



# Solaris 9 4/04 インストールガイド

---

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

Part No: 817-4894-10  
2004 年 4 月

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

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

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

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

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

Sun、Sun Microsystems、Sun のロゴマーク、docs.sun.com、AnswerBook、AnswerBook2、JumpStart、Solaris Web Start、Power Management、Sun ONE Application Server、Solaris Flash、Solaris Live Upgrade、SunSolve は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サン のロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

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。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政事業庁が公開したデータを元に制作された物です (一部データの加工を行なっています)。

本製品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 '98』に添付のものを使用しています。© 1997 ビレッジセンター

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、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

DtComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

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ず、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

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

原典: Solaris 9 4/04 Installation Guide

Part No: 817-3799-10

Revision A



040307@7940



# 目次

---

- はじめに 21
  
- 1 **Solaris** のインストールおよびアップグレードの計画 (トピック) 25
  
- 2 **Solaris** のインストールおよびアップグレードの計画 (概要) 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
  - 初期インストールかアップグレードか 29
    - 初期インストール 29
    - アップグレード 29
  - 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
  - DVD メディアの使用 32
  - Sun ONE Application Server のインストール後の構成 32
  - x86: Solaris 9 Device Configuration Assistant (デバイス構成用補助) と PXE へのアクセス 33
  
- 3 **Solaris** のインストール方法の選択 (概要) 35
  - Solaris Web Start 35
  - Solaris suninstall プログラム 36
  - カスタム JumpStart 36
  - フラッシュインストール機能 37
  - WAN ブートインストール 38
  - Solaris Live Upgrade 38
  - SPARC: ファクトリ JumpStart 39

<b>4</b>	<b>Solaris のインストールおよびアップグレードの準備 (トピック)</b>	<b>41</b>
<b>5</b>	<b>システム要件とガイドライン (計画)</b>	<b>43</b>
	メモリー要件	43
	ディスク容量の割り当て	44
	ディスク容量に関する一般的な計画と推奨事項	44
	ソフトウェアグループごとの推奨ディスク容量	45
	SPARC: スワップ領域の割り当て方法	46
	Solaris INSTALLATION CD を使用するための要件	47
	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	48
<b>6</b>	<b>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)</b>	<b>51</b>
	インストール用のチェックリスト	51
	インストール用のワークシート	52
	アップグレード用のチェックリスト	57
	アップグレード用のワークシート	58
<b>7</b>	<b>システム構成情報の事前設定 (作業)</b>	<b>63</b>
	システム構成情報を事前設定することの利点	63
	システム構成情報の事前設定方法	64
	sysidcfg ファイルによる事前設定	65
	sysidcfg ファイルの構文規則	66
	sysidcfg ファイルキーワード	67
	▼ sysidcfg 構成ファイルを作成する方法	70
	ネームサービスによる事前設定	72
	▼ NIS を使ってロケールを事前設定する方法	72
	▼ NIS+ を使ってロケールを事前設定する方法	74
	DHCP サービスによるシステム構成情報の事前設定 (作業)	75
	Solaris インストールパラメータ用の DHCP オプションとマクロの作成	76
	▼ Solaris のインストールをサポートするオプションを作成する方法 (DHCP マネージャ)	83
	▼ Solaris のインストールをサポートするマクロを作成する方法 (DHCP マネージャ)	84
	SPARC: 電源管理情報の事前設定	85

<b>8</b>	<b>Solaris</b> オペレーティング環境のアップグレード (計画)	<b>87</b>
	アップグレード (概要)	87
	Solaris のアップグレード方法	88
	Solaris Live Upgrade の利用	89
	カスタム JumpStart によるアップグレード	90
	ディスク容量の再配置を使用するアップグレード	90
	アップグレード前のシステムのバックアップ	91
<b>9</b>	インストール時の <b>RAID-1</b> ボリューム (ミラー) の作成 (トピック)	<b>93</b>
<b>10</b>	インストール時の <b>RAID-1</b> ボリューム (ミラー) の作成 (概要)	<b>95</b>
	ミラー化を行う理由	95
	ミラー化の機能	96
	ミラー化コンポーネントの概要	99
	状態データベースと状態データベースの複製	99
	RAID-0 ボリューム (連結)	100
	RAID-1 ボリューム (ミラー)	101
	ミラー化されたファイルシステムのレイアウト例	102
<b>11</b>	インストール時の <b>RAID-1</b> ボリューム (ミラー) の作成 (計画)	<b>105</b>
	システム要件	105
	ボリューム名の要件とガイドライン	106
	状態データベースの複製のガイドラインと要件	106
	状態データベースの複製用のスライスの選択	107
	状態データベースの複製の数の選択	107
	コントローラ間で状態データベースの複製を分散	108
	ミラーとサブミラーの要件およびガイドライン	108
	カスタム JumpStart と Solaris Live Upgrade のガイドライン	108
	ディスクとコントローラを選択のガイドライン	109
	スライスの選択のガイドライン	109
	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	110
<b>12</b>	ネットワークインストールの準備 (トピック)	<b>111</b>
<b>13</b>	ネットワークインストールの準備 (概要)	<b>113</b>
	ネットワークインストールの計画	113

ネットワークインストールに必要なサーバー 113  
DHCPによるネットワークインストールパラメータの指定 116

- 14 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) 117**  
作業マップ: DVDメディアを使用してネットワーク上でインストールするための準備 117  
DVDメディアを用いたインストールサーバーの作成 118
- ▼ SPARC または x86 DVDメディアを使用して SPARC インストールサーバーを作成する方法 119
  - ▼ x86: SPARC または x86 DVDメディアを使用して x86 インストールサーバーを作成する方法 122
- DVDイメージを用いたサブネット上でのブートサーバーの作成 126
- ▼ DVDイメージを使用してサブネット上にブートサーバーを作成する方法 126
- 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を追加する 128
- ▼ `add_install_client` を実行してネットワークからインストールするシステムを追加する方法 (DVD) 129
- 15 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) 135**  
作業マップ: CDメディアを使用してネットワーク上でインストールするための準備 136  
CDメディアによる SPARC インストールサーバーの作成 137
- ▼ SPARC: SPARC CDメディアを使用して SPARC インストールサーバーを作成する方法 137
- CDメディアによる x86 インストールサーバーの作成 142
- ▼ x86: x86 CDメディアを使用して x86 インストールサーバーを作成する方法 142
- クロスプラットフォームの CDメディアを使用したインストールサーバーの作成 147
- ▼ x86 CDメディアを使用して SPARC システムに x86 インストールサーバーを作成する方法 147
  - ▼ SPARC CDメディアを使用して x86 システムに SPARC インストールサーバーを作成する方法 153
- CDイメージを用いたサブネット上でのブートサーバーの作成 159
- ▼ CDイメージを使用してサブネット上にブートサーバーを作成する方法 159
- 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を追加する 161
- ▼ `add_install_client` を実行してネットワークからインストールするシステムを追加する方法 (CD) 162

<b>16</b>	<b>ネットワークインストールの準備 (コマンドリファレンス)</b>	<b>167</b>
<b>17</b>	<b>Solaris Web Start の使用 (作業)</b>	<b>169</b>
	Solaris Web Start の実行方法	169
	SPARC: Solaris Web Start によるインストールまたはアップグレード	170
	SPARC: 作業マップ: Solaris Web Start インストールの実行	171
	▼ SPARC: Solaris Web Start 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	171
	x86: Solaris Web Start によるインストールまたはアップグレード	175
	x86: 作業マップ: Solaris Web Start インストールの実行	176
	▼ x86: Solaris Web Start 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	176
	Solaris Web Start によるインストールまたはアップグレード後に必要な作業	182
	▼ アップグレード後にローカル変更を復元する方法	182
<b>18</b>	<b>Solaris suninstall プログラムの使用 (作業)</b>	<b>185</b>
	Solaris suninstall プログラム	185
	SPARC: Solaris suninstall プログラムによるインストールまたはアップグレー ド	186
	SPARC: 作業マップ: Solaris suninstall プログラムインストールの実行	186
	▼ SPARC: Solaris suninstall プログラムを使用してインストールまたはアップ グレードするには	187
	x86: Solaris suninstall プログラムによるインストールまたはアップグレー ド	189
	x86: 作業マップ: suninstall プログラムを実行する	190
	▼ x86: Solaris suninstall プログラムを使用してインストールまたはアップグ レードするには	190
	Solaris suninstall プログラムによるアップグレード後に必要な作業	196
	▼ アップグレード後にローカル変更を復元する方法	196
<b>19</b>	<b>フラッシュインストール機能 (トピック)</b>	<b>197</b>
<b>20</b>	<b>フラッシュ (概要と計画)</b>	<b>199</b>
	フラッシュの概要	199
	初期インストールでクローンシステムにインストールする	199
	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更新する	201
	フラッシュインストールの計画	204

	マスターシステムへの初期インストールの設計	204
	フラッシュアーカイブの作成の計画	207
	フラッシュアーカイブのインストールの計画	211
<b>21</b>	<b>フラッシュアーカイブの作成 (作業)</b>	<b>213</b>
	作業マップ: フラッシュアーカイブを作成する	213
	マスターシステムへのインストール	214
	▼ 初期インストールでマスターシステムにインストールする方法	215
	カスタムスクリプトの作成	215
	▼ 作成前スクリプトの作成方法	216
	▼ 配置前スクリプトの作成方法	218
	▼ 配置後スクリプトの作成方法	219
	▼ 再起動スクリプトの作成方法	219
	フラッシュアーカイブの作成	220
	▼ 初期インストール用フラッシュアーカイブの作成方法	220
	▼ 更新されたマスターイメージを使用してフラッシュ差分アーカイブを作成する方法	225
	▼ Solaris Live Upgrade を使用してフラッシュ差分アーカイブを作成する方法	228
<b>22</b>	<b>フラッシュアーカイブのインストールと管理 (作業)</b>	<b>233</b>
	フラッシュアーカイブをインストールする方法	233
	フラッシュアーカイブの管理	234
	フラッシュアーカイブの分割	234
	フラッシュアーカイブの結合	235
	アーカイブからの情報の抽出	236
<b>23</b>	<b>フラッシュ (リファレンス)</b>	<b>239</b>
	フラッシュアーカイブセクションの説明	239
	フラッシュのキーワード	241
	一般的なキーワード	241
	識別セクションのキーワード	242
	ユーザー定義セクションのキーワード	246
	フラッシュ flar create コマンド	246
	flar create	246



24	カスタム <b>JumpStart</b> インストール (トピック)	251
25	カスタム <b>JumpStart</b> (概要)	253
	カスタム <b>JumpStart</b> とは	253
	<b>JumpStart</b> 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	254
26	カスタム <b>JumpStart</b> インストールの準備 (作業)	259
	作業マップ: カスタム <b>JumpStart</b> インストールの準備	260
	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	261
	▼ <b>JumpStart</b> ディレクトリをサーバー上に作成する方法	262
	▼ 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	263
	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	265
	▼ SPARC: プロファイルフロッピーディスクを作成する方法	266
	▼ x86: プロファイルフロッピーディスクを作成する方法	268
	rules ファイルの作成	270
	rules ファイルの構文	271
	▼ rules ファイルを作成する方法	271
	rules ファイルの例	273
	プロファイルの作成	274
	プロファイルの構文	274
	▼ プロファイルを作成する方法	275
	プロファイルの例	276
	プロファイルのテスト	284
	▼ 一時的な Solaris 9 環境を作成してプロファイル进行测试する方法	284
	▼ プロファイル进行测试する方法	285
	プロファイルテストの例	288
	rules ファイルの妥当性を検査する	288
	▼ rules ファイルの妥当性を検査する方法	289
27	カスタム <b>JumpStart</b> オプション機能の使用 (作業)	291
	begin スクリプトの作成	291
	begin スクリプトに関する注意事項	292
	begin スクリプトによる動的プロファイルの作成	292
	finish スクリプトの作成	293
	finish スクリプトに関する注意事項	293
	▼ finish スクリプトでファイルを追加する方法	294

- finish スクリプトによるパッケージとパッチの追加 294
- finish スクリプトによるルート環境のカスタマイズ 297
- finish スクリプトによるシステムのルートパスワードの設定 297
- finish スクリプトを使用して Web Start インストールプログラムでソフトウェアをインストールする 299
- 圧縮された構成ファイルの作成 299
  - ▼ 圧縮された構成ファイルを作成する方法 300
  - 圧縮された構成ファイルの例 300
- ディスク構成ファイルの作成 301
  - ▼ SPARC: ディスク構成ファイルを作成する方法 301
  - SPARC: ディスク構成ファイルの例 302
  - ▼ x86: ディスク構成ファイルを作成する方法 302
  - x86: ディスク構成ファイルの例 304
- サイト固有のインストールプログラムの使用 306
  
- 28 カスタムルールおよびプローブキーワードの作成 (作業) 307**
  - プローブキーワード 307
  - custom\_probes ファイルの作成 308
    - custom\_probes ファイルの構文 308
    - custom\_probes の関数名の構文 309
    - ▼ custom\_probes ファイルを作成する方法 309
    - custom\_probes ファイルとキーワードの例 309
  - custom\_probes ファイルの妥当性検査 311
    - ▼ custom\_probes ファイルを妥当性検査する方法 311
  
- 29 カスタム JumpStart インストールの実行 (作業) 313**
  - SPARC: カスタム JumpStart インストールの実行 313
    - 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 313
    - ▼ カスタム JumpStart インストールを使用してフラッシュアーカイブをインストールする方法 315
    - ▼ 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 316
  - x86: カスタム JumpStart インストールの実行 319
    - 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 319
    - ▼ 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 320

<b>30</b>	<b>カスタム JumpStart によるインストール (例)</b>	<b>325</b>
	サイトの設定例	325
	インストールサーバーの作成	327
	x86: マーケティングシステム用のブートサーバーの作成	328
	JumpStart ディレクトリの作成	329
	JumpStart ディレクトリの共有	329
	SPARC: エンジニアリンググループのプロファイルの作成	329
	x86: マーケティンググループのプロファイルの作成	330
	rules ファイルの更新	330
	rules ファイルの妥当性を検査する	331
	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	332
	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	332
	SPARC: エンジニアリングシステムのブートと Solaris 9 ソフトウェアのインストー ル	333
	x86: マーケティングシステムのブートと Solaris 9 ソフトウェアのインストール	334
<b>31</b>	<b>カスタム JumpStart (リファレンス)</b>	<b>335</b>
	ルールキーワードと値の説明	335
	プロファイルキーワードと値	340
	プロファイルキーワードのリスト	340
	プロファイルキーワードの説明と例	342
	カスタム JumpStart の環境変数	371
	プローブキーワードと値	373
<b>32</b>	<b>Solaris Live Upgrade (トピック)</b>	<b>377</b>
<b>33</b>	<b>Solaris Live Upgrade (概要)</b>	<b>379</b>
	Solaris Live Upgrade の紹介	379
	Solaris Live Upgrade の実行手順	380
	ブート環境の作成	380
	ミラー化されたファイルシステムを持つブート環境の作成	384
	ブート環境のアップグレード	390
	ブート環境のアクティブ化	392
	元のブート環境へのフォールバック	394
	ブート環境の保守	395

<b>34</b>	<b>Solaris Live Upgrade (計画)</b>	<b>397</b>
	Solaris Live Upgrade の要件	397
	Solaris Live Upgrade のシステム要件	397
	Solaris Live Upgrade のインストール	398
	Solaris Live Upgrade のディスク容量の要件	399
	RAID-1 ボリューム (ミラー) を作成する場合の Solaris Live Upgrade の要件	399
	Solaris Live Upgrade によるパッケージやパッチの管理	400
	必要なパッケージ	400
	パッケージとパッチによるシステムのアップグレード	401
	Solaris Live Upgrade の実行に必要なパッチ	402
	lucreate コマンドを用いたファイルシステムの作成のための指針	402
	ファイルシステムのスライスを選択するための指針	403
	ルート (/) ファイルシステムのスライスを選択するための指針	403
	ミラー化されたファイルシステムのスライスを選択するための指針	404
	スワップファイルシステムのスライスを選択するための指針	406
	共有可能なファイルシステムのスライスを選択するための指針	407
	新しいブート環境の内容のカスタマイズ	408
	ブート環境間でのファイルの同期	409
	/etc/lu/synclist へのファイルの追加	409
	ブート環境間での強制的な同期	410
	リモートシステムからの Solaris Live Upgrade の使用	411
<b>35</b>	<b>Solaris Live Upgrade によるブート環境の作成 (作業)</b>	<b>413</b>
	Solaris Live Upgrade インタフェースの概要	413
	Solaris Live Upgrade メニューの使用	414
	作業マップ: Solaris Live Upgrade のインストールとブート環境の作成	415
	Solaris Live Upgrade のインストール	415
	▼ pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法	416
	▼ Solaris Web Start インストーラを使用して Solaris Live Upgrade をインストールする方法	416
	Solaris Live Upgrade の起動と停止 (キャラクタインタフェース)	417
	▼ Solaris Live Upgrade メニューを起動する	417
	▼ Solaris Live Upgrade のメニューを終了するには	418
	新しいブート環境の作成	418
	▼ ブート環境を作成する (キャラクタインタフェース)	419
	▼ ブート環境を初めて作成する (コマンド行インタフェース)	424

▼	ブート環境を作成しファイルシステムをマージする (コマンド行インタフェース)	426
▼	ブート環境を作成しファイルシステムを分割する (コマンド行インタフェース)	428
▼	ブート環境を作成しスワップを再構成する (コマンド行インタフェース)	430
▼	リストを使用してブート環境を作成しスワップを再構成する (コマンド行インタフェース)	431
▼	ブート環境を作成し共有可能ファイルシステムをコピーする (コマンド行インタフェース)	433
▼	別々のソースから単一のブート環境を作成 (コマンド行インタフェース)	435
▼	フラッシュアーカイブ用の空のブート環境の作成 (コマンド行インタフェース)	436
▼	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)	438
▼	ブート環境の作成と内容のカスタマイズ (コマンド行インタフェース)	444
<b>36</b>	<b>Solaris Live Upgrade によるアップグレード (作業)</b>	<b>447</b>
	作業マップ: ブート環境のアップグレード	448
	ブート環境のアップグレード	448
	アップグレードのガイドライン	449
	▼ ブート環境のオペレーティングシステムイメージをアップグレードする (キャラクタインタフェース)	450
	▼ ブート環境のオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)	451
	▼ 複数の CD を使用してオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)	451
	▼ Solaris Live Upgrade で使用されるプロファイルを作成する	453
	▼ Solaris Live Upgrade で使用されるプロファイルをテストする	457
	▼ プロファイルを使用して Solaris Live Upgrade でアップグレードする (コマンド行インタフェース)	458
	ブート環境へのフラッシュアーカイブのインストール	459
	▼ ブート環境へのフラッシュアーカイブのインストール (キャラクタインタフェース)	460
	▼ ブート環境への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)	461
	▼ プロファイルを使用した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)	462
	▼ プロファイルキーワードを使用した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)	463
	ブート環境のアクティブ化	464

- ▼ x86: (省略可能) アクティブ化の前にブート用フロッピーディスクを更新する 465
- ▼ ブート環境のアクティブ化 (キャラクタインタフェース) 465
- ▼ ブート環境をアクティブにする (コマンド行インタフェース) 466
- ▼ ブート環境をアクティブにしてファイルを同期させる (コマンド行インタフェース) 467
- 問題の解決: 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) 468
  - ▼ 新しいブート環境の作成に成功した場合のフォールバック 469
  - ▼ SPARC: ブート環境のアクティブ化に失敗した場合のフォールバック 469
  - ▼ SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする 470
  - ▼ x86: 別のディスクに存在するブート環境をフォールバックする 471
  - ▼ x86: 同じディスクに存在するブート環境をフォールバックする 472
- 37 Solaris Live Upgrade ブート環境の管理 (作業) 475**
  - Solaris Live Upgrade 管理作業の概要 475
  - すべてのブート環境のステータスの表示 476
    - ▼ すべてのブート環境のステータスを表示する (キャラクタインタフェース) 477
    - ▼ すべてのブート環境のステータスを表示する (コマンド行インタフェース) 477
  - 以前に構成されたブート環境の更新 478
    - ▼ 以前に構成されたブート環境を更新する (キャラクタインタフェース) 478
    - ▼ 以前に構成されたブート環境を更新する (コマンド行インタフェース) 479
  - スケジュールされた処理 (作成/アップグレード/コピー) の取り消し 480
    - ▼ スケジュールされた処理 (作成/アップグレード/コピー) を取り消す (キャラクタインタフェース) 480
    - ▼ スケジュールされた処理 (作成/アップグレード/コピー) を取り消す (コマンド行インタフェース) 480
  - ブート環境の比較 481
    - ▼ ブート環境を比較する (キャラクタインタフェース) 481
    - ▼ ブート環境を比較する (コマンド行インタフェース) 482
  - 非アクティブブート環境の削除 482
    - ▼ 非アクティブブート環境を削除する (キャラクタインタフェース) 483
    - ▼ 非アクティブブート環境を削除する (コマンド行インタフェース) 483
  - アクティブブート環境の名前の表示 483
    - ▼ アクティブブート環境の名前を表示する (キャラクタインタフェース) 484
    - ▼ アクティブブート環境の名前を表示する (コマンド行インタフェース) 484
  - ブート環境の名前の変更 484

- ▼ 非アクティブブート環境の名前を変更する (キャラクタインタフェース) 485
- ▼ 非アクティブブート環境の名前を変更する (コマンド行インタフェース) 485
- ブート環境名に関連付ける説明の作成または変更 486
  - ▼ テキストを使用してブート環境名の説明を作成または変更する方法 486
  - ▼ ファイルを使用してブート環境名の説明を作成または変更する方法 486
  - ▼ テキストで記述された説明からブート環境名を確認する方法 487
  - ▼ ファイル内の説明からブート環境名を確認する方法 487
  - ▼ 名前からブート環境説明を確認する方法 488
- ブート環境の構成の表示 488
  - ▼ 非アクティブブート環境の構成を表示する (キャラクタインタフェース) 488
  - ▼ ブート環境の構成を表示する (コマンド行インタフェース) 489

### 38 Solaris Live Upgrade (例) 491

- Solaris Live Upgrade によるアップグレードの使用例 (コマンド行インタフェース) 491
  - アクティブブート環境で Live Upgrade をインストールする 492
  - ブート環境を作成する 492
  - 非アクティブブート環境をアップグレードする 493
  - ブート環境がブート可能か確認する 493
  - 非アクティブブート環境をアクティブにする 493
  - ソースブート環境へフォールバックする 493
- RAID 1 ボリューム (ミラー) の一方を切り離してアップグレードする例 (コマンド行インタフェース) 494
- 既存のボリュームから Solaris ボリュームマネージャ RAID-1 ボリュームへの移行例 (コマンド行インタフェース) 498
- 空のブート環境を作成してフラッシュアーカイブをインストールする例 (コマンド行インタフェース) 498
  - 空のブート環境の作成 499
  - 新しいブート環境へのフラッシュアーカイブのインストール 500
  - 新しいブート環境のアクティブ化 501
- Solaris Live Upgrade によるアップグレードの例 (キャラクタインタフェース) 501
  - アクティブブート環境で Live Upgrade をインストールする 501
  - ブート環境を作成する 502
  - 非アクティブブート環境をアップグレードする 504
  - 非アクティブブート環境をアクティブにする 505

<b>39</b>	<b>Solaris Live Upgrade (コマンドリファレンス)</b>	<b>507</b>
<b>40</b>	<b>WAN ブートによる広域ネットワーク経由のインストール (トピック)</b>	<b>509</b>
<b>41</b>	<b>WAN ブート (概要)</b>	<b>511</b>
	WAN ブートとは	511
	どのような場合に WAN ブートを使用するか	512
	WAN ブートのしくみ (概要)	513
	WAN ブートインストールでのイベントの順序	513
	WAN ブートインストール時のデータの保護	515
	WAN ブートでサポートされているセキュリティ構成 (概要)	517
	セキュリティ保護された WAN ブートインストール構成	517
	セキュリティ保護されていない WAN ブートインストール構成	518
<b>42</b>	<b>WAN ブートによるインストールの準備 (計画)</b>	<b>519</b>
	WAN ブートの要件とガイドライン	519
	Web サーバーソフトウェアの要件とガイドライン	521
	サーバー構成オプション	521
	ドキュメントルートディレクトリへのインストールファイルと構成ファイルの保存	522
	/etc/netboot ディレクトリへの構成情報とセキュリティ情報の保存	524
	wanboot-cgi プログラムの保存	527
	デジタル証明書の要件	527
	WAN ブートのセキュリティ限界	528
	WAN ブートインストールに必要な情報の収集	528
<b>43</b>	<b>WAN ブートによるインストールの準備 (作業)</b>	<b>531</b>
	広域ネットワーク経由でインストールを行うための準備 (作業マップ)	531
	WAN ブートサーバーの構成	535
	ドキュメントルートディレクトリの作成	536
	WAN ブートミニルートの作成	536
	WAN ブートサーバーへの wanboot プログラムのインストール	539
	WAN ブートサーバーに /etc/netboot ディレクトリを作成する	540
	WAN ブートサーバーへの WAN ブート CGI プログラムのコピー	543
	(省略可能) WAN ブートログサーバーの構成	544
	(省略可能) HTTPS によるデータの保護	545



カスタム JumpStart インストールファイルの作成	550
フラッシュアーカイブの作成	551
sysidcfg ファイルの作成	552
プロファイルの作成	553
rules ファイルの作成	554
(省略可能) begin スクリプトと finish スクリプトの作成	556
構成ファイルの作成	556
システム構成ファイルの作成	557
wanboot.conf ファイルの作成	559
(省略可能) DHCP による構成情報の提供	563
<b>44 SPARC: WAN ブートによるインストール (作業)</b>	<b>565</b>
作業マップ: WAN ブートによるクライアントのインストール	565
WAN ブートインストールを行うためのクライアントの準備	566
クライアント OBP での WAN ブート対応の確認	567
クライアント OBP の net デバイス別名の確認	568
クライアントに対するキーのインストール	569
クライアントのインストール	574
自動 WAN ブートインストールの実行	575
対話式 WAN ブートインストールの実行	577
DHCP サーバーを使ったインストール	581
ローカルの CD メディアを使ったインストール	583
<b>45 SPARC: WAN ブートによるインストール (例)</b>	<b>589</b>
サイトの設定例	590
ドキュメントルートディレクトリの作成	591
WAN ブートミニルートの作成	591
WAN ブートサーバーへの wanboot プログラムのインストール	592
/etc/netboot ディレクトリの作成	592
WAN ブートサーバーへの wanboot-cgi プログラムのコピー	593
WAN ブートサーバーをログサーバーとして構成	593
HTTPS を使用するための WAN ブートサーバーの構成	593
クライアントに対する信頼できる証明書の提供	594
(省略可能) クライアント認証用の非公開鍵と証明書の使用	594
サーバーとクライアントのキーの作成	595
フラッシュアーカイブの作成	596
sysidcfg ファイルの作成	596

