインストールを利用する場合でも、選択したソフトウェア製品だけをインストールすることもできます。
 - その他の **Solaris Operating System CD** – これらの CD には次のものが含まれます。
 - Solaris パッケージ (必要に応じてインストールします)
 - サポートまたは非サポートのソフトウェアが入っている ExtraValue ソフトウェア
 - インストーラ
 - ローカライズされたインターフェースソフトウェアおよびマニュアル
- Solaris Installation CD はなくなりました。
- CD または DVD のどちらでインストールする場合も、GUI インストールがデフォルトです (十分なメモリーがある場合)。ただし、text ブートオプションを使用してコンソールベースのインストールを指定することもできます。
- インストール手順が単純化され、ブート時に言語サポートを選択し、ロケールを後で選択できます。

注 – GUI またはコンソールを使用しない Solaris カスタム JumpStart™ インストール方式には変更はありません。

OS をインストールするには、Solaris Software - 1 CD または Solaris Operating System DVD を挿入してから、次のいずれかのコマンドを入力します。

- デフォルトの GUI インストールの場合 (システムメモリーが十分にある場合) は、**boot cdrom** と入力します。
- コンソールベースのインストールの場合は、**boot cdrom - text** と入力します。

新しい text ブートオプションのある CD または DVD メディアを使用して Solaris OS をインストールする方法について

『Solaris 10 インストールガイド (基本編)』

CD メディアによるインストールサーバーの設定方法の変更について

『Solaris 10 インストールガイド (ネットワークインストール)』

GUI インストールまたはコンソールベースのインストールを利用する

Solaris 10 3/05 以降のリリースでは、ソフトウェアのインストールに GUI を使用するか、ウィンドウ表示環境を使用するか、またはウィンドウ表示環境を使用しないかを選択できます。十分なメモリーがある場合は、デフォルトで GUI が表示されます。メモリー不足により GUI を表示できない場合はデフォルトで別の環境が表示されます。ブートオプション `nowin` または `text` を使用すると、デフォルト動作を変更できます。しかし、システム内のメモリー容量による制約、またはリモートインストールによる制約があります。また、ビデオアダプタが検出されない場合、Solaris インストールプログラムは、自動的にコンソールベースの環境で表示されます。

具体的なメモリー要件については、[37 ページ](#)の「システム要件と推奨事項」を参照してください。

カスタム JumpStart インストールのパッケージとパッチの機能拡張

Solaris 10 3/05 以降のリリースでは、カスタム JumpStart インストール方式を使用して Solaris OS をインストールおよびアップグレードした場合、新しくカスタマイズすることで次のことが可能になります。

- Solaris フラッシュインストールと追加パッケージ
カスタム JumpStart プロファイルの `package` キーワードが拡張され、Solaris フラッシュアーカイブを追加パッケージとともにインストールできるようになりました。たとえば、2 台のマシンに同じ基本アーカイブをインストールし、それぞれのマシンに別のパッケージを追加することができます。これらのパッケージは、Solaris OS ディストリビューションに含まれている必要はありません。
- Solaris ディストリビューションに含まれない追加パッケージのインストール
`package` キーワードが拡張され、Solaris ディストリビューションに含まれないパッケージもインストールできるようになりました。追加パッケージをインストールするために、インストール後スクリプトを作成する必要がなくなりました。
- Solaris OS パッチをインストールする機能
カスタム JumpStart プロファイルの新しい `patch` キーワードを使用して、Solaris OS のパッチをインストールできます。この機能を利用して、パッチファイルに指定されているパッチをインストールできます。

詳細は、『Solaris 10 インストールガイド (カスタム JumpStart/ 上級編)』を参照してください。

インストール時に複数のネットワークインタフェースを構成する

Solaris 10 3/05 以降のリリースでは、Solaris インストールプログラムを使用してインストール時に複数のインタフェースを構成できます。これらのインタフェースは、システムの `sysidcfg` ファイルに事前に構成できます。また、インストール時に構成することもできます。詳細については、次のドキュメントを参照してください。

- 『Solaris 10 インストールガイド (ネットワークインストール)』
- `sysidtool(1M)`
- `sysidcfg(4)`

SPARC: 64 ビットパッケージに関する変更点

以前の Solaris リリースでは、Solaris ソフトウェアは 32 ビットコンポーネントと 64 ビットコンポーネントがそれぞれ別のパッケージで提供されていました。**Solaris 10 3/05** 以降のリリースでは、パッケージが簡略化され、32 ビットコンポーネントと 64

ビットコンポーネントのほとんどが1つのパッケージで配布されています。統合されたパッケージでは、元の32ビットパッケージの名前がそのまま使用されています。64ビットパッケージは提供されなくなりました。

64ビットパッケージがなくなったことで、インストールが簡素化され、パフォーマンスも向上します。

- パッケージ数が減ったことで、パッケージのリストが含まれるカスタム JumpStart スクリプトが簡素化されます
- ソフトウェア機能を1つのパッケージにまとめるだけなので、パッケージシステムも簡素化されます
- インストールするパッケージ数が少ないため、インストール時間が短縮されます

64ビットパッケージの名前は、次の規則に従って変更されます。

- 64ビットパッケージに対応する32ビットパッケージがある場合は、その32ビットパッケージの名前が64ビットパッケージに付けられます。たとえば、`/usr/lib/sparcv9/libc.so.1` などの64ビットライブラリの場合、以前は `SUNWcs1x` で提供されていましたが、現在は `SUNWcs1` で提供されます。64ビットの `SUNWcs1x` パッケージは提供されなくなりました。
- 対応する32ビットパッケージが存在しない場合は、名前から接尾辞「x」が削除されます。たとえば、`SUNW1394x` は `SUNW1394` になります。

このため、カスタム JumpStart スクリプトやほかのパッケージインストールスクリプトで64ビットパッケージを参照している場合は、これらのスクリプトを変更して参照を削除する必要があります。

カスタム JumpStart インストール方式による新しいブート環境の作成

Solaris 10 3/05 以降のリリースでは、Solaris オペレーティングシステムをインストールする場合に、JumpStart インストール方式を使用して空のブート環境を作成できます。空のブート環境には、必要なときに備えて Solaris フラッシュアーカイブを格納しておくことができます。

詳細については、[第11章](#)を参照してください。

限定ネットワークソフトウェアグループ

Solaris 10 3/05 以降のリリースでは、インストール時に限定ネットワークソフトウェアグループ (`SUNWCrnet`) を選択または指定することにより、有効なネットワークサービスが少なくても、よりセキュリティー保護されたシステムを構築できます。限定ネットワークソフトウェアグループでは、システム管理ユーティリティとマルチユーザーのテキストベースコンソールが利用できます。`SUNWCrnet` は、ネット

ワークインタフェースを有効にします。インストール時に、ソフトウェアパッケージを追加したり、必要に応じてネットワークサービスを使用可能にすることによって、システムの構成をカスタマイズすることができます。

詳細は、『Solaris 10 インストールガイド (カスタム JumpStart/ 上級編)』を参照してください。

Virtual Table of Contents を使用してディスクパーティションテーブルを変更する

Solaris 10 3/05 以降のリリースでは、Solaris インストールプログラムにより、Virtual Table of Contents (VTOC) から既存のスライスをロードできます。インストールのデフォルトのディスクレイアウトを使用するのではなく、インストール時にシステムの既存のディスクスライステーブルをそのまま使用できるようになりました。

x86: デフォルトブートディスクパーティションレイアウトの変更

Solaris 10 3/05 以降のリリースでは、Solaris インストールプログラムの新機能として、ブートディスクパーティションレイアウトが採用されています。ブートディスクパーティションのデフォルトのレイアウトは、Sun x86 ベースのシステムのサービスパーティションと調和します。このインストールプログラムを使用すれば、既存のサービスパーティションをそのまま使用できます。

デフォルトのブートディスクレイアウトには、次のパーティションが含まれます。

- 1 番目のパーティション – サービスパーティション (システムの既存サイズ)
- 2 番目のパーティション – x86 ブートパーティション (約 11M バイト)
- 3 番目のパーティション – Solaris オペレーティングシステムパーティション (ブートディスクの残りの領域)

このデフォルトのレイアウトを使用する場合は、Solaris インストールプログラムからブートディスクレイアウトの選択を要求されたときに、「デフォルト」を選択します。

注 – サービスパーティションが現在作成されていないシステムに Solaris OS x86 ベースのシステムをインストールすると、Solaris インストールプログラムは新しいサービスパーティションを作成しません。このシステムにサービスパーティションを作成するには、最初にシステムの診断 CD を使用してサービスパーティションを作成してください。サービスパーティションを作成してから、Solaris オペレーティングシステムをインストールします。

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。

詳細は、『Solaris 10 インストールガイド (カスタム JumpStart/ 上級編)』を参照してください。

第 2 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)

この章では、Solaris オペレーティングシステム (OS) のインストールやアップグレードを実施する前に行うべき決定に関して説明します。この章の内容は次のとおりです。

- 27 ページの「作業マップ: Solaris ソフトウェアのインストールまたはアップグレード」
- 30 ページの「ネットワークからインストールするか DVD または CD を使ってインストールするか」
- 31 ページの「初期インストールかアップグレードか」
- 32 ページの「Solaris インストール方法の選択」
- 35 ページの「Sun Java System Application Server Platform Edition 8」

注 - このマニュアルでは「スライス」という用語を使用しますが、一部の Solaris のマニュアルとプログラムでは、スライスのことを「パーティション」と呼んでいる場合があります。

x86: 混同を避けるため、このマニュアルでは、x86 fdisk パーティションと Solaris fdisk パーティション内の分割とを区別しています。x86 fdisk の分割を「パーティション」と呼びます。Solaris fdisk パーティション内の分割を「スライス」と呼びます。

作業マップ: Solaris ソフトウェアのインストールまたはアップグレード

次の作業マップは、任意のインストールプログラムを使った Solaris OS のインストールやアップグレードに必要な作業の概要を示したものです。インストールしようとする環境にとってもっとも効率的なインストールを行うためにこういった選択をすべきかを、この作業マップを参考にして判断してください。

表 2-1 作業マップ: Solaris ソフトウェアのインストールまたはアップグレード

作業	説明	参照先
初期インストールかアップグレードかを選択します。	初期インストールまたはアップグレードのいずれかを選択します。	31 ページの「初期インストールかアップグレードか」。
インストールプログラムを選択します。	Solaris OS には、インストールまたはアップグレード用のプログラムがいくつか用意されています。使用している環境にもっとも適したインストール方法を選択してください。	32 ページの「Solaris インストール方法の選択」。
(Solaris 対話式インストールプログラム) デフォルトインストールかカスタムインストールを選択します。	環境に適したインストール方法を決定します。 ■ グラフィカルユーザーインターフェース (GUI) を使用する場合、デフォルトインストールまたはカスタムインストールを選択できます。 ■ デフォルトインストールでは、ハードディスクのフォーマット後、あらかじめ選択されたソフトウェアセットがインストールされます。 ■ カスタムインストールでは、ハードディスクレイアウトを自由に変更し、インストールするソフトウェアを選択することができます。 ■ テキストインストーラ (グラフィカルでないインターフェース) を使用する場合は、デフォルト値を選択するか、この値を編集してインストールするソフトウェアを選択できます。	Solaris インストールプログラムの選択肢については、『Solaris 10 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「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)」を参照してください。
システム要件を検討します。また、ディスク容量およびスワップ空間を計画を立てて割り当てます。	インストールまたはアップグレードの最小要件をシステムが満たしているかどうかを判断します。インストールする Solaris OS のコンポーネントに必要なディスク容量をシステムに割り当てます。使用しているシステムにとって適切なスワップ空間レイアウトを判断します。	第 3 章。
ローカルメディアとネットワークのどちらを使ってインストールするかを選択します。	使用している環境にもっとも適切なインストールメディアを選択します。	30 ページの「ネットワークからインストールするか DVD または CD を使ってインストールするか」。

表 2-1 作業マップ: Solaris ソフトウェアのインストールまたはアップグレード (続き)

作業	説明	参照先
システム情報を収集します。	■ Solaris インストールプログラムの場合は、ワークシートを使用して、インストールまたはアップグレードに必要なすべての情報を収集します。 ■ カスタム JumpStart インストールの場合は、プロファイルで使用するプロファイルキーワードを決定します。次に、キーワードの説明を読み、必要なシステム情報を確認します。	■ Solaris インストールプログラムの場合は、次のいずれかのマニュアルを参照してください。 ■ 初期インストールの場合: 『Solaris 10 インストールガイド (ネットワークインストール)』の「インストール用のチェックリスト」 ■ アップグレードの場合: 『Solaris 10 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 4 章「アップグレードの前に収集すべき情報 (計画)」 ■ カスタム JumpStart インストールについては、第 11 章を参照してください。
(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 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「システム構成情報の事前設定 (作業)」。

表 2-1 作業マップ: Solaris ソフトウェアのインストールまたはアップグレード (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。	ネットワークから Solaris ソフトウェアをインストールする方法を選択する場合は、次の作業をすべて実行してください。 ■ (x86 システム) 使用しているシステムが PXE をサポートしていることを確認します ■ インストールサーバーを作成します ■ ブートサーバーを作成します (必要な場合) ■ DHCP サーバーを構成します (必要な場合) ■ ネットワークからインストールするシステムを設定します	ローカルエリアネットワークからインストールする方法については、『Solaris 10 インストールガイド (ネットワークインストール)』の第 9 章「CD メディアを使用したネットワークインストールの準備 (作業)」を参照してください。 広域ネットワークからインストールする方法については、『Solaris 10 インストールガイド (ネットワークインストール)』の第 13 章「WAN ブートによるインストールの準備 (作業)」を参照してください。
(アップグレードのみ) アップグレードの前に必要な作業を行います。	システムのバックアップをとり、アップグレード時にディスク容量の再配置が行われるかを確認します。	43 ページの「アップグレード」。
インストールまたはアップグレードを実行します。	選択した Solaris インストール方法を使って Solaris ソフトウェアのインストールまたはアップグレードを行います。	インストールプログラムの詳細な手順を説明している章。
インストールのトラブルシューティングを行います。	インストールのトラブルシューティングについては、問題発生時の解決方法を参照してください。	付録 A。

ネットワークからインストールするか DVD または CD を使ってインストールするか

DVD-ROM または CD-ROM ドライブにアクセスできるシステムへのインストールまたはアップグレードができるように、Solaris ソフトウェアは DVD または CD メディアで配布されます。

リモートの DVD イメージまたは CD イメージを使ってネットワークからインストールを行うように、システムを設定できます。このようなシステム設定は、次のような場合に行います。

- システムにローカルの DVD-ROM ドライブまたは CD-ROM ドライブがない場合

- Solaris ソフトウェアを複数のシステムにインストールする必要があるが、各システムのローカルドライブにディスクを挿入する手間を省きたい場合

ネットワークからインストールする場合、どの Solaris インストール方法でも使用できます。しかし、Solaris フラッシュインストール機能やカスタム JumpStart インストールを使ってネットワークからインストールを行うと、大規模の企業におけるインストールプロセスの一元化と自動化が可能になります。さまざまなインストール方法の詳細については、[32 ページの「Solaris インストール方法の選択」](#)を参照してください。

ネットワークから Solaris ソフトウェアをインストールする場合は、初期設定が必要です。ネットワークからインストールする場合の準備については、次のいずれかを参照してください。

ローカルエリアネットワークからインストールする場合の準備の詳細	『Solaris 10 インストールガイド (ネットワークインストール)』の第 9 章「CD メディアを使用したネットワークインストールの準備 (作業)」
広域ネットワークからインストールする場合の準備	『Solaris 10 インストールガイド (ネットワークインストール)』の第 13 章「WAN ブートによるインストールの準備 (作業)」
PXE を使用してネットワーク経由で x86 ベースのクライアントをインストールする場合	『Solaris 10 インストールガイド (ネットワークインストール)』の「PXE を使用したネットワーク経由のブートとインストールの概要」

初期インストールかアップグレードか

初期インストールを行うか、アップグレードを行う (Solaris OS がシステム上ですでに動作している場合) かを選択できます。

初期インストール

初期インストールでは、システムのディスクが Solaris OS の新しいバージョンで上書きされます。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 Solaris OS がすでに動作している場合でも、初期インストールを行うことができます。ローカルに行なった変更を維持する場合は、インストールを行う前にローカル変更のバックアップをとる必要があります。インストールが終わったあとでローカル変更を復元できます。

初期インストールは、Solaris のどのインストール方法を使っても実行できます。さまざまな Solaris のインストール方法の詳細については、[32 ページの「Solaris インストール方法の選択」](#)を参照してください。

アップグレード

Solaris OS のアップグレードには、2 つの方法があります。標準の方法と Solaris Live Upgrade を使う方法です。標準アップグレードでは、現在の Solaris OS の既存の構成パラメータが最大限保存されます。Solaris Live Upgrade では、現在のシステムのコピーが作成されます。このコピーを、標準アップグレードでアップグレードできます。リブートするだけで、アップグレード済みの Solaris OS に切り替えることができます。エラーが発生した場合、リブートして、元の Solaris OS に戻すことができます。Solaris Live Upgrade では、アップグレードの間システムを停止する必要がなく、新旧の Solaris OS 間で切り替えることができます。

アップグレードの詳細と、さまざまなアップグレード方法の一覧については、[43 ページの「アップグレード」](#)を参照してください。

Solaris インストール方法の選択

Solaris OS には、インストールまたはアップグレード用のプログラムがいくつか用意されています。それぞれのインストール方法には、特定のインストール要件やインストール環境を意図したさまざまな機能があります。次の表を参考にして、使用するインストール方法を決定してください。

表 2-2 インストール方法の選択

作業	インストール方法	各プログラムの特長	参照先
対話式プログラムを使って、CD-ROM か DVD-ROM から単一のシステムにインストールします。	Solaris インストールプログラム	■ このプログラムでは、作業をパネルに分割し、情報の入力を求めるプロンプトを表示して、デフォルト値を提供します。 ■ このプログラムは、複数のシステムをインストールまたはアップグレードする必要がある場合には効率のよい方法ではありません。複数のシステムをバッチインストールする場合は、カスタム JumpStart または Solaris フラッシュインストール機能を使用してください。	『Solaris 10 インストールガイド (基本編)』
ローカルエリアネットワークから単一のシステムにインストールします。	ネットワーク経由で Solaris インストールプログラムを実行	このプログラムを使用すると、インストールするソフトウェアのイメージをサーバーに設定し、このイメージをリモートシステムにインストールできます。複数のシステムをインストールする必要がある場合は、カスタム JumpStart および Solaris フラッシュインストール方式でネットワークインストールイメージを使用すると、ネットワーク上でシステムを効率的にインストールおよびアップグレードできます。	『Solaris 10 インストールガイド (ネットワークインストール)』のパート II 「ローカルエリアネットワーク経由のインストール」
プロファイルを作成し、このプロファイルに基づいて、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化します。	カスタム JumpStart	このプログラムを使用すると、複数のシステムを効率的にインストールできます。ただし、システムの数が少ない場合は、カスタム JumpStart 環境の作成は時間の浪費になる可能性があります。システムの数が少ない場合は、Solaris 対話式インストールプログラムを使用してください。	第 6 章

表 2-2 インストール方法の選択 (続き)

作業	インストール方法	各プログラムの特長	参照先
同じソフトウェアとその構成を複数のシステムに複製します。	Solaris フラッシュアーカイブ	■ このプログラムを使用すると、Solaris パッケージすべてを一度にシステムにインストールすることで時間を節約できます。ほかのプログラムでは、各 Solaris パッケージを個別にインストールし、パッケージごとにパッケージマップをアップグレードします。 ■ Solaris フラッシュアーカイブはサイズの大きいファイルであるため、大量のディスク容量が必要です。別のインストール構成を管理したり、使用しているインストール構成を変更したりするには、カスタム JumpStart インストール方式の使用を検討することをお勧めします。代わりに、JumpStart 終了スクリプトまたは、組み込まれている Solaris フラッシュ配備後スクリプトを使用して、システム固有のカスタマイズを実行できます。	『Solaris 10 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)」
広域ネットワーク (WAN) またはインターネットから複数のシステムにインストールします。	WAN ブート	ネットワーク経由で Solaris フラッシュアーカイブをインストールする場合は、このプログラムを使用するとセキュリティ保護されたインストールが可能です。	『Solaris 10 インストールガイド (ネットワークインストール)』の第 11 章「WAN ブート (概要)」
システムを稼働させたままアップグレードします。	Solaris Live Upgrade	■ このプログラムを使用すると、アップグレードするかパッチを追加して、標準アップグレードに関連したシステム停止時間が発生するのを避けることができます ■ このプログラムを使用すると、現行の OS に影響を与えずにアップグレードまたは新しいパッチのテストをすることができます	『Solaris 10 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 6 章「Solaris Live Upgrade (概要)」

表 2-2 インストール方法の選択 (続き)

作業	インストール方法	各プログラムの特長	参照先
Solaris OS のインストール後、隔離されたアプリケーション環境を作成します。	Solaris ゾーン区分技術	このプログラムを使用すると、隔離された非大域ゾーンが作成され、セキュリティ保護されたアプリケーション環境が提供されます。このように隔離されているので、あるゾーンで実行中のプロセスが、ほかのゾーンで実行中のプロセスから監視または操作されることがありません。	『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」

Sun Java System Application Server Platform Edition 8

Sun Java System Application Server Platform Edition 8 では、アプリケーションサービスと Web サービスを広範囲に配備できます。このソフトウェアは、Solaris OS のインストール時に自動的にインストールされます。このサーバーについては、次のマニュアルがあります。

サーバーの起動に関するマニュアル	インストールディレクトリ内の『Sun Java System Application Server Platform Edition 8 クイックスタートガイド』 (/docs/QuickStart.html)
アプリケーションサーバーのマニュアルセット一式	http://docs.sun.com/db/coll/ApplicationServer8_04q2
チュートリアル	http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html

第 3 章

Solaris のインストールおよびアップグレード (計画)

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。この章の内容は次のとおりです。

- 37 ページの「システム要件と推奨事項」
- 39 ページの「ディスク容量とスワップ空間の割り当て」
- 43 ページの「アップグレード」
- 47 ページの「システムで稼働中の Solaris OS のバージョンを確認する方法」
- 48 ページの「ロケールの値」
- 48 ページの「プラットフォーム名とグループ」
- 51 ページの「非大域ゾーンを使用している場合のインストールとアップグレード」
- 53 ページの「SPARC: 64 ビットパッケージに関する変更点」
- 54 ページの「x86: パーティション分割の推奨事項」

システム要件と推奨事項

表 3-1 SPARC: メモリー、スワップ、およびプロセッサの推奨事項

SPARC システム	サイズ
インストールまたはアップグレードするために必要なメモリー	推奨サイズは 256M バイトです。最小サイズは 128M バイトです。 注 - オプションのインストール機能の中には、十分な量のメモリーがないと使用できないものがあります。たとえば、メモリーが足りないシステムに DVD からインストールする場合、Solaris インストールプログラムの GUI ではなく、テキストインストーラを使用する必要があります。これらのメモリー要件については、表 3-3 を参照してください。

表 3-1 SPARC: メモリー、スワップ、およびプロセッサの推奨事項 (続き)

SPARC システム	サイズ
スワップ領域	デフォルトのサイズは 512M バイトです。 注 – スワップ空間は必要に応じてカスタマイズできます。スワップ空間のサイズは、システムのハードディスクのサイズに基づいて決定されます。
プロセッサ要件	200 MHz 以上のプロセッサが必要です。

表 3-2 x86: メモリー、スワップ、およびプロセッサの推奨事項

x86 システム	サイズ
インストールまたはアップグレードするために必要なメモリー	■ Solaris 10 1/06 以降のリリースでは、推奨サイズは 512M バイトです。最小サイズは 256M バイトです。 ■ Solaris 10 3/05 リリースでは、推奨サイズは 256M バイトです。最小サイズは 128M バイトです。 注 – オプションのインストール機能の中には、十分な量のメモリーがないと使用できないものがあります。たとえば、メモリーが足りないシステムに DVD からインストールする場合、Solaris インストールプログラムの GUI ではなく、テキストインストーラを使用する必要があります。これらのメモリー要件については、表 3-3 を参照してください。
スワップ領域	デフォルトのサイズは 512M バイトです。 注 – スワップ空間は必要に応じてカスタマイズできます。スワップ空間のサイズは、システムのハードディスクのサイズに基づいて決定されます。
プロセッサ要件	120 MHz 以上のプロセッサを推奨します。ハードウェアによる浮動小数点サポートが必要です。

ソフトウェアのインストールに GUI を使用するか、ウィンドウ表示環境を使用するか、またはウィンドウ表示環境を使用しないかを選択できます。十分なメモリーがある場合は、デフォルトで GUI が表示されます。GUI を表示するためのメモリーが不足している場合は、デフォルトでその他の環境が表示されます。ブートオプション `nowin` または `text` を使用すると、デフォルト動作を変更できます。しかし、システム内のメモリー容量による制約、またはリモートインストールによる制約があります。また、ビデオアダプタが検出されない場合、Solaris インストールプログラムは、自動的にコンソールベースの環境で表示されます。表 3-3 に、これらの環境の説明と、環境を表示するための最小メモリー要件の一覧を示します。