	クライアントのプロファイルの作成	597
	rules ファイルの作成と妥当性検査	597
	システム構成ファイルの作成	598
	wanboot.conf ファイルの作成	599
	クライアント OBP での WAN ブート対応の確認	600
	OBP の net デバイス別名の確認	601
	クライアントに対するキーのインストール	601
	クライアントのインストール	602
<b>46</b>	<b>WAN ブート (リファレンス)</b>	<b>605</b>
	WAN ブートインストールのコマンド	605
	OBP コマンド	608
	システム構成ファイルの設定と構文	609
	wanboot.conf ファイルのパラメータと構文	610
<b>A</b>	<b>Solaris ソフトウェアに関するリファレンス情報 (トピック)</b>	<b>615</b>
<b>B</b>	<b>Solaris 9 メディアの構成 (リファレンス)</b>	<b>617</b>
	SPARC: Solaris 9 メディア	617
	SPARC: Solaris 9 メディアのディレクトリ構造	618
	SPARC: Solaris DVD (SPARC 版) のディレクトリ構造	619
	SPARC: Solaris INSTALLATION CD のディレクトリ構造	620
	SPARC: Solaris SOFTWARE (SPARC 版) CD のディレクトリ構造	621
	SPARC: Solaris LANGUAGES CD (SPARC 版) のディレクトリ構造	622
	x86: Solaris 9 メディア	624
	x86: Solaris 9 メディアのディレクトリ構造	625
	Solaris DVD (x86 版) ディレクトリの構造	625
	x86: Solaris INSTALLATION CD のディレクトリ構造	627
	x86: Solaris SOFTWARE (x86 版) CD のディレクトリ構造	628
	x86: Solaris LANGUAGES CD (x86 版) のディレクトリ構造	630

<b>C</b>	プラットフォーム名とプラットフォームグループ (リファレンス)	<b>633</b>
<b>D</b>	ロケールの値 (リファレンス)	<b>635</b>
<b>E</b>	問題発生時の解決方法 (作業)	<b>643</b>
	ネットワークインストールの設定に関する問題	643
	システムのブートに関する問題	644
	メディアからのブート時のエラーメッセージ	644
	メディアからのブート時の一般的な問題	645
	ネットワークからのブート時のエラーメッセージ	648
	ネットワークからのブート時の一般的な問題	651
	Solaris 9 オペレーティング環境の初期インストール時の問題	652
	▼ x86: IDE ディスクの不良ブロックの検査	652
	Solaris 9 オペレーティング環境のアップグレード時の問題	654
	アップグレード時のエラーメッセージ	654
	アップグレード時の一般的な問題	655
	▼ 問題発生後にアップグレードを継続する方法	656
	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	656
	x86: 既存の Service 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 Service パーティションが作成されない	658
<b>F</b>	リモートからのインストールまたはアップグレード (作業)	<b>661</b>
	SPARC: Solaris Web Start を使用してリモート DVD-ROM または CD-ROM からインストールまたはアップグレードする	661
	▼ SPARC: リモート DVD-ROM または CD-ROM 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	662
<b>G</b>	その他の SVR4 パッケージ要件 (リファレンス)	<b>665</b>
	パッケージ化要件の概要	665
	カスタム JumpStart プログラムと Solaris Live Upgrade の非アクティブブート環境要件	666
	\$PKG_INSTALL_ROOT と \$BASEDIR の相違点の概要	667
	Solaris Live Upgrade 非アクティブブート環境の要件	668
	カスタム JumpStart プログラムのアップグレード要件	670

<b>H</b>	<b>Solaris Update</b>	リリースへのアップグレード (作業)	<b>673</b>
	Solaris Update	リリースへのアップグレード	673
		▼ analyze_patches スクリプトを実行する方法	674
		▼ パッチアナライザの出力の確認方法	675
<b>I</b>	<b>x86: Solaris Device Configuration Assistant</b>	(デバイス構成用補助) またはネットワークからのブート準備 (作業)	<b>677</b>
	x86:	ブートソフトウェアのフロッピーディスクへのコピー	677
		▼ x86: ブートソフトウェアをフロッピーディスクにコピーする方法	678
	x86:	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	679
		PXE の概要	679
	x86:	PXE によるブートに関するガイドライン	679
		PXE によるブート (作業マップ)	680
	x86:	x86 インストールサーバーの作成	681
	x86:	PXE を使用してネットワーク経由でインストールするシステムの追加	684
	x86:	PXE を使用したネットワーク経由でのクライアントのブート	689
<b>J</b>	<b>Solaris 9</b>	での更新情報	<b>691</b>
	Solaris 9	9/02 リリース	691
	Solaris 9	12/02 リリース	691
	Solaris 9	4/03 リリース	692
	Solaris 9	8/03 リリース	693
	Solaris 9	12/03 リリース	694
	Solaris 9	4/04 リリース	695
	用語集		<b>697</b>
	索引		<b>713</b>

## 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に、Solaris™ 9 オペレーティング環境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 - Solaris オペレーティング環境は、SPARC と x86 の 2 種類のハードウェア (プラットフォーム) で動作します。特別な章、節、注、リスト、図、表、例、またはコード例でない限り、このマニュアルの情報は両方のプラットフォームに適用されます。

---

---

注 - このマニュアルでは、「x86」という用語は、Intel 32 ビット系列のマイクロプロセッサチップ、および AMD が提供する互換マイクロプロセッサチップを意味します。

---

---

## 対象読者

このマニュアルは、Solaris オペレーティング環境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参考となる関連情報の一覧を示します。

表 P-1 関連情報

インフォメーション	説明
『Solaris のシステム管理 (基本編)』	システムファイルのバックアップ方法が解説されています。
『Solaris 9 4/04 ご使用にあたって』	Solaris リリースに関する、バグ、既知の問題、サポートが中止されたソフトウェア、パッチなどが解説されています。
SPARC: 『Solaris 9 Sun ハードウェアマニュアル』	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 9 4/04 パッケージリスト』	Solaris 9 4/04 オペレーティング環境に含まれるパッケージの一覧と説明です。
x86: 『Solaris (x86 Platform Edition) 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。

---

## Sun のオンラインマニュアル

docs.sun.com では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。URL は、<http://docs.sun.com> です。

---

## 表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-2 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。  system%
<b>AaBbCc123</b>	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% <b>su</b> password:
<i>AaBbCc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。  この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% <b>grep</b> `^#define \ XV_VERSION_STRING`

コード例は次のように表示されます。

■ C シェル

```
machine_name% command y|n [filename]
```

■ C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

■ Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

■ Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

---

## 一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が、適宜併記されています。



## 第 1 章

---

# Solaris のインストールおよびアップグレードの計画 (トピック)

---

以下の章では、Solaris オペレーティング環境のインストールやアップグレードの計画について説明します。

第 2 章	インストールやアップグレードを行う前に決定すべき項目について説明します。
第 3 章	Solaris をインストールするさまざまな方法について詳しく説明します。インストール環境にとってどの方法が最適であるかを判断する際の参考になります。



## 第 2 章

---

# Solaris のインストールおよびアップグレードの計画 (概要)

---

この章では、Solaris オペレーティング環境のインストールやアップグレードを行う前に決定すべき項目について説明します。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」
- 29 ページの「初期インストールかアップグレードか」
- 31 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」
- 32 ページの「DVD メディアの使用」
- 32 ページの「Sun ONE Application Server のインストール後の構成」
- 33 ページの「x86: Solaris 9 Device Configuration Assistant (デバイス構成用補助) と PXE へのアクセス」

---

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。混同を避けるため、このマニュアルでは、x86 版 Solaris でのみサポートされる fdisk パーティションと、スライスまたはパーティションと呼ばれる Solaris fdisk パーティション内の分割とを区別しています。

---

---

## 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、Solaris オペレーティング環境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ど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する	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。	29 ページの「初期インストールかアップグレードか」
インストール方法を選択する	Solaris オペレーティング環境では、インストールやアップグレードをいくつかの方法で行うことができます。インストール環境に最も適した方法を選択してください。	第 3 章
システム要件を検討する。また、ディスク容量およびスワップ領域の計画を立てて割り当てる	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 オペレーティング環境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。システムに適したスワップ領域の配置を決定します。	第 5 章
インストール媒体を選択する	Solaris ソフトウェアのインストールは、ローカル媒体から行うこともネットワークから行うこともできます。インストール環境に最も適したインストール媒体を選択してください。	31 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」
システム情報を収集する	チェックリストやワークシートを使って、インストールやアップグレードに必要なすべての情報を収集します。	第 6 章
(省略可能) システム構成情報を事前設定する	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。	第 7 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う	Solaris ソフトウェアをネットワークからインストールする場合は、インストールサーバーを作成し、必要に応じてブートサーバーを作成し、ネットワークからインストールできるように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、第 15 章を参照 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、第 43 章を参照
(アップグレードのみ) アップグレード前に必要な作業を行う	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どうかを判断し、Solaris Update リリースによって上書きされる可能性があるパッチを検索します。	第 8 章
インストールまたはアップグレードを行う	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	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 E

---

## 初期インストールかアップグレードか

まず、初期インストールを行うのかアップグレードを行うのかを選択する必要があります。アップグレードを選択する場合は、Solaris オペレーティング環境がすでにシステム上で動作している必要があります。

### 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 オペレーティング環境の新しいバージョンで上書きされます。システム上で Solaris オペレーティング環境がまだ動作していない場合は、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 オペレーティング環境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したい場合は、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り、インストールが終わった後でローカル変更を復元する必要があります。

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。Solaris のインストール方法については、第 3 章を参照してください。

### アップグレード

アップグレードでは、システムのディスクにある既存のファイルと Solaris オペレーティング環境の新しいバージョンがマージされます。アップグレードでは、既存の Solaris オペレーティング環境に対して行なった変更はできるだけ保存されます。

アップグレードできるシステムは、Solaris 2.6、Solaris 7、または Solaris 8 ソフトウェアが動作しているものです。システムで動作している Solaris のバージョンを確認するには、次のコマンドを入力します。

```
$ uname -a
```

Solaris オペレーティング環境のアップグレードでは、次のインストール方法を使用できます。

---

注 - ディスクレスクライアントをアップグレードする場合は、`smosservice patch` コマンドを使用する必要があります。詳細は、『Solaris のシステム管理 (基本編)』または `smosservice(1M)` のマニュアルページを参照してください。

---

## Solaris のアップグレード方法

表 2-2 SPARC: Solaris のアップグレード方法

現在の Solaris オペレーティング環境	Solaris のアップグレード方法
Solaris 2.6、Solaris 7、Solaris 8、Solaris 9	<ul style="list-style-type: none"><li>■ Solaris™ Web Start プログラム</li><li>■ Solaris suninstall プログラム</li><li>■ カスタム JumpStart™</li><li>■ Solaris Live Upgrade</li></ul>

表 2-3 x86: Solaris のアップグレード方法

現在の Solaris オペレーティング環境	Solaris のアップグレード方法
Solaris 2.6	DVD メディアまたはネットインストールイメージからのインストール: <ul style="list-style-type: none"><li>■ Solaris Web Start プログラム</li><li>■ Solaris suninstall プログラム</li><li>■ カスタム JumpStart</li></ul> CD メディアからのインストール: <ul style="list-style-type: none"><li>■ Solaris suninstall プログラム</li><li>■ カスタム JumpStart</li></ul>
Solaris 7	DVD メディアまたはネットインストールイメージからのインストール: <ul style="list-style-type: none"><li>■ Solaris Web Start プログラム</li><li>■ Solaris suninstall プログラム</li><li>■ カスタム JumpStart</li><li>■ Solaris Live Upgrade</li></ul> CD メディアからのインストール: <ul style="list-style-type: none"><li>■ Solaris suninstall プログラム</li><li>■ カスタム JumpStart</li><li>■ Solaris Live Upgrade</li></ul>
Solaris 8、Solaris 9	DVD/CD メディアまたはネットインストールイメージからのインストール: <ul style="list-style-type: none"><li>■ Solaris Web Start プログラム</li><li>■ Solaris suninstall プログラム</li><li>■ カスタム JumpStart</li><li>■ Solaris Live Upgrade</li></ul>

## アップグレードに関する制限事項

---

注 – Solaris Live Upgrade を使用したアップグレードの制限については、397 ページの「Solaris Live Upgrade の要件」を参照してください。

---

アップグレード時には、次の点に注意してください。

- 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 Update リリースへのアップグレード: すでに Solaris 9 オペレーティング環境を実行していて、システムに個別のパッチがインストールされている場合、Solaris 9 Update リリースへのアップグレードを行うと、パッチは次のようになります。
  - Solaris 9 Update リリースの一部として組み込まれているパッチは自動的に再適用されます。これらのパッチをバックアウトすることはできません。
  - システムにすでにインストールされているパッチのうち、Solaris 9 Update リリースに組み込まれていないパッチは削除されます。

パッチアナライザを使用すれば、Solaris 9 Update リリースにアップグレードしたときに、これらのパッチのうちどれが削除されるかを確認することができます。パッチアナライザの詳しい使い方については、673 ページの「Solaris Update リリースへのアップグレード」を参照してください。

---

## 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。

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や、Solaris ソフトウェアを複数のシステムにインストールする際に、それぞれのシステムに対してローカルドライブにディスクを挿入したくない場合は、リモートの DVD イメージまたは CD イメージからインストールするようにシステムを設定できます。

ネットワークからインストールする場合は、どの Solaris インストール方法でも使用できます。ただし、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。各インストール方法の詳細は、第 3 章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、次のど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については、第 15 章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場合の準備については、第 43 章を参照してください。
- 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方法については、679 ページの「x86: 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」を参照してください。

---

## DVD メディアの使用

DVD メディアを使用している場合、ok プロンプトからブートするように求められたときには、常に次のコマンドを入力してください。

```
ok boot cdrom
```

---

## Sun ONE Application Server のインストール後の構成

Sun™ ONE Application Server を使用している場合、Solaris オペレーティング環境のインストール後にサーバーを構成する必要があります。Application Server の構成に関する詳細は、『Sun ONE Application Server 7 入門ガイド』を参照してください。



---

## x86: Solaris 9 Device Configuration Assistant (デバイス構成用補助) と PXE へのアクセス

Solaris Device Configuration Assistant (デバイス構成用補助) は、さまざまなハードウェア構成作業とブート作業を行うためのプログラムです。このプログラムを使用して、DVD、CD、ネットインストールイメージ、またはフロッピーディスク上のソフトウェアコピーからブートできます。Solaris Device Configuration Assistant (デバイス構成用補助) には、次の方法でアクセスします。

- Solaris DVD、Solaris INSTALLATION CD、または Solaris SOFTWARE 1 of 2 CD からブートする。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。
- PXE (Preboot Execution Environment) を使用して Solaris Software のネットインストールイメージからブートする。PXE を使用すると、ブート用フロッピーディスクを使用せずにネットワークから直接システムをブートできます。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。PXE ネットワークブート用にシステムを構成する方法については、679 ページの「x86: 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」を参照してください。
- フロッピーディスクからブートする。付録 I のコピー手順に従って Solaris DVD (x86 版) または Solaris SOFTWARE 2 of 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。



## 第 3 章

---

# Solaris のインストール方法の選択 (概要)

---

この章では、Solaris をインストールする方法について説明します。Solaris オペレーティング環境では、インストールやアップグレードをいくつかの方法で行うことができます。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。したがって、インストール環境に最も適した方法を選択してください。

- 35 ページの「Solaris Web Start」
- 36 ページの「Solaris suninstall プログラム」
- 36 ページの「カスタム JumpStart」
- 37 ページの「フラッシュインストール機能」
- 38 ページの「WAN ブートインストール」
- 38 ページの「Solaris Live Upgrade」
- 39 ページの「SPARC: ファクトリ JumpStart」

---

## Solaris Web Start

Solaris DVD または Solaris INSTALLATION CD に含まれる Solaris Web Start インストールプログラムは、グラフィカルユーザーインターフェース (GUI) でも、コマンド行インターフェース (CLI) でも起動できます。Solaris Web Start では、Solaris ソフトウェアや追加ソフトウェアのインストールやアップグレードに必要な手順がステップごとに示されます。システム管理者は、デフォルトオプションを使ってインストールすることも、カスタマイズオプションを使って必要なソフトウェアだけをインストールすることもできます。

Solaris Web Start では、Solaris オペレーティング環境や UNIX の初心者であっても、インストールの際に先に進んだり前に戻って必要な変更を簡単に行うことができます。インストール作業は複数の画面に分かれており、各画面にはユーザーが選択できるようにデフォルトの値が表示されます。

Solaris Web Start プログラムでは情報を入力する必要があるため、ユーザーはインストールプログラムと対話方式で処理を進める必要があります。したがって、複数のシステムのインストールやアップグレードを行う際には、このインストール方法が最適であるとは限りません。多数のシステムを対象としたバッチインストールには、カスタム JumpStart かフラッシュインストール機能を使用してください。

詳細は、第 17 章を参照してください。

---

## Solaris suninstall プログラム

Solaris SOFTWARE 1 of 2 CD に含まれる Solaris suninstall プログラムは、コマンド行インタフェース (CLI) で動作します。Solaris suninstall プログラムでは、Solaris 9 ソフトウェアのインストールやアップグレードに必要な手順がステップごとに示されます。このインストール方法は、GUI を実行するのに十分なメモリーがなく、国際ロケールを使用している場合に便利です。

Solaris suninstall プログラムでは Solaris オペレーティング環境だけがインストールされます。このプログラムは、サードパーティアップグレードやネットワーク上でダウンロードできるソフトウェアなどを認識できません。サードパーティアプリケーションは、Solaris オペレーティング環境のインストール後にインストールしてください。また、インストール時にはシステム構成情報を入力するように求められます。このため、複数のシステムをインストールする場合は、suninstall は最適とは言えません。サードパーティアプリケーションのインストールには Solaris Web Start プログラムを使用できます。多数のシステムを対象としたバッチインストールには、カスタム JumpStart かフラッシュインストール機能を使用してください。

詳細は、第 18 章を参照してください。

---

## カスタム JumpStart

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 は、指定されたプロファイルとスクリプトに従ってシステムのインストールやアップグレードを行います。

Solaris オペレーティング環境に関する知識を持っていて、複数のシステムをインストールする必要がある場合には、カスタム JumpStart が最適であるかもしれません。インストールするシステムが 2、3 台だけの場合には、このインストール方法が最適であるとは限りません。カスタム JumpStart 環境の作成に時間がかかる可能性があるからです。オペレーティング環境に関する知識を持っていない場合や、少数のシステムのインストールを行う場合は、Solaris Web Start インストールプログラムを使用してください。

詳細は、第 26 章を参照してください。

---

## フラッシュインストール機能

フラッシュインストール機能では、マスターシステムにインストールする構成を使用して、多数のシステムをインストールすることができます。それには、マスターシステムのインストールと構成を行なったあとに、マスターシステムからフラッシュアーカイブを作成する必要があります。フラッシュアーカイブは、必要に応じていくつでも作成できます。それぞれのシステムにインストールする際に、使用するフラッシュアーカイブを選択します。このインストール方法では、同じソフトウェアと構成を持つ多数のシステムを効率的にインストールできます。

フラッシュアーカイブを使用しない Solaris インストール方法では、各 Solaris パッケージが個別にインストールされます。パッケージベースのインストールではパッケージごとにパッケージマップの更新が必要になるため、時間がかかります。フラッシュアーカイブによるインストールは、個々の Solaris パッケージをインストールする場合よりもずっと早く終わります。

どの Solaris インストール方法を使用する場合でも、フラッシュアーカイブの初期インストールを複製できます。Solaris Web Start と Solaris suninstall プログラムでは、インストールするフラッシュアーカイブを選択するように指示されます。カスタム JumpStart では、インストールするフラッシュアーカイブをプロファイルに指定します。Solaris Live Upgrade では、非アクティブブート環境にインストールするフラッシュアーカイブを指定します。

インストール済みのクローンシステムを更新する場合は、差分アーカイブを使用して相違部分だけをインストールできます。差分アーカイブには、2 つのシステムイメージの相違部分のみが含まれています。差分アーカイブをインストールするには、カスタム Jumpstart インストールか Solaris Live Upgrade を使用します。

複数のシステムに多くの異なる構成でインストールしたい場合には、システムごとにフラッシュアーカイブが必要になります。フラッシュアーカイブはファイルサイズが大きいため、大量のディスク容量が必要です。多数の異なるインストール構成が必要であったり、インストール構成を変更する柔軟性を残しておきたい場合には、カスタム JumpStart インストールを使用することを検討してください。また、JumpStart finish スクリプトまたは組み込み済みのフラッシュ配置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することも可能です。

初期インストールや更新にあたってのアーカイブのインストールやスクリプトの使用を含む、フラッシュインストール機能の概要は、第 20 章を参照してください。

カスタム JumpStart を使用したアーカイブのインストール方法の詳細は、315 ページの「カスタム JumpStart インストールを使用して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。

---

## WAN ブートインストール

WAN ブートインストールでは、ハイパーテキストトランスポートプロトコル (HTTP) を使用して、広域ネットワーク (WAN) 経由でソフトウェアのブートとインストールを行うことができます。WAN ブートを使用すると、インターネットなどの大規模なパブリックデータネットワークを介して Solaris オペレーティング環境をシステムにインストールできますが、これらのネットワークはインフラストラクチャの信頼性が低い場合があります。さまざまなセキュリティ機能を使ってデータの機密性とインストールイメージの完全性を保護できます。

WAN ブートインストールでは、暗号化された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。インストールの完全性を保護するために、非公開鍵を使ってデータの認証および暗号化を行うことができます。また、デジタル証明書を使用するようにシステムを構成すると、HTTPS 接続を介してインストールデータやファイルを転送できます。

WAN ブートインストールの概要と詳細については、第 40 章を参照してください。

---

## Solaris Live Upgrade

Solaris Live Upgrade では、アクティブブート環境が稼動している間に複製ブート環境のアップグレードが行え、稼動中の環境のダウンタイムをなくすことができます。Solaris Live Upgrade は、CUI または CLI のどちらでも行えます。まず、複製ブート環境を作成する必要があります。複製ブート環境を作成した後で、このブート環境をアップグレードするか、あるいは、この非アクティブブート環境にフラッシュアーカイブまたは差分アーカイブをインストールできます。準備が整った時点で、この非アクティブブート環境をアクティブにします。次のリブート時には、この非アクティブブート環境がアクティブブート環境に切り替わります。何か問題が発生する場合は、本来のブート環境をアクティブにしてリブートするだけで元どおりに復元できます。

詳しい説明と操作方法については、第 32 章を参照してください。

---

## SPARC: ファクトリ JumpStart

ファクトリ JumpStart インストールでは、Solaris DVD または Solaris SOFTWARE 1 of 2 CD をドライブに挿入して、システムの電源を入れるだけで、新しい SPARC システムに Solaris ソフトウェアを自動的にインストールできます。その際、システムの機種とディスクサイズに基づいてデフォルトのプロファイルが選択されます。システムにどのソフトウェアコンポーネントをインストールするかは、このプロファイルで決まります。システム構成情報を入力するように求められることはなく、インストールするソフトウェアを選択することはできません。

SPARC ベースの新しいシステムには、このインストール方法を使用する場合に必要な JumpStart ブートイメージがあらかじめインストールされています。古い SPARC ベースシステムの場合は、`re-preinstall (1M)` コマンドを使用すれば、JumpStart ブートイメージをシステムにインストールできます。x86 ベースのシステムでは、ファクトリ JumpStart インストールを使用することはできません。





## 第 4 章

---

# Solaris のインストールおよびアップグレードの準備 (トピック)

---

以下の章では、Solaris オペレーティング環境のインストールおよびアップグレードを行うための準備について説明します。

第 5 章	Solaris オペレーティング環境のインストールやアップグレードの際に必要なディスク容量を計画するための指針を示します。
第 6 章	Solaris のインストールを始める前にどのようなシステム情報を取得すべきかを説明します。
第 7 章	Solaris のインストールに必要なシステム構成情報を <code>sysidcfg</code> ファイルを使って事前設定する方法を説明します。
第 8 章	Solaris オペレーティング環境をアップグレードする前に必要な作業について説明します。



## 第 5 章

---

# システム要件とガイドライン (計画)

---

この章では、Solaris オペレーティング環境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、ディスク容量の計画に関しての一般的な指針や、スワップ領域のデフォルトの割り当てについても説明します。この章の内容は次のとおりです。

- 43 ページの「メモリー要件」
- 44 ページの「ディスク容量の割り当て」
- 46 ページの「SPARC: スワップ領域の割り当て方法」
- 47 ページの「Solaris INSTALLATION CD を使用するための要件」
- 48 ページの「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」

---

## メモリー要件

Soralis オペレーティング環境をインストールまたはアップグレードするには、128M バイト以上のメモリーを用意することをお勧めします。

---

注 - オプションのインストール機能の中には、メモリーが十分でないと有効にできないものもあります。たとえば、メモリーが十分に存在しないシステムに DVD からインストールする場合、Solaris Web Start インストールプログラムのグラフィカルユーザーインターフェースではなく、Solaris Web Start のコマンド行インターフェースを使用する必要があります。

---

---

## ディスク容量の割り当て

Solaris 9 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。必要なディスク容量が事前にわかれば、Solaris 9 ソフトウェアを実際にインストールする前にディスクを増設しておくことができます。アップグレードを行う場合は、90 ページの「ディスク容量の再配置を使用するアップグレード」を参照してください。

## 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。ディスク容量の計画にあたっては、次の一般的な事項を考慮に入れてください。

- 中国語、日本語、韓国語など、インストールする言語を追加するごとにディスク容量を増やします。
- 印刷やメールをサポートする場合は、/var ファイルシステムのディスク容量を増やします。
- 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。
- サーバーがほかのシステム上のユーザーに対してホームディレクトリファイルシステムを提供する場合は、サーバーのディスク容量を増やします。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。
- スワップ領域に 512M バイト以上を割り当てます。
- インストールする Solaris ソフトウェアグループ用にディスク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は表 5-1 のとおりです。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。
- 作成するファイルシステムの数を最小限に抑えます。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。ファイルシステムを割り当てる場合には、将来の Solaris バ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。新しい Solaris 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が必要になります。ファイルシステムごとに 30% のディスク容量を追加しておくことによって、システムのスライスを切り直さなくても Solaris をアップグレードできます。
- 追加ソフトウェアやサードパーティソフトウェア用にディスク容量を増やします。

Solaris Live Upgrade を使用する場合は、399 ページの「Solaris Live Upgrade のディスク容量の要件」を参照してください。

## 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。したがって、システムでどの機能を実行するかを考慮して、インストールするソフトウェアグループを選択する必要があります。

- 「コアシステムサポート」ソフトウェアグループ – 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が含まれています。
- 「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 – 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が含まれています。
- 「開発者システムサポート」ソフトウェアグループ – 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ものです。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。
- 「全体ディストリビューション」ソフトウェアグループ – 開発者システムサポートソフトウェアグループに、サーバーに必要なソフトウェアが追加されたものです。
- 「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 – 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ものです。これには、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。

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。

次の表に、各 Solaris ソフトウェアグループごとに、推奨するディスク容量を示します。

---

注 – 推奨ディスク容量には、次の項目の容量も含まれています。

- スワップ領域
- パッチ
- 追加のソフトウェアパッケージ

各ソフトウェアグループに必要なディスク容量は、次の表に記載されている容量より少ない場合があります。

---

表 5-1 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	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	2.9G バイト
全体ディストリビューション	2.7G バイト
開発者システムサポート	2.4G バイト
エンドユーザーシステムサポート	2.0G バイト
コアシステムサポート	2.0G バイト

## SPARC: スワップ領域の割り当て方法

デフォルトでは、Solaris インストールプログラムは、利用可能な最初のディスクシリンダ (SPARC システムでは通常シリンダ 0) でスワップが開始されるようにスワップ領域を割り当てます。この配置により、デフォルトのディスクレイアウトでルート (/) ファイルシステムに最大の容量を確保し、アップグレード時にルート (/) ファイルシステムを拡張できます。インストールプログラムによるディスクスライスの自動レイアウトを選択し、スワップファイルのサイズを手作業で変更しなければ、Solaris インストールプログラムはデフォルトのスワップ領域 (512M バイト) を割り当てます。

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 Web Start および Solaris suninstall 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。
- Solaris カスタム JumpStart インストールプログラムの場合、プロファイルファイル内でスワップスライスを割り当てることができます。Solaris カスタム JumpStart プロファイルファイルの詳細は、274 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 (基本編)』の「追加スワップ空間の構成 (手順)」を参照してください。

# Solaris INSTALLATION CD を使用する ための要件

Solaris INSTALLATION CD を使用してインストールまたはアップグレードを行う場合、SPARC スライスおよび x86 の fdisk パーティションに関連した特別な要件が存在します。DVD またはネットインストールイメージからインストールを行う場合、これらの要件を考慮する必要はありません。

表 5-2 Solaris INSTALLATION CD の要件

プラットフォーム	要件
アップグレードに伴うスライス要件	Solaris INSTALLATION CD と Solaris Web Start プログラムを使ってアップグレードを行うためには、ファイルが保存されていないスライスがディスク上になければなりません。望ましいスライスは swap スライスですが、/etc/vfstab にリストされている「アップグレード対象」のルートスライスのどれにも属していないスライスであれば、どのスライスでもかまいません。このスライスのサイズは少なくとも 512M バイトが必要です。
x86 システムの fdisk パーティションの要件	<p>Solaris INSTALLATION CD を使う場合は、Solaris Web Start プログラムはインストールまたはアップグレードのためにシステムディスク上に 2 つの fdisk パーティションを必要とします。</p> <ul style="list-style-type: none"><li>■ Solaris fdisk パーティション 通常の Solaris fdisk パーティションです。システム上に Solaris fdisk パーティションがない場合は、その作成を求めるプロンプトが Solaris Web Start から出されます。 注意:既存の fdisk パーティションのサイズを変更すると、そのパーティションにあるすべてのデータが自動的に削除されます。Solaris fdisk パーティションを作成する場合は、データのバックアップをとってください。</li><li>■ x86BOOT fdisk パーティション x86 システムをミニルートから起動するための、10M バイトの fdisk パーティションです。ミニルートは、Solaris fdisk パーティションに新たに作成されるスワップスライスに置かれます。 注意:x86BOOT fdisk パーティションを手動で作成しないでください。 Solaris Web Start インストールプログラムは、Solaris fdisk パーティションを分割し、10M バイトの x86BOOT fdisk パーティションを作成します。このように x86BOOT fdisk パーティションの作成をインストールプログラムに任せることによって、既存の fdisk パーティションを変更してしまう危険が防止されます。</li></ul> <p>注-Service パーティションを保持するシステムをインストールまたはアップグレードする場合、Solaris Web Start インストールプログラムは Service パーティションを保持したまま、Solaris および x86BOOT fdisk パーティションを作成します。Service パーティションの保持については、48 ページの「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」を参照してください。</p>

表 5-2 Solaris INSTALLATION CD の要件 (続き)

プラットフォーム	要件
x86 システムのアップグレードに関する制限事項	<p>Solaris INSTALLATION CD を使用する場合、Solaris Web Start プログラムを使用して Solaris 2.6 または Solaris 7 オペレーティング環境からアップグレードすることはできません。Solaris INSTALLATION CD では、Solaris 2.6 または Solaris 7 リリースでは作成されなかった 10M バイトの x86 ブートパーティションが別に必要です。DVD またはネットワークインストールイメージから Solaris Web Start プログラムを使用するか、Solaris suninstall プログラムまたはカスタム JumpStart を使用して、アップグレードを行う必要があります。</p>
x86 システムの LBA 要件	<p>システムが 1024 シリンダの制限を超えてブートできない場合は、Solaris INSTALLATION CD は使用しないでください。LBA (Logical block addressing) を利用すると、この制限を超えるマシンのブートや、複数の Solaris ディスクドライブに渡るブートが可能になります。システムの BIOS およびデフォルトブートディスクの SCSI ドライバが LBA をサポートする場合は、Solaris INSTALLATION CD を使用してください。</p> <p>LBA がサポートされているかどうかを調べるには、次のコマンドを使用します。</p> <pre data-bbox="511 829 852 861"># prtconf -pv   grep -i lba</pre> <p>デフォルトブートディスクの BIOS および SCSI ドライバが LBA をサポートしている場合は、次のメッセージが出力されます。</p> <pre data-bbox="511 955 690 987">lba-access-ok:</pre> <p>デフォルトブートディスクの SCSI ドライバが LBA をサポートしていない場合は、次のメッセージが出力されます。</p> <pre data-bbox="511 1081 730 1113">no-bef-lba-access</pre> <p>BIOS およびデフォルトブートディスクの SCSI ドライバが LBA をサポートしない場合、Solaris DVD またはネットワークインストールイメージを使用してインストールまたはアップグレードを実行します。</p>

## 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

x86 版 Solaris 9 オペレーティング環境では、Solaris Web Start と suninstall インストールプログラムは、新しい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、Service パーティションを利用可能にします。システムに Service パーティションが含まれる場合、新しい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 - Service パーティションを含まないシステムに x86 版 Solaris 9 オペレーティング環境をインストールする場合、インストールプログラムは、デフォルトでは新たに Service パーティションを作成しません。システムに Service 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。

---

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 - Service 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 オペレーティング環境パーティション (ブートディスクの残りの領域)

このデフォルトレイアウトを使用する場合は、インストールプログラムによりブートディスクレイアウトを選択するよう求められたら、「デフォルト」を選択してください。



## 第 6 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)

この章では、システムのインストールやアップグレードを行う前に収集すべきすべての情報が列挙されたチェックリストとワークシートについて説明します。

- 51 ページの「インストール用のチェックリスト」
- 52 ページの「インストール用のワークシート」
- 57 ページの「アップグレード用のチェックリスト」
- 58 ページの「アップグレード用のワークシート」

## インストール用のチェックリスト

Solaris オペレーティング環境をインストールするための準備では、次のチェックリストを使用します。

- tip (1) 接続を使用して Solaris ソフトウェアをシステムにインストールする場合は、ウィンドウ表示が少なくとも横 80 桁、縦 24 行あることを確認します。  
tip ウィンドウの現在の大きさを調べるには、`stty` コマンドを使用します。詳細については、`stty(1)` のマニュアルページを参照してください。
- 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。
- Solaris INSTALLATION CD に入っている Solaris Web Start プログラムを使用してインストールする場合は、47 ページの「Solaris INSTALLATION CD を使用するための要件」を参照してください。
- ディスク容量が十分あることを確認します。詳細は、第 5 章を参照してください。
- SPARC: DVD メディアを使用している場合、`ok` プロンプトからブートするように求められたときには、常に次のコマンドを入力してください。**`boot cdrom`**
- 『Solaris 9 4/04 ご使用にあたって』やベンダーのリリースノートを参照して、使用するソフトウェアが新しい Solaris リリースでもサポートされていることを確認します。

- 『Solaris 9 Sun ハードウェアマニュアル』を参照して、使用するハードウェアがサポートされていることを確認します。
- システムに添付されている資料を参照して、使用するシステムやデバイスが Solaris リリースでサポートされていることを確認します。

## インストール用のワークシート

Solaris オペレーティング環境のインストールに必要な情報を収集する際に、次のワークシートを使用します。ただし、ワークシートに記載されているすべての情報を収集する必要はありません。使用するシステムに関連する情報だけを収集してください。

注- ネットワークに接続されないスタンドアロンシステムをインストールする場合は、\* のついた項目については無視してください。

表 6-1 インストール用ワークシート

インストールに必要な情報	説明/例	答を記入する
ネットワーク	このシステムはネットワークに接続されていますか。	はい/いいえ
DHCP *	このシステムでは、DHCP (Dynamic Host Configuration Protocol) を使ってネットワークインタフェースを構成しますか。	はい/いいえ
ホスト名	このシステムのホスト名	
IP アドレス *	DHCP を使用しない場合は、このシステムの IP アドレスを指定します。 例: 129.200.9.1	
サブネット *	DHCP を使用しない場合、このシステムはサブネットの一部ですか。 「はい」の場合は、サブネットのネットマスクを指定します。 例: 255.255.0.0	はい/いいえ
IPv6 *	このマシンで IPv6 を使用可能にしますか。	はい/いいえ

表 6-1 インストール用ワークシート (続き)

インストールに必要な情報	説明/例	答を記入する
Kerberos *	<p>このマシンに Kerberos セキュリティを構成しますか。</p> <p>「はい」の場合は、次の情報を収集します。</p> <p style="text-align: right;">デフォルトのレルム: 管理サーバー: 一次 KDC: (省略可能) 追加 KDC:</p>	はい/いいえ
ネームサービス *	このシステムではどのネームサービスを使用しますか。	NIS+/NIS/DNS/LDAP/使用しない
ドメイン名 *	このシステムでネームサービスを使用する場合は、システムが属するドメインの名前を指定します。	
NIS+ および NIS *	<p>ネームサーバーを指定しますか、それともインストールプログラムにネームサーバーの検索を任せますか。</p> <p>ネームサーバーを指定する場合は、次の情報を指定します。</p> <p style="text-align: right;">サーバーのホスト名: サーバーの IP アドレス:</p>	指定/検索
DNS *	<p>DNS サーバーの IP アドレスを指定します。DNS サーバーの IP アドレスを少なくとも1つ、最大3つまで指定します。</p> <p style="text-align: right;">サーバーの IP アドレス:</p> <p>DNS 検索を行うときに検索するドメインのリストを入力できます。</p> <p style="text-align: right;">検索ドメイン: 検索ドメイン: 検索ドメイン:</p>	

表 6-1 インストール用ワークシート (続き)

インストールに必要な情報	説明/例	答を記入する
LDAP *	<p>LDAP プロファイルに関する次の情報を指定します。</p> <p style="text-align: center;">プロファイル名:</p> <p style="text-align: center;">プロファイルサーバーのホスト名:</p> <p>LDAP プロファイルでプロキシ資格レベルを指定した場合、この情報を収集します。</p> <p style="text-align: center;">プロキシバインドの識別名:</p> <p style="text-align: center;">プロキシバインドのパスワード:</p>	
デフォルトルート *	<p>デフォルトルート IP アドレスを指定しますか、それとも Solaris Web Start インストールプログラムまたは Solaris suninstall プログラムに IP アドレスの検索を任せますか。</p> <p>デフォルトルートは、2つの物理ネットワーク間のトラフィック転送用のブリッジを提供します。IP アドレスは、ネットワーク上の各ホストを識別する一意の番号です。</p> <p>次のうちから選択できます。</p> <ul style="list-style-type: none"> <li>■ IP アドレスを指定できます。指定された IP アドレスを使用して /etc/defaultrouter ファイルが作成されます。システムをリポートすると、指定された IP アドレスがデフォルトルートになります。</li> <li>■ ソフトウェアに IP アドレスを検出させることができます。 <ul style="list-style-type: none"> <li>■ Solaris suninstall プログラムは、システムのリポート時に IP アドレスを検出します。</li> <li>■ Solaris Web Start インストールプログラムを使用して、デフォルトルートを検出できます。ただし、システムは、ICMP ルーター発見プロトコルを使用して自らを通知するルーターの存在するサブネット上になければなりません。</li> </ul> </li> <li>■ ルーターが存在しない場合、または今回はソフトウェアに IP アドレスを検出させない場合は、「なし」を選択します。リポート時に、ソフトウェアが自動的に IP アドレスの検出を試みます。</li> </ul> <p style="text-align: right;">ホスト IP アドレス:</p>	指定/検出/なし

表 6-1 インストール用ワークシート (続き)

インストールに必要な情報	説明/例	答を記入する
時間帯	デフォルトの時間帯をどのように指定しますか。	地域 GMT との時間差 時間帯ファイル
ロケール	どの地域のサポートをインストールしますか。	
SPARC: 電源管理 (電源管理システムをサポートする SPARC システムの場合のみ)	電源管理システムを使用しますか。 注 - 使用するシステムが Energy Star バージョン 3 以降に対応している場合、このプロンプトは表示されません。	はい/いいえ
プロキシサーバーの構成* (Solaris Web Start の場合のみ)	インターネットに直接接続できますか、それともインターネットへのアクセスにプロキシサーバーを経由する必要がありますか。 プロキシサーバーを使用する場合は、次の情報を指定します。  プロキシサーバーのホスト名: ポート:	直接接続/プロキシサーバー
自動的なリポートまたは CD/DVD 取り出し	ソフトウェアをインストールした後に自動的にリポートしますか。 ソフトウェアをインストールした後に CD/DVD を自動的に取り出しますか。	はい/いいえ はい/いいえ
ソフトウェアグループ	どの Solaris ソフトウェアグループをインストールしますか。	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 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 コアシステムサポート
カスタムパッケージ選択	インストールする Solaris ソフトウェアグループに対してソフトウェアパッケージの追加や削除を行いますか。 注 - パッケージの追加や削除を行う場合には、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。	

表 6-1 インストール用ワークシート (続き)

インストールに必要な情報	説明/例	答を記入する
SPARC: 64-ビット	64 ビットアプリケーションのサポートをインストールしますか。	はい/いいえ
ディスクの選択	Solaris ソフトウェアをどのディスクにインストールしますか。 例: c0t0d0	
x86: fdisk によるパーティション分割	Solaris fdisk パーティションの作成、削除、または変更を行いますか。 ファイルシステムを配置するディスクには、Solaris fdisk パーティションが必要です。1 つのシステムに作成できる x86 Boot パーティションは 1 つだけです。 システムに現在 Service パーティションがある場合、Solaris Web Start および suninstall インストールプログラムはデフォルトで Service パーティションを保存します。Service パーティションを保存しない場合、fdisk パーティションをカスタマイズする必要があります。Service パーティションの保持については、48 ページの「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」を参照してください。  fdisk パーティションのカスタマイズのためにディスクを選択しますか。  fdisk パーティションをカスタマイズしますか。	はい/いいえ  はい/いいえ
データの保存	Solaris ソフトウェアをインストールするために選択したディスク上の既存のデータを保存しますか。	はい/いいえ
ファイルシステムの自動配置	ディスク上にあるファイルシステムを自動的に配置するために自動配置を使用しますか。 「はい」の場合は、どのファイルシステムを自動配置しますか。 例: /、/opt、/var 「いいえ」の場合は、手作業でファイルシステムを配置する必要があります。	はい/いいえ



表 6-1 インストール用ワークシート (続き)

インストールに必要な情報	説明/例	答を記入する
リモートファイルシステムのマウント* (Solaris suninstall プログラムの場合のみ利用可能)	このシステムからほかのファイルシステムにあるソフトウェアにアクセスする必要があります。 必要な場合、リモートファイルシステムに関する次の情報を用意します。 サーバー: プロファイルサーバーの IP アドレス: リモートファイルシステム: ローカルマウントポイント:	はい/いいえ

## アップグレード用のチェックリスト

Solaris オペレーティング環境をアップグレードするための準備では、次のチェックリストを使用します。

- tip (1) 接続を使用してシステムの Solaris ソフトウェアをアップグレードする場合は、ウィンドウ表示が少なくとも横 80 桁、縦 24 行あることを確認します。  
tip ウィンドウの現在の大きさを調べるには、stty コマンドを使用します。詳細については、stty (1) のマニュアルページを参照してください。
- 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。
- Solaris INSTALLATION CD に入っている Solaris Web Start プログラムを使用する場合は、ディスクに 512M バイトのスライスが存在することを確認します。詳細 (x86 システムの要件など) は、表 5-2 を参照してください。DVD またはインストールイメージからインストールを実行する場合は、これらの要件を無視できません。
- Solaris Live Upgrade を使用する場合は、新しいブート環境を作成してアップグレードするためのリソース要件を決定します。詳細は、第 34 章を参照してください。
- Solaris Live Upgrade と Solaris ボリュームマネージャを使用する場合は、ルートのみラーリングをオフにします。詳細は、403 ページの「ファイルシステムのスライスを選択するための指針」を参照してください。
- ディスク容量が十分あることを確認します。詳細は、第 5 章を参照してください。
- SPARC: DVD メディアを使用している場合、ok プロンプトからブートするように求められたときには、常に次のコマンドを入力してください。 **boot cdrom**

- 『Solaris 9 4/04 ご使用にあたって』やベンダーのリリースノートを参照して、使用するソフトウェアが新しい Solaris リリースでもサポートされていることを確認します。
- 『Solaris 9 Sun ハードウェアマニュアル』を参照して、使用するハードウェアがサポートされていることを確認します。
- システムに添付されている資料を参照して、使用するシステムやデバイスが Solaris リリースでサポートされていることを確認します。
- ベンダーやサードパーティのソフトウェア資料を参照して、アップグレードに関するその他の情報を入手します。
- 使用可能なパッチで必要なパッチがあるかを確認します。最新のパッチリストは <http://sunsolve.sun.com> から入手できます。
- Prestoserve ソフトウェアがシステムに存在するかを確認します。init 0 コマンドを使ってシステムをシャットダウンしてからアップグレードプロセスを開始すると、データが失われることがあります。シャットダウンについての説明は、Prestoserve の資料を参照してください。
- x86: Linux オペレーティングシステムを使用している場合は、Solaris fdisk パーティションと Linux swap パーティションの識別子が同じ (0x82) になります。この問題を解決するには、次のいずれかを行う必要があります。
  - メモリーが十分にある場合は Linux swap パーティションをまったく使用しない。
  - Linux swap パーティションを別のドライブに置く。
  - 保存したい Linux データを記憶媒体にバックアップし、Solaris オペレーティング環境をインストールした後に、Linux をインストールし直す。




---

注意 - Solaris オペレーティング環境のインストール後に Linux をインストールする場合、Linux swap パーティション (実際には、Solaris fdisk パーティション) を swap ファイルとしてフォーマットするかどうかを Linux インストールプログラムから尋ねられたら、フォーマットしないを選択してください。

---

## アップグレード用のワークシート

Solaris オペレーティング環境のアップグレードに必要な情報を収集する際には、次のワークシートを使用します。ただし、ワークシートに記載されているすべての情報を収集する必要はありません。使用するシステムに関連する情報だけを収集してください。アップグレードをネットワークを使用して行う場合は、インストールプログラムが現在のシステム構成から情報を取得します。

ホスト名や IP アドレスのような、システムの基本的な識別情報は変更できません。インストールプログラムによってシステムの基本的な識別情報を入力するように求められる場合がありますが、元の値を入力する必要があります。Solaris Web Start プログラムを使用してアップグレードする場合は、そのような識別情報の値をどれか変更しようとするするとアップグレードは失敗します。

---

注- ネットワークに接続されないスタンドアロンシステムをインストールする場合は、\* のついた項目については無視してください。

---

表 6-2 アップグレード用ワークシート

アップグレードに必要な情報	説明/例	答を記入する
ネットワーク	このシステムはネットワークに接続されていますか。	はい/いいえ
DHCP *	このシステムでは、DHCP (Dynamic Host Configuration Protocol) を使ってネットワークインタフェースを構成しますか。	はい/いいえ
ホスト名	このシステムのホスト名	
IP アドレス *	DHCP を使用しない場合は、このシステムの IP アドレスを指定します。 例: 129.200.9.1	
サブネット *	DHCP を使用しない場合、このシステムはサブネットの一部ですか。 「はい」の場合は、サブネットのネットマスクを指定します。 例: 255.255.0.0	はい/いいえ
IPv6 *	このマシンで IPv6 を使用可能にしますか。	はい/いいえ
Kerberos *	このマシンに Kerberos セキュリティを構成しますか。 「はい」の場合は、次の情報を収集します。 デフォルトのレルム: 管理サーバー: 一次 KDC: (省略可能) 追加 KDC:	はい/いいえ
ネームサービス *	このシステムではどのネームサービスを使用しますか。	NIS+/NIS/DNS/LDAP/使用しない

表 6-2 アップグレード用ワークシート (続き)

アップグレードに必要な情報	説明/例	答を記入する
ドメイン名 *	このシステムでネームサービスを使用する場合は、システムが属するドメインの名前を指定します。	
NIS+ および NIS *	<p>ネームサーバーを指定しますか、それともインストールプログラムにネームサーバーの検索を任せますか。</p> <p>ネームサーバーを指定する場合は、次の情報を指定します。</p> <p style="text-align: right;">サーバーのホスト名:</p> <p style="text-align: right;">サーバーの IP アドレス:</p>	指定/検索
DNS *	<p>DNS サーバーの IP アドレスを指定します。DNS サーバーの IP アドレスを少なくとも1つ、最大3つまで指定します。</p> <p style="text-align: right;">サーバーの IP アドレス:</p> <p>DNS 検索を行うときに検索するドメインのリストを入力できます。</p> <p style="text-align: right;">検索ドメイン:</p>	
LDAP *	<p>LDAP プロファイルに関する次の情報を指定します。</p> <p style="text-align: right;">プロファイル名:</p> <p style="text-align: right;">プロファイルサーバーのホスト名:</p> <p style="text-align: right;">プロファイルサーバーの IP アドレス:</p> <p>LDAP プロファイルでプロキシ資格レベルを指定した場合、この情報を収集します。</p> <p style="text-align: right;">プロキシバインドの識別名:</p> <p style="text-align: right;">プロキシバインドのパスワード:</p>	

表 6-2 アップグレード用ワークシート (続き)

アップグレードに必要な情報	説明/例	答を記入する
デフォルトルート*	<p>デフォルトルート IP アドレスを指定しますか、それとも Solaris Web Start インストールプログラムまたは Solaris suninstall プログラムに IP アドレスの検索を任せますか。</p> <p>デフォルトルートは、2つの物理ネットワーク間のトラフィック転送用のブリッジを提供します。IP アドレスは、ネットワーク上の各ホストを識別する一意の番号です。</p> <p>次のうちから選択できます。</p> <ul style="list-style-type: none"> <li>■ IP アドレスを指定できます。指定された IP アドレスを使用して /etc/defaultrouter ファイルが作成されます。システムをリブートすると、指定された IP アドレスがデフォルトルートになります。</li> <li>■ ソフトウェアに IP アドレスを検出させることができます。 <ul style="list-style-type: none"> <li>■ Solaris suninstall プログラムは、システムのリブート時に IP アドレスを検出します。</li> <li>■ Solaris Web Start インストールプログラムを使用して、デフォルトルートを検出できます。ただし、システムは、ICMP ルーター発見プロトコルを使用して自らを通知するルーターの存在するサブネット上になければなりません。</li> <li>■ ルーターが存在しない場合、または今回はソフトウェアに IP アドレスを検出させない場合は、「なし」を選択します。リブート時に、ソフトウェアが自動的に IP アドレスの検出を試みます。</li> </ul> </li> </ul> <p style="text-align: right;">ホスト IP アドレス:</p>	指定/検索/なし
時間帯	デフォルトの時間帯をどのように指定しますか。	地域 GMT との時間差 時間帯ファイル
ロケール	どの地域のサポートをインストールしますか。	
SPARC: 電源管理 (電源管理システムをサポートする SPARC システムの場合のみ)	<p>電源管理システムを使用しますか。</p> <p>注 - 使用するシステムが Energy Star バージョン 3 以降に対応している場合、このプロンプトは表示されません。</p>	はい/いいえ

表 6-2 アップグレード用ワークシート (続き)

アップグレードに必要な情報	説明/例	答を記入する
プロキシサーバーの構成 *  (Solaris Web Start の場合のみ)	インターネットに直接接続できますか、それともインターネットへのアクセスにプロキシサーバーを経由する必要がありますか。  プロキシサーバーを使用する場合は、次の情報を指定します。  プロキシサーバーのホスト名:  ポート:	直接接続/プロキシサーバー
自動的なリブートまたは CD/DVD 取り出し	ソフトウェアをインストールした後に自動的にリブートしますか。  ソフトウェアをインストールした後に CD/DVD を自動的に取り出しますか。	はい/いいえ  はい/いいえ
ディスク容量の再割り当て	ディスク上にあるファイルシステムを自動的に再配置するために自動配置を使用しますか。  「はい」の場合は、どのファイルシステムを自動配置しますか。  例: /、/opt、/var  「いいえ」の場合は、ファイルシステム構成用の情報を指定する必要があります。	はい/いいえ
SPARC: 64 ビット	64 ビットアプリケーションのサポートをインストールしますか。	はい/いいえ

## 第 7 章

---

# システム構成情報の事前設定 (作業)

---

この章では、システム構成情報の事前設定について説明します。事前に設定を行うと、Solaris オペレーティング環境をインストールする際に、システム構成情報の入力を求めるプロンプトが表示されません。この章では、さらに、電源管理システム情報の事前設定についても説明します。この章の内容は次のとおりです。

- 63 ページの「システム構成情報を事前設定することの利点」
- 64 ページの「システム構成情報の事前設定方法」
- 65 ページの「sysidcfg ファイルによる事前設定」
- 72 ページの「ネームサービスによる事前設定」
- 75 ページの「DHCP サービスによるシステム構成情報の事前設定 (作業)」
- 85 ページの「SPARC: 電源管理情報の事前設定」

---

## システム構成情報を事前設定することの利点

どのインストール方法でも、周辺機器、ホスト名、IP (インターネットプロトコル) アドレス、ネームサービスなどのシステム構成情報が必要です。インストールプログラムは、構成情報の入力を求める前に、まず sysidcfg ファイルの情報を調べ、次にネームサービスデータベースの情報を調べます。

Solaris Web Start プログラム、Solaris suninstall プログラム、およびカスタム JumpStart インストールプログラムは、事前設定されたシステム構成情報を検出すると、その情報の入力を求めることはありません。たとえば、複数のシステムに Solaris 9 ソフトウェアをインストールするとき、システムごとに時間帯を入力したくない場合があります。この時間帯を sysidcfg ファイル中またはネームサービスデータベース中に指定できます。Solaris 9 ソフトウェアのインストール時に時間帯の入力を求められなくなります。

## システム構成情報の事前設定方法

システム構成情報を事前設定するには、2つの方法があります。システム構成情報は以下の場所に設定できます。

- `sysidcfg` ファイル (リモートシステム上またはフロッピーディスク)
- 自分のサイトで使用しているネームサービスデータベース

サイトで DHCP を使用している場合は、サイトの DHCP サーバーで一部のシステム情報の事前構成を行うこともできます。DHCP サーバーを使ってシステム情報の事前構成を行う方法の詳細については、75 ページの「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。

次の表を使って、システム構成情報の事前設定に `sysidcfg` ファイルを使用するかネームサービスデータベースを使用するかを決定してください。

表 7-1 システム構成情報を事前設定するための方法

事前設定できるシステム構成情報	<code>sysidcfg</code> ファイルでの事前設定の可否	ネームサービスでの事前設定の可否
ネームサービス	可能	可能
ドメイン名	可能	不可
ネームサーバー	可能	不可
ネットワークインタフェース	可能	不可
ホスト名	可能	可能
	この情報はシステムに固有なため、各システム用に異なる <code>sysidcfg</code> ファイルを作成するよりも、ネームサービスを編集してください。	
IP アドレス	可能	可能
	この情報はシステムに固有なため、各システム用に異なる <code>sysidcfg</code> ファイルを作成するよりも、ネームサービスを編集してください。	
ネットマスク	可能	不可
DHCP	可能	不可
IPv6	可能	不可
デフォルトルーター	可能	不可



表 7-1 システム構成情報を事前設定するための方法 (続き)