表 3-3 SPARC: 表示オプションごとのメモリー要件

SPARC: メモリー	インストールの種類	説明
128 ~ 383M バイト	テキストベース	グラフィックスは含まれませんが、ウィンドウを1つ提供します。また、その他のウィンドウを開くこともできます。 text ブートオプションを使用している場合でシステムに十分なメモリーがあるときは、ウィンドウ表示環境でインストールされます。tip ラインを介してリモートでインストールする場合や、nowin ブートオプションを使用してインストールする場合は、コンソールベースのインストールに限定されます。
384M バイト以上	GUI ベース	ウィンドウ、プルダウンメニュー、ボタン、スクロールバー、およびアイコンが提供されます。

表 3-4 x86: 表示オプションごとのメモリー要件

x86: メモリー	インストールの種類	説明
■ Solaris 10 1/06 以降のリリース: 256 ~ 511M バイト ■ Solaris 10 3/05 リリース: 128 ~ 383M バイト	テキストベース	グラフィックスは含まれませんが、ウィンドウを1つ提供します。また、その他のウィンドウを開くこともできます。 text ブートオプションを使用してインストールする場合、十分なメモリーがあれば、ウィンドウ表示環境になります。tip ラインを介してリモートでインストールする場合や、nowin ブートオプションを使用してインストールする場合は、コンソールベースのインストールに限定されます。
■ Solaris 10 1/06 以降のリリース: 512M バイト ■ Solaris 10 3/05 リリース: 384M バイト	GUI ベース	ウィンドウ、プルダウンメニュー、ボタン、スクロールバー、およびアイコンが提供されます。

ディスク容量とスワップ空間の割り当て

Solaris ソフトウェアをインストールする前に、ディスク容量の計画を立てて、システムに十分なディスク容量があるかどうかを調べることができます。

ディスク容量に関する一般的な計画と推奨事項

ディスク容量の計画のたて方は、ユーザーによって異なります。必要に応じて、次の条件で容量を割り当てます。

表 3-5 ディスク容量とスワップ空間に関する一般的な計画

容量の割り当ての条件	説明
ファイルシステム	ファイルシステムを割り当てる場合には、将来の Solaris パージョンにアップグレードするときのために、現在必要な容量よりも 30% 多く割り当ててください。 デフォルトでは、ルート(/) とスワップ領域 (/swap) だけが作成されます。OS サービスのためにディスク容量が割り当てられたときは、/export ディレクトリも作成されます。Solaris のメジャーリリースにアップグレードする場合は、システムスライスを切り直す必要や、インストールに要する容量の 2 倍を割り当てて必要が生じる可能性があります。Update リリースにアップグレードする場合、将来のアップグレードに備えて余分のディスク容量を割り当てておけば、システムのスライスを切り直す手間を軽減できます。Solaris Update リリースにアップグレードするたびに、直前のリリースに比べておよそ 10% のディスク容量が追加で必要になります。何回分かの Solaris Update リリースを見込んで、各ファイルシステムにそれぞれ 30% ずつ多めにディスク容量を割り当てることができます。
/var ファイルシステム	クラッシュダンプ機能 savecore (1M) を使用する場合は、/var ファイルシステムの容量を物理メモリーの倍のサイズに設定します。
スワップ	Solaris インストールプログラムは、次の場合、デフォルトのスワップ領域 512M バイトを割り当てます。 ■ インストールプログラムのディスクスライス自動配置機能を使用する場合 ■ 手動でスワップスライスのサイズを変更したくない場合 デフォルトでは、Solaris インストールプログラムは、最初の使用可能なディスクシリンダ (通常、SPARC システムではシリンダ 0) からスワップ空間を割り当てます。この配置によって、デフォルトのディスクレイアウト時にはルート (/) ファイルシステムに最大のスワップ空間を割り当てることができ、アップグレード時にはルート (/) ファイルシステムを拡張できます。 将来スワップ領域を拡張することを考えている場合、次のいずれかの手順を実行してスワップスライスを配置することにより、別のディスクシリンダでスワップスライスを開始できます。 ■ Solaris インストールプログラムの場合、シリンダモードでディスクレイアウトをカスタマイズして、スワップスライスを目的の位置に手動で割り当てることができます。 ■ カスタム JumpStart インストールプログラムの場合、スワップスライスをプロファイルファイル内に割り当てることができます。JumpStart プロファイルの詳細は、96 ページの「プロファイルの作成」を参照してください。 スワップ領域の概要については、『Solaris のシステム管理 (デバイスとファイルシステム)』の第 21 章「追加スワップ空間の構成 (手順)」を参照してください。

表 3-5 ディスク容量とスワップ空間に関する一般的な計画 (続き)

容量の割り当ての条件	説明
ホームディレクトリファイルシステムを提供しているサーバー	ホームディレクトリは、通常デフォルトで /export ファイルシステムにあります。
インストールする Solaris ソフトウェアグループ	ソフトウェアグループとは、ソフトウェアパッケージの集まりのことです。ディスク容量を計画する際には、選択したソフトウェアグループから個々のソフトウェアパッケージを個別に追加したり削除したりできることを覚えておいてください。ソフトウェアグループの詳細については、 41 ページの「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。
アップグレード	■ Solaris Live Upgrade を使用して非アクティブブート環境をアップグレードする場合、ディスク容量の計画については、『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「Solaris Live Upgrade のディスク容量の要件」を参照してください ■ その他の Solaris インストール方法を使ってディスク容量計画を行う場合は、46 ページの「ディスク容量の再配置を伴うアップグレード」を参照してください。
言語サポート	たとえば、中国語、日本語、韓国語などです。単一の言語をインストールする場合は、約 0.7G バイトのディスク容量をその言語用に追加して割り当ててください。すべての言語サポートをインストールする場合は、インストールするソフトウェアグループに応じて、最大で約 2.5G バイトのディスク容量を言語サポート用に追加して割り当てる必要があります。
印刷やメールのサポート	追加の容量を割り当てます。
追加ソフトウェアや Sun 以外のソフトウェア	追加の容量を割り当てます。

ソフトウェアグループごとの推奨ディスク容量

Solaris ソフトウェアグループは Solaris パッケージの集まりです。それぞれのソフトウェアグループには、異なる機能やハードウェアドライバのサポートが含まれています。

- 初期インストールの場合、システム上で実行する機能に基づいて、インストールするソフトウェアグループを選択します。
- アップグレードの場合、システムにすでにインストールされているソフトウェアグループをアップグレードする必要があります。たとえば、システムに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がインストールされている場合には、開発者システムサポートソフトウェアグループにアップグレードするオプションはありません。ただし、アップグレード中に、インストール済みのソフトウェアグループに属していないソフトウェアをシステムに追加することはできます。

Solaris ソフトウェアのインストール時には、選択した Solaris ソフトウェアグループに対してパッケージを追加したり、削除したりすることができます。追加や削除するパッケージを選択する際には、ソフトウェアの依存関係や Solaris ソフトウェアがどのようにパッケージ化されているかを知っている必要があります。

次に、ソフトウェアパッケージのグループの成り立ちを図示します。限定ネットワークサポートソフトウェアグループには、最小限のパッケージだけが含まれています。これに対して、全体ディストリビューションと OEM サポートソフトウェアグループには、すべてのパッケージが含まれています。

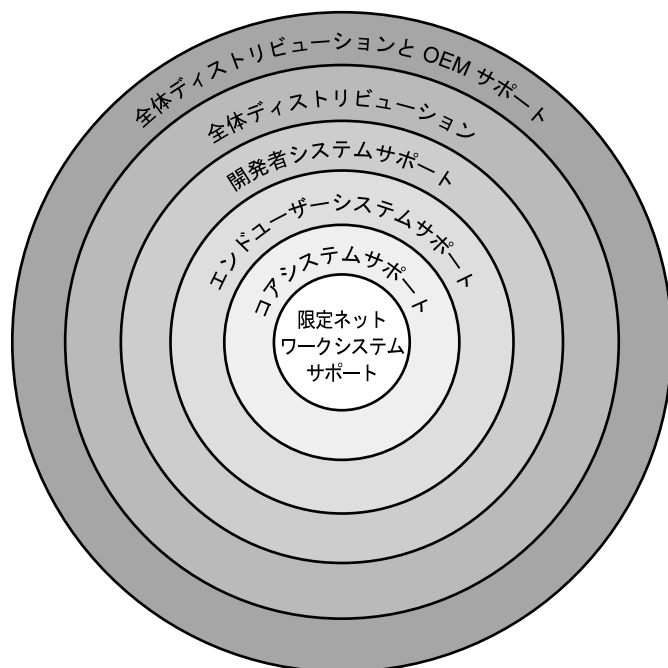


図 3-1 Solaris ソフトウェアグループ

表 3-6 に、Solaris ソフトウェアグループと、各グループのインストールに必要な推奨ディスク容量を示します。

注 – 表 3-6 に示す推奨ディスク容量には、次の項目のための容量が含まれています。

- スワップ領域
- パッチ
- 追加のソフトウェアパッケージ

ソフトウェアグループに必要なディスク容量が、この表に一覧表示されている容量より少ない場合もあります。

表 3-6 ソフトウェアグループごとの推奨ディスク容量

ソフトウェアグループ	説明	推奨ディスク容量
全体ディストリビューションと OEM サポート	全体ディストリビューションソフトウェアグループに、ハードウェアドライバが追加されたパッケージです。ここには、インストール時にシステムに存在していないハードウェアのドライバも含まれます。	6.8G バイト
全体ディストリビューション	開発者システムサポートソフトウェアグループに、サーバー用のソフトウェアが追加されたパッケージです。	6.7G バイト
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に、ソフトウェア開発のサポートが追加されたパッケージです。ソフトウェア開発のサポートとして、ライブラリ、インクルードファイル、マニュアルページ、プログラミングツールなどが追加されています。ただし、コンパイラは含まれていません。	6.6G バイト
エンドユーザーシステムサポート	ネットワークに接続された Solaris システムと共通デスクトップ環境 (CDE) の起動と実行に必要な最小限のコードを提供するパッケージです。	5.3G バイト
コアシステムサポート	ネットワークに接続された Solaris システムの起動と実行に必要な最小限のコードを提供するパッケージです。	2.0G バイト
限定ネットワークシステムサポート	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) の起動と実行に必要な最小限のコードを提供するパッケージです。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。	2.0G バイト

アップグレード

システムのアップグレードには、3 つの方法があります。Solaris Live Upgrade、Solaris インストールプログラム、およびカスタム JumpStart です。

表 3-7 Solaris のアップグレード方法

現在の Solaris OS	Solaris のアップグレード方法
Solaris 8、Solaris 9、および Solaris 10	■ Solaris Live Upgrade – 稼働中のシステムのコピーを作成し、それをアップグレードすることでシステムをアップグレードします ■ 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フェースまたはコマンド行インタフェースを使用して対話形式でアップグレードを行うことができます ■ カスタム JumpStart – 自動アップグレードを行うことができます

アップグレードに関する制限事項

問題	説明
別のソフトウェアグループへのアップグレード	システムのソフトウェアグループを、アップグレード時に別のソフトウェアグループに変更することはできません。たとえば、システムに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がインストールされている場合には、開発者システムサポートソフトウェアグループにアップグレードするオプションはありません。ただし、アップグレード中に、インストール済みのソフトウェアグループに属していないソフトウェアをシステムに追加することはできます。
Solaris 10 1/06 以降のリリース: 非大域ゾーンがインストールされている場合のアップグレード	Solaris OS をアップグレードするときに、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。Solaris 対話式インストールプログラムおよびカスタム JumpStart プログラムを使用するとアップグレードが可能です。アップグレードする場合の制限については、 52 ページの「非大域ゾーンがインストールされている場合のアップグレード」 を参照してください。

アップグレードプログラム

標準的な対話式アップグレードには Solaris インストールプログラム、無人アップグレードにはカスタム JumpStart インストールを使用します。Solaris Live Upgrade では、実行中のシステムをアップグレードできます。

アップグレードプログラム	説明	参照先
Solaris Live Upgrade	現在実行中のシステムのコピーを作成できます。このコピーをアップグレードし、再起動します。すると、このアップグレード済みコピーが現在実行中のシステムと切り替わります。Solaris Live Upgrade を使用すると、Solaris OS のアップグレード時のダウンタイムを短縮できます。Solaris Live Upgrade では、アップグレードに伴うさまざまな問題を防ぐこともできます。たとえば、アップグレード中に電源障害が発生すると、アップグレードを回復することができなくなります。しかし、Solaris Live Upgrade では、現在実行中のシステムではなく、コピーをアップグレードするので、この問題は起こりません。	Solaris Live Upgrade を使用する場合はディスク容量の割り当て計画については、『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「Solaris Live Upgrade の要件」を参照してください。
Solaris インストールプログラム	対話式 GUI のガイドに従ってアップグレードを実行できます。	『Solaris 10 インストールガイド (基本編)』の第 2 章「Solaris インストールプログラムによるインストール (作業)」。
カスタム JumpStart プログラム	自動アップグレードを行います。必要な情報は、プロファイルファイルと、オプションのプリインストールスクリプトならびにポストインストールスクリプトによって提供されます。アップグレード用にカスタム JumpStart プロファイルを作成するときは、install type upgrade を指定します。さらに、実際にアップグレードを行う前に、システムの現在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、カスタム JumpStart プロファイルが目的どおりのことを実行しようとしているかを確認する必要があります。アップグレードしようとしているシステム上で、pfinstall - D コマンドを実行して、プロファイル进行测试します。ディスク構成ファイルを使用してアップグレード用プロファイル进行测试することはできません。	■ アップグレードオプションのテスト方法については、110 ページの「プロファイルのテスト」を参照してください ■ アップグレードプロファイルの作成方法については、98 ページの「プロファイルの例」を参照してください ■ アップグレードの実行方法については、141 ページの「SPARC: カスタム JumpStart インストールの実行」を参照してください

アップグレードではなく Solaris フラッシュアーカイブをインストール

Solaris フラッシュのインストール機能では、マスターシステムからインストール全体のコピーを作成し、これを多数のクローンシステムに複製できます。このコピーを「Solaris フラッシュアーカイブ」と呼びます。アーカイブは、任意のインストールプログラムでインストールできます。



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

アーカイブのインストールについては、次の表を参照してください。

Solaris Live Upgrade	『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「ブート環境への Solaris フラッシュアーカイブのインストール」
カスタム JumpStart	141 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」
Solaris 対話式インストール	『Solaris 10 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 4 章「Solaris フラッシュアーカイブのインストールと管理 (作業)」
WAN ブートインストール	『Solaris 10 インストールガイド (ネットワークインストール)』の第 14 章「WAN ブートによるインストール (作業)」

ディスク容量の再配置を伴うアップグレード

Solaris インストールプログラムのアップグレードオプションとカスタム JumpStart プログラムの upgrade キーワードはどちらも、ディスク容量の再配置機能を提供します。この再配置により、ディスクスライスのサイズが自動的に変更されます。アップグレードするのに十分な容量が現在のファイルシステムにない場合、ディスク容量を割り当て直すことができます。たとえば、アップグレードに伴ってファイルシステムの容量を増やす必要があるのは、次のような場合です。

- 新しいリリースで、システムに現在インストールされている Solaris ソフトウェアグループに新たにソフトウェアが追加されている。ソフトウェアグループに新しいソフトウェアが含まれている場合は、アップグレード時に自動的に選択されてインストールされます。
- 新しいリリースで、システム上の既存のソフトウェアのサイズが大きくなった。

自動配置機能を使用すると、ファイルシステムに必要な容量を確保するようにディスク容量の再配置が行われます。自動配置機能では、デフォルトの制約にもとづいて容量の再配置が試みられます。このため、この機能によって容量の再配置が行われない場合は、ファイルシステムの制約を変更する必要があります。

注 – 自動配置機能では、ファイルシステムの「サイズを大きくする」ことはできません。自動配置機能は、次のように容量を割り当て直します。

1. 変更する必要があるファイルシステム上の必要なファイルのバックアップを作成する。
2. ファイルシステムの変更内容に基づいて、ディスクパーティションを割り当て直す。
3. アップグレード前にバックアップファイルを復元する。

-
- Solaris インストールプログラムの自動配置機能がディスク容量をどのように再配置するかを決定できない場合は、カスタム JumpStart プログラムを使用してアップグレードを行う必要があります。
 - カスタム JumpStart によるアップグレードでアップグレード用プロファイルを作成する際に、ディスク容量が問題になることがあります。現在のファイルシステムのディスク容量が不足していてアップグレードできない場合は、キーワード `backup_media` および `layout_constraint` を使ってディスク容量を割り当て直すことができます。キーワード `backup_media` と `layout_constraint` の使用例については、98 ページの「プロファイルの例」を参照してください。

アップグレード前のシステムのバックアップ

Solaris OS へのアップグレードを実行する前に、既存のファイルシステムのバックアップを作成しておくことを強くお勧めします。ファイルシステムをテープなどのリムーバブルメディアにコピーすれば、データの損失や損傷、破壊などを防止できます。システムのバックアップの詳しい作成方法については、『Solaris のシステム管理 (デバイスとファイルシステム)』の第 24 章「ファイルシステムのバックアップと復元 (概要)」を参照してください。

システムで稼働中の Solaris OS のバージョンを確認する方法

システムで稼働中の Solaris ソフトウェアのバージョンを確認するには、次のいずれかのコマンドを入力します。

```
$ uname -a
```

cat コマンドを使用すると、より詳しい情報を得られます。

```
$ cat /etc/release
```

ロケールの値

インストールの一環として、システムで使用するロケールを事前構成できます。「ロケール」によって、オンライン情報を特定の言語と地域で表示する方法が決まります。また、日付と時間の表記、数字や通貨、綴りなどの地域的差異を表すために、1つの言語に対して複数のロケールが存在することもあります。

システムのロケールを事前構成するには、カスタム JumpStart プロファイルか sysidcfg ファイルを使用します。

プロファイルを使ったロケールの設定	96 ページの「プロファイルの作成」
sysidcfg ファイルを使ったロケールの設定	『Solaris 10 インストールガイド (ネットワークインストール)』の「sysidcfg ファイルによる事前設定」
ロケール値のリスト	『国際化対応言語環境の利用ガイド』

プラットフォーム名とグループ

ネットワークインストール用にクライアントを追加する際は、システムアーキテチャー (プラットフォームグループ) の情報が必要です。カスタム JumpStart インストールの rules ファイルを作成する場合は、プラットフォーム名が必要になります。

プラットフォーム名とプラットフォームグループの例を下記の表に示します。SPARC システムの完全な一覧については、『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。

表 3-8 プラットフォーム名とプラットフォームグループの例

システム	プラットフォーム名	プラットフォームグループ
Sun Fire	T2000	sun4v
Sun Blade™	SUNW,Sun-Blade-100	sun4u

表 3-8 プラットフォーム名とプラットフォームグループの例 (続き)

システム	プラットフォーム名	プラットフォームグループ
x86 ベース	i86pc	i86pc

注 – システムが動作している場合、システムのプラットフォーム名は `uname -i` コマンドで、システムのプラットフォームグループは `uname -m` コマンドで、それぞれ調べることができます。

ゾーンのインストールと構成

ここでは、大域ゾーンと非大域ゾーンの大まかな計画について紹介します。概要と計画情報、および具体的な手順については、『Solaris のシステム管理 (Solaris コンテナ : 資源管理と Solaris ゾーン)』の第 16 章「Solaris ゾーンの紹介」を参照してください。

Solaris ゾーン区分技術 (概要)

Solaris OS をインストールした後、ゾーンをインストールして構成できます。大域ゾーンは稼働中のオペレーティングシステムの単一インスタンスであり、各 Solaris システムに 1 つ含まれています。大域ゾーンは、システムのデフォルトのゾーンであり、システム全体の管理に使用されるゾーンでもあります。非大域ゾーンは仮想化されたオペレーティングシステム環境です。

Solaris ゾーンはソフトウェア区分技術で、オペレーティングシステムサービスを仮想化し、安全で隔離されたアプリケーション実行環境を提供します。ゾーンを作成すると、そのアプリケーション実行環境で実行されるプロセスは、ほかのゾーンから隔離されます。このように隔離されているので、あるゾーンで実行中のプロセスが、ほかのゾーンで実行中のプロセスから監視または操作されることがありません。非大域ゾーンで実行中のプロセスは、スーパーユーザー資格で実行されていても、ほかのゾーンの活動を監視したり操作したりすることはできません。大域ゾーンでスーパーユーザー資格で実行されるプロセスは、任意のゾーンの任意のプロセスを操作できます。

大域ゾーンと非大域ゾーンについて

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。システムハードウェアから起動できるのは、大域ゾーンだけです。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、ほかのどのゾーンに関連付けられているオブジェクトにもアクセスできます。次の表に、大域ゾーンと非大域ゾーンの特性をまとめます。

大域ゾーン	非大域ゾーン
システムによって ID 0 が割り当てられます	ゾーンの起動時にシステムによってゾーン ID が割り当てられます
システムで起動され実行される Solaris カーネルの単一のインスタンスを提供します	大域ゾーンから起動される Solaris カーネルの下で処理を共有します
Solaris システムソフトウェアパッケージの完全なインストールが含まれています	Solaris オペレーティングシステムソフトウェアパッケージの完全なインストールの一部が含まれています
追加のソフトウェアパッケージや、パッケージを通してインストールされない追加のソフトウェア、ディレクトリ、ファイル、その他のデータが含まれている場合もあります	大域ゾーンから共有された Solaris ソフトウェアパッケージが含まれています
大域ゾーンにインストールされているすべてのソフトウェア構成要素に関する情報を格納した、一貫性のある完全な製品データベースを提供します	大域ゾーンから共有されたものではない、インストールされた追加のソフトウェアパッケージが含まれていることもあります 追加のソフトウェア、ディレクトリ、ファイル、非大域ゾーンで作成されたその他のデータなど、パッケージを通してインストールされないもの、あるいは、大域ゾーンからの共有でないものが含まれていることもあります
大域ゾーンのホスト名やファイルシステムテーブルなど、大域ゾーンのものに固有の構成情報を保持します	非大域ゾーンのホスト名やファイルシステムテーブルなど、その非大域ゾーンのものに固有の構成情報を保持します
すべてのデバイスとすべてのファイルシステムが認識される、唯一のゾーンです	ゾーンにインストールされているすべてのソフトウェア構成要素に関する情報を格納した、一貫性のある完全な製品データベースを持ちます。非大域ゾーンに置かれている構成要素と、読み取り専用モードで大域ゾーンから共有される構成要素があります
非大域ゾーンの存在と構成が認識される、唯一のゾーンです	ほかのゾーンの存在を認識できません
非大域ゾーンの構成、インストール、管理、またはアンインストールを行うことができます、唯一のゾーンです	自身を含め、ゾーンのインストール、管理、アンインストールを行うことはできません

詳細については、以下を参照してください。

- 『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」
- [49 ページの「ゾーンのインストールと構成」](#)

Solaris ゾーン (計画)

Solaris OS をインストールしたあと、ゾーンをインストールして構成できます。大域ゾーンは稼働中のオペレーティングシステムの単一インスタンスであり、各 Solaris システムに 1 つ含まれています。大域ゾーンは、システムのデフォルトのゾーンであり、システム全体の管理に使用されるゾーンでもあります。非大域ゾーンは仮想化されたオペレーティングシステム環境です。



注意 - 次の条件がいずれも成立する場合は、コマンドに `-R` オプションまたは同等のオプションを使用して代替ルート (`/`) ファイルシステムを指定してはいけません。

- コマンドが大域ゾーン内で実行される。
- 代替ルート (`/`) ファイルシステムが非大域ゾーンにあるすべてのパスを参照する。

たとえば、`pkgadd` ユーティリティに `-R root_path` オプションで非大域ゾーンのルート (`/`) ファイルシステムへのパスを指定して、大域ゾーンから実行する場合があります。

代替ルート (`/`) ファイルシステムが指定可能なユーティリティの一覧およびゾーンの詳細については、『Solaris のシステム管理 (Solaris コンテナ: 資源管理と Solaris ゾーン)』の「大域ゾーンから非大域ゾーンにアクセスする際の制限」を参照してください。

非大域ゾーンを使用している場合のインストールとアップグレード

Solaris OS がインストールされている場合、大域ゾーンにインストールされたソフトウェアグループのパッケージはすべての非大域ゾーンで共有されます。たとえば、全体ディストリビューションをインストールした場合、これらのパッケージはすべてのゾーンに含まれます。デフォルトでは、大域ゾーンに追加パッケージをインストールすると、これらも非大域ゾーンに追加されます。アプリケーション、ネームスペース、サーバー、NFS や DHCP などのネットワーク接続、その他のソフトウェアを、非大域ゾーンに分離できます。それぞれの非大域ゾーンは、ほかの非大域ゾーンを認識せず、それぞれ独立して動作可能です。たとえば、大域ゾーンに全体ディストリビューションをインストールし、個々の非大域ゾーンでは Java Enterprise System Messaging Server、データベース、DHCP、および Web サーバーを実行することができます。非大域ゾーンをインストールするときは、各ゾーンで実行されるアプリケーションのパフォーマンス要件に留意してください。



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
-

非大域ゾーンがインストールされている場合のアップグレード

Solaris 10 1/06 以降のリリースでは、Solaris OS をアップグレードするときに、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。Solaris 対話式インストールプログラムとカスタム JumpStart プログラムにより、アップグレードが可能です。

- Solaris 対話式インストールプログラムで、「「アップグレード」または「初期」インストールの選択」パネルで、「アップグレードインストール」を選択して、非大域ゾーンにあるシステムをアップグレードできます。インストールプログラムがシステムを分析してアップグレード可能かどうかを判定し、分析の概要を表示します。その後インストールプログラムは、アップグレードの続行を求めるプロンプトを表示します。このプログラムを使用する場合には、次の制限事項があります。
 - アップグレードのカスタマイズはできません。たとえば、追加のソフトウェア製品のインストール、追加のロケールパッケージのインストール、またはディスクレイアウトの変更を行うことはできません。
 - Solaris Operating System DVD または DVD に作成したネットワークインストールイメージを使用する必要があります。Solaris SOFTWARE CD を使用してシステムをアップグレードすることはできません。このプログラムによるインストール方法については、『Solaris 10 インストールガイド (基本編)』の第 2 章「Solaris インストールプログラムによるインストール (作業)」を参照してください。
- カスタム JumpStart インストールプログラムでは、アップグレードに使用できるキーワードは `install_type` および `root_device` のみです。

いくつかのキーワードは非大域ゾーンに影響を与えるため、これらのキーワードはプロファイルに含められません。たとえば、パッケージを追加するキーワード、ディスク容量を再配置するキーワード、またはロケールを追加するキーワードを使用すると非大域ゾーンに影響を与えます。これらのキーワードを使用すると、キーワードが無視されるか JumpStart アップグレードが失敗します。これらのキーワードの一覧については、[219 ページの「非大域ゾーンでアップグレードする際のプロファイルキーワードの制限」](#)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場合は、Solaris Live Upgrade を使用してシステムをアップグレードすることはできません。lucreate コマンドでブート環境を作成できますが、luupgrade コマンドを使用するとアップグレードに失敗します。エラーメッセージが表示されます。

非大域ゾーンのディスク容量要件

大域ゾーンをインストールするときには、作成するすべてのゾーンに十分なディスク容量を必ず確保してください。非大域ゾーンごとに、ディスク容量要件は異なる場合があります。ここでは、計画情報の概略について説明します。計画の要件と推奨事項の詳細は、『Solaris のシステム管理 (Solaris コンテナ : 資源管理と Solaris ゾーン)』の第 18 章「非大域ゾーンの計画と構成 (手順)」を参照してください。

1 つのゾーンで消費できるディスク容量に制限はありません。容量制限は大域ゾーンの管理者が行います。単一プロセッサの小規模なシステムでも、多数のゾーンを同時に実行できます。

非大域ゾーンを作成するときの容量要件は、大域ゾーンにインストールされたパッケージの種類によって異なります。パッケージ数と容量要件がその要因になります。ディスク容量に関する一般的なガイドラインを次に示します。

- 大域ゾーンに Solaris の標準パッケージをすべてインストールした場合は、ディスクに約 100M バイトの空き容量を確保することをお勧めします。大域ゾーンに追加パッケージをインストールした場合は、この量を増やしてください。デフォルトでは、大域ゾーンに追加パッケージをインストールすると、これらも非大域ゾーンに追加されます。非大域ゾーンでこれらの追加パッケージが格納されるディレクトリは、inherit-pkg-dir リソースで指定されます。
- システムに十分なスワップ空間がある場合は、1 つのゾーン当たり 40M バイトの RAM を追加します。各ゾーンが正常に稼働できるように、この追加をお勧めします。システムサイズを計画するときは、この RAM の追加を考慮してください。

SPARC: 64 ビットパッケージに関する変更点

Solaris OS の以前のリリースでは、32 ビットコンポーネントと 64 ビットコンポーネントがそれぞれ別のパッケージで提供されていました。Solaris 10 OS ではパッケージが簡略化され、32 ビットコンポーネントと 64 ビットコンポーネントのほとんどが単一のパッケージで提供されています。統合されたパッケージでは、元の 32 ビット

パッケージの名前がそのまま使用されています。64 ビットパッケージは提供されなくなりました。この変更によってパッケージ数が減り、インストールが簡単になります。このため、カスタム JumpStart スクリプトやほかのパッケージインストールスクリプトで 64 ビットパッケージを参照している場合は、これらのスクリプトを変更して参照を削除する必要があります。

64 ビットパッケージの名前は、次の規則に従って変更されます。

- 64 ビットパッケージに対応する 32 ビットパッケージがある場合は、その 32 ビットパッケージの名前が 64 ビットパッケージに付けられます。たとえば、`/usr/lib/sparcv9/libc.so.1` などの 64 ビットライブラリの場合、以前は SUNWcslx で提供されていましたが、現在は SUNWcsl で提供されます。64 ビットの SUNWcslx パッケージは提供されなくなりました。
- 対応する 32 ビットパッケージが存在しない場合は、名前から接尾辞「x」が削除されます。たとえば、SUNW1394x は SUNW1394 になります。

x86: パーティション分割の推奨事項

x86 システムで Solaris OS を使用する場合、次のガイドラインに従ってシステムのパーティション分割を行ってください。

Solaris インストールプログラムは、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使用します。これらのパーティションを「fdisk パーティション」と呼びます。fdisk パーティションは、x86 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です。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。

表 3-9 x86: デフォルトパーティション

パーティション	パーティション名	パーティションサイズ
1 番目のパーティション (一部のシステムのみ)	診断・サービスパーティション	システム上の既存のサイズ

表 3-9 x86: デフォルトパーティション (続き)

パーティション	パーティション名	パーティションサイズ
2 番目のパーティション (一部のシステムのみ)	x86 ブートパーティション	Solaris 10 3/05 リリースの場合: x86 ブートパーティションが作成されます。サイズはシステムの既存サイズです。 Solaris 10 1/06 以降のリリースでは、次のようになります。 ■ 初期インストールの場合は、このパーティションは作成されません。 ■ アップグレードするときに、使用しているシステムに既存の x86 ブートパーティションがない場合は、このパーティションは作成されません。 ■ アップグレードするときに、システムに x86 ブートパーティションがある場合: ■ あるブートデバイスから別のブートデバイスにブートストラップするためにパーティションが必要な場合は、x86 ブートパーティションがシステムに保持されます。 ■ 追加のブートデバイスのブートにパーティションが不要な場合は、x86 ブートパーティションは削除されます。パーティションの内容は、ルートパーティションに移されます。
3 番目のパーティション	Solaris OS パーティション	ブートディスク上に残っている容量

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でサービスパーティションを保持

Solaris インストールプログラムは、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使って、診断・サービスパーティションに対応することができます。システムに診断・サービスパーティションが含まれている場合、デフォルトブートディスクパーティションレイアウトを使用して、このパーティションを保持できます。

注 – 診断・サービスパーティションが含まれない x86 システムに Solaris OS をインストールする場合、デフォルトでは、新しい診断・サービスパーティションは生成されません。システムに診断・サービスパーティションを作成する場合は、ハードウェアのマニュアルを参照してください。

第 4 章

x86: Solaris インストールのための GRUB ベースのブート

この章では、Solaris インストールに関連する、x86 システムでの GRUB ベースのブートについて説明します。この章の内容は次のとおりです。

- 57 ページの「x86: GRUB ベースのブート (概要)」
- 60 ページの「x86: GRUB ベースのブート (計画)」
- 65 ページの「x86: GRUB メニューの menu.lst ファイルの検出 (作業)」

x86: GRUB ベースのブート (概要)

Solaris 10 1/06 以降のリリースでは、オープンソースのブートローダーである GRUB が、Solaris OS のデフォルトのブートローダーとして採用されています。

注 – GRUB ベースのブートは、SPARC システムでは使用できません。

ブートローダーは、システムの電源を入れたあと最初に実行されるソフトウェアプログラムです。x86 システムの電源を入れると、BIOS (Basic Input/Output System) により、CPU、メモリー、およびプラットフォームハードウェアが初期化されます。初期化フェーズが完了すると、BIOS が構成済みブートデバイスからブートローダーをロードし、システムの制御をブートローダーに移します。

GRUB 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、メニューには構成ファイルに定義されたブートオプションが表示されます。また、GRUB はコマンド行インタフェースも備えており、メニューインタフェースからアクセスしてさまざまなブートコマンドを実行できます。Solaris OS では、GRUB 実装はマルチブート仕様に準拠しています。マルチブート仕様について詳細は、<http://www.gnu.org/software/grub/grub.html> を参照してください。

Solaris カーネルはマルチブート仕様に完全に準拠しているため、GRUB を使用して Solaris x86 システムをブートできます。GRUB を使用すると、さまざまなオペレーティングシステムのブートおよびインストールがより簡単にできます。たとえば、1 つのシステムで次のオペレーティングシステムを別々にブートできます。

- Solaris OS
- Microsoft Windows

注 – GRUB は Microsoft Windows のパーティションを検出しますが、OS がブート可能かどうかは確認しません。

GRUB の主な利点は、ファイルシステムおよびカーネル実行可能ファイルの形式に対して直観的であるため、ディスク上のカーネルの物理的位置を記録せずにオペレーティングシステムをロードできることです。GRUB ベースのブートでは、カーネルのファイル名、ドライブ、およびカーネルがあるパーティションを指定することでカーネルがロードされます。GRUB ベースのブートは Solaris Device Configuration Assistant (デバイス構成用補助) を置き換え、GRUB メニューによってブート処理を簡略化します。

x86: GRUB ベースのブートの動作

GRUB がシステムの制御を取得すると、コンソールにメニューが表示されます。GRUB メニューでは次の操作を実行できます。

- エントリを選択してシステムをブートします
- 組み込まれている GRUB 編集メニューを使用してブートエントリを変更します
- コマンド行から手動で OS カーネルをロードします

デフォルトで登録されている OS ブートでは、設定可能なタイムアウトを利用することができます。いずれかのキーを押すと、デフォルトの OS エントリのブートが中止されます。

GRUB メニューの例については、61 ページの「GRUB メインメニューについて」を参照してください。

x86: GRUB デバイス命名規則

GRUB が使用するデバイス命名規則は、以前の Solaris OS バージョンの場合と若干異なっています。GRUB デバイス命名規則を理解すると、使用しているシステムで GRUB を構成するときに、ドライブとパーティションの情報を正しく指定できます。

次の表に、GRUB デバイス命名規則を示します。

表 4-1 GRUB デバイスの命名規則

デバイス名	説明
(fd0), (fd1)	最初のプロッピーディスク、2 番目のプロッピーディスク
(nd)	ネットワークデバイス
(hd0, 0), (hd0, 1)	最初の bios ディスクの 1 番目と 2 番目の fdisk パーティション
(hd0, 0, a), (hd0, 0, b)	最初の bios ディスクの 1 番目の fdisk パーティションの Solaris/BSD スライス 0 および 1

注 – GRUB デバイス名はすべて括弧で囲む必要があります。パーティション番号は、1 からではなく 0 (ゼロ) から数えます。

fdisk パーティションの詳細については、『Solaris のシステム管理 (デバイスとファイルシステム)』の「fdisk パーティションの作成上のガイドライン」を参照してください。

x86: GRUB ベースのインストールについての情報の参照先

変更内容の詳細については、次の関連情報を参照してください。

表 4-2 GRUB ベースのインストールについての情報の参照先

トピック	GRUB メニューでの作業	参照先
インストール	Solaris OS CD または DVD メディアからインストールする方法	『Solaris 10 インストールガイド (基本編)』
	ネットワークインストールイメージからインストールする方法	『Solaris 10 インストールガイド (ネットワークインストール)』のパート II 「ローカルエリアネットワーク経由のインストール」
	ネットワークインストールのために DHCP サーバーを構成する方法	『Solaris 10 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)」

表 4-2 GRUB ベースのインストールについての情報の参照先 (続き)

トピック	GRUB メニューでの作業	参照先
	カスタム JumpStart プログラムでインストールする方法	147 ページの「x86: カスタム JumpStart インストールの実行」
	Solaris Live Upgrade を使用してブート環境のアクティブ化またはフォールバックを行う方法	■ 『Solaris 10 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」 ■ 『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「障害回復: 元のブート環境へのフォールバック (コマンド行インタフェース)」
システム管理	GRUB および管理作業についての詳細	『Solaris のシステム管理 (基本編)』の第 11 章「GRUB ベースのブート (手順)」

x86: GRUB ベースのブート (計画)

この節では、GRUB ベースのブートの基本と、GRUB メニューについて説明します。

Solaris OS のインストール時に、デフォルトで 2 つの GRUB メニューエントリがシステムにインストールされます。最初のエントリは Solaris OS エントリです。2 番目のエントリはフェイルセーフブートアーカイブで、システムの回復に使用されます。

Solaris GRUB メニューエントリは、Solaris ソフトウェアのインストールおよびアップグレード処理の一環として自動的にインストールおよびアップグレードされます。これらのエントリは OS によって直接管理されるため、手動で編集しないでください。

Solaris OS の標準インストール中に、システム BIOS の設定を変更せずに GRUB が Solaris fdisk パーティションにインストールされます。この OS が BIOS ブートディスクにない場合は、次のいずれかの操作を行う必要があります。

- BIOS の設定を変更します。
- ブートマネージャーを使用して Solaris パーティションにブートストラップするようにします。詳細については、使用しているブートマネージャーの使用方法を参照してください。

ブートディスクに Solaris OS をインストールする方法をお勧めします。マシンに複数のオペレーティングシステムがインストールされている場合は、エントリを menu.lst ファイルに追加できます。これらのエントリは、システムを次にブートしたときに GRUB メニューに表示されます。

複数のオペレーティングシステムの詳細については、『Solaris のシステム管理 (基本編)』の「GRUB ブート環境で複数のオペレーティングシステムをサポートする方法」を参照してください。

x86: ネットワークからの GRUB ベースのインストールの実行

GRUB ベースのネットワークブートを実行するには、PXE クライアント用に構成された DHCP サーバーと、tftp サービスを提供するインストールサーバーが必要です。DHCP サーバーには、DHCP クラスである PXEClient と GRUBClient に応答する機能が必要です。DHCP 応答には、次の情報が含まれている必要があります。

- ファイルサーバーの IP アドレス
- ブートファイルの名前 (pxegrub)

注 - rpc.bootparamd は、通常、ネットワークブートを実行する場合にサーバー側で必要とされるファイルですが、GRUB ベースのネットワークブートでは不要です。

PXE も DHCP サーバーも使用できない場合は、CD-ROM またはローカルディスクから GRUB をロードできます。次に GRUB でネットワークを手動で構成し、ファイルサーバーからマルチブートプログラムとブートアーカイブをダウンロードできます。

詳細については、『Solaris 10 インストールガイド (ネットワークインストール)』の「PXE を使用したネットワーク経由のブートとインストールの概要」を参照してください。

GRUB メインメニューについて

x86 システムをブートすると、GRUB メニューが表示されます。このメニューには、選択可能なブートエントリの一覧が表示されます。「ブートエントリ」は、使用しているシステムにインストールされている OS インスタンスです。GRUB メニューは、構成ファイルの menu.lst ファイルに基づいています。menu.lst ファイルは、Solaris インストールプログラムによって作成され、インストール後に変更できます。menu.lst ファイルには、GRUB メニューに表示される OS インスタンスの一覧が記述されています。

- Solaris OS をインストールまたはアップグレードすると、GRUB メニューが自動的に更新されます。これにより、Solaris OS が新しいブートエントリとして表示されます。
- Solaris OS 以外の OS をインストールする場合は、menu.lst 構成ファイルに変更を加えて新しい OS インスタンスを含める必要があります。新しい OS インスタンスを追加すると、システムを次にブートしたときに、新しいブートエントリが GRUB メニューに表示されます。

例 4-1 GRUB メインメニュー

次の例では、GRUB メインメニューに Solaris オペレーティングシステムと Microsoft Windows オペレーティングシステムが表示されています。また、Solaris Live Upgrade ブート環境も `second_disk` という名前で一覧に表示されています。各メニューの項目については、続く説明を参照してください。

```
GNU GRUB version 0.95 (616K lower / 4127168K upper memory)
+-----+
|Solaris                                     |
|Solaris failsafe                           |
|second_disk                                |
|second_disk failsafe                       |
|Windows                                    |
+-----+
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.
```

Solaris	Solaris OS を指定します。
Solaris failsafe	Solaris OS が損傷した場合に、回復のために使用できるブートアーカイブを指定します。
second_disk	Solaris Live Upgrade ブート環境を指定します。second_disk ブート環境は、Solaris OS のコピーとして作成されました。luactivate コマンドによってアップグレードおよびアーカイブされています。ブート環境は、ブート時に使用できます。
Windows	Microsoft Windows OS を指定します。GRUB はこれらのパーティションを検出しますが、OS がブート可能かどうかは確認しません。

GRUB menu.lst ファイルについて

GRUB menu.lst ファイルには、GRUB メインメニューの内容の一覧が記述されています。GRUB メインメニューには、Solaris Live Upgrade ブート環境を含め、使用しているシステムにインストールされているすべての OS インスタンスのブートエントリが一覧表示されます。Solaris のソフトウェアアップグレード処理では、このファイルに対して行なった変更内容がすべて保持されます。

menu.lst ファイルに対する修正は、Solaris Live Upgrade エントリと一緒に、すべて GRUB メインメニューに表示されます。ファイルに対する変更点はすべて、システムを次にリブートしたときに有効になります。このファイルは、次の目的で変更できます。

- Solaris 以外のオペレーティングシステム用の GRUB メニューエントリを追加するため
- GRUB メニューでのデフォルトの OS の指定など、ブート動作をカスタマイズするため



注意 – GRUB menu.lst ファイルを使用して Solaris Live Upgrade エントリを変更しないでください。変更により、Solaris Live Upgrade の実行に失敗する可能性があります。

menu.lst ファイルを使用してカーネルデバッグによるブートなどのブート動作をカスタマイズできますが、カスタマイズする場合は eeprom コマンドを使用する方法をお勧めします。menu.lst ファイルを使用してカスタマイズすると、ソフトウェアのアップグレード時に Solaris OS エントリが変更される可能性があります。これにより、ファイルに加えた変更内容が失われます。

eeprom コマンドの用法については『Solaris のシステム管理 (基本編)』の「eeprom コマンドを使用して Solaris ブートパラメータを設定する方法」を参照してください。

例 4-2 menu.lst ファイル

次に menu.lst ファイルの例を示します。

```
default 0
timeout 10
title Solaris
    root (hd0,0,a)
    kernel /platform/i86pc/multiboot -B console=ttya
    module /platform/i86pc/boot_archive
title Solaris failsafe
    root (hd0,0,a)
    kernel /boot/multiboot -B console=ttya -s
    module /boot/x86.miniroot.safe
#----- second_disk - ADDED BY LIVE UPGRADE - DO NOT EDIT -----
title second_disk
    root (hd0,1,a)
    kernel /platform/i86pc/multiboot
    module /platform/i86pc/boot_archive
title second_disk failsafe
    root (hd0,1,a)
    kernel /boot/multiboot kernel/unix -s
    module /boot/x86.miniroot-safe
#----- second_disk ----- END LIVE UPGRADE -----
title Windows
    root (hd0,0)
    chainloader -1
```

default

タイムアウトした場合にブートする項目を指定します。デフォルトの設定を変更する場合は、番号を変更して一覧の別の項目を指定できます。最初のタイトルを 0 として数えます。たとえば、デフォルトを 2 に変更すると、second_disk ブート環境が自動的にブートします。

例 4-2 menu.lst ファイル (続き)

timeout

デフォルトのエントリをブートするまで、ユーザーの入力を待つ時間を秒単位で指定します。タイムアウトを指定しない場合は、エントリを選択する必要があります。

title *OS name*

オペレーティングシステムの名前を指定します。

- これが Solaris Live Upgrade ブート環境の場合、*OS name* は新しいブート環境の作成時にその環境に指定した名前です。前の例では、Solaris Live Upgrade ブート環境の名前は `second_disk` です。
- これがフェイルセーフブートアーカイブの場合、このブートアーカイブは主 OS が損傷したときの回復処理に使用されます。前の例では、Solaris フェイルセーフと `second_disk` フェイルセーフが Solaris および `second_disk` オペレーティングシステムの回復用ブートアーカイブです。

root (hd0,0,a)

ファイルのロード先となるディスク、パーティション、およびスライスを指定します。GRUB は、ファイルシステムの種類を自動的に検出します。

kernel /platform/i86pc/multiboot

マルチブートプログラムを指定します。kernel コマンドのあとに、必ずマルチブートプログラムを指定する必要があります。マルチブートのあとの文字列は、解釈されずに Solaris OS に渡されます

複数のオペレーティングシステムの詳細については、『Solaris のシステム管理 (基本編)』の「GRUB ブート環境で複数のオペレーティングシステムをサポートする方法」を参照してください。

GRUB メニューを変更するための menu.lst ファイルの検出

GRUB メニューの menu.lst ファイルを検出するには、常に bootadm コマンドを使用する必要があります。アクティブな GRUB メニューを見つけるには、list-menu サブコマンドを使用します。menu.lst ファイルには、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一覧表示されています。このファイルの内容は、GRUB メニューに表示されるオペレーティングシステムの一覧を記述したものです。このファイルに変更を加える場合は、[65 ページの「x86: GRUB メニューの menu.lst ファイルの検出 \(作業\)」](#)を参照してください。

x86: GRUB メニューの menu.lst ファイルの検出 (作業)

Solaris 10 1/06 以降のリリースでは、GRUB メニューを更新することができます。たとえば、デフォルトの OS がブートするまでのデフォルトの時間を変更できます。また、別の OS を GRUB メニューに追加することができます。

通常、アクティブな GRUB メニューの menu.lst ファイルは /boot/grub/menu.lst に置かれています。場合によっては、GRUB menu.lst ファイルが別の場所に置かれていることもあります。たとえば、Solaris Live Upgrade を使用するシステムの場合は、GRUB menu.lst ファイルが現在稼働中のブート環境ではないブート環境に存在することもあります。また、x86 ブートパーティションがあるシステムをアップグレードした場合、menu.lst ファイルは /stubboot ディレクトリに置かれている場合があります。システムのブートには、アクティブな GRUB menu.lst ファイルのみが使用されます。システムのブート時に表示される GRUB メニューを変更するには、アクティブな GRUB menu.lst ファイルに変更を加える必要があります。それ以外の GRUB menu.lst ファイルに変更を加えても、システムのブート時に表示されるメニューに影響はありません。アクティブな GRUB menu.lst ファイルの場所を特定するには、bootadm コマンドを使用します。list-menu サブコマンドを実行すると、アクティブな GRUB メニューの場所が表示されます。次の手順で、GRUB メニューの menu.lst ファイルの場所を特定します。

bootadm コマンドの詳細については、bootadm(1M) のマニュアルページを参照してください。

▼ GRUB メニューの menu.lst ファイルの検出

次の手順では、システムに次の 2 つのオペレーティングシステムが含まれています。Solaris、および Solaris Live Upgrade ブート環境である second_disk です。Solaris OS はブート済みで、GRUB メニューが含まれています。

- 手順 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. menu.lst ファイルを検出するには、次のように入力します。

```
# /sbin/bootadm list-menu
```

ファイルの場所と内容が表示されます。

```
The location for the active GRUB menu is: /boot/grub/menu.lst
default 0
timeout 10
```

```
0 Solaris
1 Solaris failsafe
2 second_disk
3 second_disk failsafe
```

▼ アクティブな menu.lst ファイルが別のブート環境にある場合の GRUB メニューの menu.lst ファイルの検出

次の手順では、システムに次の 2 つのオペレーティングシステムが含まれています。Solaris、および Solaris Live Upgrade ブート環境である second_disk です。この例では、現在稼働中のブート環境には menu.lst ファイルは存在しません。second_disk ブート環境がブートされています。Solaris ブート環境には GRUB メニューが含まれています。Solaris ブート環境はマウントされていません。

- 手順 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. **menu.lst** ファイルを検出するには、次のように入力します。

```
# /sbin/bootadm list-menu
```

ファイルの場所と内容が表示されます。

```
The location for the active GRUB menu is: /dev/dsk/device_name(not mounted)
The filesystem type of the menu device is <ufs>
default 0
timeout 10
0 Solaris
1 Solaris failsafe
2 second_disk
3 second_disk failsafe
```

3. **menu.lst** ファイルを含むファイルシステムがマウントされていないため、ファイルシステムをマウントします。**UFS** ファイルシステムとデバイス名を指定します。

```
# /usr/sbin/mount -F ufs /dev/dsk/device_name /mnt
```

device_name は、マウントするブート環境のディスクデバイスにあるルート (/) ファイルシステムの場所を指定します。デバイス名は、/dev/dsk/cwtxdysz の形式で入力します。次に例を示します。

```
# /usr/sbin/mount -F ufs /dev/dsk/c0t1d0s0 /mnt
```

GRUB メニューには /mnt/boot/grub/menu.lst でアクセスできます。

4. ファイルシステムをマウント解除します。

```
# /usr/sbin/umount /mnt
```

注 - ブート環境またはブート環境のファイルシステムをマウントする場合は、使用後に必ずファイルシステムをマウント解除してください。これらのファイルシステムをマウント解除しないと、その後 Solaris Live Upgrade を同じブート環境で実行したときに失敗する可能性があります。