事前設定できるシステム構成情報	sysidcfg ファイルでの事前設定の可否	ネームサービスでの事前設定の可否
root パスワード	可能	不可
セキュリティポリシー	可能	不可
インストールプログラムとデスクトップで表示する言語 (ロケール)	可能	NIS または NIS+ の場合、可能 DNS または LDAP の場合、不可
端末タイプ	可能	不可
時間帯	可能	可能
日付と時刻	可能	可能
Web プロキシ	不可	不可
	Solaris Web Start のインストール時にこの情報を構成できますが、sysidcfg ファイルやネームサービスを使用せずに行う必要があります。	
x86: モニタータイプ	可能	不可
x86: キーボード言語、キーボード配置	可能	不可
x86: グラフィックスカード、カラー深度、表示解像度、画面サイズ	可能	不可
x86: ポインティングデバイス、ボタン数、IRQ レベル	可能	不可
SPARC: 電源管理システム (autoshtutdown)	不可	不可
sysidcfg ファイルやネームサービスを通して電源管理システムの事前構成を行うことはできません。詳細は、85 ページの「SPARC: 電源管理情報の事前設定」を参照してください。		

## sysidcfg ファイルによる事前設定

sysidcfg ファイルに一連のキーワードを指定すると、システムを事前設定できます。表 7-2 は、これらのキーワードを示しています。

異なる構成情報を必要とするシステムごとに、固有の `sysidcfg` ファイルを作成する必要があります。すべてのシステムに同じ時間帯を割り当てる場合は、同じ `sysidcfg` ファイルを使用して、一連のシステムに時間帯を事前設定することができます。ただし、これらの各システムに異なる `root` (スーパーユーザー) パスワードを事前設定する場合は、各システムに固有の `sysidcfg` ファイルを作成する必要があります。

`sysidcfg` ファイルは、次のどれかに置くことができます。

- NFS ファイルシステム – `sysidcfg` ファイルを共有 NFS ファイルシステムに置く場合は、ネットワークからインストールできるようにシステムを設定するときに、`add_install_client (1M)` コマンドの `-p` オプションで Solaris 9 ソフトウェアのインストール時に `sysidcfg` ファイルが検索される場所を指定する必要があります。
- UFS フロッピーディスクまたは PCFS フロッピーディスク – `sysidcfg` ファイルをフロッピーディスクのルート (`/`) ディレクトリに置きます。
- HTTP または HTTPS サーバー – WAN ブートインストールを実行する場合は、`sysidcfg` ファイルを Web サーバーのドキュメントルートディレクトリに置きます。

---

注 – カスタム JumpStart インストールを実行していて、フロッピーディスク上の `sysidcfg` ファイルを使用したい場合は、プロファイルフロッピーディスク上に `sysidcfg` ファイルを置く必要があります。プロファイルフロッピーディスクを作成するには、265 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」を参照してください。

---

1 つのディレクトリまたはフロッピーディスクには、1 つの `sysidcfg` ファイルだけを入れることができます。複数の `sysidcfg` ファイルを作成する場合は、各ファイルを異なるディレクトリまたは異なるフロッピーディスクに置く必要があります。

## sysidcfg ファイルの構文規則

`sysidcfg` ファイルで使用するキーワードには、非依存型と依存型の 2 種類があります。依存型キーワードは、非依存型キーワード内でのみ固有であることが保証されています。依存型キーワードは、対応する非依存型キーワードによって識別される場合にのみ存在します。

次の例では、`name_service` が非依存型キーワードであり、`domain_name` と `name_server` が依存型キーワードです。

```
name_service=NIS {domain_name=marquee.central.sun.com
name_server=connor(129.152.112.3)}
```

構文規則	例
非依存型キーワードは任意の順序で指定可能	pointer=MS-S display=ati {size=15-inch}
キーワードは大文字と小文字のどちらでもよい	TIMEZONE=US/Central terminal=PC Console
関連する非依存型キーワードを結合するには、すべての依存型キーワードを中括弧 {} で囲む	name_service=NIS {domain_name=marquee.central.sun.com name_server=connor(129.152.112.3)}
値は単一引用符 (') または二重引用符 (") で囲んで指定可能	network_interface='none'
キーワードは 1 回だけ指定可能。キーワードを複数回指定した場合は最初のキーワードだけが有効	network_interface=none network_interface=le0

## sysidcfg ファイルキーワード

表 7-2 に、sysidcfg ファイルで使用できるキーワードを示します。

表 7-2 sysidcfg キーワード

構成情報	キーワード	使用例または設定値
ネームサービス、ドメイン名、ネームサーバー	name_service=NIS, NIS+, DNS, LDAP, NONE	
	NIS と NIS+ 用オプション: {domain_name=domain_name name_server=hostname(ip_address)}	name_service=NIS {domain_name=west.arp.com name_server=timber(129.221.2.1)}
		name_service=NIS+ {domain_name=west.arp.com. name_server=timber(129.221.2.1)}

表 7-2 sysidcfg キーワード (続き)

構成情報	キーワード	使用例または設定値
	DNS 用オプション: ション:{domain_name=domain_name name_server=ip_address,ip_address, ip_address (最大 3 個) search=domain_name, domain_name,domain_name, domain_name,domain_name,domain_name (最大 6 個、合計の長さが 250 文字以下)}	<pre>name_service=DNS {domain_name=west.arp.com name_server=10.0.1.10,10.0.1.20 search=arp.com,east.arp.com}</pre> <p>注 - name_service は 1 つの値だけを選択します。必要に応じて、domain_name と name_server キーワードのどちらか一方または両方を設定するか、あるいはどちらも含めません。どちらのキーワードも使用しない場合、中括弧 {} は省略します。</p>
	LDAP 用オプション: {domain_name=domain_name profile=profile_name profile_server=ip_address proxy_dn="proxy_bind_dn" proxy_password=password}	<pre>name_service=LDAP {domain_name=west.arp.com profile=default profile_server=129.221.2.1 proxy_dn="cn=proxyagent,ou=profile, dc=west,dc=arp,dc=com" proxy_password=password}</pre> <p>注 - proxy_dn および proxy_password キーワードは省略可能です。proxy_dn 値は、二重引用符で囲む必要があります。</p>
ネットワークインタフェース、ホスト名、IP アドレス、ネットマスク、DHCP、IPv6	network_interface=NONE、PRIMARY または value	<ul style="list-style-type: none"> <li>■ NONE は、ネットワーク接続をオフにします。</li> <li>■ PRIMARY は、システム上に存在するインタフェースの中で最初に稼動する非ループバックインタフェースです。順序は ifconfig の場合と同様に、稼動しているインタフェースが存在しない場合には、最初の非ループバックインタフェースが使用されることとなります。非ループバックインタフェースが見つからない場合は、システムは NON-NETWORKED に設定されます。</li> <li>■ value には、le0 や hme0 などのインタフェースを指定します。</li> </ul>
	DHCP を使用する場合、次のように指定する。{dhcp protocol_ipv6=yes_or_no}	<pre>network_interface=primary {dhcp protocol_ipv6=yes}</pre>

表 7-2 sysidcfg キーワード (続き)

構成情報	キーワード	使用例または設定値
	<p>DHCP を使用しない場合、次のように指定する。 {hostname=host_name default_route=ip_address または NONE ip_address=ip_address netmask=netmask protocol_ipv6=yes_or_no}</p> <p>注 - DHCP を使用しない場合、 protocol_ipv6=yes と default_route は指定する必要がありません。ただし、次の点に注意してください。</p> <ul style="list-style-type: none"> <li>■ JumpStart インストールの場合、この時点で protocol_ipv6 を指定する必要があります。指定がないと、後で対話方式で入力进行要求されます。</li> <li>■ default_route が指定されておらず、ICMP ルーター発見プロトコルを使用して検出できない場合、ルート (route) を選択するよう要求されます。</li> </ul> <p>WAN ブートインストールで sysidcfg ファイルを使用する場合は、 protocol_ipv6=no を設定する必要があります。</p>	<pre>network_interface=le0 {hostname=feron default_route=129.146.88.1 ip_address=129.146.88.210 netmask=255.255.0.0 protocol_ipv6=no}</pre> <p>注 - network_interface は、1つの値だけを選択します。必要に応じて、hostname、ip_address、netmask キーワードのどれかを組み合わせて設定するか、あるいはどれも設定しません。どのキーワードも使用しない場合、中括弧 {} は省略します。</p>
root パスワード	root_password=root_password	/etc/shadow にある暗号化された文字列
セキュリティポリシー	<p>security_policy=kerberos, NONE</p> <p>Kerberos 用オプション: {default_realm=FQDN admin_server=FQDN kdc=FQDN1, FQDN2, FQDN3}</p> <p>(FQDN は完全修飾ドメイン名です)</p>	<pre>security_policy=kerberos {default_realm=Yoursite.COM admin_server=krbadmin.Yoursite.COM kdc=kdcl.Yoursite.COM, kdc2.Yoursite.COM}</pre> <p>注 - 最大3つの鍵発行センター (KDC) をリストできます。少なくとも1つは必須です。</p>
インストールプログラムとデスクトップで表示する言語	system_locale=locale	有効なロケール値が、/usr/lib/locale ディレクトリまたは付録 D にあります。
端末タイプ	terminal=terminal_type	有効な端末値が /usr/share/lib/terminfo ディレクトリのサブディレクトリにあります。

表 7-2 sysidcfg キーワード (続き)

構成情報	キーワード	使用例または設定値
時間帯	<code>timezone=timezone</code>	有効な時間帯値が <code>/usr/share/lib/zoneinfo</code> ディレクトリのサブディレクトリとファイルにあります。時間帯値は <code>/usr/share/lib/zoneinfo</code> ディレクトリからの相対パス名です。たとえば、米国の山岳部標準時の時間値は <code>US/Mountain</code> です。日本の時間帯値は <code>Japan</code> です。また、有効な Olson 時間帯も指定できます。
日付と時刻	<code>timeserver=localhost, hostname, ip_addr</code>	<code>localhost</code> を指定した場合は、そのシステムの時刻が正しいものと仮定し、時刻が設定されます。ネームサービスを実行してなくて、あるシステムの <code>hostname</code> または <code>ip_addr</code> を指定した場合は、そのシステムの時刻を使って時刻が設定されます。
x86: モニタータイプ	<code>monitor=monitor_type</code>	インストールしたいシステム上で <code>kdmconfig -d filename</code> を実行すると、その出力が <code>sysidcfg</code> ファイルに追加されます。
x86: キーボード言語、キーボード配置	<code>keyboard=keyboard_language {layout=value}</code>	インストールしたいシステム上で <code>kdmconfig -d filename</code> を実行すると、その出力が <code>sysidcfg</code> ファイルに追加されます。
x86: グラフィックスカード、画面サイズ、カラー深度、表示解像度	<code>display=graphics_card {size=screen_size depth=color_depth resolution=screen_resolution}</code>	インストールしたいシステム上で <code>kdmconfig -d filename</code> を実行すると、その出力が <code>sysidcfg</code> ファイルに追加されます。
x86: ポインティングデバイス、ボタン数、IRQ レベル	<code>pointer=pointing_device {nbuttons=number_buttons irq=value}</code>	インストールしたいシステム上で <code>kdmconfig -d filename</code> を実行すると、その出力が <code>sysidcfg</code> ファイルに追加されます。

## ▼ sysidcfg 構成ファイルを作成する方法

1. テキストエディタで、**sysidcfg** という名前のファイルを作成します。
2. 必要な **sysidcfg** のキーワードを入力します。
3. **sysidcfg** ファイルを保存します。

---

注 - 複数の `sysidcfg` ファイルを作成する場合は、それぞれのファイルを別々のディレクトリまたは別々のフロッピーディスクに保存する必要があります。

---

4. クライアントから次のファイルシステムまたはディレクトリを介して **sysidcfg** ファイルにアクセスできるようにします。

- 共有 NFS ファイルシステム。ネットワークからインストールできるようにシステムを設定するには、`add_install_client (1M)` に `-p` オプションを指定します。
- UFS フロッピーディスクまたは PCFS フロッピーディスクのルート(/) ディレクトリ

## SPARC: sysidcfg ファイルの例

一連の SPARC システムのための `sysidcfg` ファイルの例を次に示します。これらのシステムのホスト名、IP アドレス、およびネットマスクは、ネームサービスを編集することにより、すでに事前設定されています。このファイルにはすべてのシステム構成情報が事前設定されているので、カスタム JumpStart プロファイルを使ってカスタム JumpStart インストールを実行できます。

```
system_locale=en_US
timezone=US/Central
terminal=sun-cmd
timeserver=localhost
name_service=NIS {domain_name=marquee.central.sun.com
                  name_server=connor(129.152.112.3)}
root_password=m4QPOWNY
network_interface=le0 {hostname=feron
                       default_route=129.146.88.1
                       ip_address=129.146.88.210
                       netmask=255.255.0.0
                       protocol_ipv6=no}
security_policy=kerberos {default_realm=Yoursite.COM
                          admin_server=krbadmin.Yoursite.COM
                          kdc=kdc1.Yoursite.COM,
                          kdc2.Yoursite.COM}
```

## x86: sysidcfg ファイルの例

一連の x86 システムで、キーボード、グラフィックスカード、ポインティングデバイスがすべて同じ場合の `sysidcfg` ファイルの例を次に示します。これらのデバイス情報 (`keyboard`、`display`、`pointer`) は、`kdmconfig (1M) -d` コマンドを実行して取得したものです。この例では、Solaris インストールプログラムで使用される言語 (`system_locale`) を選択するプロンプトがインストール前に表示されます。

```
keyboard=ATKBD {layout=US-English}
display=ati {size=15-inch}
pointer=MS-S
timezone=US/Central
timeserver=connor
terminal=ibm-pc
name_service=NIS {domain_name=marquee.central.sun.com
                  name_server=connor(129.152.112.3)}
root_password=URFUni9
```

## ネームサービスによる事前設定

次の表は、システム構成情報を事前設定するために編集および入力を行う必要があるネームサービスデータベースの概要を示したものです。

事前設定するシステム情報	ネームサービスデータベース
ホスト名と IP アドレス	hosts
日付と時刻	hosts インストール対象のシステムに、日付と時刻を提供するホスト名に続けて timehost という別名を記述します。
時間帯	timezone
ネットマスク	netmasks

DNS や LDAP のネームサービスでは、システムのロケールを事前設定することはできません。NIS や NIS+ のネームサービスを使用する場合は、そのネームサービスの手順に従ってシステムのロケールを事前設定してください。

- 72 ページの「NIS を使ってロケールを事前設定する方法」
- 74 ページの「NIS+ を使ってロケールを事前設定する方法」

### ▼ NIS を使ってロケールを事前設定する方法

1. ネームサーバー上でスーパーユーザーになります。
2. `/var/yp/Makefile` ファイルを編集して、ロケールマップを追加します。
  - a. エントリの後に、以下を追加します。

```
locale.time: $(DIR)/locale
-@if [ -f $(DIR)/locale ]; then \
    sed -e "/^#/d" -e s/#.*$$// $(DIR)/locale \
    | awk '{for (i = 2; i<=NF; i++) print $$i, $$0}' \
    | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/locale.byname; \
    touch locale.time; \
    echo "updated locale"; \
    if [ ! $(NOPUSH) ]; then \
        $(YPPUSH) locale.byname; \
        echo "pushed locale"; \
    else \
        : ; \
    fi \
else \
    echo "couldn't find $(DIR)/locale"; \
```



fi

- b. 文字列 **all:** を検索し、変数リストの最後に **locale** という語を挿入します。

```
all: passwd group hosts ethers networks rpc services protocols \  
    netgroup bootparams aliases publickey netid netmasks c2secure \  
    timezone auto.master auto.home locale
```

- c. ファイルの下の方にある同じようなエントリの後に、文字列 **locale:**  
**locale.time** を追加します。

```
passwd: passwd.time  
group: group.time  
hosts: hosts.time  
ethers: ethers.time  
networks: networks.time  
rpc: rpc.time  
services: services.time  
protocols: protocols.time  
netgroup: netgroup.time  
bootparams: bootparams.time  
aliases: aliases.time  
publickey: publickey.time  
netid: netid.time  
passwd.adjunct: passwd.adjunct.time  
group.adjunct: group.adjunct.time  
netmasks: netmasks.time  
timezone: timezone.time  
auto.master: auto.master.time  
auto.home: auto.home.time  
locale: locale.time
```

- d. ファイルを保存します。

3. **/etc/locale** というファイルを作成し、ドメインまたは特定のシステムに対して1つのエントリを作成します。

*locale domain\_name*

または

*locale system\_name*

---

注 - 使用できるロケール値のリストについては、付録 D を参照。

---

たとえば次の行は、worknet.com ドメインに対してデフォルト言語として日本語を指定しています。

```
ja worknet.com
```

たとえば次の行は、charlie というシステムに対してデフォルトロケールとして ja\_JP.UTF-8 ロケールを指定しています。

```
ja_JP.UTF-8
```

---

注 - ロケールは、Solaris DVD または Solaris SOFTWARE 1 of 2 CD に入っています。

---

4. マップを作成します。

```
# cd /var/yp; make
```

これでドメインまたは locale マップで個別に指定したシステムは、デフォルトのロケールを使用するように設定されました。ここで指定したデフォルトのロケールは、インストール時に使用されるとともに、システムのリポート後のデスクトップでも使用されます。

## ▼ NIS+ を使ってロケールを事前設定する方法

この手順は、NIS+ ドメインが設定されていると仮定しています。NIS+ ドメインの設定については、『Solaris のシステム管理 (ネーミングとディレクトリサービス: DNS、NIS、LDAP 編)』を参照してください。

1. ネームサーバーに、スーパーユーザーまたは **NIS+ admin** グループのユーザーとしてログインします。

2. **locale** テーブルを作成します。

```
# nistbladm -D access=og=rmcd,nw=r -c locale_tbl name=SI,nogw=
locale=,nogw= comment=,nogw= locale.org_dir.'nisdefaults -d'
```

3. **locale** に必要なエントリを追加します。

```
# nistbladm -a name=name locale=locale comment=comment
locale.org_dir.'nisdefaults -d'
```

*name* デフォルトのロケールを事前設定したいドメイン名または特定のシステム名。

*locale* システムにインストールし、システムのリポート後にデスクトップ上で使用するロケール。使用できるロケール値のリストについては、付録 D を参照。

*comment* コメントフィールド。複数の単語を使ったコメントは、前後を二重引用符で囲むこと。

---

注 - ロケールは、Solaris DVD または Solaris SOFTWARE 1 of 2 CD に入っています。

---

これでドメインまたは locale テーブルで個別に指定したシステムは、デフォルトロケールを使用するように設定されました。ここで指定したデフォルトロケール

は、インストール時に使用されるとともに、システムのリブート後のデスクトップでも使用されます。

---

## DHCP サービスによるシステム構成情報の事前設定 (作業)

DHCP (Dynamic Host Configuration Protocol) を使用すると、TCP/IP ネットワーク内のホストシステムを、ブート時にネットワークに合わせて自動的に構成できます。DHCP では、クライアント/サーバーメカニズムが使用されます。サーバーは、クライアントの構成情報を格納、管理し、クライアントの要求に応じてその構成情報を提供します。構成情報には、クライアントの IP アドレスと、クライアントが使用可能なネットワークサービス情報が含まれます。

DHCP の主な利点は、リースを通して IP アドレス割り当てを管理できることです。リースを使用すると、使用されていない IP アドレスを回収し、他のクライアントに割り当て直すことができます。この機能によって、1つのサイトで使用する IP アドレスプールは、すべてのクライアントに常時アドレスを割り当てた場合に比べて、小さくなります。

DHCP を使用してネットワーク上で Solaris オペレーティング環境をインストールできるのは、一定の条件のクライアントシステムに限られます。この機能を使用できるのは、Sun Enterprise Ultra システムと Solaris オペレーティング環境を実行するためのハードウェア要件を満たしている x86 システムだけです。

次の作業マップに、クライアントが DHCP を使用してインストールパラメータを取得するために必要な作業を示します。

表 7-3 作業マップ: DHCP サービスによるシステム構成情報の事前設定

作業	説明	参照先
インストールサーバーの構成	Solaris サーバーを設定して、ネットワークから Solaris オペレーティング環境をインストールする必要があるクライアントをサポートする	第 13 章

表 7-3 作業マップ: DHCP サービスによるシステム構成情報の事前設定 (続き)

作業	説明	参照先
DHCP を使用してネットワーク経由で Solaris をインストールできるようにクライアントシステムを構成する	<code>add_install_client -d</code> を使用して、特定のクラスあるいはマシンタイプに属するクライアントや特定のクライアント ID に対して、DHCP ネットワークインストールのサポートを追加する	Solaris DVD を使用する場合 128 ページの「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を追加する」  Solaris CD を使用する場合 161 ページの「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を追加する」  <code>add_install_client (1M)</code>
DHCP サービスを使用するためにネットワークを準備する	DHCP サーバーをどのように構成するかを決定する	『Solaris のシステム管理 (IP サービス)』の「DHCP サービスの使用計画 (手順)」
DHCP サーバーを構成する	DHCP マネージャを使用して DHCP サーバーを構成する	『Solaris のシステム管理 (IP サービス)』の「DHCP サービスの構成 (手順)」
インストールパラメータ用の DHCP オプションとそのオプションを含むマクロの作成	DHCP マネージャまたは <code>dhtadm</code> を使用して、DHCP サーバーがインストール情報をクライアントに渡すときに使用できる、新しいベンダーオプションとマクロを作成する	76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」

## Solaris インストールパラメータ用の DHCP オプションとマクロの作成

インストールサーバー上で `add_install_client -d` スクリプトを使用してクライアントを追加するとき、そのスクリプトは DHCP 構成情報を標準出力にレポートします。この情報は、ネットワークインストール情報をクライアントに伝えるために必要なオプションとマクロを作成する際に使用できます。

ネットワークから DHCP クライアントに対してインストールを行うには、ベンダーカテゴリオプションを作成して、Solaris オペレーティング環境をインストールするために必要な情報を伝える必要があります。表 7-4 に、作成する必要があるオプションと、その作成に必要なプロパティを示します。

DHCP サービス内のマクロをカスタマイズして、次の種類のインストールを実行できます。

- 特定のクラスのインストール - DHCP サービスに、特定のクラスのすべてのクライアントのネットワークインストールを行うように指示できます。たとえば、ネットワーク上のすべての Sun Blade システム上で同じインストールを実行するような DHCP マクロを定義できます。add\_install\_client -d コマンドの出力を使用して、特定のクラスのインストールを設定します。
- 特定のクライアントのインストール - DHCP サービスに、特定の Ethernet アドレスのクライアントのネットワークインストールを行うように指示できます。たとえば、Ethernet アドレス 00:07:e9:04:4a:bf のクライアント上でインストールを行うような DHCP マクロを定義できます。add\_install\_client -d -e ethernet\_address コマンドの出力を使用して、特定のクライアントのインストールを設定します。

ネットワークからインストールするようにクライアントを設定する方法については、次の手順を参照してください。

- DVD メディアを使用するネットワークインストールについては、128 ページの「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を追加する」を参照してください。
- CD メディアを使用するネットワークインストールについては、161 ページの「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を追加する」を参照してください。

次の表のベンダークライアントクラスは、各オプションを使用できるクライアントのクラスを表します。ここに示されているベンダークライアントクラスは単なる例です。ネットワークからインストールする必要がある実際のクライアントについて、クライアントクラスを指定する必要があります。クライアントのベンダークライアントクラスを調べる方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP オプションを使用した作業 (作業マップ)」を参照してください。

DHCP オプションの詳細については、『Solaris のシステム管理 (IP サービス)』の「DHCP のオプション」を参照してください。

表 7-4 Solaris クライアント用にベンダーカテゴリオプションを作成するための値

名前	コード	データ型	データの単位 数 (Granularity)	最大値	ベンダークライアントクラス	説明
次のベンダーカテゴリオプションは、クライアントへの Solaris のインストールを DHCP サーバーでサポートするために必須のものです。これらのオプションは、Solaris クライアントの起動スクリプトで使用されます。						
SrootIP4	2	IP アドレス	1	1	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	ルートサーバーの IP アドレス
SrootNM	3	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	ルートサーバーのホスト名

表 7-4 Solaris クライアント用にベンダーカテゴリオプションを作成するための値 (続き)

名前	コード	データ型	データの単位 数 (Granularity)	最大値	ベンダークライアントクラス	説明
SrootPTH	4	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	ルートサーバーにあるクライアントのルートディレクトリへのパス
SinstIP4	10	IP アドレス	1	1	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	JumpStart インストールサーバーの IP アドレス
SinstNM	11	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	インストールサーバーのホスト名
SinstPTH	12	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	インストールサーバーにあるインストールイメージへのパス
次のオプションは、クライアントの起動スクリプトで使用できますが、必須ではありません。						
SrootOpt	1	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	クライアントのルートファイルシステム用の NFS マウントオプション
SbootFIL	7	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	クライアントのブートファイルへのパス
SbootRS	9	数値	2	1	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	カーネルを読み込む際にスタンドアロンのブートプログラムが使用する NFS 読み込みサイズ
SsysidCF	13	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	<i>server:/path</i> という形式での、 <i>sysidcfg</i> ファイルへのパス
SjumpsCF	14	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	<i>server:/path</i> という形式での、JumpStart 構成ファイルへのパス

表 7-4 Solaris クライアント用にベンダーカテゴリオプションを作成するための値 (続き)

名前	コード	データ型	データの単位 数 (Granularity)	最大値	ベンダークライアントクラス	説明
SbootURI	16	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	<p>スタンドアロンのブートファイルへのパスまたは WAN ブートファイルへのパス。スタンドアロンのブートファイルの場合は、次の形式を使用する</p> <p><code>tftp://inetboot.sun4u</code></p> <p>WAN ブートファイルの場合は、次の形式を使用する</p> <p><code>http://host.domain/path-to-file</code></p> <p>このオプションを使用すると、BootFile と siaddr の設定を無効にし、スタンドアロンのブートファイルを検出させることができる。サポートされているプロトコルは、tftp (inetboot) および http (wanboot)。たとえば、次の形式を使用する</p> <p><code>tftp://inetboot.sun4u</code></p>
SHTTPproxy	17	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	<p>ネットワーク上で使用されているプロキシサーバーの IP アドレスとポート番号。このオプションが必要なのは、クライアントが WAN を介してブートされる場合で、ローカルネットワークでプロキシサーバーが使用されているときだけです。たとえば、次の形式を使用する</p> <p><code>198.162.10.5:8080</code></p>

次のオプションは、Solaris クライアントの起動スクリプトで現在は使用されていません。起動スクリプトを編集する場合だけ、これらのオプションを使用できます。

表 7-4 Solaris クライアント用にベンダーカテゴリオプションを作成するための値 (続き)

名前	コード	データ型	データの単位 数 (Granularity)	最大値	ベンダークライアントクラス	説明
SswapIP4	5	IP アドレス	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	スワップサーバーの IP アドレス
SswapPTH	6	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	スワップサーバーにあるクライアントのスワップファイルへのパス
Stz	8	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	クライアントのタイムゾーン
Sterm	15	ASCII テキスト	1	0	SUNW.Sun-Blade-1000、SUNW.Sun-Fire-880、SUNW.i86pc	端末タイプ

オプションがすでに作成されていれば、これらのオプションを含んだマクロを作成することができます。次に、クライアントに対する Solaris のインストールをサポートするために作成できるマクロの例を示します。

表 7-5 ネットワークインストールクライアントをサポートするマクロの例

マクロ名	含まれるオプションとマクロ
Solaris	SrootIP4、SrootNM、SinstIP4、SinstNM
sparc	SrootPTH、SinstPTH
sun4u	Solaris マクロと sparc マクロ
i86pc	Solaris マクロ、SrootPTH、SinstPTH、SbootFIL
SUNW.i86pc	i86pc マクロ
SUNW.Sun-Blade-1000	sun4u マクロ、SbootFIL
SUNW.Sun-Fire-880	sun4u マクロ、SbootFIL
xxx.xxx.xxx.xxx ネットワークアドレスマクロ	BootSrvA オプションは既存のネットワークアドレスマクロに追加できる。BootSrvA の値は tftboot サーバーを示す必要がある



上記の表に示されているマクロ名は、ネットワークからインストールする必要のあるクライアントのベンダークライアントクラスと一致します。これらの名前は、ネットワーク上にあるクライアントの例です。クライアントのベンダークライアントクラスを調べる方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP オプションを使用した作業 (作業マップ)」を参照してください。

これらのオプションとマクロは、次の方法で作成できます。

- dhtadm コマンドを使ってオプションとマクロを作成するスクリプトを記述する。これらのオプションとマクロを作成するスクリプトの記述方法については、81 ページの「dhtadm を使用してオプションとマクロを作成するスクリプトの作成」を参照
- オプションとマクロを DHCP マネージャで作成する。オプションとマクロを DHCP マネージャで作成する方法については、83 ページの「DHCP マネージャを使用したインストールオプションとマクロの作成」を参照

## dhtadm を使用してオプションとマクロを作成するスクリプトの作成

例 7-1 の例を変更して Korn シェルスクリプトを作成し、表 7-4 に示されているすべてのオプションと、いくつかの便利なマクロを作成できます。引用符に囲まれたすべての IP アドレスと値を、実際のネットワークの IP アドレス、サーバー名、およびパスに変更してください。また、Vendor= キーを編集して、使用するクライアントのクラスを示す必要もあります。add\_install\_client -d の情報を使って、スクリプトを変更するために必要なデータを取得します。

### 例 7-1 ネットワークインストールをサポートするスクリプトの例

```
# Solaris ベンダー固有のオプションを読み込みます。
# Sun-Blade-1000、Sun-Fire-880、および i86 プラットフォームのサポートを開始します。
# -A を -M に変更すると、値を追加するのではなく、現在の値に上書きします。
dhtadm -A -s SrootOpt -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,1,ASCII,1,0'
dhtadm -A -s SrootIP4 -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,2,IP,1,1'
dhtadm -A -s SrootNM -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,3,ASCII,1,0'
dhtadm -A -s SrootPTH -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,4,ASCII,1,0'
dhtadm -A -s SswapIP4 -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,5,IP,1,0'
dhtadm -A -s SswapPTH -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,6,ASCII,1,0'
dhtadm -A -s SbootFIL -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,7,ASCII,1,0'
dhtadm -A -s Stz -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,8,ASCII,1,0'
dhtadm -A -s SbootRS -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,9,NUMBER,2,1'
dhtadm -A -s SinstIP4 -d \
```

例 7-1 ネットワークインストーンをサポートするスクリプトの例 (続き)

```
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,10,IP,1,1'
dhtadm -A -s SinstNM -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,11,ASCII,1,0'
dhtadm -A -s SinstPTH -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,12,ASCII,1,0'
dhtadm -A -s SsysidCF -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,13,ASCII,1,0'
dhtadm -A -s SjumpsCF -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,14,ASCII,1,0'
dhtadm -A -s Sterm -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,15,ASCII,1,0'
dhtadm -A -s SbootURI -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,16,ASCII,1,0'
dhtadm -A -s SHTTPproxy -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,17,ASCII,1,0'
# 役立つマクロ定義をいくつか読み込みます。
# Solaris 汎用のオプションをすべて、この Solaris というマクロに定義します。
dhtadm -A -m Solaris -d \
':SrootIP4=10.21.0.2:SrootNM="blue2":SinstIP4=10.21.0.2:SinstNM="red5":'
# sparc プラットフォーム固有のオプションをすべて、この sparc というマクロに定義します。
dhtadm -A -m sparc -d \
':SrootPTH="/export/sparc/root":SinstPTH="/export/sparc/install":'
# sun4u アーキテクチャ固有のオプションをすべて、この sun4u というマクロに定義します。
# (Solaris マクロと sparc マクロも含む)
dhtadm -A -m sun4u -d ':Include=Solaris:Include=sparc:'
# Solaris on IA32 プラットフォーム固有のパラメータを、この i86pc というマクロに定義します。
dhtadm -A -m i86pc -d \
':Include=Solaris:SrootPTH="/export/i86pc/root":SinstPTH="/export/i86pc/install"\
:SbootFIL="/platform/i86pc/kernel/unix":'
# Solaris on IA32 マシンは、"SUNW.i86pc" クラスによって識別されます。
# このクラスのメンバーとして識別されるクライアントはすべて、
# SUNW.i86pc というマクロ内のパラメータを認識できます。このマクロには、i86pc マクロが含まれています。
dhtadm -A -m SUNW.i86pc -d ':Include=i86pc:'
# Sun-Blade-1000 プラットフォームは、
# "SUNW.Sun-Blade-1000" クラスの一部として識別されます。
# このクラスのメンバーとして識別されるクライアントはすべて、
# これらのパラメータを認識できます。
dhtadm -A -m SUNW.Sun-Blade-1000 -d \
':SbootFIL="/platform/sun4u/kernel/sparcv9/unix":\
Include=sun4u:'
# Sun-Fire-880 プラットフォームは、"SUNW.Sun-Fire-880" クラスの一部として識別されます。
# このクラスのメンバーとして識別されるクライアントはすべて、これらのパラメータを認識できます。
dhtadm -A -m SUNW.Sun-Fire-880 -d \
':SbootFIL="/platform/sun4u/kernel/sparcv9/unix":Include=sun4u:'
# DHCP サーバーで管理されるトポロジーについて、各ネットワークマクロにブートサーバー IP を追加します。
# ブートサーバーは DHCP サーバーと同じマシンで実行されています。
dhtadm -M -m 10.20.64.64 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.20.64.0 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.20.64.128 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.21.0.0 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.22.0.0 -e BootSrvA=10.21.0.2
# ホスト名がクライアントに返されるようにします。
dhtadm -M -m DHCP-servername -e Hostname=_NULL_VALUE_
```

### 例 7-1 ネットワークインストーラをサポートするスクリプトの例 (続き)

```
# この MAC アドレスを持つクライアントはディスククライアントです。
# インストール用のネットワーク範囲設定ルートを、クライアントのルートディレクトリで置き換えます。
dhtadm -A -m 0800201AC25E -d \
':SrootIP4=10.23.128.2:SrootNM="orange-svr-2":SrootPTH="/export/root/10.23.128.12":'
```

スーパーユーザーになり、dhtadm をバッチモードで実行します。オプションとマクロを dhcptab に追加するためのスクリプトの名前を指定します。たとえば、スクリプトの名前が netinstalloptions の場合、次のコマンドを入力します。

```
# dhtadm -B netinstalloptions
```

Vendor= 文字列に指定されているベンダークライアントクラスを持つクライアントが、DHCP を使ってネットワーク経由でインストールできるようになります。

dhtadm コマンドの使用の詳細については、dhtadm(1M) のマニュアルページを参照してください。dhcptab ファイルの詳細は、dhcptab(4) のマニュアルページを参照してください。

## DHCP マネージャを使用したインストールオプションとマクロの作成

DHCP マネージャを使って、表 7-4 のオプションと表 7-5 のマクロを作成できます。

### ▼ Solaris のインストールをサポートするオプションを作成する方法 (DHCP マネージャ)

この手順では、DHCP サーバーを構成済みであると仮定します。DHCP サーバーの構成を完了していない場合は、『Solaris のシステム管理 (IP サービス)』の「DHCP サービスの使用計画 (手順)」を参照してください。

1. DHCP サーバーシステム上でスーパーユーザーになります。
2. DHCP マネージャを起動します。

```
# /usr/sadm/admin/bin/dhccpmgr &
「DHCP マネージャ (DHCP Manager)」ウィンドウが表示されます。
```

3. DHCP マネージャで「オプション (Options)」を選択します。
4. 「編集 (Edit)」メニューから「作成 (Create)」を選択します。  
「オプションの作成 (Create Option)」ダイアログボックスが開きます。
5. 最初のオプションのオプション名を入力し、そのオプションに値を入力します。  
表 7-4 を使用して、作成する必要があるオプションの名前と値を調べます。ベンダークライアントクラスは推奨値に過ぎないことに注意してください。DHCP

サービスから Solaris インストールパラメータを取得する必要がある実際のクライアントのタイプを示すクラスを作成する必要があります。クライアントのベンダークライアントクラスを調べる方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP オプションを使用した作業 (作業マップ)」を参照してください。

6. すべての値を入力したら、「了解 (OK)」をクリックします。
7. 「オプション (Options)」タブで、今作成したオプションを選択します。
8. 「編集 (Edit)」メニューから「複製 (Duplicate)」を選択します。  
「オプションの複製 (Duplicate Option)」ダイアログボックスが開きます。
9. 別のオプションの名前を入力し、その他の値を適宜変更します。  
コード、データ型、データの単位数、最大値は通常は変更する必要があります。これらの値については、表 7-4 を参照してください。
10. すべてのオプションを作成するまで、手順 7 から手順 9 までを繰り返します。  
次の手順の説明に従って、ネットワークインストールクライアントにオプションを渡すマクロを作成できます。

---

注 - これらのオプションはすでに Solaris クライアントの /etc/dhcp/inittab ファイルに含まれているので、わざわざ追加する必要はありません。

---

## ▼ Solaris のインストールをサポートするマクロを作成する方法 (DHCP マネージャ)

この手順では、DHCP サーバーを構成済みであると仮定します。DHCP サーバーの構成を完了していない場合は、『Solaris のシステム管理 (IP サービス)』の「DHCP サービスの使用計画 (手順)」を参照してください。

1. DHCP マネージャで「マクロ (Macros)」を選択します。
2. 「編集 (Edit)」メニューから「作成 (Create)」を選択します。  
「マクロの作成 (Create Macro)」ダイアログボックスが開きます。
3. マクロの名前を入力します。  
使用できるマクロ名については、表 7-5 を参照してください。
4. 「選択 (Select)」ボタンをクリックします。  
「オプションの選択 (Select Option)」ダイアログボックスが開きます。
5. 「カテゴリ (Category)」リストで「ベンダー (Vendor)」を選択します。  
作成したベンダーオプションがリストされます。

6. マクロに追加するオプションを選択して、「了解 (OK)」をクリックします。
7. オプションの値を入力します。  
オプションのデータ型については表 7-4 を参照してください。  
`add_install_client -d` の情報も参照してください。
8. すべてのオプションを追加するまで、手順 6 から手順 7 までを繰り返します。  
別のマクロを追加するには、オプション名に **Include** と入力し、オプション値にそのマクロ名を入力します。
9. マクロが完成したら、「了解 (OK)」をクリックします。

---

## SPARC: 電源管理情報の事前設定

Solaris の電源管理ソフトウェアを使用すると、システムが 30 分間アイドル状態になると自動的にシステム状態を保存し電源を切ることができます。EPA の省電力 (Energy Star) ガイドラインのバージョン 2 に準拠したシステム (sun4u システムなどに Solaris 9 ソフトウェアをインストールするときは、デフォルトで電源管理ソフトウェアもインストールされます。Solaris Web Start のインストール時に、インストールプログラムは、電源管理ソフトウェアを有効にするか、無効にするかの指定を求めます。suninstall インストールプログラムは、インストールが完了し、システムがリポートした後で、電源管理ソフトウェアを有効にするか無効にするかを指定するよう求めます。

---

注 - 使用するシステムが Energy Star バージョン 3 以降に対応している場合、このプロンプトは表示されません。

---

対話式インストールを実行している場合は、電源管理情報を事前設定してプロンプトを回避する方法はありません。カスタム JumpStart インストールでは、finish スクリプトを使ってシステムに /autoshtutdown または /noautoshtutdown ファイルを作成することで、電源管理情報を事前設定できます。システムのリポート時に、/autoshtutdown は電源管理ソフトウェアを有効にし、/noautoshtutdown ファイルは電源管理ソフトウェアを無効にします。

たとえば、finish スクリプトに次の行を入れておくと電源管理ソフトウェアが有効になり、システムリポート後にプロンプトが表示されないようにすることができます。

```
touch /a/autoshtutdown
```

finish スクリプトの詳細は、293 ページの「finish スクリプトの作成」を参照してください。



## 第 8 章

---

# Solaris オペレーティング環境のアップグレード (計画)

---

この章では、Solaris オペレーティング環境をアップグレードする前に行う必要がある作業についての情報や手順を説明します。

- 87 ページの「アップグレード (概要)」
- 89 ページの「Solaris Live Upgrade の利用」
- 90 ページの「カスタム JumpStart によるアップグレード」
- 90 ページの「ディスク容量の再配置を使用するアップグレード」
- 91 ページの「アップグレード前のシステムのバックアップ」

---

## アップグレード (概要)

アップグレードでは、システムのディスクにある既存のファイルと Solaris オペレーティング環境の新しいバージョンがマージされます。アップグレードでは、既存の Solaris オペレーティング環境に対して行なった変更はできるだけ保存されます。

アップグレードできるシステムは、Solaris 2.6、Solaris 7、または Solaris 8 ソフトウェアが動作しているものです。システムで動作している Solaris のバージョンを確認するには、次のコマンドを入力します。

```
$ uname -a
```

Solaris オペレーティング環境のアップグレードでは、次のインストール方法を使用できます。

---

注 - ディスクレスクライアントをアップグレードする場合は、`smosservice patch` コマンドを使用する必要があります。詳細は、『Solaris のシステム管理 (基本編)』または `smosservice(1M)` のマニュアルページを参照してください。

---

すでに Solaris 9 オペレーティング環境を実行していて、システムに個別のパッチがインストールされている場合、Solaris 9 Update リリースへのアップグレードを行うと、パッチは次のようになります。

- Solaris 9 Update リリースの一部として組み込まれているパッチは自動的に再適用されます。これらのパッチをバックアウトすることはできません。
- システムにすでにインストールされているパッチのうち、Solaris 9 リリースに組み込まれていないパッチは削除されます。

パッチアナライザを使用すれば、Solaris 9 Update リリースにアップグレードしたときに、これらのパッチのうちどれが削除されるかを確認することができます。パッチアナライザの詳しい使い方については、673 ページの「Solaris Update リリースへのアップグレード」を参照してください。

## Solaris のアップグレード方法

表 8-1 SPARC: Solaris のアップグレード方法

プラットフォーム	現在の Solaris オペレーティング環境	Solaris のアップグレード方法
SPARC システム	Solaris 2.6、Solaris 7、Solaris 8、Solaris 9	<ul style="list-style-type: none"> <li>■ Solaris Web Start プログラム</li> <li>■ Solaris suninstall プログラム</li> <li>■ カスタム JumpStart</li> <li>■ Solaris Live Upgrade</li> </ul>

表 8-2 x86: Solaris のアップグレード方法

現在の Solaris オペレーティング環境	Solaris のアップグレード方法
Solaris 2.6	DVD メディアまたはネットインストールイメージからのインストール: <ul style="list-style-type: none"> <li>■ Solaris Web Start プログラム</li> <li>■ Solaris suninstall プログラム</li> <li>■ カスタム JumpStart</li> </ul> CD メディアからのインストール: <ul style="list-style-type: none"> <li>■ Solaris suninstall プログラム</li> <li>■ カスタム JumpStart</li> </ul>



表 8-2 x86: Solaris のアップグレード方法 (続き)

現在の Solaris オペレーティング環境	Solaris のアップグレード方法
Solaris 7	DVD メディアまたはネットインストールイメージからのインストール: <ul style="list-style-type: none"> <li>■ Solaris Web Start プログラム</li> <li>■ Solaris suninstall プログラム</li> <li>■ カスタム JumpStart</li> <li>■ Solaris Live Upgrade</li> </ul> CD メディアからのインストール: <ul style="list-style-type: none"> <li>■ Solaris suninstall プログラム</li> <li>■ カスタム JumpStart</li> <li>■ Solaris Live Upgrade</li> </ul>
Solaris 8, Solaris 9	DVD/CD メディアまたはネットインストールイメージからのインストール: <ul style="list-style-type: none"> <li>■ Solaris Web Start プログラム</li> <li>■ Solaris suninstall プログラム</li> <li>■ カスタム JumpStart</li> <li>■ Solaris Live Upgrade</li> </ul>

---

注 – Solaris Live Upgrade を使用したアップグレードの制限については、397 ページの「Solaris Live Upgrade の要件」を参照してください。

---

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 Live Upgrade の利用

Solaris Live Upgrade を利用すると、複製された非アクティブオペレーティング環境をアップグレードすることによりオペレーティングシステムのダウンタイムを減らすことができます。

Solaris Live Upgrade の計画と実施の方法については、第 32 章を参照してください。

---

## カスタム JumpStart によるアップグレード

カスタム JumpStart インストールを使用してアップグレードを行う場合は、カスタム JumpStart プロファイルに `install_type upgrade` を指定する必要があります。

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。プロファイル进行测试するには、アップグレードするシステムで `pfinstall -D` コマンドを使用します。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。アップグレード用プロファイルのテストについての詳細は、284 ページの「プロファイルのテスト」を参照してください。

JumpStart を使用する場合、フラッシュ差分アーカイブを利用してクローンシステムを更新できます。フラッシュアーカイブの概要は、第 20 章を参照してください。

---

## ディスク容量の再配置を使用するアップグレード

Solaris Web Start のアップグレードオプション、Solaris `suninstall` プログラムおよびカスタム JumpStart プログラムにはディスクの容量を再配置する機能があります。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。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。特定のソフトウェアグループに含まれる新しいソフトウェアは、インストールの対象となるようにアップグレード時に自動的に選択される。
- 新しいリリースに、システム上の既存のソフトウェアよりもサイズが大きいソフトウェアが存在する。

自動再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。自動再配置機能では、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 – 自動再配置機能には、ファイルシステムの容量を増やす能力はありません。自動再配置機能による容量の再配置では、変更の必要なファイルシステム上の必須ファイルのバックアップ、ファイルシステムの変更にもとづくディスクパーティションの再分割、およびバックアップファイルの復元がアップグレードの前に行われます。

---

- Solaris Web Start 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、Solaris suninstall プログラムまたはカスタム 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。
- Solaris suninstall 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、移動または変更してもよいファイルシステムを指定してから再び自動配置機能を実行します。
- 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に十分なディスク容量がない場合は、`backup_media` と `layout_constraint` キーワードを使用してディスク容量を再配置することができます。`backup_media` と `layout_constraint` キーワードの使用例については、例 26-5 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 オペレーティング環境のアップグレードを行う前に既存システムのバックアップを行う必要があります。ファイルシステムをテープなどの取り外し可能媒体にコピーすれば、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手順についての詳細は、『Solaris のシステム管理 (基本編)』を参照してください。



## 第 9 章

---

# インストール時の RAID-1 ボリューム (ミラー) の作成 (トピック)

---

以下の章では、Solaris ボリュームマネージャコンポーネントの概要を説明します。インストール時またはアップグレード時にファイルシステムをミラー化するためのガイドラインおよび要件についても説明します。

### 第 10 章

ファイルシステムのミラー化に必要な Solaris ボリュームマネージャコンポーネントの概要を説明します。

### 第 11 章

インストール時またはアップグレード時にミラー化されたファイルシステムを作成するためのガイドラインおよび要件について説明します。



## 第 10 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。

- 95 ページの「ミラー化を行う理由」
- 96 ページの「ミラー化の機能」
- 99 ページの「ミラー化コンポーネントの概要」
- 102 ページの「ミラー化されたファイルシステムのレイアウト例」

ミラー化されたファイルシステムの作成に Solaris Live Upgrade を使用する方法については、404 ページの「ミラー化された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。

ミラー化されたファイルシステムの作成にカスタム JumpStart インストールを使用する方法については、359 ページの「filesys プロファイルキーワード (ミラー化されたファイルシステムの作成)」および365 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。

---

## ミラー化を行う理由

インストール時またはアップグレード時に、ミラー化されたファイルシステムを作成して、システムデータを複数の物理ディスクに複製できます。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 インストールでは、ミラー化されたファイルシステムの作成に Solaris ボリュームマネージャを使用します。Solaris ボリュームマネージャでは、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では、連結、ストライプなど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。ミラー化されたファイルシステムを、インストール時またはアップグレード時に作成すれば、インストール後に作成する必要はなくなります。

---

注 - カスタム JumpStart および Solaris Live Upgrade インストールでは、RAID-0 ボリュームと RAID-1 ボリュームの作成だけがサポートされます。その他の Solaris ボリュームマネージャコンポーネント、たとえば RAID-5 ボリュームは、サポートされません。

---

カスタム JumpStart インストールでは、初期インストール時のみ、ミラー化されたファイルシステムの作成がサポートされます。Solaris Live Upgrade では、アップグレード時のミラー化されたファイルシステムの作成がサポートされます。

Solaris ボリュームマネージャのソフトウェアとコンポーネントについては、『Solaris ボリュームマネージャの管理』を参照してください。

---

## ミラー化の機能

Solaris ボリュームマネージャは、仮想ディスクを使用して、物理ディスクとその関連データの管理を行います。Solaris ボリュームマネージャでは、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。実際には、ボリュームは標準 UNIX® の擬似または仮想デバイスと同義です。

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) または他の Solaris ボリュームマネージャ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。

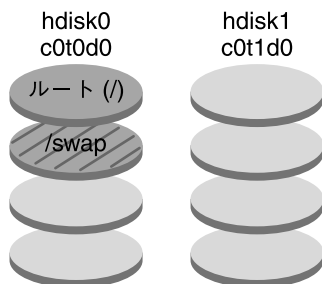


カスタム JumpStart インストールおよび Solaris Live Upgrade では、ミラー化されたファイルシステムの作成用としてブロックデバイスがサポートされます。ボリューム名の詳細は、106 ページの「ボリューム名の要件とガイドライン」を参照してください。

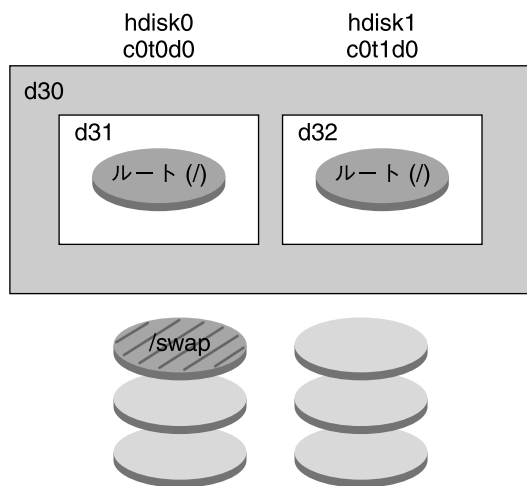
ミラー化されたファイルシステムを作成すると、RAID-0 ボリューム (単一スライスの連結) と RAID-1 ボリューム (ミラー) が作成されます。Solaris ボリュームマネージャは、連結 (サブミラー) 上にデータを複製し、これらのサブミラーを 1 つのミラーボリュームとして処理します。

図 10-1 は、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 (サブミラー)  
からなるミラー化されたルートファイルシステム



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 10-1 ルートファイルシステムを2つのディスクにミラー化

図 10-1 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。

---

## ミラー化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、ファイルシステムのミラー化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は動作できません。Solaris ボリュームマネージャ構成には、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。そのために、Solaris ボリュームマネージャは「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、コンセンサスが得られたこととなります。

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、複製を格納した後で、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。

状態データベースと状態データベースの複製の要件の計画については、106 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。

状態データベースと状態データベースの複製の詳細は、『Solaris ボリュームマネージャの管理』を参照してください。

## RAID-0 ボリューム (連結)

カスタム JumpStart および Solaris Live Upgrade インストールでは、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 ボリュームを接続することができます。「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 ボリュームの要件の計画については、108 ページの「ミラーとサブミラーの要件およびガイドライン」を参照してください。

RAID-0 ボリュームの詳細は、『Solaris ボリュームマネージャの管理』を参照してください。

## RAID-1 ボリューム (ミラー)

RAID-1 ボリューム (「ミラー」) とは、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) で保持しているボリュームのことです。ミラー化するためには、より多くのディスク容量が必要です。少なくとも、ミラー化するデータ量の 2 倍のディスク容量が必要になります。また、ミラー化ではデータがすべてのサブミラーに書き込まれるため、書き込み要求の処理時間が長くなります。

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。

構成したミラーは、物理スライスと同じように使用できます。

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ミラー化できます。また、ミラーは、データベースなど、どのようなアプリケーションにも使用できます。

RAID-1 ボリュームの要件の計画については、108 ページの「ミラーとサブミラーの要件およびガイドライン」を参照してください。

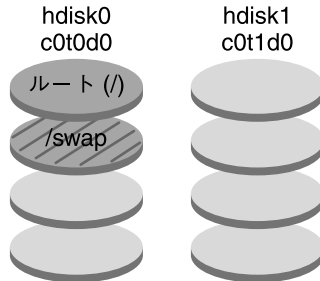
RAID-1 ボリュームの詳細は、『Solaris ボリュームマネージャの管理』を参照してください。

---

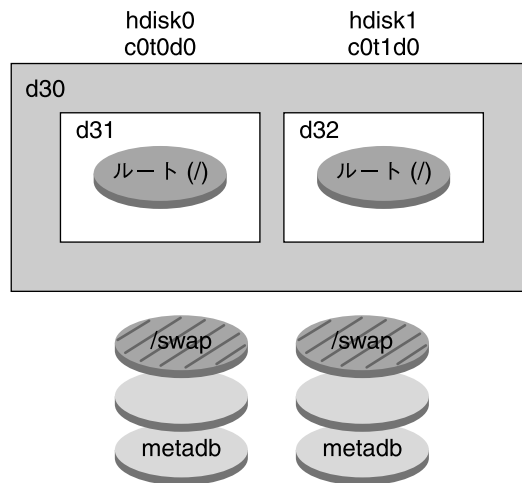
## ミラー化されたファイルシステムのレイアウト例

次の図に、ルートファイルシステム (/) を 2 つの物理ディスクに複製したミラーを示します。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 10-2 ミラー化されたルートファイルシステムのレイアウト例

図 10-2 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この構成を作成するプロファイル例については、例 26-10を参照してください。

Solaris Live Upgrade を使用し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、438 ページの「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。



## 第 11 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 インストールを使用してミラー化されたファイルシステ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。

- 105 ページの「システム要件」
- 106 ページの「ボリューム名の要件とガイドライン」
- 106 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」
- 108 ページの「ミラーとサブミラーの要件およびガイドライン」
- 110 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」

ミラー化されたファイルシステムの作成に Solaris Live Upgrade インストールを使用する計画の詳細は、404 ページの「ミラー化された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。

ミラー化されたファイルシステムの作成にカスタム JumpStart インストールを使用する手順については、359 ページの「filesys プロファイルキーワード (ミラー化されたファイルシステムの作成)」および365 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

特定のスライス上にミラー化されたファイルシステムを作成する場合は、インストール時に、ミラー化に使用するディスクを直接システムに接続し、使用可能にする必要があります。

---

## ボリューム名の要件とガイドライン

カスタム JumpStart インストールでは、インストール時に RAID-0 サブミラーにボリューム名が自動的に割り当てられます。オプションとして、JumpStart キーワード `filesys` を指定して、RAID-1 ボリューム (ミラー) に名前を割り当てることができます。