▼ Solais Live Upgrade ブート環境がマウントされている場合の GRUB メニューの menu.lst ファイルの検出

次の手順では、システムに次の 2 つのオペレーティングシステムが含まれています。Solaris、および Solaris Live Upgrade ブート環境である second_disk です。second_disk ブート環境がブートされています。Solaris ブート環境には GRUB メニューが含まれています。Solaris ブート環境は /.alt.Solaris にマウントされています。

- 手順 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. menu.lst ファイルを検出するには、次のように入力します。

```
# /sbin/bootadm list-menu
```

ファイルの場所と内容が表示されます。

```
The location for the active GRUB menu is:
/.alt.Solaris/boot/grub/menu.lst
default 0
timeout 10
0 Solaris
1 Solaris failsafe
2 second_disk
3 second_disk failsafe
```

GRUB メニューを含むブート環境がすでにマウントされているため、/.alt.Solaris/boot/grub/menu.lst で menu.lst ファイルにアクセスできます。

▼ 使用しているシステムに x86 ブートパーティションがある場合の GRUB メニューの menu.lst ファイルの検出

次の手順では、システムに次の 2 つのオペレーティングシステムが含まれています。Solaris、および Solaris Live Upgrade ブート環境である second_disk です。second_disk ブート環境がブートされています。システムはアップグレードされており、x86 ブートパーティションが残されています。ブートパーティションは /stubboot にマウントされ、GRUB メニューが含まれています。x86 ブートパーティションについては、54 ページの「x86: パーティション分割の推奨事項」を参照してください。

手順 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. menu.lst ファイルを検出するには、次のように入力します。

```
# /sbin/bootadm list-menu
```

ファイルの場所と内容が表示されます。

```
The location for the active GRUB menu is:
/stubboot/boot/grub/menu.lst
default 0
timeout 10
0 Solaris
1 Solaris failsafe
2 second_disk
3 second_disk failsafe
```

menu.lst ファイルには、/stubboot/boot/grub/menu.lst でアクセスできます。

パート II 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。

第 5 章

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。

- 71 ページの「カスタム JumpStart とは」
- 73 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検査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 5-1 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。

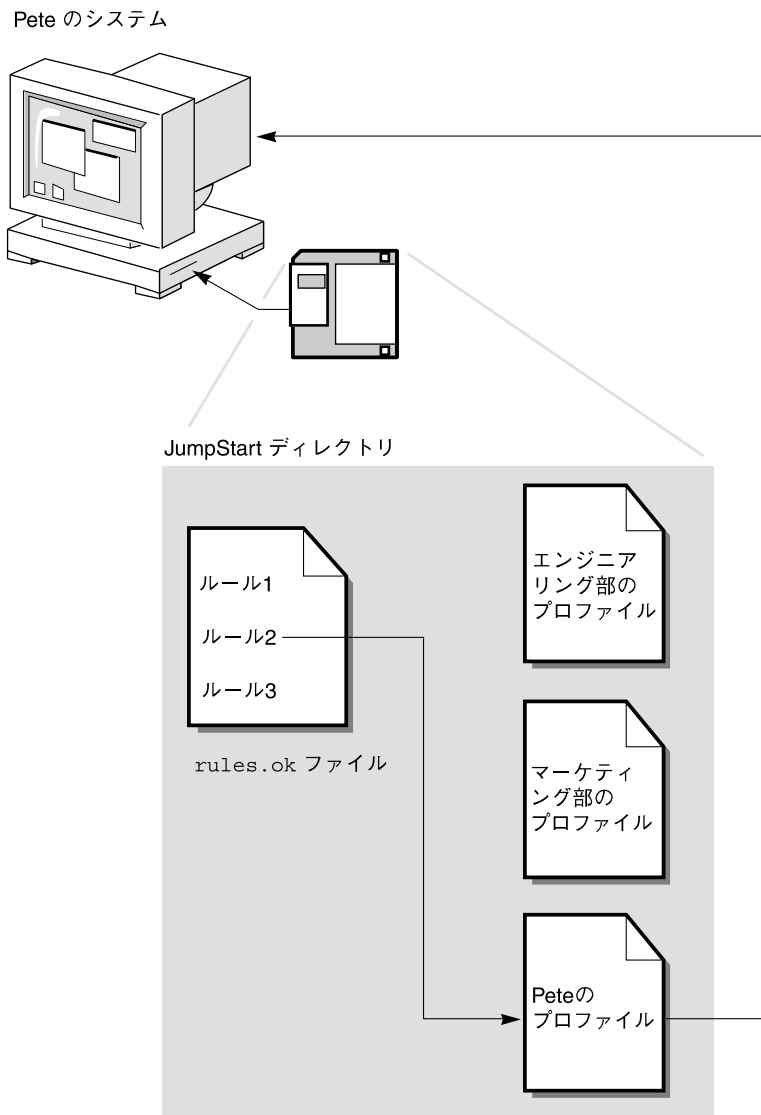


図 5-1 カスタム JumpStart インストールの流れ: ネットワークに接続されていない場合の例

図 5-2 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。

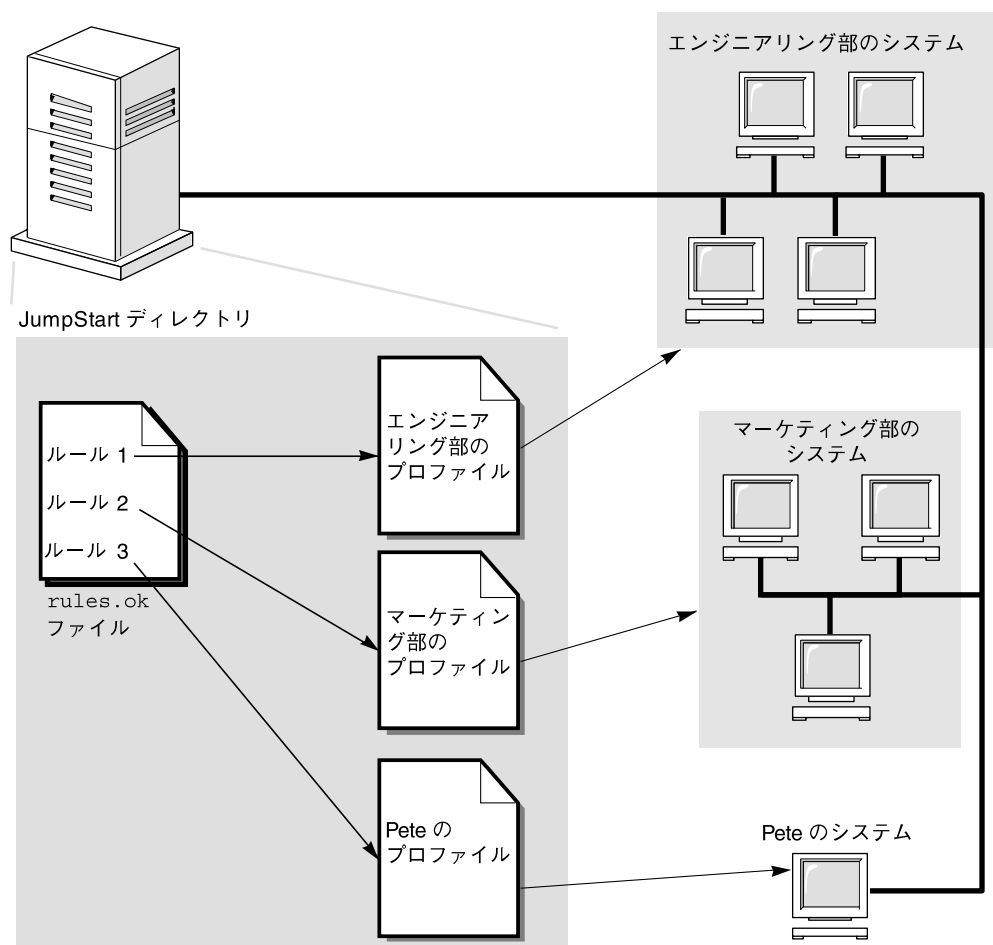


図 5-2 カスタム JumpStart インストールの流れ: ネットワークに接続されている場合の例

図 5-3 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。

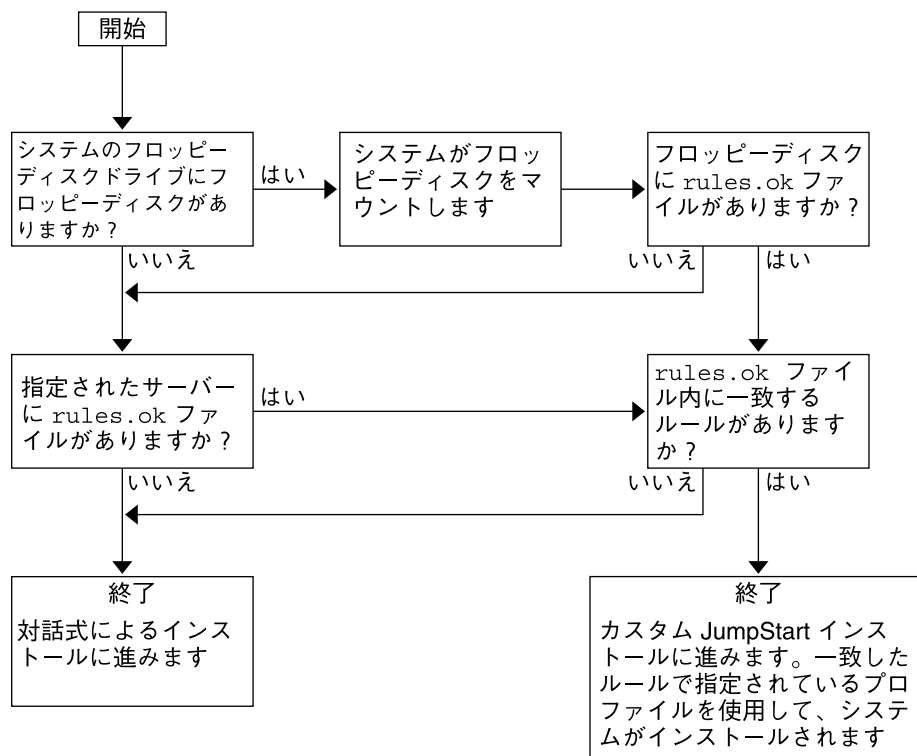


図 5-3 カスタム JumpStart インストール実行時の流れ

第 6 章

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。

- 78 ページの「作業マップ: カスタム JumpStart インストールの準備」
- 79 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」
- 85 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」
- 93 ページの「rules ファイルの作成」
- 96 ページの「プロファイルの作成」
- 110 ページの「プロファイルのテスト」
- 115 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 6-1 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。	43 ページの「アップグレード」
JumpStart ディレクトリを作成します。	サーバーの場合 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。 フロッピーディスクの場合 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。	79 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」 85 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」
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。	93 ページの「rules ファイルの作成」

表 6-1 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
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。	96 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。	110 ページの「プロファイルのテスト」
rules ファイルの妥当性を検査します。	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検査するには、check スクリプトを使用する必要があります。	115 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際には、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、83 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。

手順 1. **JumpStart** ディレクトリを作成するサーバーに、スーパーユーザーとしてログインします。

2. 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
```

3. 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
```

4. **shareall** と入力し、**Enter** キーを押します。

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 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 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。 cd /export/install

6. 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** ディレクトリへコピーします。

```
# 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:

- Solaris 10 1/06 以降のリリース:

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

- Solaris 10 3/05 リリース:

```
cp -r /cdrom/cdrom0/s2/Solaris_10/Misc/jumpstart_sample/* /jumpstart
```

7. 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。
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. 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、83 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	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。 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。 ■ SPARC システムでは <code>boot</code> コマンドを使用します ■ x86 システムの場合: ■ Solaris 10 1/06 以降のリリースでは GRUB メニューを編集します。 ■ Solaris 10 3/05 リリースでは <code>boot</code> コマンドを使用します。	『Solaris 10 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」
■ SPARC: <code>boot</code> コマンド ■ x86: <code>boot</code> コマンド	次のリリースの場合: ■ SPARC ■ x86 版 Solaris 10 3/05 リリース 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、NFS サーバー、HTTP サーバー、またはシステムがローカルにアクセスできるメディアに、圧縮されたその構成ファイルを保存します。 システムをブートしてカスタム JumpStart インストールを開始する場合は、この圧縮ファイルの場所を指定してください。	■ 125 ページの「圧縮された構成ファイルの作成」 ■ 143 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

コマンドまたはファイル	アクセスの提供	参照先
x86: GRUB メニュー	Solaris 10 1/06 以降のリリースでは、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、NFS サーバー、HTTP サーバー、またはシステムがローカルにアクセスできるメディアに、圧縮されたその構成ファイルを保存します。 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。	■ 125 ページの「圧縮された構成ファイルの作成」 ■ 150 ページの「x86: GRUB boot コマンドの編集によるカスタム JumpStart インストールの実行」
/etc/bootparams ファイル	/etc/bootparams ファイルにワイルドカードを追加します。	83 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

上記のような場合には boot コマンドを使用するか、x86 システムの Solaris 10 1/06 リリースでは 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、[144 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 サーバーおよび PXE を使用するように指定します。

Solaris 10 1/06 以降のリリースの場合、GRUB メニューの GRUB エントリを編集して `dhcp` オプションを追加できます。GRUB エントリを編集する手順については、[150 ページの「x86: GRUB boot 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

- 手順
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については、[249 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、必要なカスタム JumpStart ファイルがすべて収められています。カスタム JumpStart ファイルには、たとえば、rules ファイル、rules.ok ファイル、プロファイルなどがあります。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。

- 85 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」
- 87 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」
- 89 ページの「x86: Solaris 10 3/05 リリースの場合: 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、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。

- 手順 1. フロッピーディスクドライブを持つ **SPARC** システムに、スーパーユーザーとしてログインします。
2. 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。
3. フロッピーディスクをマウントします。

```
# volcheck
```

4. 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 6](#) 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。

5. フロッピーディスク上に **UFS** を作成します。

```
# newfs /vol/dev/aliases/floppy0
```

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
Solaris Operating System DVD (SPARC 版) または Solaris SOFTWARE - 1 CD (SPARC 版)	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。 ボリュームマネージャーが自動的に CD をマウントします。
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	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。 <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フロッピーディスクの **JumpStart** ディレクトリへコピーします。

```
# 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/sol_10_sparc/s0/Solaris_10/Misc/jumpstart_sample/* /floppy/scrap
```

8. 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。
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. 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、[93 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

Solaris 10 1/06 以降のリリースの場合は、次の手順を実行します。システムをブートするのに Solaris Device Configuration Assistant (デバイス構成用補助) ソフトウェアは使用されなくなりました。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。

Solaris 10 3/05 リリースを使用している場合は、[89 ページ](#) の「[x86: Solaris 10 3/05 リリースの場合: プロファイルフロッピーディスクを作成する方法](#)」を参照してください。

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、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。

- 手順
1. フロッピーディスクドライブを持つ **x86** システムに、スーパーユーザーとしてログインします。
 2. 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。
 3. フロッピーディスクをマウントします。

volcheck
 4. 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 7](#)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
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 をマウントします。
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。 cd /export/install

5. 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** ディレクトリへコピーします。

```
# 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 – プロファイルフロッピーディスクを使用するときは、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 (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/sol_10_x86/Solaris_10/Misc/jumpstart_sample/* /floppy/scrap
```

6. 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。
7. **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。
8. 「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。
9. 「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。
10. フロッピーディスクを手動で取り出します。

参照 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、[93 ページの「rules ファイルの作成」](#)へ進みます。

▼ x86: Solaris 10 3/05 リリースの場合: プロファイルフロッピーディスクを作成する方法

Solaris 10 3/05 リリースを使用している場合は、この手順を実行します。

Solaris 10 1/06 以降のリリースでは、システムをブートするのに Solaris Device Configuration Assistant (デバイス構成用補助) ソフトウェアは使用されなくなりました。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。手順については、[87 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」](#)を参照してください。

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、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。

- 手順
1. フロッピーディスクドライブを持つ **x86** システムに、スーパーユーザーとしてログインします。
 2. **Solaris 10 3/05 Device Configuration Assistant** (デバイス構成用補助) フロッピーディスクをフロッピーディスクドライブ (通常はドライブ **A**) に挿入します。このフロッピーディスクは、プロファイルフロッピーディスクとして使用します。

x86 のみ – 『Solaris 10 インストールガイド (ネットワークインストール)』の「x86 版 Solaris 10 3/05: ブートソフトウェアのフロッピーディスクへのコピー」の手順に従って、Solaris Operating System DVD (x86 版) または Solaris SOFTWARE - 2 CD (x86 版) からフロッピーディスクへ Device Configuration Assistant (デバイス構成用補助) ソフトウェアをコピーできます。

3. フロッピーディスクをマウントします。

```
# volcheck
```

4. **Solaris 10 3/05 Device Configuration Assistant** (デバイス構成用補助) のイメージをシステムのハードディスクへコピーします。

```
# dd if=/vol/dev/aliases/floppy0 of=boot_image
```

コマンドの *boot_image* の部分には、Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) のイメージのコピー先ファイル名を入力します。絶対パス名を指定できます。

たとえば次のコマンドは、ブートフロッピーディスクを *boot_save* ファイルにコピーします。

```
dd if=/vol/dev/aliases/floppy0 of=boot_save
```

5. 「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。
6. 「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。

7. **Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助)** フロッピーディスクを手動で取り出します。
8. 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。
9. フロッピーディスクをマウントします。
- ```
volcheck
```
10. フロッピーディスクをフォーマットします。
- ```
# fdformat -d -U
```
11. **Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助)** イメージを、システムのハードディスクからフォーマットしたフロッピーディスクにコピーします。
- ```
dd if=boot_image of=/vol/dev/aliases/floppy0
```
- コマンドの *boot\_image* の部分には、Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) のイメージのコピー先ファイル名を入力します。絶対パス名を指定できます。
12. 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 7](#)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                                                                                                                                      |
|------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| 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。<br><br>ボリュームマネージャーが自動的に CD をマウントします。 |
| 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。<br><br>cd /export/install       |

13. 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** ディレクトリへコピーします。
- ```
# 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 – プロファイルフロッピーディスクを使用するときは、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 (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/sol_10_x86/s2/Solaris_10/Misc/jumpstart_sample/* /floppy/scrap
```

14. 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。
15. **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。
16. 「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。
17. 「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。
18. フロッピーディスクを手動で取り出します。

参照 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、[93 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 - 85 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 79 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

- 手順 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、[171 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。

<code>!rule_keyword rule_value && !rule_keyword rule_value ... begin profile finish</code>	
<code>!</code>	ルールキーワードの前で使用し、否定を示す記号。
<code>rule_keyword</code>	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。 <code>rule_keyword</code> 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、 171 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。
<code>rule_value</code>	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、 171 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。
<code>&&</code>	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。
<code>begin</code>	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しい作成方法については、 117 ページの「開始スクリプトの作成」 を参照してください。
<code>profile</code>	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には、別の使用方法もあります。詳細は、132 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 118 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、119 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。

例 6-1 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	-

例 6-1 rule ファイル (続き)

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 (SUNW, Sun-Blade-100) 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。このルールは、93 ページの「rules ファイルの構文」に説明されているルールラップの例も示しています。
model	このルールは、システムが SPARCstation LX 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、x86 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 64 ～ 128M バイトのメモリを持つ x86 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、上記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 rules 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ストール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。

- 79 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」
 - 85 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 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

- 手順 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、[175 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、[110 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。[175 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 6-2 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
filesystem	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、すべてのプロファイルに必要です。
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。

例 6-2 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)

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す。

例 6-3 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
filesystem             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、すべてのプロファイルに必要です。
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	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 6-4 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 6-5 パッチのアップグレードとインストール

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

例 6-5 パッチのアップグレードとインストール (続き)

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、183 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 6-6 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
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。

例 6-6 アップグレード用ディスク容量の再配置 (続き)

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、183 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	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。 ■ 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。
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 6-7 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。

例 6-7 HTTP サーバーから Solaris フラッシュアーカイブを取得する (続き)

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 6-8 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。

# profile keywords	profile values
# -----	-----
install_type	flash_install
archive_location	https://192.168.255.255/solarisupdate.flar
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	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。
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。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 6-9 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。

例 6-9 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)

```
# 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
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。

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。
package	SUNWnew パッケージは HTTP サーバー 192.168.254.255 の Solaris_10 ディレクトリから追加されます。

例 6-10 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。

例 6-10 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)

install_type	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。
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す。
no_master_check	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、元のマスターシステムを使用して構築されているはずです。

例 6-11 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 上に設定されます。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

例 6-11 空のブート環境の作成 (続き)

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、175 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。
- 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については、『Solaris 10 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 6 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基本的な使用方法については、『Solaris 10 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 6-12 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
archive_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。
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す。

例 6-12 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)

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、第 12 章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、第 13 章を参照してください。
- キーワード値については、197 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 203 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 6-13 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ー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
filesys                mirror:d30 c0t1d0s0 c0t0d0s0  /
```

例 6-13 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 (続き)

```
filesys          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。
- filesys 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。
- filesys 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、[第 12 章](#)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、[第 13 章](#)を参照してください。
 - キーワード値については、[197 ページの「filesys プロファイルキーワード \(RAID-1 ボリュームの作成\)」](#)および [203 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 6-14 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
filesys                mirror:d100 c0t1d0s0 c0t0d0s0 200 /
filesys                c0t1d0s5 500 /var
filesys                c0t0d0s5 500
filesys                mirror c0t0d0s1 512 swap
metadb                 c0t0d0s3 size 8192 count 5
filesys                mirror c0t1d0s4 c0t0d0s4 2000 /usr
```

例 6-14 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)

```
filesys          c0t1d0s7 free /export/home
filesys          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。
filesys	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。
filesys	/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。
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 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。
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 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。

- 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、[第 12 章](#)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、[第 13 章](#)を参照してください。
- キーワード値については、197 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 203 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 6-15 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
```

例 6-15 x86: fdisk キーワードの使用方法 (続き)

```
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ー (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、[127 ページの「ディスク構成ファイルの作成」](#)と[129 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。

▼ プロファイルをテストするために、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をテストする場合は、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をテストする場合は、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 のアップグレードプロファイルをテストする場合。
- Solaris 10 の初期インストールプロファイルをテストするための Solaris 10 システムをまだインストールしていない場合。

手順 1. 次を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。

```
Solaris Web Start 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、[112 ページの「プロファイル进行测试する方法」の手順 7](#) を参照してください。

▼ プロファイル进行测试する方法

- 手順 1. プロファイル进行测试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进行测试する場合は、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 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 が動作しているシステムがない場合)。	一時的な Solaris 10 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、 111 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 [127 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をテスト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 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ので、Solaris SOFTWARE - 1 CD (SPARC 版) または Solaris SOFTWARE - 1 CD (x86 版) イメージへのパスが指定されています。

例 6-16 Solaris 10 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 システム上で `basic_prof` というプロファイルをテスト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 6-17 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有効な場合は、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 6-2 に、check スクリプトの実行内容を示します。

表 6-2 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> フィールドは、ファイル名でなくマイナス記号 (-) でも可能です。
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

手順 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** ディレクトリへ移動します。

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、カスタム JumpStart オプション機能については、[第 7 章](#)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 9 章](#)を参照してください。

第 7 章

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。

- 117 ページの「開始スクリプトの作成」
- 119 ページの「終了スクリプトの作成」
- 125 ページの「圧縮された構成ファイルの作成」
- 127 ページの「ディスク構成ファイルの作成」
- 132 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) に利用できます。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す。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す。

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 に對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、219 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。

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對應させる 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 7-1 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}
```

例 7-1 派生プロファイルを作成する開始スクリプト (続き)

```
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、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、システムがリブ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、このログファイルは `/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、219 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。これは、SI_CONFIG_DIR 変数で指定されるディレクトリ (デフォルトは /tmp/install_config) に JumpStart ディレクトリがマウントされるためです。

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。

- 手順
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 7-2 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 7-3 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 7-4 に、パッチを追加する終了スクリプトの例を示します。

例 7-3 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

/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 7-4 終了スクリプトでパッチを追加する

```
#!/bin/sh

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

例 7-4 終了スクリプトでパッチを追加する (続き)

```
#
#####

# 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

# 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 7-5 の終了スクリプトは、ルート (/) ディレクトリ内の .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 7-5 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 7-6 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 7-6 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 7-6 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=dKO5IBkSF42lw
#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
```

- 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 と 1 の後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 7-1 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せ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` コマンドのファイルの場所を指定します。
- x86 システムの場合:
 - Solaris 10 1/06 以降のリリースでは、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。
 - Solaris 10 3/05 リリースでは、`boot` コマンドの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

手順 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 7-7 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进行测试できます。

▼ SPARC: ディスク構成ファイルを作成する方法

- 手順
1. テストするディスクのある **SPARC** 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保存して、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 7-8 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
*   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、[110 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

- 手順 1. テストするディスクがある x86 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/rdisk/device_name
```
- disk\_config\_file*                      ディスク構成ファイルの名前。  
*/dev/rdisk/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/rdisk/device_name >>disk_config
```
- /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 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 7-9 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
*
* 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。

例 7-9 x86: ディスク構成ファイルの作成 (続き)

```
# 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
* 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
*           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
```

例 7-9 x86: ディスク構成ファイルの作成 (続き)

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进行测试する方法については、[110 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
```

第 8 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)

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。

- [133 ページの「プローブキーワード」](#)
- [134 ページの「custom_probes ファイルの作成」](#)
- [137 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、[221 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

171 ページの「ルールキーワードと値の説明」と 221 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が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

- 手順
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 - custom_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 8-1 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 | awk '{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 8-2 rules ファイルで使用す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 8-1 に、check スクリプトの実行内容を示します。

表 8-1 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

手順 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** ディレクトリへ移動します。

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。

第 9 章

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。

- 143 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」
- 148 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」

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 9-1 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ル』(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。	第 3 章

表 9-1 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) 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 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「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 6 章
(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 7 章および第 8 章
(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 インストールガイド (ネットワークインストール)』の第 8 章「DVD メディアを使用したネットワークインストールの準備 (作業)」 『Solaris 10 インストールガイド (ネットワークインストール)』の第 9 章「CD メディアを使用したネットワークインストールの準備 (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。	141 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。	143 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 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 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た Solaris フラッシュアーカイブのインストール (コマンド行インタフェース)」を参照してください。

手順 1. 次の制限事項を確認します。

説明	例
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。	たとえば、アーカイブが Solaris 10 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 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。

説明	例
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	

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 6 章](#)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、[98 ページ](#)の「[プロファイルの例](#)」を参照してください。

表 11-2 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)	○	○
filesys	○	
注 – filesys 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、[177 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、[205 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 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 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、[143 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

- 手順
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</i> <i>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。 ■ <i>cd-dvd</i> - <i>cdrom</i> 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。
[<i>url</i> <i>ask</i>]	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。 ■ <i>url</i> - 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 <pre>http://server_name:IP_address/jumpstart_dir_path/ compressed_config_file&proxy_info</pre> ■ <i>sysidcfg</i> 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>http://131.141.2.32/jumpstart/config.tar</pre> ■ 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 <i>proxy</i> 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>http://www.shadow.com/jumpstart/ config.tar&proxy=131.141.6.151</pre> ■ <i>ask</i> - 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。
<i>options</i>	■ <i>dhcp</i> - 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。<i>dhcp</i> で DHCP サーバーの使用を指定しないと、<i>/etc/bootparams</i> ファイル、またはネームサービスの <i>bootparams</i> 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、<i>dhcp</i> を指定しません。 ■ <i>nowin</i> オプションおよび <i>text</i> 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。詳細については、『Solaris 10 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 9-2 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 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 <code>fdisk (1M)</code> キーワードをどのように指定するかが決まります。	191 ページの「x86: <code>fdisk</code> 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。	第 3 章
(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 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「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。	<code>rules</code> 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 6 章
(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 7 章および第 8 章

表 9-2 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 インストールガイド (ネットワークインストール)』の第 9 章「CD メディアを使用したネットワークインストールの準備 (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。	141 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。	148 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、[141 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。

使用しているリリースに応じて、次の手順のいずれかを選択します。

- **Solaris 10 1/06** 以降のリリースでは、次に示す手順のいずれかを参照してください。

- 標準的なカスタム JumpStart の手順については、148 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、150 ページの「x86: GRUB boot 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。
- **Solaris 10 3/05** リリースの場合は、152 ページの「x86: 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。

▼ 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

Solaris 10 1/06 以降のリリースでは、この手順を使用して Solaris OS をインストールします。

- 手順
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。
Solaris 10 1/06 以降のリリースでは、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。フロッピーディスクを用意することはできませんが、収録されているのは JumpStart ディレクトリのみです。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。
 - 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 を使用するように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。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 image_directory                                     |
|                                                                 |
+-----+
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、[150 ページの「x86: GRUB boot 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. 「**Boot Solaris**」画面が表示されたら、システムをブートするデバイス(「**DVD**」、「**CD**」、「**Net**」、または「**Disk**」)を選択します。
8. 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 インストールガイド (基本編)』を参照してください。
 - これらのインストールの詳細については、『Solaris 10 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。
-

9. **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。

10. 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 boot コマンドの編集によるカスタム JumpStart インストールの実行

Solaris 10 1/06 以降のリリースでは、GRUB メニューを編集できます。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 boot コマンドを変更する方法

- 手順 1. インストールを開始するには、[148 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、[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、インストールされる Solaris インスタンスです。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
|Solaris 10 image_directory|
+-----+
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、[154 ページの「x86: b コマンドまたは 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。

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: 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

Solaris 10 3/05 リリースの場合は、この手順を使用して Solaris OS をインストールします。

- 手順
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場合は、システムのプロ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。

注 - プロファイルフロッピーディスクには、プロファイル情報に加え、Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) のコピーが入っています。PXE ネットワークブートを利用してネットワークを介してシステムをブートする場合は、フロッピーディスクからではなくネットワークからシステムがブートするようにシステムを構成する必要があります。

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 を使用するようにシステムを設定します。
- フロッピーディスクからブートする場合は、[手順 3](#) でドライブに挿入したプロファイルフロッピーディスクを使用するか、あるいはシステムのフロッピーディスクドライブに Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスクを挿入します。

x86 のみ – 『Solaris 10 インストールガイド (ネットワークインストール)』の「x86 版 Solaris 10 3/05: ブートソフトウェアのフロッピーディスクへのコピー」の手順に従って、Solaris Operating System DVD (x86 版) または Solaris SOFTWARE - 2 CD (x86 版) からフロッピーディスクへ Device Configuration Assistant (デバイス構成用補助) ソフトウェアをコピーできます。

5. 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。

Device Configuration Assistant プログラムがシステムのデバイスを調べます。

6. 「**Boot Solaris**」画面が表示されたら、システムをブートするデバイス(「**DVD**」、「**CD**」、「**Net**」、または「**Disk**」)を選択します。
7. プrompt で、次のどちらかを実行します。

```
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)
```

```
Enter the number of your choice followed by the <ENTER> key.
Alternatively, enter custom boot arguments directly.
```

```
If you wait for 30 seconds without typing anything,
an interactive installation will be started.
```

カスタム JumpStart 方式を選択するには、次のどちらかを行います。

注 -

- 30 秒以内に「2」またはブートコマンドを入力しないと、Solaris の対話式インストールプログラムが開始されます。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 インストールガイド (基本編)』を参照してください。

- 2 を入力し、Enter キーを押します。
- **b** コマンドに適切なオプションを入力します。このコマンドの構文は次のとおりです。

b 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 **b install http://131.141.2.32/jumpstart/config.tar**

b コマンドのオプションの詳細は、下記の表を参照してください。

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。

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: b コマンドまたは boot コマンドのコマンドリファレンス

boot コマンドのコマンド行オプションの説明については、使用しているリリースに応じて、次のいずれかの節を参照してください。

- **Solaris 10 1/06** 以降のリリースの場合は、155 ページの「x86: GRUB boot メニューのコマンドリファレンス」を参照してください。
- **Solaris 10 3/05** リリースの場合は、157 ページの「x86: b (ブート) コマンドリファレンス」を参照してください。

x86: GRUB boot メニューのコマンドリファレンス

Solaris 10 1/06 以降のリリースでは、boot コマンドの構文が次のようになります。

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

次の表に、JumpStart インストールに適した GRUB の boot メニュー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
install	カスタム JumpStart インストールを実行します。 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。 ■ 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>

オプション	説明
[url ask]	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。 ■ url – ファイルのパスを指定します。次の場所にあるファイルを URL で指定できます。 ■ ローカルハードディスク <pre>file://jumpstart_dir_path/compressed_config_file</pre> 次に例を示します。 <pre>kernel /I86pc.Solaris_11.8/multiboot install \ file://jumpstart/config.tar \ -B install_media=dvdrom module /I86Solaris_11.8/x86.new</pre> ■ ネットワークファイルシステム (NFS) サーバー <pre>nfs://server_name:IP_address/jumpstart_dir_path/compressed_config_file</pre> 次に例を示します。 <pre>kernel /I86pc.Solaris_11.8/multiboot install \ myserver: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> ■ HTTP サーバー <pre>http://server_name:IP_address/jumpstart_dir_path/ compressed_config_file&proxy_info</pre> ■ sysidcfg 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> ■ 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 proxy 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>

オプション	説明
	ask – 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。 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。 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。 kernel /boot/multiboot kernel/unix install ask -B \ install_media=192.168.2.1:export/sol_11_x86/boot module \ /I86PC.Solaris_11.8_
options	- dhcp – 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。dhcp で DHCP サーバーの使用を指定しないと、/etc/bootparams ファイル、またはネームサービスの bootparams 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、dhcp を指定しません。次に例を示します。 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 - nowin オプションおよび text 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。詳細については、『Solaris 10 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: b (ブート) コマンドリファレンス

Solaris 10 3/05 リリースの場合、b コマンドの構文は次のようになります。

ok **b 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>url: ask</i>	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。 ■ <i>url</i> - ファイルのパスを指定します。HTTP または HTTPS サーバーに存在しているファイルの 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> ■ <i>sysidcfg</i> 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 proxy 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> ■ <i>ask</i> - システムが起動し、ネットワークへの接続が完了すると、インストールプログラムから圧縮構成ファイルの場所を入力す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。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。
<i>options</i>	<i>dhcp</i> - 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。 <i>dhcp</i> 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 <i>bootparams</i> 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、<i>dhcp</i> を指定しません。 その他のオプション <i>nowin</i> および <i>text</i> は、JumpStart インストールには適用されません。

第 10 章

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。

- 159 ページの「サイトの設定例」
- 161 ページの「インストールサーバーの作成」
- 162 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」
- 163 ページの「JumpStart ディレクトリの作成」
- 163 ページの「JumpStart ディレクトリの共有」
- 163 ページの「SPARC: エンジニアリング部のプロファイルの作成」
- 164 ページの「x86: マーケティング部のプロファイルの作成」
- 165 ページの「rules ファイルの更新」
- 165 ページの「rules ファイルの妥当性を検査する」
- 166 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」
- 167 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」
- 168 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」
- 168 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 10-1 は、この例で使用するサイトの設定を示しています。

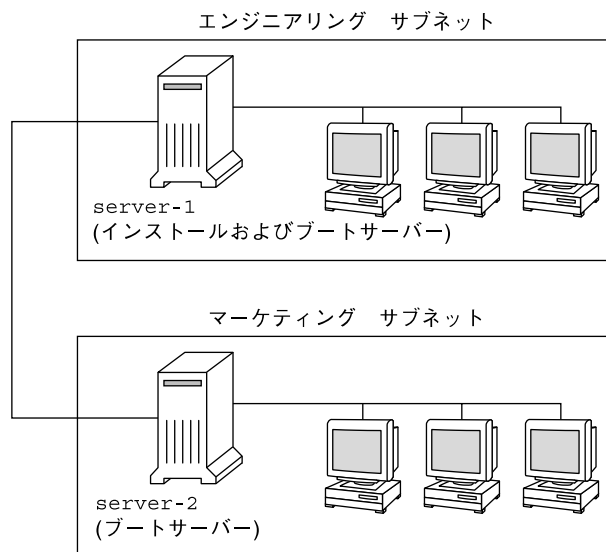


図 10-1 サイトの設定例

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 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 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 10-1 SPARC: Solaris 10 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、上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 10-2 x86: Solaris 10 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。

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

例 10-2 x86: Solaris 10 CD のコピー (続き)

インストールする Solaris SOFTWARE ごとに、上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 10-3 SPARC: Solaris 10 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 10-4 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 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
fileSYS       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。

install_type	アップグレードではなく、初期インストールを行います。
system_type	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。
partitioning	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。
cluster	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。
fileSYS	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 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。

install_type	アップグレードではなく、初期インストールを行います。
system_type	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 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 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 Solaris 10 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ることも示しています。

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 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 11-1](#) 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 インストールサーバー (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:
 - **Solaris 10 1/06** 以降のリリースでは、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
 - **Solaris 10 3/05** リリースの場合は、boot コマンドにパスを指定します
-

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 ソフトウェアのパス (/export/install/x86_10) を指定します。 |
| -c | インストールサーバー (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:
 - Solaris 10 1/06 以降のリリースでは、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
 - Solaris 10 3/05 リリースの場合は、`boot` コマンドにパスを指定します
-

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)
- プロファイルフロッピーディスク
- Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスク

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。

第 11 章

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。

- 171 ページの「ルールキーワードと値の説明」
- 175 ページの「プロファイルキーワードと値」
- 219 ページの「カスタム JumpStart の環境変数」
- 221 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 11-1 に、rules ファイルで利用できるキーワードと値の説明を記載します。
rules ファイルの詳しい作成方法については、93 ページの「rules ファイルの作成」を参照してください。

表 11-1 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: ■ 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 11-1 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	<i>actual_disk_name size_range</i> <i>actual_disk_name</i> - cxy dz 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。 ■ 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順で検索される) <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。 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。	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)。 例: <pre>disksize c0t3d0 250-300</pre> 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す。このディスクは 250M から 300M バイトの情報を保持できます。 例: <pre>disksize rootdisk 750-1000</pre> 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。 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>	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。 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。
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>	システムのホスト名を照合します。 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。

表 11-1 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	<i>slice version</i> <i>slice</i> - <i>cwt</i><i>x</i> <i>dysz</i> の形式のディスクスライス名 (<i>c0t3d0s5</i> など)、あるいは特別なワード <i>any</i> または <i>rootdisk</i>。 <i>any</i> 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)。 <i>rootdisk</i> 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。 ■ 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 ■ <i>c0t3d0s0</i> 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) <i>version</i> - バージョン名、あるいは特別なワード <i>any</i> または <i>upgrade</i>。 <i>any</i> 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。 <i>upgrade</i> 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。 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 <i>SystemV</i> になります。	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。 例: <pre>installed c0t3d0s1 Solaris 10</pre> この例では、<i>c0t3d0s1</i> 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。
karch	<i>actual_platform_group</i> 有効な値は、<i>sun4u</i>、<i>i86pc</i>、および <i>prep</i>。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。	システムのプラットフォームグループを照合します。 システムがインストール済みの場合は、<i>arch -k</i> コマンドまたは <i>uname -m</i> 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。
memsize	<i>physical_mem</i> 値は M バイト単位の範囲 (<i>x-x</i>) または 1 つの M バイト値で指定します。	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)。 例: <pre>memsize 64-128</pre> 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。 システムがインストール済みの場合は、<i>prtconf</i> 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。

表 11-1 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>	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。 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。 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 (<u>) で置き換える必要があります。</u> 例: SUNW,Sun_4_50
network	<i>network_num</i>	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。 例: network 192.168.2.0 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)。
osname	Solaris_x	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。 例: osname Solaris 10 この例では、JumpStart プログラムは Solaris 10 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。

表 11-1 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>	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。 例: <pre>probe disks</pre> 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 c0t3d0s1、c0t4d0s0)。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。 注 - probe 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。probe キーワードは、値を返すだけです。したがって、probe 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。 プローブキーワードについては、第 8 章を参照してください。
totaldisk	<i>size_range</i>	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。 例: <pre>totaldisk 300-500</pre> 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。 値は M バイト単位の範囲 (x-x) で指定する必要があります。 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。

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、[96 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 11-2 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 11-2 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 11-2 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。

- 178 ページの「NFS サーバーに格納されたアーカイブ」
- 179 ページの「HTTP または HTTPS サーバーに格納されたアーカイブ」
- 180 ページの「FTP サーバーに格納されたアーカイブ」
- 181 ページの「ローカルテープに格納されたアーカイブ」
- 182 ページの「ローカルデバイスに格納されたアーカイブ」
- 183 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 11-1 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`

<i>server_name</i>	アーカイブを格納するサーバーの名前です。
<i>port</i>	オプションのポートです。 <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。
<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 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 11-3 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 11-3 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>	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。 ■ 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>	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。

例 11-2 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。

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。

<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 11-4 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>	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。 ■ 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>	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。

例 11-3 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
```

<i>device</i>	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。
<i>position</i>	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 11-4 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
```

<i>device</i>	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。
<i>path</i>	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。
<i>filename</i>	Solaris フラッシュアーカイブファイルの名前です。
<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、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 11-5 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。

例 11-5 ローカルデバイスに格納されたアーカイブ (続き)

```
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 11-6 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 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。

- 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。
- 非大域ゾーンがインストールされている場合は、このキーワードを使用しないでください。このキーワードが JumpStart プロファイルで使用されると、アップグレードが停止してエラーメッセージが表示されます。

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/cwtxdysz</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/cwtxdysz</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 11-7 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
```


例 11-7 backup_media プロファイルキーワード (続き)

```
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: <code>cwtxdysz</code> または <code>cxdysz</code>	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 <code>c0t0d0s0</code>) です。
x86: <code>cwtxdy</code> または <code>cxdy</code>	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 <code>c0d0</code>) です。
<code>existing</code>	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。
<code>any</code>	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 11-8 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になります。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 ボリュームマネージャーのボリュームや Veritas Volume Manager のボリュームを

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 6-11
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 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 6 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 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、`/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せん。

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せん。

例 11-9 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 11-5 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 11-5 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 11-5 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参照します。

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` 名の一覧は次の表のとおりです。

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>
エンドユーザーシステムサポート	<code>SUNWCuser</code>
開発者システムサポート	<code>SUNWCprog</code>
全体ディストリビューション	<code>SUNWCall</code>
全体ディストリビューションと OEM サポート	<code>SUNWCXall</code>

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については、41 ページの「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。

<code>cluster_name</code>	クラスタ名は <code>SUNWCname</code> の形式で指定します。
<code>add_delete_switch</code>	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 <code>add</code> または <code>delete</code> を使用します。値 <code>add</code> も <code>delete</code> も指定しなかった場合、デフォルトで <code>add</code> 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 – 非大域ゾーンがインストールされている場合、このキーワードを使用してアップグレードしないでください。このキーワードが使用されると、アップグレードは続行しますが、キーワードは無視されます。

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参照します。

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、cxydz 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 が all の場合は、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値を含む変数。217 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。ディスク上にすでに指定した *type* の *fdisk* パーティションがあると、その既存の *fdisk* パーティションを使用します(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。

filesys 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)

`filesys server:path server_address mount_pt_name mount_options`

これらの値を指定して *filesys* 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。*filesys* 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: /usr、/export/home など)
<i>server_address</i>	サーバーの IP アドレス。 <i>server:path</i> 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 <i>server_address</i> を使用して、サーバーのホスト名と IP アドレスを /etc/hosts

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。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 <i>mount_pt_name</i> 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

例 11-10 filesys プロファイルキーワード

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

filesys 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)

filesys slice size file_system optional_parameters

これらの値を指定して filesys 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。filesys は 2 回以上指定できません。

<i>slice</i>	次のいずれかの値を使用します。
any	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。

注 - size が existing、all、free、start:size、または ignore の場合は、any は指定できません。

cwtxdysz または cxdysz	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、c0t0d0s0 または c0d0s0。
---------------------	---

	<code>rootdisk.sn</code>	システムのルートディスクの値を含む変数は、JumpStart プログラムによって決定されます。詳細については、 217 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。拡張子 <code>sn</code> は、ディスク上の特定のスライスを示します。
<code>size</code>		次のいずれかの値を使用します。
	<code>num</code>	ファイルシステムのサイズを <code>num</code> (M バイト単位) で設定します。
	<code>existing</code>	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 - <code>existing</code> の値を使用すると、別の <code>mount_pt_name</code> として <code>file_system</code> 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。
	<code>auto</code>	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。
	<code>all</code>	指定した <code>slice</code> 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。
	<code>free</code>	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 - <code>filesys</code> の値として <code>free</code> を使用する場合は、 <code>filesys</code> 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。
	<code>start:size</code>	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。 <code>start</code> はスライスが始まるシリンダであり、 <code>size</code> はそのスライスのシリンダ数です。
<code>file_system</code>		<code>file_system</code> 値はオプションで、 <code>slice</code> に <code>any</code> または <code>cwtxdysz</code> 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 <code>unnamed</code> がデフォルトで設定されますが、この場合 <code>optional_parameters</code> 値を指定できません。次のいずれかの値を使用します。

<i>mount_pt_name</i>	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)。
<i>swap</i>	指定した <i>slice</i> が swap として使用されます。
<i>overlap</i>	指定した <i>slice</i> 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 - <i>size</i> に existing、all、または <i>start:size</i> を指定した場合だけ <i>overlap</i> を指定できます。	
<i>unnamed</i>	指定した <i>slice</i> が raw スライスとして定義されるので、 <i>slice</i> にはマウントポイント名がありません。 <i>file_system</i> を指定しないと、デフォルトで unnamed が設定されます。
<i>ignore</i>	指定した <i>slice</i> 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。
<i>optional_parameters</i>	次のいずれかの値を使用します。
<i>preserve</i>	指定した <i>slice</i> 上のファイルシステムを保存します。
注 - <i>size</i> に existing、 <i>slice</i> に cwtxdysz を指定した場合だけ <i>preserve</i> を指定できます。	
<i>mount_options</i>	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 <i>mount_pt_name</i> 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: d100)。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、 239 ページの「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: c0t0d0s0、c0t0d0s5 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>/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。

optional_parameters 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 *file_system* の /etc/vfstab 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ンマで区切ってください (例: ro, quota)。

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については、[第 12 章](#)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 を指定すると、クローンシステムがカスタム JumpStart の想定するシステムと異なる場合でも 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。

注 – 非大域ゾーンがインストールされている場合、このキーワードを使用してアップグレードしないでください。このキーワードが使用されると、アップグレードは続行しますが、キーワードは無視されます。

値	説明
N_Africa	北アフリカ。エジプトを含みます

値	説明
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
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

上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。
非大域ゾーンがインストールされている場合は、このキーワードを使用しないでください。	このキーワードが使用されると、アップグレードは停止し、エラーメッセージが表示されます。

制約	説明
layout_constraint 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。 ■ 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、changeable 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 (/etc/vfstab ファイルでマウントされる) も changeable とマークされます。 ■ 残りのファイルシステムは fixed 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 layout_constraint 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。 ■ 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、changeable とマークされます。 ■ layout_constraint 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、fixed とマークされます。
ファイルシステムが changeable 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 (changeable 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 layout_constraint キーワードを使用すれば、その <i>minimum_size</i> 値を変更できます。
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、changeable または movable 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、<i>cwtxdysz</i> または <i>cxdsz</i> 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。 changeable 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。この制約は、/etc/vfstab 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。<i>minimum_size</i> 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。 ファイルシステムを changeable とマークして、<i>minimum_size</i> 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。 <i>minimum_size</i> 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。
<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。この制約は、<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 <code>/usr/share</code> が存在する場合、<code>/usr/share</code> 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 <code>/usr</code> (その親) に移動します。この制約は、<code>/etc/vfstab</code> 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。
<i>minimum_size</i>	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。<i>minimum_size</i> 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 <code>changeable</code> 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 11-11 `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 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`

注 - `locale` 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。

`locale` は、指定した `locale_name` に対して、どのロケールパッケージをインストール (アップグレードの場合は追加) するかを指定します。`locale_name` 値は、`$LANG` 環境変数で使用されるのと同じです。有効なロケール値のリストは、『国際化対応言語環境の利用ガイド』に記載されています。

`locale` 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。

- デフォルトロケールを事前設定している場合は、そのロケールは自動的にインストールされます。English 言語パッケージはデフォルトでインストールされます。
- `locale` キーワードは、システムに追加するロケールごとに指定できます。
- 非大域ゾーンがインストールされている場合、このキーワードを使用してアップグレードしないでください。このキーワードが使用されると、アップグレードは続行しますが、キーワードは無視されます。すでにシステムにインストールされているロケールは、自動的にアップグレードされます。

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)

`metadb slice [size size-in-blocks] [count number-of-replicas]`

metadb キーワードでは、カスタム JumpStart インストール時に、Solaris ボリュームマネージャーの状態データベースの複製 (mediates) を作成できます。metadb キーワードをプロファイルファイル内で複数回使用して、複数のディスクスライス上に状態データベースの複製を作成することができます。

<i>slice</i>	カスタム JumpStart プログラムが状態データベースの複製を配置するディスクスライスを指定する必要があります。 <i>slice</i> の値の形式は、 <i>cwtxdysz</i> です。
<i>size size-in-blocks</i>	<i>size</i> オプションキーワードでは、作成する状態データベースの複製のサイズをブロック単位で指定できます。 <i>size</i> を指定しないと、カスタム JumpStart プログラムは、状態データベースの複製のデフォルトのサイズ、8192 ブロックを使用します。
<i>count number-of-replicas</i>	プロファイル内にオプションの <i>count</i> キーワードを設定することにより、作成する状態データベースの複製の数を指定できます。 <i>count</i> を指定しないと、カスタム JumpStart プログラムは、デフォルトで、状態データベースの複製を 3 つ作成します。

インストール時に Solaris ボリュームマネージャーの状態データベースの複製を作成する方法については、[236 ページの「状態データベースの複製のガイドラインと要件」](#)を参照してください。

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、[141 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、[141 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。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す。

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。

<i>package_name</i>	パッケージ名は <i>SUNWname</i> 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。
<code>add delete</code>	指定されたパッケージを追加または削除します。値 <code>add</code> も <code>delete</code> も指定しなかった場合、デフォルトで <code>add</code> 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。