ボリューム名の割り当て時には、次の規則に従ってください。

- 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`)。
- 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。

ボリュームの管理を単純化するため、次の標準命名規則を参考にしてください。

- 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。
- Solaris Live Upgrade を使用してミラーを作成するときは、いつも同じ命名関係を使用します。たとえば、ミラーの名前を 0 で終わる数字にし、サブミラーの名前を 1 や 2 で終わる数字にします。この方法で、ミラーの名前を `d10` にした場合、サブミラーの名前は `d11` および `d12` になります。ミラーの名前を `d20` にした場合、サブミラーの名前は `d21` および `d22` になります。

カスタム JumpStart インストールを使用してミラーを作成する場合、サブミラーにはミラー名に対応したサブミラー名が自動的に割り当てられます。

- 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。
- Solaris ボリュームマネージャでは、0 から 127 までの 128 個の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`

Solaris ボリュームマネージャ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。

---

## 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 (基本編)』の「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、他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擇する前に、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 つのスライスに置く

- 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つずつ複製を置く
- 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つずつ複製を置く
- 状態データベースの複製を追加して、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。

---

## ミラーとサブミラーの要件およびガイドライン

RAID-1 ボリューム(ミラー) および RAID-0 ボリューム(単一スライスの連結)は、次のガイドラインに従って操作してください。

## 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。

- RAID-0 ボリュームという用語は、ディスクドライブまたはディスク連結を表します。カスタム JumpStart と Solaris Live Upgrade インストールでは、単一スライスの連結のみ作成できます。インストール時またはアップグレード時に RAID-0 ストライプボリュームを作成することはできません。
- カスタム JumpStart インストールでは、各ミラーにつき最大2つのサブミラーを作成できます。Solaris Live Upgrade では、各ミラーにつき最大3つの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すみます。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。

- カスタム JumpStart インストールでミラー化されたファイルシステムを作成する場合は、ミラーを作成する前にミラー化するファイルシステムを作成する必要はありません。

## ディスクとコントローラの実験のガイドライン

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には、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。

## スライスの実験のガイドライン

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。
- サブミラーのスライス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。

## 第 12 章

# ネットワークインストールの準備 (トピック)

---

以下の章では、DVD メディアや CD メディアを利用するのではなくローカルエリアネットワークを使用して Solaris ソフトウェアをインストールする場合のシステム設定の方法を説明します。

第 13 章	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方法の概要と計画について説明します。
第 14 章	DVD メディアからインストールサーバーに Solaris ソフトウェアをコピーする手順を説明します。
第 15 章	CD メディアからインストールサーバーに Solaris ソフトウェアをコピーする手順を説明します。
第 16 章	ネットワークインストールを設定するコマンドについて説明します。





## 第 13 章

# ネットワークインストールの準備 (概要)

この章では、DVD または CD メディアからではなくローカルエリアネットワークから Solaris ソフトウェアをインストールする場合にネットワークとシステムをどのように設定する必要があるかを説明します。

広域ネットワーク経由でクライアントをインストールする方法については、第 41 章を参照してください。

## ネットワークインストールの計画

この節では、ネットワークからインストールを行う前に認識しておくべき事柄を説明します。ネットワークインストールでは、Solaris 9 ディスクイメージにアクセスできる「インストールサーバー」というシステムから Solaris ソフトウェアをインストールできます。その場合には、まず Solaris 9 DVD または CD メディアの内容をインストールサーバーのハードディスクにコピーする必要があります。これで、任意の Solaris インストール方法を使用して Solaris ソフトウェアをネットワーク上でインストールできます。

## ネットワークインストールに必要なサーバー

ネットワーク上で Solaris オペレーティング環境をインストールするには、次のサーバーがネットワーク上に存在する必要があります。

- インストールサーバー – Solaris 9 ディスクイメージ (CD イメージまたは DVD イメージ) が格納された、ネットワークに接続されたシステム。このシステムからネットワーク上の別のシステムに Solaris 9 をインストールできます。インストールサーバーを作成するには、以下のメディアの 1 つからイメージをコピーします。
  - Solaris DVD

- Solaris SOFTWARE 1 of 2 CD および Solaris SOFTWARE 2 of 2 CD

Solaris SOFTWARE 1 of 2 と Solaris SOFTWARE 2 of 2 CD からイメージをコピーした後、インストール要件に応じて、Solaris INSTALLATION と Solaris LANGUAGES CD のイメージもコピーできます。

1 台のインストールサーバーで、異なる Solaris リリースや異なるプラットフォームに対して別々のディスクイメージを提供できます。それには、提供する複数のイメージをインストールサーバーのハードディスクにコピーする必要があります。たとえば、1 台のインストールサーバーに SPARC プラットフォームと x86 プラットフォームのディスクイメージを格納できます。

インストールサーバーの作成についての詳細は、次に示す節のどれかを参照してください。

- 119 ページの「SPARC または x86 DVD メディアを使用して SPARC インストールサーバーを作成する方法」
- 122 ページの「x86: SPARC または x86 DVD メディアを使用して x86 インストールサーバーを作成する方法」
- 137 ページの「SPARC: SPARC CD メディアを使用して SPARC インストールサーバーを作成する方法」
- 147 ページの「クロスプラットフォームの CD メディアを使用したインストールサーバーの作成」
- ブートサーバー - オペレーティング環境をインストールするため、同一ネットワークサブネット上の複数のクライアントに対して、ブートに必要な情報を提供するサーバーシステム。ブートサーバーとインストールサーバーは、通常は同じシステムです。ただし、インストールされる Solaris 9 ソフトウェアのシステムがインストールサーバーとは異なるサブネット上に存在する場合で、DHCP を使用していないときは、そのサブネットにブートサーバーが必要です。

1 台のブートサーバーで、複数のリリースの Solaris 9 ブートソフトウェア (異なるプラットフォームの Solaris 9 ブートソフトウェアも含む) を提供できます。たとえば、SPARC ブートサーバーは、SPARC システムに Solaris 8 と Solaris 9 のブートソフトウェアを提供できます。同時に x86 システムに Solaris 9 のブートソフトウェアを提供することもできます。

---

注 - DHCP を使用する場合は、ブートサーバーを個別に作成する必要はありません。詳細は、116 ページの「DHCP によるネットワークインストールパラメータの指定」を参照してください。

---

ブートサーバーの作成についての詳細は、次に示す節のいずれかを参照してください。

- 126 ページの「DVD イメージを用いたサブネット上でのブートサーバーの作成」
- 159 ページの「CD イメージを用いたサブネット上でのブートサーバーの作成」

- (省略可能) ネームサーバー - 分散ネットワークデータベース (DNS、NIS、NIS+、LDAP など) を管理するシステム。これには、同じネットワーク上のユーザーやほかのシステムの情報が含まれています。

ネームサーバーの作成方法については、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

---

注 - インストールサーバーとネームサーバーは、同じシステムであっても別のシステムであってもかまいません。

---

図 13-1 は、ネットワークインストールに使用される一般的なサーバー構成を示したものです。

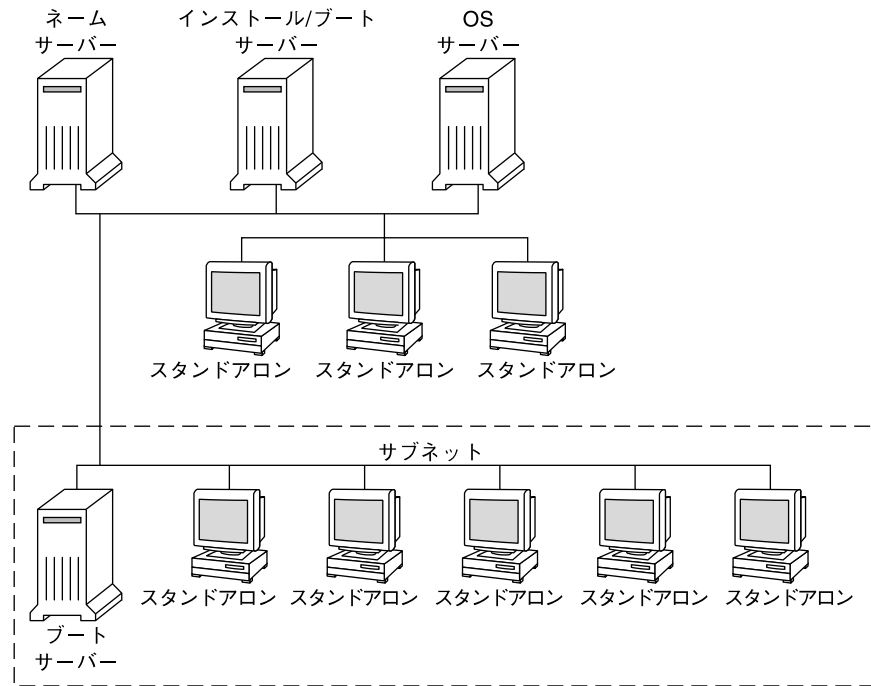


図 13-1 ネットワークインストールサーバー

## DHCP によるネットワークインストールパラメータの指定

DHCP (Dynamic Host Configuration Protocol) は、インストールに必要なネットワークパラメータを提供します。DHCP を使用する場合は、ブートサーバーを個別に作成する必要はありません。インストールサーバーの作成後、`add_install_client -d` コマンドを使用してネットワークにクライアントを追加します。-d オプションを指定すると、DHCP を使用してネットワーク経由で Solaris インストールできるようにクライアントシステムを設定できます。

インストールパラメータ用の DHCP オプションについては、75 ページの「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。

## 第 14 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)

この章では、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として DVD メディアを使用してネットワークとシステムを設定する方法を説明します。ネットワークインストールを使用すると、Solaris 9 ディスクイメージにアクセスできるシステム (インストールサーバーと呼ばれる) から、ネットワーク上のほかのシステムに Solaris ソフトウェアをインストールできます。その場合には、まず Solaris 9 DVD メディアの内容をインストールサーバーのハードディスクにコピーする必要があります。これで、任意の Solaris インストール方法を使用して Solaris ソフトウェアをネットワーク上でインストールできます。この章の内容は次のとおりです。

- 117 ページの「作業マップ: DVD メディアを使用してネットワーク上でインストールするための準備」
- 118 ページの「DVD メディアを用いたインストールサーバーの作成」
- 126 ページの「DVD イメージを用いたサブネット上でのブートサーバーの作成」
- 128 ページの「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を追加する」

## 作業マップ: DVD メディアを使用してネットワーク上でインストールするための準備

表 14-1 作業マップ: DVD メディアを使用するインストールサーバーの設定

作業	説明	参照先
インストールサーバーを作成する	<code>setup_install_server</code> (1M) コマンドを使用して Solaris DVD をインストールサーバーのハードディスクにコピーします。	118 ページの「DVD メディアを用いたインストールサーバーの作成」

表 14-1 作業マップ: DVD メディアを使用するインストールサーバーの設定 (続き)

作業	説明	参照先
(省略可能) ブートサーバーを作成する	インストールサーバーと異なるサブネット上にあるシステムをネットワークからインストールする場合、システムをブートするためのブートサーバーをサブネット上に作成する必要があります。ブートサーバーを設定するには、 <code>setup_install_server</code> コマンドを <code>-b</code> オプションを付けて実行します。DHCP (Dynamic Host Configuration Protocol) を使用する場合は、ブートサーバーは必要ありません。	126 ページの「DVD イメージを用いたサブネット上でのブートサーバーの作成」
ネットワークインストールするシステムを追加する	<code>add_install_client</code> コマンドを使用して、ネットワークからインストールする各システムを設定します。インストールする各システムには、ネットワーク上にあるインストールサーバー、ブートサーバー、および構成情報を認識させる必要があります。	128 ページの「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を追加する」

## DVD メディアを用いたインストールサーバーの作成

インストールサーバーには、ネットワーク上でシステムをインストールするために必要なインストールイメージが含まれます。Solaris ソフトウェアをネットワークからインストールするためには、インストールサーバーを作成する必要があります。ブートサーバーは必ずしも設定する必要はありません。

- DHCP を使用してインストールパラメータを設定している場合、あるいは、インストールサーバーとクライアントが同じサブネット上にある場合、ブートサーバーは必要ありません。
- インストールサーバーとクライアントが同じサブネット上に存在せず、DHCP を使用していない場合は、サブネットごとに異なるブートサーバーを作成する必要があります。インストールサーバーはサブネットごとに作成できます。しかし、インストールサーバーはより多くのディスク容量を必要とします。

---

注 – Solaris DVD メディアを使用して Solaris 2.6 または 7 オペレーティング環境の稼働するシステム上でインストールサーバーを設定する場合、最初に次のパッチのいずれかを適用する必要があります。

- Solaris 2.6 オペレーティング環境 (SPARC 版) — Patch ID 107618-03
  - Solaris 2.6 オペレーティング環境 (Intel 版) — Patch ID 107619-03
  - Solaris 7 オペレーティング環境 (SPARC 版) — Patch ID 107259-03
  - Solaris 7 オペレーティング環境 (Intel 版) — Patch ID 107260-03
- 

## ▼ SPARC または x86 DVD メディアを使用して SPARC インストールサーバーを作成する方法

---

注 – SPARC: Solaris 2.3 以前にリリースされた SunOS バージョンを使用するシステムは使用できません。

---

---

注 – この手順では、システムでボリュームマネージャを実行していると仮定しています。ボリュームマネージャを使用せずにメディアを管理する方法については、『Solaris のシステム管理 (基本編)』を参照してください。

---

1. インストールサーバーとして使用する **SPARC** システムでスーパーユーザーになります。  
このシステムには DVD-ROM ドライブが必要です。さらに、システムは、このサイトのネットワークに接続されネームサービスに登録されている必要があります。ネームサービスを使用する場合、システムはすでに NIS、NIS+、DNS、または LDAP のいずれかのサービスに登録されている必要があります。ネームサービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。
2. **SPARC** システムのドライブに **Solaris DVD** を挿入します。
3. **DVD** イメージをコピーするディレクトリを作成します。  

```
# mkdir -p install_dir_path
```
4. マウントされたディスクの **Tools** ディレクトリに移動します。
  - **SPARC** DVD メディアの場合、次のように入力します。  

```
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
```
  - **x86** DVD メディアの場合、次のように入力します。

```
# cd /cdrom/cdrom0/Solaris_9/Tools
```

上の例では、**cdrom0** は、Solaris オペレーティング環境の DVD メディアを含むドライブへのパスです。

5. ドライブ内の **DVD** イメージをインストールサーバーのハードディスクにコピーします。

```
# ./setup_install_server install_dir_path
```

*install\_dir\_path* DVD イメージをコピーするディレクトリを指定します。

---

注 - `setup_install_server` コマンドは、Solaris SOFTWARE ディスクイメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、`df -k1` コマンドを使用します。

---

6. 他のシステムがインストールサーバーをマウントできるように設定する必要があるかどうかを判断します。

- インストールサーバーとクライアント (インストールしているシステム) が同じサブネット上にある場合、あるいは、DHCP を使用している場合、ブートサーバーを作成する必要はありません。手順 10 に進みます。
- インストールサーバーがインストールするシステムと同じサブネット上に存在せず、DHCP を使用していない場合は、以下の手順を実行してください。

7. インストールサーバーのイメージへのパスが適切に共有されていることを確認します。

```
# share | grep install_dir_path
```

*install\_dir\_path* DVD イメージがコピーされているインストールイメージへのパスを指定します。

- インストールサーバーのディレクトリへのパスが表示される場合、かつ、`anon=0` がオプションに表示される場合、手順 10 に進みます。
  - インストールサーバーのディレクトリのパスが表示されないか、オプション内に `anon=0` と示されない場合は、次の作業へ進みます。
8. 次のエントリを `/etc/dfs/dfstab` ファイルに追加し、ブートサーバーからインストールサーバーが利用できるように設定します。

```
share -F nfs -o ro,anon=0 -d "install server directory" install_dir_path
```

9. インストールサーバーのディレクトリパスが正しく共有されているか確認します。

```
# ps -ef | grep nfsd
```

- `nfsd` デーモンが稼働していない場合は、このデーモンを起動してください。

```
# /etc/init.d/nfs.server start
```



- nfsd デーモンがすでに稼働している場合は、インストールサーバーを共有してください。

```
# shareall
```

10. ルート (/) ディレクトリへ移動します。

```
# cd /
```

11. Solaris DVD を取り出します。

12. `setup_install_server` によって作成されたネットインストールイメージ上のミニルート (`/install_dir_path/Solaris_9/Tools/Boot`) にあるファイルにパッチを適用するか決定します。ブートイメージに問題がある場合は、ファイルにパッチを適用する必要があります。

- パッチを適用しない場合は、このまま続けます。
- 適用する場合は、`patchadd -C` コマンドを使用して、ミニルートにあるファイルにパッチを適用します。



---

注意 `-patchadd -C` を使用する前には必ず、パッチの README を読むか、ご購入先におたずねください。

---

13. ブートサーバーを作成する必要があるかどうかを判断します。

- DHCP を使用しているか、あるいはインストール対象のシステムと同じサブネット上にインストールサーバーが存在する場合は、ブートサーバーを作成する必要はありません。128 ページの「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を追加する」へ進みます。
- DHCP を使用しておらず、インストールサーバーとクライアントが別のサブネットに存在する場合は、ブートサーバーを作成する必要があります。126 ページの「DVD イメージを用いたサブネット上でのブートサーバーの作成」へ進みます。

例 14-1 SPARC: SPARC DVD を使用した SPARC インストールサーバーの作成

次に、Solaris DVD をインストールサーバーの `/export/home/dvdsparc` ディレクトリにコピーして、インストールサーバーを作成する方法を示します。

```
# mkdir -p /export/home/dvdsparc
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
# ./setup_install_server /export/home/dvdsparc
```

別のブートサーバーが必要な場合、以下のコマンドを入力します。

次のパスを `/etc/dfs/dfstab` ファイルに追加します。

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/dvdsparc
```

例 14-1 SPARC: SPARC DVD を使用した SPARC インストールサーバーの作成 (続き)

nfsd デーモンが稼働しているかどうかを確認します。nfsd デーモンが稼働していない場合、このデーモンを起動して共有します。

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
# cd /
```

例 14-2 x86: x86 DVD を使用した SPARC インストールサーバーの作成

次に、Solaris DVD をインストールサーバーの /export/home/dvdx86 ディレクトリにコピーして、インストールサーバーを作成する方法を示します。

```
# mkdir -p /export/home/dvdx86
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./setup_install_server /export/home/dvdx86
```

次のパスを /etc/dfs/dfstab ファイルに追加します。

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/dvdx86
```

nfsd デーモンが稼働しているかどうかを確認します。nfsd デーモンが稼働していない場合、このデーモンを起動して共有します。

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
# cd /
```

## ▼ x86: SPARC または x86 DVD メディアを使用して x86 インストールサーバーを作成する方法

---

注 - この手順では、システムでボリュームマネージャを実行していると仮定していません。ボリュームマネージャを使用せずにメディアを管理する方法については、『Solaris のシステム管理 (基本編)』を参照してください。

---

1. インストールサーバーとして使用する x86 システムでスーパーユーザーになります。  
このシステムには DVD-ROM ドライブが必要です。さらに、システムは、このサイトのネットワークに接続されネームサービスに登録されている必要があります。ネームサービスを使用する場合、システムのネームサービスは、NIS、NIS+、DNS、LDAP のいずれかでなければなりません。ネームサービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。

2. システムのドライブに **Solaris DVD** を挿入します。

3. ブートイメージをコピーするディレクトリを作成します。

```
# mkdir -p install_dir_path
```

*install\_dir\_path* DVD イメージをコピーするディレクトリを指定します。

4. マウントされたディスクの **Tools** ディレクトリに移動します。

- x86 DVD メディアの場合、次のように入力します。

```
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
```

- SPARC DVD メディアの場合、次のように入力します。

```
# cd /cdrom/cdrom0/Solaris_9/Tools
```

上の例では、**cdrom0** は、Solaris オペレーティング環境の DVD メディアを含むドライブへのパスです。

5. **setup\_install\_server** コマンドを使用して、ドライブ内のディスクをインストールサーバーのハードディスクにコピーします。

```
# ./setup_install_server install_dir_path
```

*install\_dir\_path* DVD イメージをコピーするディレクトリを指定します。

---

注 - **setup\_install\_server** コマンドは、Solaris SOFTWARE ディスクイメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、**df -k1** コマンドを使用します。

---

6. 他のシステムがインストールサーバーをマウントできるように設定する必要があるかどうかを判断します。

- インストールサーバーとクライアント (インストールしているシステム) が同じサブネット上にある場合、あるいは、DHCP を使用している場合、ブートサーバーを作成する必要はありません。手順 10 に進みます。

- インストールサーバーがインストールするシステムと同じサブネット上に存在せず、DHCP を使用していない場合は、以下の手順を実行してください。

7. インストールサーバーのイメージへのパスが適切に共有されていることを確認します。

```
# share | grep install_dir_path
```

*install\_dir\_path* DVD イメージがコピーされているインストールイメージを指定します。

- インストールサーバーのディレクトリへのパスが表示されて、**anon=0** がオプションに表示される場合、手順 10 に進みます。

- インストールサーバーのディレクトリのパスが表示されないか、オプション内に `anon=0` と示されない場合は、次の作業へ進みます。
8. 次のエントリを `/etc/dfs/dfstab` ファイルに追加し、ブートサーバーからインストールサーバーが利用できるように設定します。

```
share -F nfs -o ro,anon=0 -d "install server directory" install_dir_path
```

9. インストールサーバーのディレクトリパスが正しく共有されているかを確認します。

```
# ps -ef | grep nfsd
```

- `nfsd` デーモンが稼働していない場合は、このデーモンを起動してください。

```
# /etc/init.d/nfs.server start
```

- `nfsd` デーモンがすでに稼働している場合は、インストールサーバーを共有してください。

```
# shareall
```

10. ルート (`/`) ディレクトリへ移動します。

```
# cd /
```

11. **Solaris DVD** を取り出します。

12. `setup_install_server` によって作成されたネットインストールイメージ上のミニルート (`Solaris_9/Tools/Boot`) にあるファイルにパッチを適用するか決定します。

- パッチを適用しない場合は、そのまま継続します。
- 適用する場合は、`patchadd -c` コマンドを使用して、ミニルートにあるファイルにパッチを適用します。

13. ブートサーバーを作成する必要があるかどうかを判断します。

- インストールサーバーとクライアント (インストールしているシステム) が同じサブネット上にある場合、あるいは、DHCP を使用している場合、ブートサーバーを作成する必要はありません。128 ページの「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を追加する」を参照してください。
- インストールサーバーがインストールするシステムと同じサブネット上に存在せず、DHCP を使用していない場合は、ブートサーバーを作成する必要があります。ブートサーバーの作成方法については、126 ページの「DVD イメージを用いたサブネット上でのブートサーバーの作成」を参照してください。

例 14-3 x86: x86 DVD を使用して x86 インストールサーバーを作成する

次に、インストールサーバーの `/export/home/dvdx86` ディレクトリに Solaris DVD (x86 版) をコピーして x86 インストールサーバーを作成する方法を示します。

例 14-3 x86: x86 DVD を使用して x86 インストールサーバーを作成する (続き)

```
# mkdir -p /export/home/dvdx86
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
# ./setup_install_server /export/home/dvdx86
```

次のパスを /etc/dfs/dfstab ファイルに追加します。

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/dvdx86
```

nfsd デーモンが稼働しているかどうかを確認します。nfsd デーモンが稼働していない場合、このデーモンを起動して共有します。

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
# cd /
```

例 14-4 SPARC DVD を使用して x86 インストールサーバーを作成する

次に、インストールサーバーの /export/home/dvdsparc ディレクトリに Solaris DVD (SPARC 版) をコピーして x86 インストールサーバーを作成する方法を示します。

```
# mkdir -p /export/home/dvdsparc
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./setup_install_server /export/home/dvdsparc
```

次のパスを /etc/dfs/dfstab ファイルに追加します。

```
share -F nfs -o ro,anon=0 -d "install server directory" /export/home/dvdsparc
```

nfsd デーモンが稼働しているかどうかを確認します。nfsd デーモンが稼働していない場合、このデーモンを起動して共有します。

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
# cd /
```

---

## DVD イメージを用いたサブネット上でのブートサーバーの作成

Solaris ソフトウェアをネットワークからインストールするためには、インストールサーバーを作成する必要があります。ブートサーバーは必ずしも設定する必要はありません。ブートサーバーには、ネットワーク上でシステムをブートするために必要なブートソフトウェアが含まれています。一方、インストールサーバーは Solaris ソフトウェアのインストールを行います。

- DHCP を使用してインストールパラメータを設定している場合、あるいは、インストールサーバーとクライアントが同じサブネット上にある場合、ブートサーバーは必要ありません。128 ページの「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を追加する」へ進みます。
- インストールサーバーとクライアントが同じサブネット上に存在せず、DHCP を使用していない場合は、サブネットごとに異なるブートサーバーを作成する必要があります。インストールサーバーはサブネットごとに作成できます。しかし、インストールサーバーはより多くのディスク容量を必要とします。

### ▼ DVD イメージを使用してサブネット上にブートサーバーを作成する方法

1. サブネット内でブートサーバーとして使用するシステムにログインし、スーパーユーザーになります。  
システムはリモートの Solaris 9 ディスクイメージ (通常は、インストールサーバー) へのアクセス権を持っている必要があります。ネームサービスを使用する場合、システムはすでにネームサービスに登録されている必要があります。ネームサービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。

2. インストールサーバーから **Solaris DVD** をマウントします。

```
# mount -F nfs -o ro server_name:path /mnt
```

*server\_name:path*

インストールサーバー名とディスクイメージの絶対パスを指定します。

3. ブートイメージ用のディレクトリを作成します。

```
# mkdir -p boot_dir_path
```

*boot\_dir\_path* ブートソフトウェアをコピーするディレクトリを指定します。

4. **Solaris DVD** イメージ上の **Tools** ディレクトリに移動します。

```
# cd /mnt/Solaris_9/Tools
```

5. ブートソフトウェアをブートサーバーにコピーします。

```
# ./setup_install_server -b boot_dir_path
```

-b システムをブートサーバーとして設定することを示します。

boot\_dir\_path ブートソフトウェアをコピーするディレクトリを指定します。

---

注 -setup\_install\_server コマンドは、イメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、df -k1 コマンドを使用します。

---

6. ルート (/) ディレクトリへ移動します。

```
# cd /
```

7. インストールイメージをマウント解除します。

```
# umount /mnt
```

以上の操作で、ネットワークからインストールするためにシステムを設定する用意が整いました。128 ページの「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を追加する」を参照してください。

#### 例 14-5 サブネット上にブートサーバーを作成する

次の例は、ブートサーバーをサブネット上に作成する方法を示したものです。これらのコマンドを実行すると、Solaris DVD イメージのブートソフトウェアが crystal という名前のシステムのローカルディスク上の /export/home/dvdsparc にコピーされます。

```
# mount -F nfs -o ro crystal:/export/home/dvdsparc /mnt
# mkdir -p /export/home/dvdsparc
# cd /mnt/Solaris_9/Tools
# ./setup_install_server -b /export/home/dvdsparc
# cd /
# umount /mnt
```

---

## 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を追加する

インストールサーバーとブートサーバー (必要な場合) を作成したあと、ネットワークからインストールする各システムを設定する必要があります。インストールする各システムは、次の情報にアクセスできる必要があります。

- インストールサーバー
- ブートサーバー (必要な場合)
- `sysidcfg` ファイル (`sysidcfg` ファイルを使用してシステム構成情報を事前設定する場合)
- ネームサーバー (ネームサービスを使用してシステム構成情報を事前設定する場合)
- プロファイルサーバー上の `JumpStart` ディレクトリにあるプロファイル (カスタム `JumpStart` インストールを使用する場合)

インストールサーバーとクライアントの設定には、`add_install_client` を使用して以下の作業を実行してください。次の手順の例も参照してください。

- DHCP を使用してインストールパラメータを設定している場合、例 14-6 を参照してください。
- インストールサーバーとクライアントが同じサブネット上にある場合、例 14-7 を参照してください。
- インストールサーバーとクライアントが同じサブネット上にない場合、かつ、DHCP を使用していない場合、例 14-8 を参照してください。
- x86 クライアントのインストールパラメータの設定に DHCP を使用する場合は、例 14-9 を参照してください。
- x86 ベースのシステムのインストール中に特定のシリアルポートに出力を表示したい場合は、例 14-10 を参照してください。
- x86 クライアントのインストール時に、特定のネットワークインタフェースを使用するように設定したい場合は、例 14-11 を参照してください。

このコマンドで使用できるオプションについての詳細は、`add_install_client` (1M) のマニュアルページを参照してください。



## ▼ add\_install\_client を実行してネットワークからインストールするシステムを追加する方法 (DVD)

ブートサーバーを使用する場合、インストールサーバーのインストールイメージを共有していること、および適切なデーモンを起動していることを確認します。「SPARC または x86 DVD メディアを使用して SPARC インストールサーバーを作成する方法」の手順 6 を参照してください。

1. インストールサーバーまたはブートサーバー上でスーパーユーザーになります。
2. NIS、NIS+、DNS、または LDAP のネームサービスを使用する場合、インストールするシステムについての次のような情報がネームサービスに追加されていることを確認します。

- ホスト名
- IP アドレス
- Ethernet アドレス

ネームサービスの詳細は、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

3. インストールサーバーの Solaris DVD イメージ上の **Tools** ディレクトリに移動します。

```
# cd /install_dir_path/Solaris_9/Tools
```

`install_dir_path` Tools ディレクトリのパスを指定します。

4. 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。

```
# ./add_install_client [-d] [-s install_server:install_dir_path] \  
[-c jumpstart_server:jumpstart_dir_path] [-p sysid_server:path] \  
[-t boot_image_path] [-b "boot-property=value"] \  
[-e ethernet_address] client_name platform_group
```

-d

クライアントが DHCP を使用してネットワークインストールパラメータを取得することを指定します。-d だけを使用した場合、add\_install\_client コマンドは、同じクラスに属するすべてのクライアントシステム (たとえばすべての SPARC クライアントマシン) のインストール情報を設定します。特定のクライアントのインストール情報を設定する場合は、-d と -e を併用します。

	<p>x86 クライアントの場合、このオプションを指定し、PXE ネットワークブートを使用してネットワークからシステムをブートしてください。</p> <p>DHCP を使用した特定のクラスのインストールの詳細は、76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」を参照してください。</p>
-s <i>install_server:install_dir_path</i>	<p>インストーラサーバーの名前とパスを指定します。</p> <ul style="list-style-type: none"> <li>■ <i>install_server</i> は、インストーラサーバーのホスト名です。</li> <li>■ <i>install_dir_path</i> は、Solaris DVD イメージへの絶対パスです。</li> </ul>
-c <i>jumpstart_server:jumpstart_dir_path</i>	<p>カスタム JumpStart インストールの JumpStart ディレクトリを指定します。<i>jumpstart_server</i> には JumpStart ディレクトリがあるサーバーのホスト名を指定します。<i>jumpstart_dir_path</i> は、JumpStart ディレクトリへの絶対パスです。</p>
-p <i>sysid_server:path</i>	<p>システムの構成情報を事前設定するための <i>sysidcfg</i> ファイルのパスを指定します。<i>sysid_server</i> は、このファイルを持っているサーバーの有効なホスト名または IP アドレスです。<i>path</i> は、<i>sysidcfg</i> ファイルを含むディレクトリの絶対パスです。</p>
-t <i>boot_image_path</i>	<p>Solaris 9 ネットインストールイメージ、CD、または DVD 上の Tools ディレクトリにあるブートイメージ以外のブートイメージを使用したい場合、別のブートイメージへのパスを指定します。</p>
-b " <i>boot-property=value</i> "	<p><b>x86 システムのみ:</b> ネットワークからクライアントをブートするとき使用するブートプロパティ変数の値を設定できます。<i>-b</i> オプションは、<i>-e</i> オプションと組み合わせて使用してください。</p> <p>ブートプロパティについては、<i>eeprom</i> (1M) のマニュアルページを参照してください。</p>

<code>-e ethernet_address</code>	インストールするクライアントの Ethernet アドレスを指定します。このオプションを指定すると、特定のクライアントのインストール情報を設定できません。
<code>client_name</code>	DHCP を使用した特定のクライアントのインストールの詳細は、76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」を参照してください。
<code>platform_group</code>	ネットワークからインストールするシステムのホスト名を指定します。この名前はインストールサーバーのホスト名ではありません。
	インストール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。詳細は、付録 C を参照してください。

**例 14-6 SPARC: DHCP を使用している場合の SPARC インストールサーバー上の SPARC インストールクライアントの追加方法 (DVD)**

次の例に、DHCP を使用してネットワーク上のインストールパラメータを設定しているときに、インストールクライアントを追加する方法を示します。インストールクライアントは、`basil` という名前の Ultra™ 5 システムです。ファイルシステム `/export/home/dvdsparc/Solaris_9/Tools` には、`add_install_client` コマンドが入っています。

DHCP を使用してネットワークインストール用のインストールパラメータを設定する方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP サービスを使用した Solaris ネットワークインストールのサポート (作業マップ)」を参照してください。

```
sparc_install_server# cd /export/home/dvdsparc/Solaris_9/Tools
sparc_install_server# ./add_install_client -d basil sun4u
```

**例 14-7 インストールサーバーと同じサブネット上にあるインストールクライアントの追加方法 (DVD)**

次の例に、インストールサーバーと同じサブネット上にあるインストールクライアントを追加する方法を示します。インストールクライアントは、`basil` という名前の Ultra 5 システムです。ファイルシステム `/export/home/dvdsparc` には `add_install_client` コマンドが入っています。

```
install_server# cd /export/home/dvdsparc/Solaris_9/Tools
install_server# ./add_install_client basil sun4u
```

#### 例 14-8 ブートサーバーへのインストールクライアントの追加方法 (DVD)

次の例に、ブートサーバーにインストールクライアントを追加する方法を示します。インストールクライアントは、`rose` という名前の Ultra 5 システムです。次のコマンドはブートサーバー上で実行します。`-s` オプションには、`/export/home/dvdsparc` に Solaris DVD (SPARC 版) イメージを持っている `rosemary` という名前のインストールサーバーを指定しています。

```
boot_server# cd /export/home/dvdsparc/Solaris_9/Tools
boot_server# ./add_install_client -s rosemary:/export/home/dvdsparc rose sun4u
```

#### 例 14-9 x86: DHCP を使用している場合の x86 インストールサーバー上の x86 インストールクライアントの追加方法 (DVD)

次の例に、DHCP を使用してネットワーク上のインストールパラメータを設定しているときに、x86 インストールクライアントを追加する方法を示します。`-d` オプションを指定すると、クライアントのネットワークインストールパラメータを構成する際に、DHCP プロトコルが使用されます。PXE ネットワークブートを使用する場合には、DHCP プロトコルを使用する必要があります。DHCP クラス名 `SUNW.i86pc` は、このコマンドが単一のクライアントだけでなく、ネットワーク上のすべての Solaris x86 ブートクライアントに適用されることを意味します。`-s` オプションには、`rosemary` という名前のインストールサーバーからクライアントをインストールすることを指定します。このサーバーは、`/export/boot/dvdx86` に Solaris DVD (x86 版) イメージを保持しています。

DHCP を使用してネットワークインストール用のインストールパラメータを設定する方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP サービスを使用した Solaris ネットワークインストールのサポート (作業マップ)」を参照してください。

```
x86_install_server# cd /export/boot/dvdx86/Solaris_9/Tools
x86_install_server# ./add_install_client -d -s rosemary:/export/home/dvdx86 \
SUNW.i86pc i86pc
```

#### 例 14-10 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (DVD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するシリアルコンソールを指定する方法を示しています。この例では、インストールクライアントは次のように設定されます。

- `-d` オプションが指定されているので、クライアントは DHCP を使用してインストールパラメータを設定します。
- `-e` オプションが指定されているので、Ethernet アドレスが `00:07:e9:04:4a:bf` と一致するクライアントに対してのみ、インストールが行われます。
- 1 番目と 2 番目の `-b` オプションにより、インストールプログラムの入出力デバイスがシリアルポート `tttya` に設定されます。

```
install_server# cd /export/boot/dvdx86/Solaris_9/Tools
install_server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
```

例 14-10 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (DVD)  
(続き)

```
-b "input-device=ttya" -b "output-device=ttya" i86pc
```

-b オプションで使用可能なブートプロパティ変数の詳細情報は、eeprom(1M) のマニュアルページを参照してください。

例 14-11 x86: ネットワークインストール時に使用するブートデバイスの指定 (DVD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するブートデバイスを指定する方法を示しています。インストールクライアントの設定時にブートデバイスを指定すると、インストール時、Device Configuration Assistant (デバイス構成用補助) はこの情報の入力プロンプトを表示しません。

この例では、インストールクライアントは次のように設定されます。

- -d オプションが指定されているので、クライアントは DHCP を使用してインストールパラメータを設定します。
- -e オプションが指定されているので、Ethernet アドレスが 00:07:e9:04:4a:bf と一致するクライアントに対してのみ、インストールが行われます。
- 1 番目と 2 番目の -b オプションにより、インストールプログラムの出力デバイスがシリアルポート ttya に設定されます。
- 3 番目の -b オプションにより、インストールプログラムがインストール時に特定のブートデバイスを使用するように設定されます。

---

注 - ブートデバイスパスの値は、使用するハードウェアによって異なります。

---

- i86pc プラットフォーム名から、クライアントが x86 システムであることがわかります。

```
install server# cd /export/boot/dvdx86/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

-b オプションで使用可能なブートプロパティ変数の詳細情報は、eeprom(1M) のマニュアルページを参照してください。



## 第 15 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)

---

この章では、ネットワーク上で Solaris ソフトウェアをインストールする準備として CD メディアを使用してネットワークとシステムを設定する方法を説明します。ネットワークインストールを使用すると、Solaris 9 ディスクイメージにアクセスできるシステム (インストールサーバーと呼ばれる) から、ネットワーク上のほかのシステムに Solaris ソフトウェアをインストールできます。ユーザーは、まず CD メディアの内容をインストールサーバーのハードディスクにコピーする必要があります。これで、任意の Solaris インストール方法を使用して Solaris ソフトウェアをネットワーク上でインストールできます。この章の内容は次のとおりです。

- 136 ページの「作業マップ: CD メディアを使用してネットワーク上でインストールするための準備」
- 137 ページの「CD メディアによる SPARC インストールサーバーの作成」
- 142 ページの「CD メディアによる x86 インストールサーバーの作成」
- 147 ページの「クロスプラットフォームの CD メディアを使用したインストールサーバーの作成」
- 159 ページの「CD イメージを用いたサブネット上でのブートサーバーの作成」
- 161 ページの「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を追加する」

## 作業マップ: CD メディアを使用して ネットワーク上でインストールするための 準備

表 15-1 作業マップ: CD メディアによるインストールサーバーの設定

作業	説明	参照先
インストールサーバーを作成する	<p><code>setup_install_server(1M)</code> コマンドを使用して Solaris SOFTWARE 1 of 2 CD をインストールサーバーのハードディスクにコピーします。</p> <p>次に、<code>add_to_install_server(1M)</code> コマンドを使用して Solaris SOFTWARE 2 of 2 CD と Solaris LANGUAGES CD をインストールサーバーのハードディスクに追加します。</p> <p>さらに、<code>modify_install_server(1M)</code> コマンドを使用してネットワークインストールイメージに Solaris Web Start ユーザーインタフェースソフトウェアを追加します。</p>	<ul style="list-style-type: none"> <li>■ 137 ページの「CD メディアによる SPARC インストールサーバーの作成」</li> <li>■ 142 ページの「x86: x86 CD メディアを使用して x86 インストールサーバーを作成する方法」</li> <li>■ 147 ページの「クロスプラットフォームの CD メディアを使用したインストールサーバーの作成」</li> </ul>
(省略可能) ブートサーバーを作成する	<p>インストールサーバーと異なるサブネット上にあるシステムをネットワークからインストールする場合、システムをブートするためのブートサーバーをサブネット上に作成する必要があります。DHCP (Dynamic Host Configuration Protocol) を使用する場合は、ブートサーバーは必要ありません。</p>	159 ページの「CD イメージを用いたサブネット上でのブートサーバーの作成」
ネットワークインストールするシステムを追加する	<p><code>add_install_client</code> コマンドを使用して、ネットワークからインストールする各システムを設定します。インストールする各システムには、ネットワーク上にあるインストールサーバー、ブートサーバー、および構成情報を認識させる必要があります。</p>	161 ページの「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を追加する」



---

## CD メディアによる SPARC インストールサーバーの作成

インストールサーバーには、ネットワーク上でシステムをインストールするために必要なインストールイメージが含まれます。Solaris ソフトウェアをネットワークからインストールするためには、インストールサーバーを作成する必要があります。必ずしも個別のブートサーバーを設定する必要はありません。

- DHCP を使用してインストールパラメータを設定する場合や、インストールサーバーとクライアントが同じサブネット上に存在する場合は、個別のブートサーバーは不要です。
- インストールサーバーとクライアントが同じサブネット上に存在せず、DHCP を使用していない場合は、サブネットごとに異なるブートサーバーを作成する必要があります。インストールサーバーはサブネットごとに作成できます。しかし、インストールサーバーはより多くのディスク容量を必要とします。

### ▼ SPARC: SPARC CD メディアを使用して SPARC インストールサーバーを作成する方法

ここでは、SPARC CD メディアを使用して SPARC インストールサーバーを作成する方法を説明します。

インストールサーバーとは異なるプラットフォームのメディアを使用してインストールサーバーを作成する場合 (SPARC システムで x86 CD メディアを使用する場合などは、147 ページの「クロスプラットフォームの CD メディアを使用したインストールサーバーの作成」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を実行していると仮定しています。ボリュームマネージャを使用せずにメディアを管理する方法については、『Solaris のシステム管理 (基本編)』を参照してください。

---

1. インストールサーバーとして使用するシステムでスーパーユーザーになります。  
このシステムには CD-ROM ドライブが必要です。さらに、システムは、このサイトのネットワークに接続されネームサービスに登録されている必要があります。ネームサービスを使用する場合は、システムがすでに NIS、NIS+、DNS、LDAP のいずれかのネームサービスに登録されていなければなりません。ネームサービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。
2. **Solaris SOFTWARE 1 of 2 CD** をシステムのドライブに挿入します。

3. CD イメージ用のディレクトリを作成します。

```
# mkdir -p install_dir_path
```

*install\_dir\_path* CD イメージをコピーするディレクトリを指定します。

4. マウントされたディスクの **Tools** ディレクトリに移動します。

```
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
```

上の例では、**cdrom0** は、Solaris オペレーティング環境の CD メディアを含むドライブへのパスです。

5. ドライブ内のイメージをインストールサーバーのハードディスクにコピーします。

```
# ./setup_install_server install_dir_path
```

*install\_dir\_path* CD イメージをコピーするディレクトリを指定します。

---

注 - `setup_install_server` コマンドは、Solaris SOFTWARE ディスクイメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、`df -k1` コマンドを使用します。

---

6. 他のシステムがインストールサーバーをマウントできるように設定する必要があるかどうかを判断します。

- インストールサーバーとクライアント (インストールしているシステム) が同じサブネット上にある場合、あるいは、DHCP を使用している場合、ブートサーバーを作成する必要はありません。手順 10 へ進みます。
- インストールサーバーがインストールするシステムと同じサブネット上に存在せず、DHCP を使用していない場合は、以下の手順を実行してください。

7. インストールサーバーのインストールイメージのパスが適切に共有されているかを確認します。

```
# share | grep install_dir_path
```

*install\_dir\_path* ブートサーバーのインストールイメージのパスを指定します。

- インストールサーバーのディレクトリのパスが表示されてオプションに `anon=0` と示された場合は、手順 10 へ進みます。
- インストールサーバーのディレクトリのパスが表示されないか、オプション内に `anon=0` と示されない場合は、次の作業へ進みます。

8. 次のエントリを `/etc/dfs/dfstab` ファイルに追加し、ブートサーバーからインストールサーバーが利用できるように設定します。

```
share -F nfs -o ro,anon=0 -d "install server directory" install_dir_path
```

*install\_dir\_path* ブートサーバーのインストールイメージのパスを指定します。

9. インストールサーバーのディレクトリパスが正しく共有されているか確認します。

```
# ps -ef | grep nfsd
```

■ `nfsd` デーモンが稼働していない場合は、このデーモンを起動してください。

```
# /etc/init.d/nfs.server start
```

■ `nfsd` デーモンがすでに稼働している場合は、インストールサーバーを共有してください。

```
# shareall
```

10. ルート (/) ディレクトリへ移動します。

```
# cd /
```

11. Solaris SOFTWARE 1 of 2 CD を取り出します。

12. Solaris SOFTWARE 2 of 2 CD をシステムの CD-ROM ドライブに挿入します。

13. マウントされた CD の `Tools` ディレクトリに移動します。

```
# cd /cdrom/cdrom0/Solaris_9/Tools
```

14. CD-ROM ドライブ内の CD をインストールサーバーのハードディスクにコピーします。

```
# ./add_to_install_server install_dir_path
```

`install_dir_path` CD イメージをコピーするディレクトリを指定します。

15. ルート (/) ディレクトリへ移動します。

```
# cd /
```

16. Solaris SOFTWARE 2 of 2 CD を取り出します。

17. Solaris LANGUAGES CD をシステムの CD-ROM ドライブに挿入します。

18. マウントされた CD の `Tools` ディレクトリに移動します。

```
# cd /cdrom/cdrom0/Tools
```

19. CD-ROM ドライブ内の CD をインストールサーバーのハードディスクにコピーします。

```
# ./add_to_install_server install_dir_path
```

`install_dir_path` CD イメージをコピーするディレクトリを指定します。

20. ルート (/) ディレクトリへ移動します。

```
# cd /
```

21. ユーザーが **Solaris Web Start** を使用してシステムをブートし、ネットワーク上で **Solaris 9** ソフトウェアをインストールできるようにするか、決定します。

- しない場合は、Solaris LANGUAGES CD を取り出して手順 25 へ進みます。
- する場合は、Solaris LANGUAGES CD を取り出して続けます。

22. **Solaris INSTALLATION CD** をシステムの **CD-ROM** ドライブに挿入します。

23. マウントされた **CD** の **modify\_install\_server** を含むディレクトリに移動します。

```
# cd /cdrom/cdrom0/s0
```

24. **Solaris Web Start** インタフェースソフトウェアをインストールサーバーにコピーします。

```
# ./modify_install_server -p install_dir_path installer_miniroot_path
```

**-p** 既存イメージのミニルートを *install\_dir\_path/Solaris\_9/Tools/Boot.orig* に保存します。

*install\_dir\_path* Solaris Web Start インタフェースをコピーするディレクトリを指定します。

*installer\_miniroot\_path* Solaris Web Start インタフェースのコピー元となる CD 上のディレクトリを指定します。

25. **setup\_install\_server** によって作成されたネットインストールイメージ上のミニルート (*/install\_dir\_path/Solaris\_9/Tools/Boot*) にあるファイルにパッチを適用するか決定します。ブートイメージに問題がある場合は、ファイルにパッチを適用する必要があります。

- パッチを適用しない場合は、そのまま続けます。
- 適用する場合は、`patchadd -c` コマンドを使用して、ミニルートにあるファイルにパッチを適用します。



---

注意 `-patchadd -c` を使用する前には必ず、パッチの README を読むか、ご購入先におたずねください。

---

26. ブートサーバーを作成する必要があるかどうかを判断します。

- DHCP を使用しているか、あるいはインストール対象のシステムと同じサブネット上にインストールサーバーが存在する場合は、ブートサーバーを作成する必要はありません。161 ページの「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を追加する」へ進みます。
- DHCP を使用しておらず、インストールサーバーとクライアントが別のサブネットに存在する場合は、ブートサーバーを作成する必要があります。159 ページの「CD イメージを用いたサブネット上でのブートサーバーの作成」

へ進みます。

例 15-1 SPARC: SPARC CD メディアによる SPARC インストールサーバーの作成

次の例は、インストールサーバーの `/export/home/cdsparc` ディレクトリに次の CD をコピーしてインストールサーバーを作成する方法を示しています。

- Solaris SOFTWARE 1 of 2 CD (SPARC 版)
- Solaris SOFTWARE 2 of 2 CD (SPARC 版)
- Solaris LANGUAGES CD (SPARC 版)

```
# mkdir -p /export/home/cdsparc
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
# ./setup_install_server /export/home/cdsparc
```

- 個別のブートサーバーがある場合は、次の手順を追加します。

1. 次のパスを `/etc/dfs/dfstab` ファイルに追加します。

```
share -F nfs -o ro,anon=0 -d "install server directory" \
/export/home/cdsparc
```

2. `nfsd` デーモンが稼働しているかどうかを確認します。`nfsd` デーモンが稼働していない場合、このデーモンを起動して共有します。

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
```

3. 引き続き、次の手順を実行します。

- ブートサーバーが不要であるか、個別のブートサーバーを設定する手順を完了している場合は、作業を続行します。

```
# cd /
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./add_to_install_server /export/home/cdsparc
# cd /
# cd /cdrom/cdrom0/Tools
# ./add_to_install_server /export/home/cdsparc
# cd /
# cd /cdrom/cdrom0/s0
# ./modify_install_server -p /export/home/cdsparc /cdrom/cdrom0/s0
```

この例では、各コマンドを実行する前に各 CD がドライブに挿入され、自動的にマウントされるものとします。各コマンドの実行が終了すると、CD が取り出されます。

---

## CD メディアによる x86 インストールサーバーの作成

インストールサーバーには、ネットワーク上でシステムをインストールするために必要なインストールイメージが含まれます。Solaris ソフトウェアをネットワークからインストールするためには、インストールサーバーを作成する必要があります。必ずしも個別のブートサーバーを設定する必要はありません。

- DHCP を使用してインストールパラメータを設定する場合や、インストールサーバーとクライアントが同じサブネット上に存在する場合は、個別のブートサーバーは不要です。
- インストールサーバーとクライアントが同じサブネット上に存在せず、DHCP を使用していない場合は、サブネットごとに異なるブートサーバーを作成する必要があります。インストールサーバーはサブネットごとに作成できます。しかし、インストールサーバーはより多くのディスク容量を必要とします。

### ▼ x86: x86 CD メディアを使用して x86 インストールサーバーを作成する方法

ここでは、x86 CD メディアを使用して x86 インストールサーバーを作成する方法を説明します。

インストールサーバーとは異なるプラットフォームのメディアを使用してインストールサーバーを作成する場合 (x86 システムで SPARC CD メディアを使用する場合など) は、147 ページの「クロスプラットフォームの CD メディアを使用したインストールサーバーの作成」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を実行していると仮定しています。ボリュームマネージャを使用せずにメディアを管理する方法については、『Solaris のシステム管理 (基本編)』を参照してください。

---

1. インストールサーバーとして使用するシステムでスーパーユーザーになります。  
このシステムには CD-ROM ドライブが必要です。さらに、システムは、このサイトのネットワークに接続され名前サービスに登録されている必要があります。名前サービスを使用する場合は、システムがすでに NIS、NIS+、DNS、LDAP のいずれかの名前サービスに登録されていなければなりません。名前サービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。

2. Solaris SOFTWARE 1 of 2 CD をシステムのドライブに挿入します。

3. CD イメージ用のディレクトリを作成します。

```
# mkdir -p install_dir_path
```

*install\_dir\_path* CD イメージをコピーするディレクトリを指定します。

4. マウントされたディスクの **Tools** ディレクトリに移動します。

```
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
```

上の例では、**cdrom0** は、Solaris オペレーティング環境の CD メディアを含むドライブへのパスです。

5. ドライブ内のイメージをインストールサーバーのハードディスクにコピーします。

```
# ./setup_install_server install_dir_path
```

*install\_dir\_path* CD イメージをコピーするディレクトリを指定します。

---

注 - `setup_install_server` コマンドは、Solaris SOFTWARE ディスクイメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、`df -k1` コマンドを使用します。

---

6. 他のシステムがインストールサーバーをマウントできるように設定する必要があるかどうかを判断します。

- インストールサーバーとクライアント (インストールしているシステム) が同じサブネット上にある場合、あるいは、DHCP を使用している場合、ブートサーバーを作成する必要はありません。手順 10へ進みます。

- インストールサーバーがインストールするシステムと同じサブネット上に存在せず、DHCP を使用していない場合は、以下の手順を実行してください。

7. インストールサーバーのインストールイメージのパスが適切に共有されているかを確認します。

```
# share | grep install_dir_path
```

*install\_dir\_path* ブートサーバーのインストールイメージのパスを指定します。

- インストールサーバーのディレクトリへのパスが表示されて、`anon=0` がオプションに表示される場合、手順 10 に進みます。

- インストールサーバーのディレクトリのパスが表示されないか、オプション内に `anon=0` と示されない場合は、次の作業へ進みます。

8. 次のエントリを `/etc/dfs/dfstab` ファイルに追加し、ブートサーバーからインストールサーバーが利用できるように設定します。

```
share -F nfs -o ro,anon=0 -d "install server directory" install_dir_path
```

`install_dir_path` ブートサーバーのインストールイメージのパスを指定します。

9. インストールサーバーのディレクトリパスが正しく共有されているか確認します。

```
# ps -ef | grep nfsd
```

- `nfsd` デーモンが稼働していない場合は、このデーモンを起動してください。

```
# /etc/init.d/nfs.server start
```

- `nfsd` デーモンがすでに稼働している場合は、インストールサーバーを共有してください。

```
# shareall
```

10. ルート (/) ディレクトリへ移動します。

```
# cd /
```

11. **Solaris SOFTWARE 1 of 2 CD** を取り出します。

12. **Solaris SOFTWARE 2 of 2 CD** をシステムの **CD-ROM** ドライブに挿入します。

13. マウントされた **CD** の **Tools** ディレクトリに移動します。

```
# cd /cdrom/cdrom0/Solaris_9/Tools
```

14. **CD-ROM** ドライブ内の **CD** をインストールサーバーのハードディスクにコピーします。

```
# ./add_to_install_server install_dir_path
```

`install_dir_path` CD イメージをコピーするディレクトリを指定します。

15. ルート (/) ディレクトリへ移動します。

```
# cd /
```

16. **Solaris SOFTWARE 2 of 2 CD** を取り出します。

17. **Solaris LANGUAGES CD** をシステムの **CD-ROM** ドライブに挿入します。

18. マウントされた **CD** の **Tools** ディレクトリに移動します。

```
# cd /cdrom/cdrom0/Tools
```

19. **CD-ROM** ドライブ内の **CD** をインストールサーバーのハードディスクにコピーします。