<code>[retrieval_type location]</code>	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。 <i>retrieval_type</i> と <i>location</i> の値は、パッケージの格納場所によって異なります。次の節では、 <i>retrieval_type</i> と <i>location</i> に指定可能な値と、 <i>package_name</i> キーワードの使用例を示します。
--	--

注 - 非大域ゾーンがインストールされている場合、このキーワードを使用してアップグレードしないでください。このキーワードが使用されると、アップグレードは続行しますが、キーワードは無視されます。

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 <i>SUNWname</i> 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 11-12 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
```

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。
server_name	パッケージを格納したサーバーの名前を指定します。
port	オプションポートを指定します。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。
path	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 11-6 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
timeout min	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。 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 host:port	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 パッケージを取得できます。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。

例 11-13 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 11-14 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
```

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。

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 (`/`) 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 11-15 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。

例 11-15 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 (続き)

```
package SUNWnew add local_device c0t6d0s0 /Solaris_10/Product ufs
```

例 11-16 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 -1` 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 11-17 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かった場合は、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。

default	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、filesys キーワードで指定したファイルシステムを除きます。rootdisk が最初に選択され、指定したソフトウェアが rootdisk 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、filesys 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。filesys 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、filesys 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*

<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。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。
<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>patch_location</i>	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
<i>optional_keywords</i>	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。

注 - 非大域ゾーンがインストールされている場合、このキーワードを使用してアップグレードしないでください。このキーワードが使用されると、アップグレードは続行しますが、キーワードは無視されます。

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_directory [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ー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 11-18 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 11-19 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
```

```
patch patch_id_list | patch_file http server_name[:port] patch_directory optional_http_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。

server_name パッチを格納したサーバーの名前を指定します。

port オプションポートを指定します。*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。

patch_directory 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。

optional_keywords 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 11-7 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 11-20 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ず、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 11-21 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。

<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 パッチ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 11-22 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 11-23 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 11-24 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 11-25 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。217 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。

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 *crdysz* 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 11-26 `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 11-8 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 11-8 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。

表 11-8 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)
(続き)

手順	操作
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。

非大域ゾーンでアップグレードする際のプロファイルキーワードの制限

Solaris 10 1/06 以降のリリースでは、非大域ゾーンがインストールされている場合にカスタム JumpStart プログラムを使用してアップグレードできます。プロファイルには、install_type キーワードおよび root_device キーワードの 2 つのプロファイルキーワードのみを使用する必要があります。

いくつかのキーワードは非大域ゾーンに影響を与えるため、プロファイルに含められないものがあります。たとえば、パッケージを追加するキーワード、ディスク容量を再配置するキーワード、またはロケールを追加するキーワードを使用すると非大域ゾーンに影響を与えます。非大域ゾーンに悪影響を与えるキーワードを使用すると、これらのキーワードが無視されるか、アップグレードが完了せずに停止します。プロファイルに使用するべきではないキーワードの一覧については、次の表を参照してください。

表 11-9 非大域ゾーンでのアップグレードでエラーを引き起こすキーワード

プロファイルキーワード	アップグレードの動作
backup_media	このキーワードを使用するとアップグレードは停止し、エラーメッセージが表示されます。
cluster	このキーワードは無視され、アップグレードが続行します。
geo	このキーワードは無視され、アップグレードが続行します。
layout_constraint	このキーワードを使用するとアップグレードは停止し、エラーメッセージが表示されます。
locale	このキーワードは無視され、アップグレードが続行します。
package	このキーワードは無視され、アップグレードが続行します。
patch	このキーワードは無視され、アップグレードが続行します。

非大域ゾーンの詳細については、次を参照してください。

- [49 ページの「ゾーンのインストールと構成」](#)
- 『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」

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 11-10 では、これらの変数とその値について説明します。

表 11-10 インストール環境変数

環境変数	値
SI_ARCH	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。この変数は、rules ファイルで arch キーワードを使用する場合に設定されます。
SI_BEGIN	開始スクリプトを使用している場合はその名前。
SI_CLASS	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。
SI_DISKLIST	コンマで区切られた、インストールクライアント上のディスク名のリスト。この変数は、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_DISKLIST および SI_NUMDISKS 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、 217 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_DISKSIZE	コンマで区切られた、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。この変数は、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_DOMAINNAME	ドメイン名。この変数は、rules ファイルで domainname キーワードを使用して照合する場合に設定されます。
SI_FINISH	終了スクリプトを使用する場合はその名前。
SI_HOSTADDRESS	インストールクライアントの IP アドレス。
SI_HOSTNAME	インストールクライアントのホスト名。この変数は、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)。この変数は、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。

表 11-10 インストール環境変数 (続き)

環境変数	値
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。この変数は、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。この変数は、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。この変数は、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。この変数は、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。この変数は、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、 217 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。この変数は、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_ROOTDISKSIZE	論理名 rootdisk によって表されるディスクのサイズ。この変数は、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_TOTALDISK	インストールクライアント上のディスク容量の合計。この変数は、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 11-11 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robe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 11-11 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_DISKSIZE、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 ハードウェアマニュアル』(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。
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。

表 11-11 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
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 III RAID-1 ボリュームの使用

このパートでは、Solaris のインストールやアップグレードに使用する Solaris ボリュームマネージャーコンポーネントの概要について説明します。このパートでは、RAID-1 ボリュームを使用するためのガイドラインと要件についても説明します。

第 12 章

インストール時の RAID-1 ボリューム (ミラー) の作成 (概要)

この節では、ミラー化されたファイルシステムを作成する利点について説明します。ミラー化されたファイルシステムの作成に必要な Solaris ボリュームマネージャーコンポーネントについても説明します。

この章の内容は次のとおりです。

- 227 ページの「RAID-1 ボリュームを使用する理由」
- 228 ページの「RAID-1 ボリュームの機能」
- 230 ページの「Solaris ボリュームマネージャーコンポーネントの概要」
- 233 ページの「RAID-1 ボリュームのディスク配置の例」

Solaris Live Upgrade を使って、ミラー化されたファイルシステムを作成する方法については、『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「RAID-1 ボリューム (ミラー) ファイルシステムを作成するための一般的な指針」を参照してください。

カスタム JumpStart インストールを使って、ミラー化されたファイルシステムを作成する方法については、197 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)」および 203 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。

RAID-1 ボリュームを使用する理由

インストール時、またはアップグレード時に、RAID-1 ボリュームを作成して、複数の物理ディスクにシステムデータを複製できます。複数のディスクにデータを複製することにより、ディスクの破壊やディスク障害の際にデータを保護することができます。

Solaris カスタム JumpStart および Solaris Live Upgrade では、ファイルシステムをミラー化する RAID-1 ボリュームの作成に、Solaris ボリュームマネージャーを使用します。Solaris ボリュームマネージャーでは、ボリュームを使って確実にディスクやデー

タを管理できます。Solaris ボリュームマネージャーでは、連結、ストライプ、その他の複雑な構成が可能です。カスタム JumpStart および Solaris Live Upgrade インストールでは、これらの作業の一部が実行できます。たとえば、ルート (/) ファイルシステムの RAID-1 ボリュームを作成できます。RAID-1 ボリュームは、インストール時、またはアップグレード時に作成できるので、インストール後に作成する必要はありません。

- ガイドラインについては、[238 ページの「カスタム JumpStart と Solaris Live Upgrade のガイドライン」](#)を参照してください。
- Solaris ボリュームマネージャーソフトウェアとそのコンポーネントの詳細については、『Solaris ボリュームマネージャの管理』を参照してください。

RAID-1 ボリュームの機能

Solaris ボリュームマネージャーは、仮想ディスクを使って物理ディスクと関連データを管理します。Solaris ボリュームマネージャーでは、仮想ディスクを「ボリューム」と呼びます。「ボリューム」とは、システム上で単一の論理デバイスとみなされる物理スライスの集まりの名前です。ボリュームは、一般的な UNIX[®] 用語である「擬似 (仮想) デバイス」と、実質的に同義です。

アプリケーションやファイルシステム (UFS など) から見ると、ボリュームは物理ディスクと同じように機能します。Solaris ボリュームマネージャーは、ボリュームに対する入出力要求を、そのボリュームを構成するメンバーディスクに対する入出力要求に変換します。

Solaris ボリュームマネージャーのボリュームは、スライス (ディスクパーティション) またはほかの Solaris ボリュームマネージャーボリュームから作成されます。

ボリュームを使用して、パフォーマンスとデータ可用性を向上させることができます。場合によっては、ボリュームの使用により入出力パフォーマンスも向上します。ボリュームの機能は、スライスと同じです。ボリュームはスライスとよく似ていますが、エンドユーザー、アプリケーション、およびファイルシステムに対して透過的です。物理デバイスと同様に、Solaris ボリュームマネージャーを使用して、ブロックデバイス名または raw デバイス名からボリュームにアクセスできます。ボリューム名は、使用しているのがブロックデバイスなのか raw デバイスなのかによって異なります。

カスタム JumpStart インストールおよび Solaris Live Upgrade では、ミラー化されたファイルシステムの作成用としてブロックデバイスがサポートされます。ボリューム名の詳細については [239 ページの「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。

RAID-0 ボリューム (単一スライスの連結) と RAID-1 ボリュームを作成すると、Solaris ボリュームマネージャーは、連結 (サブミラー) 上にデータを複製し、これらのサブミラーを 1 つのボリュームとして扱います。

図 12-1 に、ルート (/) ファイルシステムを 2 つの物理ディスクに複製するミラーを示します。

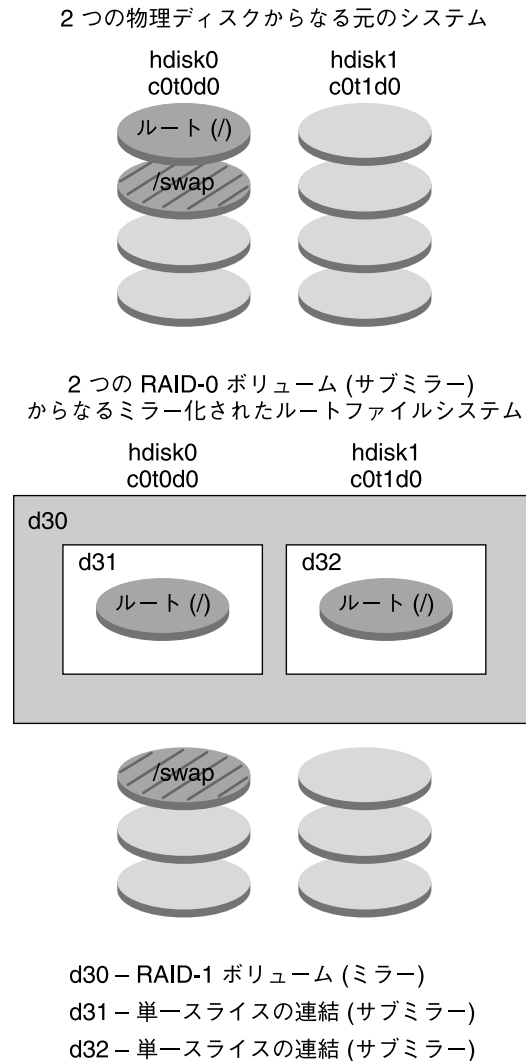


図 12-1 2 つのディスクにルートファイルシステムの RAID-1 ボリュームを作成

図 12-1 のシステムの構成は、次のとおりです。

- hdisk0 上のルートファイルシステム (/) は、d31 という名前の単一スライスの連結に含まれています。
- hdisk1 という名前のハードディスク上に、d32 という名前の単一スライスの連結が作成されています。

- d30 という名前のミラーは、d31 および d32 という名前のサブミラーで構成されています。
- このミラーは、ルートファイルシステム内のデータを 2 つのサブミラーに複製しています。

Solaris ボリュームマネージャーコンポーネントの概要

カスタム JumpStart インストールおよび Solaris Live Upgrade では、データの複製に必要な次のコンポーネントを作成できます。

- 状態データベースと状態データベースの複製 (metadbs)
- 単一スライスの連結 (サブミラー)
- RAID-1 ボリューム (ミラー)

この節では、これらのコンポーネント 1 つ 1 つについて簡単に説明します。これらのコンポーネントの詳細は、『Solaris ボリュームマネージャの管理』を参照してください。

状態データベースと状態データベースの複製

「状態データベース」は、Solaris ボリュームマネージャー構成の状態に関する情報を物理ディスクに格納するデータベースです。状態データベースは、構成に対して加えられた変更を記録および管理します。Solaris ボリュームマネージャーは、構成や状態に変化があると、状態データベースを自動的に更新します。新しいボリュームの作成は、構成の変更の一例です。サブミラーの障害は、状態の変化の一例です。

状態データベースは、実際には、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、データベース内のデータが常に有効であることを保証します。状態データベースのコピーを複数持つことにより、単一点障害からデータを保護することができます。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。

状態データベースとその状態データベースの複製が作成されるまで、Solaris ボリュームマネージャーは動作できません。Solaris ボリュームマネージャー構成には、正常に動作する状態データベースが必要です。

構成を設定するときは、状態データベースの複製を次のどちらかに配置できます。

- 専用のスライス
- (Solaris Live Upgrade のみ) あとでボリュームの一部になるスライス

複数の状態データベースのコピーを1つのスライス上に置くこともできます。しかし、複数の状態データベースの複製を1つのスライスに置くと、システムが単一点障害に対してより脆弱になる可能性があります。

状態データベースの複製は、状態データベースのデータが常に有効であることを保証します。状態データベースが更新されると、個々の状態データベースの複製も更新されます。ただし、システムクラッシュによってすべての更新が失われるのを防ぐために、更新は一度に1つずつ行われます。

システムから1つの状態データベースの複製が失われると、Solaris ボリュームマネージャーは、どの状態データベースの複製に有効なデータが格納されているかを判断する必要があります。この情報を得るために、Solaris ボリュームマネージャーは「多数決アルゴリズム」を使用します。このアルゴリズムでは、過半数 (半数 + 1) の複製が使用可能であり、一致していれば、それらの複製を有効であるとみなします。この多数決アルゴリズムがあるため、ディスク構成を設定するときに、3つ以上の状態データベースの複製を作成する必要があります。3つの状態データベースの複製のうち少なくとも2つが使用可能であれば、合意に達することができます。

個々の状態データベースの複製には、デフォルトで4M バイト (8192 ディスクセクタ) のディスク領域が使用されます。複製は、次のデバイスに格納できます。

- 専用のローカルディスクスライス
- (Solaris Live Upgrade のみ) あとでボリュームの一部になるローカルスライス
- (Solaris Live Upgrade のみ) あとで UFS ロギングデバイスの一部になるローカルスライス

複製は、ルート (/)、swap、/usr スライス、およびファイルシステムやデータがすでに格納されているスライスには格納できません。ただし、複製を格納したあとで、同じスライスにボリュームやファイルシステムを置くことができます。

カスタム JumpStart または Solaris Live Upgrade を使って RAID-1 ボリュームを作成するときに確認するガイドラインと要件	236 ページの「状態データベースの複製のガイドラインと要件」
状態データベースと状態データベースの複製の詳細情報	『Solaris ボリュームマネージャの管理』

RAID-0 ボリューム (連結)

カスタム JumpStart と Solaris Live Upgrade のどちらのインストール方法でも、RAID-0 ボリュームを作成できます。RAID-0 ボリュームの単一スライス連結は、個々のコンポーネント内にデータを順番に隣接して配置し、1つの論理記憶ユニットを構成します。カスタム JumpStart インストールおよび Solaris Live Upgrade では、ストライプの作成や、その他の複雑な Solaris ボリュームマネージャーボリュームは作成できません。

インストール時、またはアップグレード時に、RAID-1 ボリューム (ミラー) を作成し、これらのミラーに RAID-0 ボリュームを追加できます。「ミラー化された」 RAID-0 ボリュームを「サブミラー」と呼びます。ミラーは 1 個以上の RAID-0 ボリュームで構成されます。インストール後、Solaris ボリュームマネージャーを使用して RAID-1 ミラーボリュームを管理することにより、個々の RAID-0 サブミラーボリューム上のデータを管理できます。

カスタム JumpStart インストールでは、最大 2 つのサブミラーで構成されるミラーを作成できます。Solaris Live Upgrade では、最大 3 つのサブミラーで構成されるミラーを作成できます。実際には 2 面ミラーで十分です。3 つ目のサブミラーを構成すると、オンラインでバックアップをとることができます。この場合、バックアップのために 1 つのサブミラーがオフラインになっていても、データの冗長性は失われません。

RAID-0 ボリュームの計画情報

237 ページの「RAID-1 ボリュームと RAID-0 ボリュームの要件とガイドライン」

RAID-0 ボリュームの詳細情報

『Solaris ボリュームマネージャの管理』

RAID-1 ボリューム (ミラー)

RAID-1 ボリューム (ミラー) は、同じデータのコピーを複数の RAID-0 ボリューム (単一スライスの連結) に保持しているボリュームです。RAID-1 ボリュームを使用してファイルシステムをミラー化する場合、より多くのディスク容量が必要です。少なくとも、データの容量の 2 倍のディスク容量が必要です。Solaris ボリュームマネージャーソフトウェアは、すべての RAID-0 ボリュームに書き込む必要があるため、データを複製すると、書き込み要求がディスクに書き込まれるまでの時間も長くなる可能性があります。

RAID-1 ボリュームでは、両方の RAID-0 ボリュームから同時にデータを読み取ることができるので (どちらのボリュームもすべての要求に応じることができる)、パフォーマンスが向上します。1 つの物理ディスクに障害が発生しても、パフォーマンスの低下やデータの損失なしにミラーを引き続き使用できます。

RAID-1 ボリュームの構成後、このボリュームを物理スライスと同様に使用できます。

既存のファイルシステムを含め、どのようなファイルシステムでも複製できます。RAID-1 ボリュームは、データベースなど、任意のアプリケーションでも使用できます。

RAID-1 ボリューム計画情報

237 ページの「RAID-1 ボリュームと RAID-0 ボリュームの要件とガイドライン」

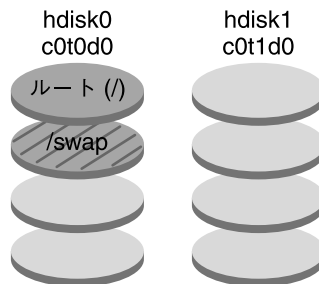
RAID-1 ボリュームの詳細情報

『Solaris ボリュームマネージャの管理』

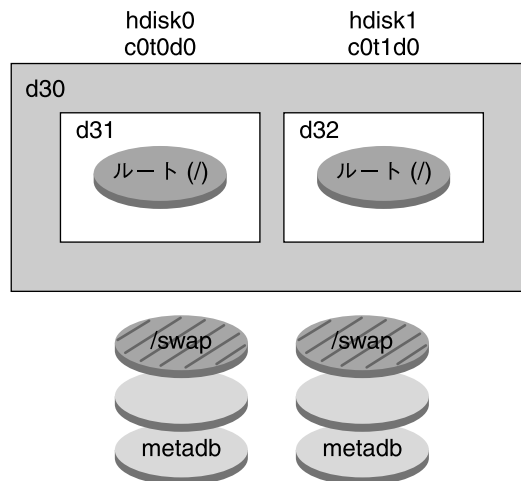
RAID-1 ボリュームのディスク配置の例

次の図は、ルートファイルシステム (/) を 2 つの物理ディスク上に複製する RAID-1 ボリュームです。状態データベースの複製 (metadb) は、両方のディスクに配置されています。

2 つの物理ディスクからなる元のシステム



2 つの RAID-0 ボリュームからなるミラー化されたファイルシステムと状態データベースの複製 (metadb)



d30 – RAID-1 ボリューム (ミラー)
d31 – 単一スライスの連結 (サブミラー)
d32 – 単一スライスの連結 (サブミラー)

図 12-2 RAID-1 ボリュームのディスク配置

図 12-2 のシステムの構成は、次のとおりです。

- `hdisk0` 上のルートファイルシステム (`/`) は、`d31` という名前の単一スライスの連結に含まれています。
- `hdisk1` という名前のハードディスク上に、`d32` という名前の単一スライスの連結が作成されています。
- `d30` という名前のミラーは、`d31` および `d32` という名前のサブミラーで構成されています。
- このミラーは、ルートファイルシステム内のデータを2つのサブミラーに複製しています。
- `hdisk0` と `hdisk1` の両方のスライスに状態データベースの複製が作成されています。

カスタム JumpStart インストールを使用して 例 6-13
この構成を作成するプロファイル例

Solaris Live Upgrade で RAID-1 ボリュームを作成する方法	『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「RAID-1 ボリューム (ミラー) を持つブート環境の作成 (コマンド行インタフェース)」
--	---

第 13 章

インストール時の RAID-1 ボリューム (ミラー) の作成 (計画)

この章では、カスタム JumpStart または Solaris Live Upgrade インストールを使って RAID-1 ボリュームを作成するために必要な条件とガイドラインについて説明します。

この章の内容は次のとおりです。

- 235 ページの「システム要件」
- 236 ページの「状態データベースの複製のガイドラインと要件」
- 237 ページの「RAID-1 ボリュームと RAID-0 ボリュームの要件とガイドライン」
- 242 ページの「シングルユーザーモードでのブートが RAID-1 ボリュームに与える影響」

Solaris Live Upgrade インストールを使って、ミラー化されたファイルシステムを作成する計画の詳細については、『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「RAID-1 ボリューム (ミラー) ファイルシステムを作成するための一般的な指針」を参照してください。

カスタム JumpStart インストールを使って、ミラー化されたファイルシステムを作成する方法については、197 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)」および 203 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。

システム要件

RAID-1 ボリュームを作成して、特定のスライスにデータを複製するには、インストール時に、使用するディスクがシステムに直接接続されていて使用可能である必要があります。

状態データベースの複製のガイドラインと要件

単一点障害を避けるため、状態データベースの複製は、複数のスライス、ドライブ、およびコントローラに分散させる必要があります。これは、単一のコンポーネントに障害が発生した場合でも、大半の複製を利用可能な状態に保つ必要があるからです。たとえばデバイス障害時などに、複製が失われた場合、Solaris ボリュームマネージャーの実行やシステムの再起動が正常に行われなくなることがあります。Solaris ボリュームマネージャーが動作するためには、少なくとも半数の複製が使用可能でなければならず、システムをマルチユーザーモードで再起動するためには過半数 (半数+1) の複製が使用可能でなければなりません。

状態データベースの複製の作成および管理方法の詳細は、『Solaris ボリュームマネージャの管理』を参照してください。

状態データベースの複製用のスライスの選択

状態データベースの複製用のスライスを選択する前に、次のガイドラインと推奨事項を参考にしてください。

- 状態データベースの複製は、4M バイト以上の容量を持つ専用スライス上に作成します。必要な場合は、あとで RAID-0 または RAID-1 ボリュームの一部とするスライス上にも、状態データベースの複製を作成できます。ただし、その場合は、スライスをボリュームに追加する前に複製を作成する必要があります。
- 状態データベースの複製のデフォルトサイズは 4M バイト (8192 ディスクブロック) です。ディスクスライスのサイズがこれより大きい場合は、状態データベースの複製を格納できるように、スライスのサイズを変更できます。スライスサイズの変更については、『Solaris のシステム管理 (デバイスとファイルシステム)』の第 12 章「ディスクの管理 (手順)」を参照してください。
- 状態データベースの複製は、未使用のスライス上に作成できます。状態データベースの複製用に予約されているスライスの部分を、ほかの目的に使用することはできません。
- 状態データベースの複製を、既存のファイルシステムや、ルート (/)、/usr、swap ファイルシステムに作成することはできません。必要であれば、swap 領域を使用して新しいスライスを作成してから (スライス名が使用可能な場合)、そのスライスに状態データベースの複製を作成できます。
- ボリュームの一部となるスライス上に状態データベースの複製が置かれている場合、ボリュームの容量は、複製によって占有される領域分だけ少なくなります。複製が占める領域はシリンダ単位で切り上げられるため、この領域はボリュームによってスキップされます。

状態データベースの複製の数の選択

状態データベースの複製の数を選択する前に、次のガイドラインを参考にしてください。

- 状態データベースの複製の数は、Solaris ボリュームマネージャーの 1 つのディスクセットに対して、最低 3 つから最高 50 までを推奨します。次のガイドラインを推奨します。
 - ドライブが 1 つだけのシステムでは、3 つの複製すべてを 1 つのスライスに置きます。
 - ドライブの数が 2 つから 4 つのシステムでは、各ドライブに 2 つずつ複製を置きます。
 - ドライブの数が 5 つ以上のシステムでは、各ドライブに 1 つずつ複製を置きます。
- 状態データベースの複製を追加することで、ミラーのパフォーマンスを向上させることができます。一般に、システムにミラーを 1 つ追加するごとに複製は 2 つ追加する必要があります。
- 小容量のランダム入出力 (データベースなど) に RAID-1 ボリュームを使用する場合は、複製の数を考慮する必要があります。RAID-1 ボリュームごとに、その RAID-1 ボリュームに接続されていない複数のスライス (および、可能であれば複数のディスクとコントローラ) 上に 2 つ以上の複製を余分に作成します。これは、最適な性能を得るために必要な作業です。

コントローラ間で状態データベースの複製を分散

複数のコントローラがある場合、できるだけすべてのコントローラに均等になるように複製を分散させます。これによって、コントローラ障害に対する冗長性が確保できるだけでなく、負荷の分散も可能になります。同じコントローラ上に複数のディスクが存在する場合は、各コントローラで 2 個以上のディスクに複製を配置します。

RAID-1 ボリュームと RAID-0 ボリュームの要件とガイドライン

RAID-1 ボリューム (ミラー) と RAID-0 ボリューム (単一スライスの連結) を使用する際は、次のガイドラインに従ってください。

カスタム JumpStart と Solaris Live Upgrade のガイドライン

カスタム JumpStart インストールと Solaris Live Upgrade は、Solaris ボリュームマネージャーで使用可能な機能の一部をサポートします。これらのインストールプログラムでミラー化されたファイルシステムを作成するときは、次のガイドラインに従ってください。

インストールプログラム	サポートされている機能	サポートされていない機能
カスタム JumpStart と Solaris Live Upgrade	■ RAID-0 および RAID-1 ボリュームはサポートされますが、RAID-5 ボリュームなどほかの Solaris ボリュームマネージャーコンポーネントはサポートされません。 ■ RAID-0 ボリュームは、単一スライスの連結としてのみサポートされています。	Solaris ボリュームマネージャーでは、RAID-0 ボリュームは、ディスクドライブまたはディスク連結を表します。インストール時またはアップグレード時に RAID-0 ストライブボリュームを作成することはできません。
カスタム JumpStart	■ 初期インストール時にのみ RAID-1 ボリュームの作成がサポートされます。 ■ 1 つの RAID-1 ボリュームに対して、最大 2 つの RAID-0 ボリューム (サブミラー) を作成できます。通常、ほとんどのアプリケーションでは、2 つのサブミラーで十分なデータの冗長性が得られます。ディスクドライブのコストも比較的小さくて済みます。	■ RAID-1 ボリュームが構成されている場合のアップグレードはサポートされません。 ■ 3 つ以上の RAID-0 ボリュームはサポートされません。
Solaris Live Upgrade	■ 1 つの RAID-1 ボリュームに対して、最大 3 つの RAID-0 ボリューム (サブミラー) を作成できます。3 つのサブミラーでは、1 つのサブミラーをオフラインにしてバックアップを実行するときも、残りの 2 つのサブミラーでデータの冗長性を確保することができます。 ■ アップグレード時の RAID-1 ボリュームの作成がサポートされます。 例については、『Solaris 10 インストールガイド (Solaris Live Upgrade とアップグレードの計画)』の「RAID-1 ボリューム (ミラー) を持つブート環境の作成 (コマンド行インタフェース)」を参照してください。	4 つ以上の RAID-0 ボリュームはサポートされません。

インストールプログラム	サポートされている機能	サポートされていない機能
RAID-1 ボリュームを使用した Solaris フラッシュの作成とインストール	Solaris 9 9/04 以降のリリースでは、RAID-1 ボリュームが構成されているマスターシステムから Solaris フラッシュアーカイブを作成してインストールすることができます。 Solaris 9 12/03 リリースと Solaris 9 4/04 リリースには、パッチをインストールする必要があります。CR 4838219 を修正するパッチを入手するには、sunsolve.sun.com を参照してください。 Solaris ボリュームマネージャー RAID-1 ボリュームが構成されていても、Solaris フラッシュアーカイブを作成できます。クローンシステムの整合性を保つため、RAID-1 ボリュームの情報はすべて、Solaris フラッシュ作成ソフトウェアによってアーカイブから削除されます。カスタム JumpStart では、JumpStart プロファイルを使用して RAID-1 ボリュームを再構築できます。 Solaris Live Upgrade では、RAID-1 ボリュームが構成されたブート環境を作成し、アーカイブをインストールできます。Solaris インストールプログラムでは、Solaris フラッシュアーカイブを使用して RAID-1 ボリュームのインストールを行うことはできません。 JumpStart プロファイルでの RAID-1 ボリュームの例については、98 ページの「プロファイルの例」を参照してください。	Veritas VxVM では、Solaris フラッシュで使えない領域に構成情報が格納されます。Veritas VxVm ファイルシステムが構成されている場合は、Solaris フラッシュアーカイブを作成しないでください。また、JumpStart と Solaris Live Upgrade も含め、Solaris インストールではインストール時の VxVM ボリュームの再構築はサポートされていません。したがって、Solaris フラッシュアーカイブを使った Veritas VxVM ソフトウェアの配備を計画している場合は、VxVM ファイルシステムを構成する前にアーカイブを作成する必要があります。その後、クローンシステムにアーカイブを適用しシステムをリブートしてから、クローンシステムの構成を個別に行う必要があります。

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

ボリュームに名前を割り当てるときには、次の規則に従ってください。

- スライス番号とディスク番号がボリューム番号に対応するような命名方法を使用します。
- ボリューム名は d で始まり、その後に 1 つの数字が続きます (たとえば、d0)。
- Solaris ボリュームマネージャーでは、0 から 127 までの 128 個のデフォルトボリューム名を使用できます。次にボリューム名の例を示します。
 - デバイス /dev/md/dsk/d0 - ブロックボリューム d0
 - デバイス /dev/md/dsk/d1 - ブロックボリューム d1
- 特定のボリュームタイプごとに範囲を指定します。たとえば、RAID-1 ボリュームには 0 から 20、RAID-0 ボリュームには 21 から 40 を割り当てます。
- 完全ボリューム名 (たとえば、/dev/md/dsk/d1) ではなく、省略名 (たとえば、d1) を指定できます。

Solaris Live Upgrade を行うときの RAID ボリュームの命名規則

物理ディスクスライスや Solaris ボリュームマネージャーのボリュームの名前は、省略形にすることができます。省略名は、デバイスを一意に識別できる最短の名前です。次に例を示します。

- Solaris ボリュームマネージャーのボリュームは `dnum` という形式で表されます。たとえば、`/dev/md/dsk/d10` は `d10` となります。
- 1 つのコントローラと複数のディスクを持つシステムでは `t0d0s0` を使用できますが、複数のコントローラがある場合は `c0t0d0s0` を使用します。

RAID-1 ボリューム (ミラー) または RAID-0 ボリューム (サブミラー) を作成するのに Solaris Live Upgrade を使用する場合は、ソフトウェアでボリューム名を検出して割り当てるか、またはユーザーがボリューム名を割り当てることができます。ソフトウェアで名前を検出すると、ソフトウェアは使用可能な最初のミラー名またはサブミラー名を割り当てます。ユーザーがミラー名を割り当てる場合は、インストール時にサブミラーに 1 および 2 で終わる名前を使用できるように、0 で終わる名前を割り当てます。ユーザーがサブミラー名を割り当てる場合は、1 または 2 で終わる名前を割り当てます。誤った番号を割り当てると、ミラーが作成されない可能性があります。たとえば、ミラー名に 1 または 2 で終わる番号 (`d1` または `d2`) を持つ名前を指定すると、ミラー名がサブミラーの名前と重複した場合、Solaris Live Upgrade はミラーの作成に失敗します。

次の例では、Solaris Live Upgrade がボリューム名を割り当てています。RAID-1 ボリュームの `d0` と `d1` だけが使用中のボリュームです。ミラー `d10` に対し、デバイス `c0t0d0s0` 用のサブミラー名として `d2` を、デバイス `c1t0d0s0` 用のサブミラー名として `d3` を、Solaris Live Upgrade が割り当てます。

```
lucreate -n newbe -m /:d10:mirror,ufs -m /:c0t0d0s0:attach -m  
/:c1t0d0s0:attach
```

次の例では、コマンドでボリューム名を割り当てています。ミラー `d10` に対し、デバイス `c0t0d0s0` 用のサブミラー名が `d11`、デバイス `c1t0d0s0` 用のサブミラー名が `d12` です。

```
lucreate -n newbe -m /:d10:mirror,ufs -m /:c0t0d0s0,d11:attach -m  
/:c1t0d0s0,d12:attach
```

Solaris ボリュームマネージャーの命名規則については、『Solaris ボリュームマネージャの管理』を参照してください。

カスタム JumpStart を行うときの RAID ボリュームの命名規則

RAID-1 ボリューム (ミラー) または RAID-0 ボリューム (サブミラー) を作成するのにカスタム JumpStart インストール方式を使用する場合は、ソフトウェアでミラーリングするボリューム名を検出して割り当てるか、またはプロファイルでボリューム名を割り当てることができます。ソフトウェアで名前を検出すると、ソフトウェアで使用可能な最初のボリューム番号を割り当てます。プロファイルでボリューム名を割り当てる場合は、インストール時にサブミラーに 1 および 2 で終わる名前を使用できるように、0 で終わるミラー名を割り当てます。誤った番号を割り当てると、ミラーが作

成されない可能性があります。たとえば、ミラー名に 1 または 2 で終わる番号 (d1 または d2) を持つ名前を指定すると、ミラー名がサブミラーの名前と重複した場合、JumpStart プログラムはミラーの作成に失敗します。次のプロファイル例では、ミラーには使用可能な最初のボリューム番号が割り当てられています。次に使用可能な 0 で終わるミラーが d10 の場合、名前 d11 および d12 はサブミラーに割り当てられます。

```
filesys                                mirror c0t0d0s1 /
```

次のプロファイル例では、プロファイルでミラー番号に d30 が割り当てられています。サブミラー名は、ミラー番号に基づいたソフトウェアおよび最初に使用可能なサブミラーによって割り当てられます。この例では、サブミラーは d31 および d32 と命名されます。

```
filesys                                mirror:d30 c0t1d0s0 c0t0d0s0 /
```

Solaris ボリュームマネージャーの命名規則については、『Solaris ボリュームマネージャーの管理』を参照してください。

ディスクとコントローラの選択のガイドライン

ファイルシステムのミラー化に使用するディスクやコントローラを選択するときは、次のガイドラインに従ってください。

- コンポーネントをそれぞれ異なるコントローラに置くと、同時に実行できる読み取りや書き込みの数が増えます。
- サブミラーのスライスは、異なるディスクとコントローラに配置します。同じミラーの 2 つ以上のサブミラーのスライスを同じディスクに置くと、データの保護機能が大幅に低下します。
- サブミラーは、別個のコントローラに配置します。これは、コントローラやそのケーブルでは、ディスクよりも障害が発生する確率が高いためです。これにより、ミラーのパフォーマンスも向上します。
- 1 つのミラーでは、同じタイプのディスクとコントローラを使用します。特に、古いタイプの SCSI 記憶装置では、ディスクやコントローラのパフォーマンスがモデルやブランドによって大幅に異なることがあります。パフォーマンスレベルが異なるデバイスが同じミラーに混在していると、パフォーマンスが大幅に低下することがあります。

スライスの選択のガイドライン

ファイルシステムのミラー化に使用するスライスを選択するときは、次のガイドラインに従ってください。

- ルート (/)、swap、/usr を含むどのファイルシステムでもミラーを使用できます。また、データベースをはじめとするどのアプリケーションでもミラーを使用できます。

- サブミラーサイズが同じサイズになっていることを確認してください。サイズが異なるサブミラーを使用すると、ディスク領域が無駄になります。
- ミラー化されたファイルシステムで、最初に接続したサブミラーがシリンダ 0 から始まらない場合、追加接続するすべてのサブミラーも、シリンダ 0 から始まらないようにする必要があります。最初のサブミラーがシリンダ 0 から始まらないミラーに、シリンダ 0 から始まるサブミラーを接続しようとする、次のエラーメッセージが表示されます。

```
can't attach
labeled submirror to an unlabeled mirror
```

1 つのミラーに接続するサブミラーは、全部シリンダ 0 から始まるか、どれもシリンダ 0 から始まらないかのどちらかにする必要があります。

開始シリンダは、すべてのサブミラーで同じにする必要はありませんが、すべてのサブミラーにシリンダ 0 が含まれるか、すべてのサブミラーにシリンダ 0 が含まれないかのどちらかでなければなりません。

シングルユーザーモードでのブートが RAID-1 ボリュームに与える影響

ルート (/)、/usr、および swap のミラーを持つシステムをシングルユーザーモードでブートした場合、これらのミラーの保守管理が必要であることが、システムから通知されます。metastat コマンドでこれらのミラーを確認すると、「Needing Maintenance」という状態情報が表示されます。システム上のすべてのミラーでこの現象が起きる場合もあります。

これは危険な状況に見えますが、心配はいりません。metasync -r コマンドは通常、ブート時にミラーの再同期のために実行されますが、システムがシングルユーザーモードでブートされた場合には実行を中断されます。システムをリブートすると、metasync -r コマンドが実行され、すべてのミラーの再同期が取られます。

この中断が問題になる場合は、手動で metasync -r コマンドを実行してください。

metasync の詳細については、metasync (1M) のマニュアルページと、『Solaris ボリュームマネージャの管理』を参照してください。

パート **IV** 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 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。

- [245 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」](#)
- [246 ページの「システムのブートに関する問題」](#)
- [253 ページの「Solaris OS の初期インストール」](#)
- [255 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` コマンドの引数に指定した *host_name* がネームサービス内のホストではありません。

説明: 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。

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 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 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 を使用)、ブートディスクを選択しませんでした。Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスクを使用するか、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: 次の手順に従ってください。

- **Solaris 10 3/05** リリースの場合は、Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスクをシステムのブートフロッピーディスクドライブ (通常はドライブ A) に挿入します。Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスクへのアクセス方法については、『Solaris 10 インストールガイド (ネットワークインストール)』の「x86 版 Solaris 10 3/05: ブートソフトウェアのフロッピーディスクへのコピー」を参照してください。
- **Solaris 10 1/06** 以降のリリースでは、ブートする BIOS を選択します。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
```

x86 システムの場合、インストールプログラムにより「Select type of installation」が表示されたら、次のコマンドを入力します。

```
b - -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。

ブート前の段階で、Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスクが、システムの IDE BIOS プライマリドライブを検出できなかった。(x86 システムのみ)

対処方法: 次の手順に従ってください。

- 古いドライブを使用している場合、サポートされていないことがあります。ハードウェアのマニュアルを参照してください。
- リボンと電源ケーブルが正しく接続されているか確認します。ハードウェアのマニュアルで確認してください。
- 1 台のドライブだけがコントローラに接続されている場合、ジャンパを設定して、そのドライブをマスタードライブとして指定します。一部のドライブでは、単一マスター用のジャンパ設定が、スレーブといっしょに動作するマスター用のジャンパ設定と異なる場合があります。未使用のコネクタがケーブルの終端にあるときに発生する信号障害を抑制するために、ケーブルの終端にあるコネクタにドライブを接続します。
- 2 台のドライブがコントローラに接続されている場合、1 台のドライブをマスター (またはスレーブと一緒に動作するマスター)、もう 1 台のドライブをスレーブとして、ジャンパ設定します。
- 1 台目のドライブがハードディスクで、2 台目のドライブが CD-ROM ドライブの場合、ジャンパを設定して、どちらかのドライブをスレーブドライブとして指定します。どちらの物理ドライブをスレーブドライブに指定してもかまいません。
- 単一のコントローラ上の 2 つのドライブで問題が継続して発生する場合、一度に 1 つのドライブを接続して、各ドライブの動作を確認します。ドライブをマスターまたは単一マスターとしてジャンパを設定して、IDE リボンケーブルの

終端にあるドライブコネクタを使用してドライブを接続します。各ドライブが動作することを確認してから、マスターとスレーブのドライブ構成にジャンパ設定を戻します。

- ドライブがディスクドライブの場合、BIOS 設定画面を使用して、ドライブタイプ (シリンダ、ヘッド、セクターの数を示す) が正しく構成されていることを確認します。一部の BIOS ソフトウェアは、ドライブタイプを自動的に検出する機能を持っています。
- ドライブが CD-ROM ドライブの場合、BIOS 設定画面を使用して、ドライブタイプを CD-ROM ドライブとして構成します (BIOS ソフトウェアがこの機能を持っている場合のみ)。
- ほとんどのシステムでは、MS-DOS CD-ROM ドライバがインストールされている場合、IDE の CD-ROM ドライブは MS-DOS でしか認識されません。別のドライブで試してみてください。

ブート前の段階で、Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスクが、システムの IDE ディスクまたは CD-ROM ドライブを検出できなかった。(x86 システムのみ)

対処方法: 次の手順に従ってください。

- **Solaris 10 3/05** リリースの場合、ディスクが BIOS で無効になっているときは、Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスクを使用してハードディスクから起動します。Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) へのアクセス方法については、『Solaris 10 インストールガイド (ネットワークインストール)』の「x86 版 Solaris 10 3/05: ブートソフトウェアのフロッピーディスクへのコピー」を参照してください。
- システムがディスクを持っていない場合は、ディスクレスクライアントにします。

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名前サービスの bootparams マップまたはテーブル (* install_config= など) にある場合は、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 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、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` 情報を探することを示しています。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` ファイル内を探します。

```
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からブートされるが、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。詳細については、『Solaris 10 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/10/SUNWxxx/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 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスク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 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスク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: 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。

手順 1. インストールメディアに応じた方法でブートします。

- **Solaris 10 3/05** リリースの場合は、シングルユーザーモードでメディアからブートします。

```
# b -s
```

- **Solaris 10 1/06** 以降のリリースでは、次の手順に従います。

a. インストールメディアからブートします。

b. 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。

2. **format (1M)** プログラムを起動します。

```
# format
```

3. 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。

```
# cxdy
```

cx コントローラ番号

dy デバイス番号

4. **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 5](#) 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
```

5. 次のように入力して、表面解析を開始します。

```
format> analyze
```

6. 次のように入力して、現在の設定を確認します。

```
analyze> config
```

7. (省略可能) 次のように入力して、設定を変更します。

```
analyze> setup
```

8. 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。

9. 次のように入力して、解析を終了します。

```
analyze> quit
```

10. 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 11](#) 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
```

11. 次のように入力して、`format` プログラムを終了します。

```
quit
```

12. マルチユーザーモードでメディアを再起動します。

- **Solaris 10 3/05** リリースの場合は、次のコマンドを入力します。

```
ok b
```

- **Solaris 10 1/06** 以降のリリースの場合は、次のコマンドを入力します。

```
# 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。

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: 1 /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: 1 /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: 2 /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: 3 /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: 46 ページの「ディスク容量の再配置を伴うアップグレード」で容量の問題をチェックし、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 (/) ファイルシステムになっている Solaris ボリュームマネージャの RAID-1 ボリュームを使用している場合に、アップグレードの問題が発生したときは、『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障害追跡 (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。

- 手順
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

Solaris 10 1/06 以降のリリースでは、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) コマンドおよび 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。

- 手順 1. 非アクティブブート環境を作成します。『**Solaris 10 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。

2. 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** 上に存在す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アウトします。**drv/vx** を含むものが対象です。

```
# 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
```

3. 非アクティブブート環境をアップグレードします。『**Solaris 10** 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 9 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。

4. 非アクティブブート環境をアクティブにします。『**Solaris 10** 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。

5. システムをシャットダウンします。

```
# init 0
```

6. 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。

7. **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 VRTSvmman VRTSvmdev
```

8. 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
```

9. 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 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 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 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。

- 手順
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。

- 手順
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レス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。

- [265 ページの「現在の OS の変更を回避」](#)
- [269 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」](#)

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	if [-f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf] ; then	if [-f /etc/myproduct.conf] ; \ then
ファイルの削除	/bin/rm -f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf	/bin/rm -f /etc/myproduct.conf
ファイルの変更	echo "test=no" > \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf	echo "test=no" > \ /etc/myproduct.conf

\$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所です。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 SUNWvxxm

\$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 SUNWtest

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他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
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。	○
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。	

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。さらに、この処理が終わったら、次のように、`pkgchk` コマンドでパッケージが正しいかチェックします。

```
# 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/SUNWvxxvm 上のパッケージのテスト

パッケージが `/export/SUNWvxxvm` にある場合は、コマンドを次のように指定します。

例 B-2 /export/SUNWvxxvm 上のパッケージのテスト (続き)

```
# pkgchk -d /export SUNWvxxvm
```

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 ${TMPDIR} SUNWvxxvm \
2>&1 > /dev/null | grep -v O_RDONLY | grep -v \
'open("${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 ユーティリティーを使用する際、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 を使用します。

```
# 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
```

参照先

次の参照先から、パッケージ要件のバックグラウンド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	<code>dircmp(1)</code> 、 <code>fssnap(1M)</code> 、 <code>ps(1)</code> 、または <code>truss(1)</code> 、 <code>pkgadd(1M)</code> 、 <code>pkgchk(1M)</code> 、または <code>pkgrm(1M)</code>
Solaris Live Upgrade の概要	『Solaris 10 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 6 章「Solaris Live Upgrade (概要)」
カスタム JumpStart の概要	第 5 章
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 ゾーンを紹介」

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 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	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	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	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。

/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。
/export	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。

IPv6 の詳細については、『Solaris のシステム管理 (IP サービス)』のパート I 「システム管理の概要: IP サービス」を参照してください。

IP アドレス インターネットプロトコル (Internet Protocol) アドレス。TCP/IP では、ネットワーク内の個々のホストを識別する 32 ビットの一意の番号。IP アドレスは、ピリオドで区切られた 4 つの数字から成る (たとえば、192.168.0.0)。通常、IP アドレスの各部は 0~225 の番号ですが、最初の番号は 224 未満とし、最後の番号は 0 以外にする必要があります。

IP アドレスは、論理的に次の 2 つの部分に分割されます。ネットワーク (市外局番のようなもの) とネットワーク上のシステム (電話番号のようなもの) です。たとえば、クラス A の IP アドレス内の数字は「network.local.local.local」を表し、クラス C の IP アドレス内の数字は「network.network.network.local」を表します。

クラス	範囲 (xxx は 0 から 255 までの数字)	使用できる IP アドレス数
クラス A	1.xxx.xxx.xxx ~ 126.xxx.xxx.xxx	1,600 万以上
クラス B	128.0.xxx.xxx ~ 191.255.xxx.xxx	65,000 以上
クラス C	192.0.0.xxx ~ 223.255.255.xxx	256

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	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	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-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。
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。

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とを照合します。rules.ok ファイルは、check スクリプトを使用して作成しなくてはなりません。
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ース (GUI) とコマンド行インターフェース (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ぶ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	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、 <code>http://www.example.com/index.html</code> は URL の一例です。
/usr	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	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ー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 パイナリが、 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ー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 の DVD イメージまたは CD 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。
カスタムプローブファイル	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、プローブと比較です。プローブ関数は、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。プローブ関数はプローブキーワードになります。比較関数は、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。「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	<code>/export/home</code> や <code>/swap</code> 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の <code>vfstab</code> 内と非アクティブブート環境の <code>vfstab</code> 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> では独立したマウントポイントになります。 <code>root (/)</code> 、 <code>/usr</code> 、 <code>/var</code> 、 <code>/opt</code> 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されたシステム。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する暗号化鍵。
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	(<code>system.conf</code>) WAN ブートインストールで使用する <code>sysidcfg</code> 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。 <code>rules</code> 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ディスクに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	再ロードが可能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 10 の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 10 のリリース全体と、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(クラスタとパッケージ)。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) 市場で使われる共通の綴り。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 を使用して、サイズの異なるディスク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	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 (sun4u SPARC システムなど) 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ソフトウェアがインストールされます。リブート後、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。 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) に入ることが要求されます。
ドキュメントルートディレクトリ	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。
ドメイン	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。
ドメイン名	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。
認証局 (CA)	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。
ネームサーバー	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。
ネームサービス	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。
ネットワークインストール	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。

ネットワークに接続されていないシステム	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。
ネットワークに接続されているシステム	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。
ハードリンク	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。
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	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。 ■ 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	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。 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。
ブートサーバー	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。
ブートローダー	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行わ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。
フェイルセーフブートアーカイブ	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。
フォーマット	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。

フォールバック	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。
復号化	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。
プラットフォームグループ	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。
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て作成できますが、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。アプリケーションやファイルシステムから見ると、ボリュームは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。
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。

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	起動可能な最小の Solaris ルート (/) ファイルシステム。ミニルートには、カーネルと、Solaris 環境をハードディスクにインストールするために必要な最小限のソフトウェアが含まれます。ミニルートは、初期インストールでマシンにコピーされるファイルシステムです。
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。アプリケーションやファイルシステムから見ると、ボリュームは物理ディスクと同じように機能します。

索引

数字・記号

#

rules ファイル内の, 93

プロファイル, 97

(/) ファイルシステム

JumpStart で設定される値, 217

A

add_install_client コマンド, JumpStart ディレクトリのアクセス, 82

add_install_client コマンドの -s オプション, 167

&& (アンパサンド) ルールフィールド, 94

AND ルールフィールド, 94

any

プローブキーワード、説明と値, 222

ルールキーワード、説明と値, 171, 222

archive_location キーワード, 177-183

arch プローブキーワード, 222

arch ルールキーワード, 171, 222

auto_install_sample ディレクトリ

check スクリプト, 115, 137

JumpStart ディレクトリへファイルをコピー, 81, 86, 88, 91

set_root_pw 終了スクリプト, 123

B

backup_media キーワード, 183-185

begin.log ファイル, 118

begin ルールフィールド、説明, 94

boot: cannot open /kernel/unix メッセージ, 246

boot_キーワード, 185

bootenv createbe キーワード, 186

bootparams ファイル

JumpStart ディレクトリのアクセスの有効化, 84

更新, 252

C

Can't boot from file/device メッセージ, 246

CHANGE DEFAULT BOOT DEVICE メッセージ, 253

check スクリプト

custom_probes.ok ファイルの作成, 137

custom_probes ファイルの妥当性検査, 137, 138

rules.ok ファイルの作成, 115

rules ファイルの妥当性検査, 115, 116, 138

派生プロファイルと, 119

ルールのテスト, 116, 138

check スクリプトの -p オプション, 115, 137

check スクリプトの -r オプション, 116, 138

client_arch キーワード, 187

client_root プロファイルキーワード, 187

CLIENT MAC ADDR エラーメッセージ, 252

clock gained xxx days メッセージ, 246

cluster プロファイルキーワード

説明と値, 189-190, 190

例, 98

CPU (プロセッサ)
 プローブキーワード, 222
 ルールキーワード, 171, 222
.cshrc ファイル, 123
custom_probes.ok ファイル
 作成, 137
 説明, 137
custom_probes ファイル
 check による妥当性検査, 137, 138
 custom_probes のテスト, 138
 名前の指定, 134
 要件, 134
-c オプション
 add_install_client コマンド, 166, 167
 pfinstall コマンド, 113

D

dd コマンド, 90
dfstab ファイル, 80, 163
disksize ルールキーワード、説明と値, 172, 222
disks プローブキーワード、説明と値, 222
domainname プローブキーワード, 222
domainname ルールキーワード, 172, 222
dontuse プロファイルキーワード, 191, 218

E

eng_profile の例, 163
/etc/bootparams ファイル
 JumpStart ディレクトリアクセスの有効化, 252
 JumpStart ディレクトリのアクセスの有効化, 84
/etc/dfs/dfstab ファイル, 80, 163
/etc/mnttab ファイル, 86

F

fdisk コマンド, 129
fdisk プロファイルキーワード
 説明と値, 191-193
 例, 98
filesys キーワード, 194-197, 197-198

filesys プロファイルキーワード
 説明と値, 193-194
 例, 98
finish.log ファイル, 119
finish ルールフィールド、説明, 95
Flash, 「アーカイブ」を参照

G

geo キーワード, 198
getfile: RPC failed: error 5: RPC Timed out
 メッセージ, 84
GRUB ベースのブート
 menu.lst ファイルの検出, 65
 インストール, 148, 150
 概要, 57-60
 計画, 60
 コマンドリファレンス, 155
 仕組み, 58
 説明
 menu.lst ファイル, 62-64
 メインメニュー, 61
 デバイス命名規則, 58-59
 ネットワーク経由, 61
 プロファイルサーバーへのアクセス, 83
 プロファイルフロッピーディスクの作成, 87

H

hostaddress プローブキーワード, 222
hostaddress ルールキーワード, 172, 222
hostname プローブキーワード、説明と値, 222
hostname ルールキーワード
 説明と値, 172, 222
 例, 171-175

I

install_config コマンド, 84
install_type キーワード, 200
install_type プロファイルキーワード
 プロファイルのテスト, 114
 要件, 97, 98
 例, 98
installed プローブキーワード、説明と値, 222

installed ルールキーワード、説明と値、173, 222

IP アドレス

プローブキーワード、222

ルールキーワード、172, 222

J

JumpStart ディレクトリ

rules ファイルの例、93

アクセス権、79, 85

共有、79, 163

作成

SPARC システム用フロッピーディスク、85

x86 システム用フロッピーディスク、85, 88, 90

サーバー、79

例、163

終了スクリプトでファイルを追加、120

ファイルのコピー

インストールファイル、81, 86, 88, 91

終了スクリプトの使用、120

JumpStart ディレクトリの共有、80, 163

K

karch プローブキーワード、222

karch ルールキーワード、173, 222

L

layout_constraint キーワード、200-202

le0: No carrier - transceiver cable problem

メッセージ、246

locale キーワード、203

M

marketing_profile の例、164

memsize プローブキーワード、説明と値、222

memsize ルールキーワード、説明と値、173, 222

menu.lst ファイル

検出、65

menu.lst ファイル (続き)

説明、62-64

metadb プロファイルキーワード、203-204

mnttab ファイル、86

model プローブキーワード、説明と値、222

model ルールキーワード、説明と値、174, 222

N

network プローブキーワード、説明と値、222

network ルールキーワード、説明と値、174, 222

no_master_check キーワード、205

No carrier - transceiver cable problem メッセージ、246

noneuclidean プロファイルキーワード、205

Not a UFS filesystem メッセージ、246

O

osname プローブキーワード、222

osname ルールキーワード、174, 222

P

partitioning, プロファイルキーワード、211

partitioning キーワード、211

pfinstall コマンド、110

probe ルールキーワード、説明と値、175

prtvtoC コマンド

SPARC: ディスク構成ファイルの作成、127

x86: ディスク構成ファイルの作成、129

R

RAID-0 ボリューム

ガイドライン、237-242

概念的な概要、231

定義、231

RAID-1 ボリューム

ガイドライン、237-242

概念的な概要、232

作成情報、237

定義、232

とディスクジオメトリ、237

- root_device キーワード, 216
- root パスワード、終了スクリプトによる設定, 123
- RPC failed: error 5: RPC Timed out メッセージ, 84
- RPC Timed out メッセージ, 84
- RPC Timed outメッセージ, 252
- rule_keyword ルールフィールド, 94
- rule_value ルールフィールド, 94
- rules
 - 構文, 94
 - 妥当性検査, 116
 - 派生プロファイル, 118, 119
 - フィールドの説明, 94, 95
 - 複数行のルール, 93
- rules.ok ファイル
 - 作成, 115
 - 説明, 115
- rules.ok ファイル, ルールの順序を一致させる, 95
- rules.ok ファイル
 - ルールの順序を一致させる, 141, 147
- rules ファイル
 - check による妥当性検査, 116
 - カスタム JumpStart の例, 165
 - 派生プロファイルと, 119
 - カスタム JumpStart の例, 165
 - 構文, 94
 - コメント, 93
 - 作成, 93
 - 説明, 93
 - 名前の指定, 93, 94
 - 複数行のルール, 93
 - ルールの追加, 94
 - ルールのテスト, 116
 - 例, 93
- rules ファイル内のバックスラッシュ, 93
- rules ファイルの行を折り返す, 93

S

- set_root_pw 終了スクリプト, 123
- setup_install_server コマンドの -b オプション, 163
- shareall コマンド, 80, 163
- share コマンド
 - JumpStart ディレクトリの共有, 80, 163

- SI_PROFILE 環境変数, 119
- Solaris 10 3/05 Device Configuration Assistant (デバイス構成用補助) フロッピーディスク, 90
- Solaris Flash, 「アーカイブ」を参照
- Solaris Live Upgrade, インストールプログラムの選択, 32
- Solaris インストールの新機能, 17
- Solaris ゾーン区分技術
 - Solaris フラッシュアーカイブによるインストール, 52
 - アップグレード, 52, 219
 - インストールの概要, 51
 - 概要, 49
 - 計画, 51
 - ディスク容量の要件, 53
- Solaris ソフトウェア
 - グループ, 189-190
 - アップグレード, 190
 - プロファイルの例, 98
 - リリースまたはバージョン
 - installed プロローブキーワード, 222
 - installed ルールキーワード, 173, 222
 - osname プロローブキーワード, 222
 - osname ルールキーワード, 174, 222
- Solaris ソフトウェアグループ全体, サイズ, 42
- Solaris ソフトウェアグループ全体と OEM サポート, サイズ, 42
- Solaris ソフトウェアのバージョン
 - installed プロローブキーワード, 222
 - installed ルールキーワード, 173, 222
 - osnameプロローブキーワード, 222
 - osname ルールキーワード, 174, 222
- Solaris ソフトウェアのリリース
 - installed プロローブキーワード, 222
 - installed ルールキーワード, 173, 222
 - osnameプロローブキーワード, 222
 - osname ルールキーワード, 174
 - osnameルールキーワード, 222
- Solaris 対話式インストールプログラム, インストールプログラムの選択, 32
- Solaris ボリュームマネージャー, カスタム JumpStart でのボリュームの作成、例, 98
- stty コマンド, 143, 148, 152
- SUNWCall グループ, 189-190
- SUNWCprog グループ, 189-190
- SUNWCreq グループ, 189-190
- SUNWCrnet グループ, 189-190

SUNWCuser グループ, 189-190
SUNWCXall グループ, 189-190
system_type プロファイルキーワード
説明と値, 218
例, 98

T

timed out RPC エラー, 252
tip 行接続表示要件, 148
tip ライン接続ディスプレイ要件, 152
tip ライン接続の要件, 143, 148
totaldisk プロンプトキーワード, 223
totaldisk ルールキーワード, 175, 223
transceiver cable problem メッセージ, 246

U

UFS, 86
Unknown client エラーメッセージ, 245
usedisk プロファイルキーワード, 説明と
値, 218

V

/var/sadm/system/logs/begin.log
ファイル, 118
/var/sadm/system/logs/finish.log
ファイル, 119
volcheck コマンド, 86, 88, 90, 91

W

WARNING: CHANGE DEFAULT BOOT
DEVICE, 253
WARNING: clock gained xxx days メッセー
ジ, 246

あ

アーカイブ
Solaris フラッシュアーカイブによるインス
トール, 52

アーカイブ (続き)

JumpStart プロファイルの例, 102, 103, 104,
105
インストール, 45
インストールプログラムの選択, 32
キーワード、カスタム JumpStart, 177-183
説明, 34

アクセス権

JumpStart ディレクトリ, 79, 85
開始スクリプト, 118
終了スクリプト, 119
アップグレード
Solaris フラッシュアーカイブを使用
説明, 45
アップグレードの失敗, 257
カスタム JumpStart インストール, 139
作業の概要, 27
初期インストールとの比較, 32
ディスク容量の推奨事項, 39-43
非大域ゾーン, 52, 219
プロファイルキーワード, 190, 200, 210
アップグレードの失敗, リブートの問題, 257
アンパサンド (&&) ルールフィールド, 94

い

一致させる
ルートディスク値, 217
ルールの順序, 95, 141, 147
インストール
Solaris フラッシュアーカイブを使用, 45
アップグレードとの比較, 31-32
作業の概要, 27
ディスク容量の推奨事項, 39-43
ネットワーク経由
計画, 30-31
インストールの準備、カスタム JumpStart, 75,
116

え

エンドユーザー Solaris ソフトウェアグループ,
サイズ, 42
エンドユーザーシステムサポート, 説明, 41-43
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, 189-190

か

開始、check スクリプト, 115, 116

開始スクリプト

アクセス権, 118

概要, 117

サイト固有のインストールプログラム, 132

派生プロファイルの作成, 118, 119

ルールフィールド, 94

開発者 Solaris ソフトウェアグループ, サイズ, 42

開発者システムサポート, 説明, 41-43

開発者システムサポートソフトウェアグループ, 189-190

プロファイルの例, 98

概要, GRUB ベースのブート, 57-60

カスタム JumpStart インストール, 139

tip ライン接続の要件, 143, 148, 152

インストールプログラムの選択, 32

オプション機能, 117

開始スクリプト, 117, 119

概要, 117

サイト固有のインストールプログラム, 132

終了スクリプト, 119, 123

概要, 75

起動とインストール, 139

準備, 75, 116

説明, 75

プロファイルキーワード, 176

例, 159, 168

check スクリプト, 165

eng_profile の作成, 163

Solaris フラッシュ プロファイル, 104

Solaris フラッシュプロファイル, 102, 103, 105

JumpStart ディレクトリ, 163

marketing_profile の作成, 164

RAID-1 ボリュームプロファイル, 106, 108

rules ファイルの編集, 165

WAN ブートインストールプロファイル, 103

エンジニアリングシステムの設定, 166

サイトの設定, 159, 160

スタンドアロンシステム, 73

ネットワークに接続された, 74

ネットワークに接続されていない, 73

ブートとインストール, 168

カスタム JumpStart インストール, 例 (続き)

マーケティングシステムの設定, 162, 167

!(感嘆符) ルールフィールド, 94

感嘆符 (!) ルールフィールド, 94

き

キーワード

Solaris フラッシュアーカイブ、カスタム
JumpStart, 177-183

プローブ, 133

け

計画

GRUB ベースのブート, 60

インストールプログラムの選択, 32

作業の概要, 27

初期インストールとアップグレードの比較, 31-32

ディスク容量, 39-43

ネットワーク経由のインストール, 30-31

限定ネットワークサポートソフトウェアグループ, 189-190

限定ネットワークシステムサポート, 説明, 41-43

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, サイズ, 42

こ

コアシステムサポート, 説明, 41-43

コアシステムサポートソフトウェアグループ, サイズ, 42

コアソフトウェアグループ, 189-190

構成, ディスク構成ファイルの作成, 127

コピー

JumpStart インストールファイル, 81, 86, 88, 91

JumpStart ディレクトリファイル, 120

Solaris 10 3/05 Device Configuration

Assistant (デバイス構成用補助) フ

ロッピーディスク, 90

コメント

rules ファイル内の, 93

コメント (続き)
プロファイル, 97

さ

サーバー

JumpStart ディレクトリの作成, 79
ルート領域, 187

サービスパーティション, インストール時に保持 (x86 システム), 55

サイズ

tip ライン接続ディスプレイサイズ, 143, 148, 152

スワップ空間

最大サイズ, 188
ディスクレスクライアント, 188

スワップ領域

プロファイルの例, 75

ハードディスク

プローブキーワード, 222, 223
ルート領域, 187
ルールキーワード, 172, 175, 222, 223

メモリー, 173, 222

サイト固有のインストールプログラム, 132

削除, アップグレード時のクラスタ, 190

作成

custom_probes.ok ファイル, 137
JumpStart ディレクトリ、サーバー上, 79
RAID-1 ボリューム, 197-198
rules.ok ファイル, 115, 137
rules ファイル, 93
UFS, 86
ディスク構成ファイル, 127
プロファイル
説明, 96
派生, 118
ローカルファイルシステム, 194-197

サブミラー, 「RAID-0 ボリュームを参照」を参照

し

終了スクリプト

システムの root パスワードの設定, 123
パッケージとパッチの追加, 120
ルート環境のカスタマイズ, 123

終了スクリプト (続き)

ルールフィールド, 95

出力ファイル

開始スクリプトログ, 118
終了スクリプトログ, 119

照合, 派生プロファイル, 118

状態データベース

概念的な概要, 230-231
定義, 230-231

状態データベースの複製, 230

基本操作, 231
最小数, 237
使用方法, 231
単一スライス上に複数作成, 236
定義, 230
場所, 236, 237

す

スクリプト

開始スクリプト, 117, 119, 132
終了スクリプト, 119, 123, 132
ルールフィールド内の Bourne シェルスクリプト, 94

スタンドアロンシステム

カスタム JumpStart インストールの例, 73
プロファイルの例, 98

スライス

プローブキーワード, 222
プロファイルの例, 98
ルールキーワード, 173, 222

スワップファイルシステム

サイズの決定, 188
ディスクレスクライアントのスワップ空間, 188
プロファイルの例, 75
メモリーサイズと, 188

せ

セキュリティー、root パスワード, 123

全体ディストリビューション, 説明, 41-43

全体ディストリビューションソフトウェアグループ, 189-190

全体ディストリビューションと OEM サポート, 説明, 41-43

全体ディストリビューションと OEM サポート
ソフトウェアグループ, 189-190

そ

ソフトウェアグループ
アップグレード, 190
サイズ, 42
説明, 42
プロファイルの例, 98
プロファイル用, 189-190

た

大域ゾーン, 説明, 49
代替インストールプログラム, 132
多数決アルゴリズム, 231
妥当性

custom_probes ファイル
テスト, 138

妥当性検査

custom_probes ファイル
check の使用, 138
rules ファイル
check の使用, 115, 116, 138
カスタム JumpStart の例, 165
派生プロファイルと, 119
ルールのテスト, 116

つ

追加

rules ファイルヘルールを, 94
アップグレード時のクラスタ, 190
終了スクリプトでパッケージとパッチ
を, 120
ソフトウェアグループからパッケージ
を, 205

て

ディスク構成ファイル
作成
SPARC システム, 127

ディスク構成ファイル, 作成 (続き)
x86 システム, 129

説明, 110, 127

ディスク容量

計画, 39-43

非大域ゾーンの計画, 53

要件、ソフトウェアグループ, 42

ディスクレスクライアント

スワップ空間, 188

プラットフォーム, 187

ディスプレイ

tip ライン接続の要件, 143, 148, 152

ディレクトリ

JumpStart

rules ファイルの例, 93

アクセス権, 79, 85

インストールファイルのコピー, 81, 86,
88, 91

システム用に作成, 85

ディレクトリの共有, 80, 163

ディレクトリの作成, 163

ファイルのコピー, 120

ファイルの追加, 120

移動

JumpStart ディレクトリへ, 115, 137

ローカルディスク上の Solaris SPARC ソ
フトウェアのイメージ, 86

ローカルディスク上の Solaris ソフト
ウェアのイメージ, 81

ローカルディスク上の x86 ベースの
Solaris イメージ, 88, 91

ディレクトリの移動

JumpStart ディレクトリへ, 115, 137

ローカルディスク上の Solaris SPARC ソフト
ウェアのイメージ, 86

ローカルディスク上の Solaris ソフトウェア
のイメージ, 81

ローカルディスク上の x86 ベースの Solaris
イメージ, 88, 91

テスト

custom_probes ファイルの妥当性検査

check の使用, 137

custom_probes のテスト, 138

rules ファイルの妥当性検査

check の使用, 115, 116, 138

カスタム JumpStart の例, 165

派生プロファイルと, 119

ルールのテスト, 116

テスト (続き)

プロファイル, 110, 114

デバイス命名規則, GRUB, 58-59

デフォルト

インストールするソフトウェアグループ, 189

パーティション分割

ディスクの指定, 218

ディスクの除外, 191

派生プロファイル名, 119

と

トークンリングカード、ブートエラー, 251

ドメイン

プローブキーワード, 222

ルールキーワード, 172, 222

な

名前/名前の指定

custom_probes ファイル, 134

rules ファイル, 93, 94

システムモデル名, 174, 222

派生プロファイル名, 119

ホスト名, 172, 222

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, 74

ネットワーク番号, 174, 222

ネットワークブート, GRUB を使用, 61

は

パーティション分割

fdisk パーティション, 98, 191-193

ディスクの除外, 191

プロファイルキーワード, 218

例, 98

ハードディスク

partitioning

プロファイルキーワード, 211

ハードディスク (続き)

サイズ

プローブキーワード, 222, 223

ルート領域, 187

ルールキーワード, 172, 175, 222, 223

スワップ空間

最大サイズ, 188

ディスクレスクライアント, 188

プロファイルの例, 98

スワップ領域

プロファイルの例, 75

パーティション分割

partitioning default のデフォルトの指定, 218

partitioning default 用に除外, 191
例, 98

マウント, 193-194

ルートディスク値, 217

パス、check スクリプト, 115, 137

パスワード、root, 123

派生プロファイル, 118, 119

パッケージ

Solaris Live Upgrade

要件, 265

カスタム JumpStart を使用する要件, 265

管理ファイル, 117

追加

chroot による, 122

終了スクリプトで, 120

パッチ

追加

chroot による, 122

終了スクリプトで, 120

ひ

非大域ゾーン

Solaris フラッシュアーカイブによるインストール, 52

アップグレード, 52, 219

インストールの概要, 51

概要, 49

計画, 51

説明, 49

ディスク容量の要件, 53

ふ

ファイルとファイルシステム

UFS の作成, 86

開始スクリプトの出力, 118

コピー

JumpStart インストールファイル, 81, 86, 88, 91

Solaris 10 3/05 Device Configuration

Assistant (デバイス構成用補助) フ

ロッピーディスク, 90

終了スクリプトによる JumpStart ディレ
クトリファイルの, 120

作成

RAID-1 ボリューム, 197-198

ローカルファイルシステム, 194-197

終了スクリプトの出力, 119

リモートファイルシステムのマウン
ト, 193-194

ブート

GRUB によるインストール, 148, 150

GRUB の使用、コマンドリファレンス, 155

GRUB を使用, 57-60

RAID-1 ボリュームとシングルユーザーモー
ド, 242

ネットワークから GRUB を使用, 61

プロファイルサーバーへのアクセス, 83

プロファイルフロッピーディスクの作成, 87

ブートディスクパーティションレイアウト、新
しいデフォルト (x86 システム), 55

ブートローダー, GRUB, 57-60

複数行ある rules ファイル, 93

複製, 230

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, 95, 141, 147

システムモデル名, 174, 222

ディスクレスクライアント, 187

プロープキーワード, 222

ルールキーワード, 173, 222

プロープキーワード

arch, 222

domainname, 222

hostaddress, 222

hostname, 222

installed, 222

karch, 222

memsize, 222

model, 222

プロープキーワード (続き)

network, 222

osname, 222

rootdisk, 222

totaldisk, 223

ディスク, 222

プロセッサ

プロープキーワード, 222

ルールキーワード, 171, 222

フロッピーディスク

JumpStart ディレクトリのアクセス, 83

Solaris 10 3/05 Device Configuration

Assistant (デバイス構成用補助) フ

ロッピーディスクのコピー, 90

x86: JumpStart ディレクトリ, 85

プロファイル

コメント, 97

作成, 96

システムを一致させる, 95, 141, 147

説明, 96, 97

テスト, 114

名前の指定, 97

派生プロファイル, 118, 119

要件, 93, 97

ルールフィールド, 94

例, 98

eng_profile, 163

marketing_profile, 164

Solaris フラッシュ, 102, 103, 104, 105

WAN ブートインストール, 103

プロファイルキーワード, 176, 218

archive_location, 177-183

backup_media, 183-185

boot_device, 185

bootenv createbe, 186

client_arch, 187

client_root, 187

client_swap, 188

cluster

説明と値, 189-190, 190

例, 98

dontuse

usedisk と, 218

説明と値, 191

fdisk

説明と値, 191-193

例, 98

プロファイルキーワード (続き)

filesystem

RAID-1 ボリューム, 197-198

説明と値, 193-194

リモートファイルシステム, 193-194

例, 98

ローカルファイルシステム, 194-197

forced_deployment, 説明と値, 198

geo

説明と値, 198

install_type

説明と値, 200

要件, 97, 98

例, 98

layout_constraint, 説明と値, 200-202

local_customization, 説明と値, 203

locale, 説明と値, 203

metadb

説明と値, 203-204

例, 98

no_master_check, 説明と値, 205

noneuclidean, 205

partitioning

説明と値, 211

root_device, 216

system_type

説明と値, 218

例, 98

usedisk, 説明と値, 218

大文字と小文字の区別, 176

状態データベースの複製 (meatball) の作成, 203-204

パーティション分割

ディスクの指定, 218

ディスクの除外, 191

例, 98

プロファイルフィールドの等号 (=), 118

へ

変数

SI_PROFILE, 119

SYS_MEMSIZE, 113

ほ

ボリューム

概念的な概要, 228

仮想ディスク, 228

使用, 228

命名規則, 239

ボリュームマネージャー

コピー, 86, 88, 91

Solaris 10 3/05 Device Configuration

Assistant (デバイス構成用補助) フ

ロッピーディスク, 90

Solaris 10 3/05 Device Configuration

Assistant (デバイス構成用補助) イメー
ジ, 91

ま

マイクロプロセッサ

プローブキーワード, 222

ルールキーワード, 171, 222

マウント

Solaris インストールによる, 119

開始スクリプトに関する注意事項, 118

リモートファイルシステム, 193-194

み

ミラー, 「RAID-1 ボリューム」を参照

め

命名規則、デバイス、GRUB, 58-59

メタデバイス, 「ボリューム」を参照

メモリー

スワップ空間と, 188

プローブキーワード, 222

ルールキーワード, 173, 222

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, 252

問題発生時の解決方法 (続き)

一般的なインストールの問題

DHCP によるネットワークからのブート, 252

システムのブート, 252

間違ったサーバーからのブート, 252

よ

要件

custom_probes ファイル, 134

ディスク容量, 39-43

プロファイル, 93, 97

メモリー, 37, 38

り

リモートファイルシステム, マウント, 193-194

る

ルート (/) ファイルシステム, プロファイルの例, 75

ルート環境、終了スクリプトによるカスタマイズ, 123

ルートディスク

filesys のスライス値, 195

JumpStart で設定される値, 217

定義, 217

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, 265

ルール

順序を一致させる, 95, 141, 147

妥当性検査, 138

ルートディスクの一致規則, 217

例, 95

ルールキーワード, 171

any、説明と値, 171, 222

arch, 171, 222

disksize、説明と値, 172, 222

domainname, 172, 222

hostaddress, 172, 222

hostname, 171-175, 222

installed、説明と値, 173, 222

karch, 173, 222

ルールキーワード (続き)

memsize, 173, 222

model, 174, 222

network, 174, 222

osname, 174, 222

probe, 175

totaldisk, 175, 223

ルールフィールド内の Bourne シェルスクリプト, 94

れ

連結

概念的な概要, 231

作成情報, 237

定義, 231

ろ

ログファイル

開始スクリプトの出力, 118

終了スクリプトの出力, 119