```
# ./add_to_install_server install_dir_path
```

`install_dir_path` CD イメージをコピーするディレクトリを指定します。

20. ルート (/) ディレクトリへ移動します。

```
# cd /
```



21. ユーザーが **Solaris Web Start** を使用してシステムをブートし、ネットワーク上で **Solaris 9** ソフトウェアをインストールできるようにするか、決定します。

- しない場合は、Solaris LANGUAGES CD を取り出して手順 25 へ進みます。
- する場合は、Solaris LANGUAGES CD を取り出して続きます。

22. **Solaris INSTALLATIONCD** をシステムの **CD-ROM** ドライブに挿入します。

23. マウントされた **CD** の **modify\_install\_server** を含むディレクトリに移動します。

```
# cd /cdrom/cdrom0/s2
```

24. **Solaris Web Start** インタフェースソフトウェアをインストールサーバーにコピーします。

```
# ./modify_install_server -p install_dir_path installer_miniroot_path
```

**-p** 既存イメージのミニルートを *install\_dir\_path/Solaris\_9/Tools/Boot.orig* に保存します。

*install\_dir\_path* Solaris Web Start インタフェースをコピーするディレクトリを指定します。

*installer\_miniroot\_path* Solaris Web Start インタフェースのコピー元となる CD 上のディレクトリを指定します。

25. **setup\_install\_server** によって作成されたネットインストールイメージ上のミニルート (*/install\_dir\_path/Solaris\_9/Tools/Boot*) にあるファイルにパッチを適用するか決定します。ブートイメージに問題がある場合は、ファイルにパッチを適用する必要があります。

- パッチを適用しない場合は、そのまま続きます。
- 適用する場合は、`patchadd -c` コマンドを使用して、ミニルートにあるファイルにパッチを適用します。



---

注意 `-patchadd -c` を使用する前には必ず、パッチの README を読むか、ご購入先におたずねください。

---

26. ブートサーバーを作成する必要があるかどうかを判断します。

- DHCP を使用しているか、あるいはインストール対象のシステムと同じサブネット上にインストールサーバーが存在する場合は、ブートサーバーを作成する必要はありません。161 ページの「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を追加する」へ進みます。
- DHCP を使用しておらず、インストールサーバーとクライアントが別のサブネットに存在する場合は、ブートサーバーを作成する必要があります。159 ページの「CD イメージを用いたサブネット上でのブートサーバーの作成」

へ進みます。

**例 15-2** x86: x86 CD メディアによる x86 インストールサーバーの作成

次の例は、インストールサーバーの `/export/home/cdx86` ディレクトリに次の CD をコピーしてインストールサーバーを作成する方法を示しています。

- Solaris SOFTWARE 1 of 2 CD (x86 版)
- Solaris SOFTWARE 2 of 2 CD (x86 版)
- Solaris LANGUAGES CD (x86 版)

```
# mkdir -p /export/home/cdx86
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
# ./setup_install_server /export/home/cdx86
```

- 個別のブートサーバーがある場合は、次の手順を追加します。

1. 次のパスを `/etc/dfs/dfstab` ファイルに追加します。

```
share -F nfs -o ro,anon=0 -d "install server directory" \
/export/home/cdx86
```

2. `nfsd` デーモンが稼働しているかどうかを確認します。 `nfsd` デーモンが稼働していない場合、このデーモンを起動して共有します。

```
# ps -ef | grep nfsd
# /etc/init.d/nfs.server start
# shareall
```

3. 引き続き、次の手順を実行します。

- ブートサーバーが不要であるか、個別のブートサーバーを設定する手順を完了している場合は、作業を続行します。

```
# cd /
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/s2
# ./modify_install_server -p /export/home/cdx86 /cdrom/cdrom0/s2
```

この例では、各コマンドを実行する前に各 CD がドライブに挿入され、自動的にマウントされるものとします。各コマンドの実行が終了すると、CD が取り出されます。

---

## クロスプラットフォームの CD メディアを使用したインストールサーバーの作成

インストールサーバーとは異なるプラットフォームの CD を使用する必要がある場合、インストールサーバー上で CD を読み取ることができません。リモートシステムを使用して CD を読み取る必要があります。たとえば、SPARC インストールサーバーを設定している状態で x86 CD メディアを使用する場合、リモート x86 システムを使用して CD を読み取る必要があります。

### ▼ x86 CD メディアを使用して SPARC システムに x86 インストールサーバーを作成する方法

ここでは、x86 CD メディアを使用して SPARC システム上に x86 インストールサーバーを作成する手順を説明します。以下が必要になります。

- SPARC システム
- CD-ROM ドライブを備えた x86 システム
- リモート x86 システム用の CD セット
  - Solaris SOFTWARE 1 of 2 CD (x86 版)
  - Solaris SOFTWARE 2 of 2 CD (x86 版)
  - Solaris LANGUAGES CD (x86 版)
  - Solaris INSTALLATION CD (x86 版)

---

注 - この手順では、システムでボリュームマネージャを実行していると仮定しています。ボリュームマネージャを使用せずにメディアを管理する方法については、『Solaris のシステム管理 (基本編)』を参照してください。

---

コマンド入力例中の *SPARC-system* はインストールサーバーとなる SPARC システムを、*remote-x86-system* は x86 CD メディアを使用するリモート x86 システムを表します。

#### 1. リモート x86 システム上で、スーパーユーザーになります。

このシステムには CD-ROM ドライブが必要です。さらに、システムは、このサイトのネットワークに接続されネームサービスに登録されている必要があります。ネームサービスを使用する場合、システムのネームサービスは、NIS、NIS+、DNS、LDAP のいずれかでなければなりません。ネームサービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。

2. リモート **x86** システム上で、システムのドライブに **Solaris SOFTWARE 1 of 2 CD (x86 版)** を挿入します。

3. リモート **x86** システム上で、次のエントリを **/etc/dfs/dfstab** ファイルに追加します。

```
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
```

4. リモート **x86** システム上で、**NFS** デーモンを起動します。

```
remote-x86-system# /etc/init.d/nfs.server start
```

5. リモート **x86** システム上で **share** コマンドを実行し、他のシステムが **CD** を利用可能であることを確認します。

```
remote-x86-system# share
- /cdrom/sol_9_x86/s0 ro,anon=0 " "
- /cdrom/sol_9_x86/s2 ro,anon=0 " "
```

上のサンプル出力では、**sol\_9\_x86** は、**x86 版 Solaris 9 オペレーティング環境**を指します。このテキスト文字列は、**Solaris オペレーティング環境のバージョン**によって異なります。

6. **x86** インストールサーバーとして使用する **SPARC** システム上で、スーパーユーザーになります。

7. **SPARC** システム上で **x86 CD** にアクセスする際のマウントポイントとなる **2 つのディレクトリ (ミニルート用と製品用)** を作成します。

```
SPARC-system# mkdir directory_name_s0
```

```
SPARC-system# mkdir directory_name_s2
```

**directory\_name\_s0** スライス 0 からのミニルート用のマウントポイントとなるディレクトリの名前です。

**directory\_name\_s2** スライス 2 からの製品用のマウントポイントとなるディレクトリの名前です。

8. **CD** がリモート **x86** システム上で適切にエクスポートされていることを確認します。

```
SPARC-system# showmount -e remote-x86-system
export list for remote-x86-system:
/cdrom/sol_9_x86/s0 (everyone)
/cdrom/sol_9_x86/s2 (everyone)
```

9. **SPARC** システム上で、リモート **x86** システムの **CD** イメージをマウントします。

```
SPARC-system# mount remote_x86_system_name:/cdrom/sol_9_x86/s0 directory_name_s0
```

```
SPARC-system# mount remote_x86_system_name:/cdrom/sol_9_x86/s2 directory_name_s2
```

10. SPARC システム上で、マウントされたディスクの **Tools** ディレクトリに移動します。

```
SPARC-system# cd /directory_name_s2/Solaris_9/Tools
```

11. SPARC システム上で、インストールサーバーのハードディスク内の、**setup\_install\_server** コマンドを使用して作成したディレクトリにドライブ内のディスクをコピーします。

```
SPARC-system# ./setup_install_server -t directory_name_s0 install_dir_path
```

-t Solaris SOFTWARE 2 of 2 CD の **Tools** ディレクトリ上のブートイメージ以外を使用する場合は、そのブートイメージのパスを指定します。

directory\_name\_s0 スライス 0 からのミニルート用のマウントポイントとなるディレクトリの名前です。

install\_dir\_path ディスクイメージをコピーするディレクトリを指定します。空のディレクトリを指定してください。

---

注 - **setup\_install\_server** コマンドは、Solaris SOFTWARE ディスクイメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、**df -k1** コマンドを使用します。

---

12. SPARC システム上で、トップディレクトリに移動します。

```
SPARC-system# cd /
```

13. SPARC システム上で、両方のディレクトリのマウントを解除します。

```
SPARC-system# umount directory_name_s0
```

```
SPARC-system# umount directory_name_s2
```

14. x86 システム上で、両方の **CD-ROM** スライスの共有を解除します。

```
remote x86 system# unshare /cdrom/sol_9_x86/s0
```

```
remote x86 system# unshare /cdrom/sol_9_x86/s2
```

15. x86 システム上で、**Solaris SOFTWARE 1 of 2 CD (x86 版)** を取り出します。

16. **Solaris SOFTWARE 2 of 2 CD (x86 版)** を SPARC システムの **CD-ROM** ドライブに挿入します。

17. SPARC システム上で、マウントされた **CD** の **Tools** ディレクトリに移動します。

```
SPARC-system# cd /cdrom/cdrom0/Solaris_9/Tools
```

18. **SPARC** システム上で、**CD** をインストールサーバーのハードディスクにコピーします。

```
SPARC-system# ./add_to_install_server install_dir_path
```

*install\_dir\_path* CD イメージをコピーするディレクトリを指定します。

19. **Solaris SOFTWARE 2 of 2 CD (x86 版)** を取り出します。

20. **Solaris LANGUAGES CD (x86 版)** を **SPARC** システムの **CD-ROM** ドライブに挿入し、**CD** をマウントします。

21. **SPARC** システム上で、マウントされた **CD** の **Tools** ディレクトリに移動します。

```
SPARC-system# cd /cdrom/cdrom0/Tools
```

22. **SPARC** システム上で、**CD** をインストールサーバーのハードディスクにコピーします。

```
SPARC-system# ./add_to_install_server install_dir_path
```

*install\_dir\_path* CD イメージをコピーするディレクトリを指定します。

23. ユーザーが **Solaris Web Start** を使用してシステムをブートし、ネットワーク上で **Solaris 9** ソフトウェアをインストールできるようにするか、決定します。

- しない場合は、**Solaris LANGUAGES CD (x86 版)** を取り出して手順 31 へ進みます。
- する場合は、**Solaris LANGUAGES CD (x86 版)** を取り出して次の作業へ進みます。

24. リモート **x86** システムの **CD-ROM** ドライブに **Solaris INSTALLATION CD (x86 版)** を挿入します。

25. リモート **x86** システム上で、次のエントリを **/etc/dfs/dfstab** ファイルに追加します。

```
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
```

26. リモート **x86** システム上で、**NFS** デーモンを起動します。

```
remote-x86-system# /etc/init.d/nfs.server start
```

27. リモート **x86** システム上で、**CD** が他のシステムから利用可能であることを確認します。

```
remote-x86-system# share
- /cdrom/multi_sol_9_x86/s0 ro,anon=0 " "
- /cdrom/multi_sol_9_x86/s2 ro,anon=0 " "
```

28. **SPARC** システム上で、**CD** 上のスライスをマウントします。

```
SPARC-system# mount remote_x86_system_name:/cdrom/multi_sol_9_x86/s0 directory_name_s0
SPARC-system# mount remote_x86_system_name:/cdrom/multi_sol_9_x86/s2 directory_name_s2
```

29. SPARC システム上で、マウントされた CD の `modify_install_server` を含むディレクトリに移動します。

```
SPARC-system# cd directory_name_s2
```

30. Solaris Web Start インタフェースソフトウェアをインストールサーバーにコピーします。

```
SPARC-system# ./modify_install_server -p install_dir_path directory_name_s0
```

-p 既存イメージのミニルートを  
`install_dir_path/Solaris_9/Tools/Boot.orig` に保存します。

`install_dir_path` Solaris Web Start インタフェースをコピーするディレクトリを指定します。

`directory_name_s0` スライス 0 からのミニルート用のマウントポイントとなるディレクトリの名前です。

31. `setup_install_server` によって作成されたネットインストールイメージ上のミニルート (`Solaris_9/Tools/Boot`) にあるファイルにパッチを適用するか決定します。

- 適用しない場合は、次の手順に進みます。
- 適用する場合は、`patchadd -c` コマンドを使用して、ミニルートにあるファイルにパッチを適用します。



---

注意 - `patchadd -c` を使用する前には必ず、パッチの README を読むか、ご購入先におたずねください。

---

32. ブートサーバーを作成する必要があるかどうかを判断します。

- インストールサーバーとクライアント (インストールしているシステム) が同じサブネット上にある場合、あるいは、DHCP を使用している場合、ブートサーバーを作成する必要はありません。161 ページの「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を追加する」を参照してください。
- インストールサーバーがインストールするシステムと同じサブネット上に存在せず、DHCP を使用していない場合は、ブートサーバーを作成する必要があります。ブートサーバーを作成する方法については、159 ページの「CD イメージを使用してサブネット上にブートサーバーを作成する方法」を参照してください。

例 15-3 x86 CD メディアを使用して SPARC システムに x86 インストールサーバーを作成する

次の例では、`rosemary` という名前の SPARC システム上に x86 インストールサーバーを作成する方法を示します。次の x86 CD が、`tadpole` という名前のリモート x86 システムから SPARC インストールサーバーの `/export/home/cdx86` ディレクトリにコピーされます。

- Solaris SOFTWARE 1 of 2 CD (x86 版)
- Solaris SOFTWARE 2 of 2 CD (x86 版)
- Solaris LANGUAGES CD (x86 版)
- Solaris INSTALLATION (MULTILINGUAL) CD (x86 版)

リモート x86 システムで、次のように入力します。

```
tadpole (remote-x86-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
tadpole (remote-x86-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
tadpole (remote-x86-system)# /etc/init.d/nfs.server start
```

SPARC システムで、次のように入力します。

```
rosemary (SPARC-system)# mkdir /x86S0
rosemary (SPARC-system)# mkdir /x86S2
rosemary (SPARC-system)# mount tadpole:/cdrom/sol_9_x86/s0 /x86S0
rosemary (SPARC-system)# mount tadpole:/cdrom/sol_9_x86/s0 /x86S2
rosemary (SPARC-system)# cd /x86S2/Solaris_9/Tools
rosemary (SPARC-system)# ./setup_install_server -t /x86S0 /export/home/cdx86
rosemary (SPARC-system)# cd /
rosemary (SPARC-system)# umount /x86S0
rosemary (SPARC-system)# umount /x86S2
```

```
tadpole (remote-x86-system) unshare /cdrom/cdrom0/s0
tadpole (remote-x86-system) unshare /cdrom/cdrom0/s2
```

```
rosemary (SPARC-system)# cd /cdrom/cdrom0/Solaris_9/Tools
rosemary (SPARC-system)# ./add_to_install_server /export/home/cdx86
rosemary (SPARC-system)# cd /cdrom/cdrom0/Tools
rosemary (SPARC-system)# ./add_to_install_server /export/home/cdx86
```

リモート x86 システムで、次のように入力します。

```
tadpole (remote-x86-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
tadpole (remote-x86-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s2
tadpole (remote-x86-system)# /etc/init.d/nfs.server start
```

SPARC システムで、次のように入力します。

```
rosemary (SPARC-system)# mount remote_x86_system_name:/cdrom/sol_9_x86/x86S0
rosemary (SPARC-system)# mount remote_x86_system_name:/cdrom/sol_9_x86/x86S2
rosemary (SPARC-system)# cd /x86S2
rosemary (SPARC-system)# ./modify_install_server -p /export/home/cdx86 /x86S0
```

この例では、各コマンドを実行する前に各 CD がドライブに挿入され、自動的にマウントされるものとします。各コマンドの実行が終了すると、CD が取り出されます。



## ▼ SPARC CD メディアを使用して x86 システムに SPARC インストールサーバーを作成する方法

ここでは、SPARC CD メディアを使用して x86 システム上に SPARC インストールサーバーを作成する手順を説明します。以下が必要になります。

- x86 システム
- CD-ROM ドライブを備えた SPARC システム
- リモート SPARC システム用の CD セット
  - Solaris SOFTWARE 1 of 2 CD (SPARC 版)
  - Solaris SOFTWARE 2 of 2 CD (SPARC 版)
  - Solaris LANGUAGES CD (SPARC 版)
  - Solaris INSTALLATION CD (SPARC 版)

---

注 - この手順では、システムでボリュームマネージャを実行していると仮定しています。ボリュームマネージャを使用せずにメディアを管理する方法については、『Solaris のシステム管理 (基本編)』を参照してください。

---

コマンド入力例中の *x86-system* はインストールサーバーとなる x86 システムを、*remote SPARC system* は SPARC CD メディアを使用するリモート SPARC システムを表します。

1. リモート **SPARC** システム上で、スーパーユーザーになります。  
このシステムには CD-ROM ドライブが必要です。さらに、システムは、このサイトのネットワークに接続されネームサービスに登録されている必要があります。ネームサービスを使用する場合、システムのネームサービスは、NIS、NIS+、DNS、LDAP のいずれかでなければなりません。ネームサービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。
2. リモート **SPARC** システム上で、システムのドライブに **Solaris SOFTWARE 1 of 2 CD (SPARC 版)** を挿入します。
3. リモート **SPARC** システム上で、次のエントリを `/etc/dfs/dfstab` ファイルに追加します。  

```
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
```
4. リモート **SPARC** システム上で、**NFS** デーモンを起動します。  

```
remote-SPARC-system# /etc/init.d/nfs.server start
```
5. リモート **SPARC** システム上で **share** コマンドを実行し、他のシステムが **CD** を利用可能であることを確認します。

```
remote-SPARC-system# share
- /cdrom/cdrom0/s0 ro,anon=0 " "
- /cdrom/cdrom0/s1 ro,anon=0 " "
```

6. **SPARC** インストールサーバーとして使用する **x86** システム上で、スーパーユーザーになります。
7. **x86** システム上で **SPARC CD** にアクセスする際のマウントポイントとなる 2 つのディレクトリ (ミニルート用と製品用) を作成します。

```
x86-system# mkdir directory_name_s0
```

```
x86-system# mkdir directory_name_s1
```

*directory\_name\_s0* スライス 0 からの製品用のマウントポイントとなるディレクトリの名前です。

*directory\_name\_s1* スライス 1 からのミニルート用のマウントポイントとなるディレクトリの名前です。

8. **CD** がリモート **x86** システム上で適切にエクスポートされていることを確認します。

```
x86-system# showmount -e remote-SPARC-system
export list for remote-SPARC-system:
/cdrom/sol_9_sparc/s0 (everyone)
/cdrom/sol_9_sparc/s1 (everyone)
```

9. **x86** システム上で、リモート **SPARC** システムの **CD** イメージをマウントします。

```
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s0 directory_name_s0
```

```
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s1 directory_name_s1
```

10. **x86** システム上で、マウントされたディスクの **Tools** ディレクトリに移動します。

```
x86-system# cd /directory_name_s0/Solaris_9/Tools
```

11. **x86** システム上で、インストールサーバーのハードディスク内の、**setup\_install\_server** コマンドを使用して作成したディレクトリにドライブ内のディスクをコピーします。

```
x86-system# ./setup_install_server -t directory_name_s1 install_dir_path
```

*-t* Solaris SOFTWARE 2 of 2 CD の **Tools** ディレクトリ上のブートイメージ以外を使用する場合は、そのブートイメージのパスを指定します。

*directory\_name\_s1* スライス 1 からのミニルート用のマウントポイントとなるディレクトリの名前です。

*install\_dir\_path* ディスクイメージをコピーするディレクトリを指定します。空のディレクトリを指定してください。

---

注 - `setup_install_server` コマンドは、Solaris SOFTWARE ディスクイメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、`df -k1` コマンドを使用します。

---

12. x86 システム上で、トップディレクトリに移動します。

```
x86-system# cd /
```

13. x86 システム上で、両方のディレクトリのマウントを解除します。

```
x86-system# umount directory_name_s0
```

```
x86-system# umount directory_name_s1
```

14. SPARC システム上で、両方の CD-ROM スライスの共有を解除します。

```
remote-SPARC-system# unshare /cdrom/cdrom0/s0
```

```
remote-SPARC-system# unshare /cdrom/cdrom0/s2
```

15. SPARC システム上で、Solaris SOFTWARE 1 of 2 CD (SPARC 版) を取り出します。

16. Solaris SOFTWARE 2 of 2 CD (SPARC 版) を x86 システムの CD-ROM ドライブに挿入します。

17. x86 システム上で、マウントされた CD の **Tools** ディレクトリに移動します。

```
x86-system# cd /cdrom/cdrom0/Solaris_9/Tools
```

18. x86 システム上で、CD をインストールサーバーのハードディスクにコピーします。

```
x86-system# ./add_to_install_server install_dir_path
```

`install_dir_path` CD イメージをコピーするディレクトリを指定します。

19. Solaris SOFTWARE 2 of 2 CD (SPARC 版) を取り出します。

20. x86 システム上で、Solaris LANGUAGES CD (SPARC 版) を x86 システムの CD-ROM ドライブに挿入し、CD をマウントします。

21. x86 システム上で、マウントされた CD の **Tools** ディレクトリに移動します。

```
x86-system# cd /cdrom/cdrom0/Tools
```

22. x86 システム上で、CD をインストールサーバーのハードディスクにコピーします。

```
x86-system# ./add_to_install_server install_dir_path
```

`install_dir_path` CD イメージをコピーするディレクトリを指定します。

23. ユーザーが **Solaris Web Start** を使用してシステムをブートし、ネットワーク上で **Solaris 9** ソフトウェアをインストールできるようにするか、決定します。

- しない場合は、Solaris LANGUAGES CD (SPARC 版) を取り出して手順 31 へ進みます。
- する場合は、Solaris LANGUAGES CD (SPARC 版) を取り出して次の作業へ進みます。

24. リモート SPARC システムの CD-ROM ドライブに **Solaris INSTALLATION CD (SPARC 版)** を挿入します。

25. リモート SPARC システム上で、次のエントリを **/etc/dfs/dfstab** ファイルに追加します。

```
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
```

26. リモート SPARC システム上で、**NFS** デーモンを起動します。

```
remote-SPARC-system# /etc/init.d/nfs.server start
```

27. リモート SPARC システム上で、**CD** が他のシステムから利用可能であることを確認します。

```
remote-SPARC-system# share
- /cdrom/multi_sol_9_x86/s0 ro,anon=0 " "
- /cdrom/multi_sol_9_x86/s1 ro,anon=0 " "
```

上のサンプル出力では、**sol\_9\_x86** は、x86 版 Solaris 9 オペレーティング環境を指します。このテキスト文字列は、Solaris オペレーティング環境のバージョンによって異なります。

28. **x86** システム上で、**CD** 上のスライスをマウントします。

```
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s0 directory_name_s0
x86-system# mount remote_SPARC_system_name:/cdrom/cdrom0/s1 directory_name_s1
```

29. **x86** システム上で、マウントされた **CD** の **modify\_install\_server** を含むディレクトリに移動します。

```
x86-system# cd directory_name_s0
```

30. **Solaris Web Start** インタフェースソフトウェアをインストールサーバーにコピーします。

```
x86-system# ./modify_install_server -p install_dir_path directory_name_s1
```

**-p** 既存イメージのミニルートを **install\_dir\_path/Solaris\_9/Tools/Boot.orig** に保存します。

**install\_dir\_path** Solaris Web Start インタフェースをコピーするディレクトリを指定します。

`directory_name_s1` スライス 1 からのミニルート用のマウントポイントとなるディレクトリの名前です。

31. `setup_install_server` によって作成されたネットインストールイメージ上のミニルート (`Solaris_9/Tools/Boot`) にあるファイルにパッチを適用するか決定します。

- 適用しない場合は、次の手順に進みます。
- 適用する場合は、`patchadd -c` コマンドを使用して、ミニルートにあるファイルにパッチを適用します。



---

注意 `-patchadd -c` を使用する前には必ず、パッチの README を読むか、ご購入先におたずねください。

---

32. ブートサーバーを作成する必要があるかどうかを判断します。

- インストールサーバーとクライアント (インストールしているシステム) が同じサブネット上にある場合、あるいは、DHCP を使用している場合、ブートサーバーを作成する必要はありません。161 ページの「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を追加する」を参照してください。
- インストールサーバーがインストールするシステムと同じサブネット上に存在せず、DHCP を使用していない場合は、ブートサーバーを作成する必要があります。ブートサーバーを作成する方法については、159 ページの「CD イメージを使用してサブネット上にブートサーバーを作成する方法」を参照してください。

例 15-4 SPARC CD メディアを使用して x86 システムに SPARC インストールサーバーを作成する

次の例では、`richards` という名前の x86 システム上に SPARC インストールサーバーを作成する方法を示します。次の SPARC CD が、`simpson` という名前のリモート SPARC システムから x86 インストールサーバーの `/export/home/cdsparc` ディレクトリにコピーされます。

- Solaris SOFTWARE 1 of 2 CD (SPARC 版)
- Solaris SOFTWARE 2 of 2 CD (SPARC 版)
- Solaris LANGUAGES CD (SPARC 版)
- Solaris INSTALLATION CD (SPARC 版)

リモート SPARC システムで、次のように入力します。

```
simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
simpson (remote-SPARC-system)# /etc/init.d/nfs.server start
```

x86 システムで、次のように入力します。

例 15-4 SPARC CD メディアを使用して x86 システムに SPARC インストールサーバーを作成する (続き)

```
richards (x86-system)# mkdir /sparcS0
richards (x86-system)# mkdir /sparcS1
richards (x86-system)# mount simpson:/cdrom/cdrom0/s0 /sparcS0
richards (x86-system)# mount simpson:/cdrom/cdrom0/s1 /sparcS1
richards (x86-system)# cd /sparcS0/Solaris_9/Tools
richards (x86-system)# ./setup_install_server -t /sparcS0 /export/home/cdsparc
richards (x86-system)# cd /
richards (x86-system)# umount /sparcS0
richards (x86-system)# umount /sparcS1
```

リモート SPARC システムで、次のように入力します。

```
simpson (remote-SPARC-system) unshare /cdrom/cdrom0/s0
simpson (remote-SPARC-system) unshare /cdrom/cdrom0/s1
```

x86 システムで、次のように入力します。

```
richards (x86-system)# cd /cdrom/cdrom0/Solaris_9/Tools
richards (x86-system)# ./add_to_install_server /export/home/cdsparc
richards (x86-system)# cd /cdrom/cdrom0/Tools
richards (x86-system)# ./add_to_install_server /export/home/cdsparc
```

リモート SPARC システムで、次のように入力します。

```
simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
simpson (remote-SPARC-system)# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s1
simpson (remote-SPARC-system)# /etc/init.d/nfs.server start
```

x86 システムで、次のように入力します。

```
richards (x86-system)# mount remote_SPARC_system_name:/cdrom/cdrom0 /sparcS0
richards (x86-system)# mount remote_SPARC_system_name:/cdrom/cdrom0 /sparcS1
richards (x86-system)# cd /sparcS0
richards (x86-system)# ./modify_install_server -p /export/home/cdsparc /sparcS1
```

この例では、各コマンドを実行する前に各 CD がドライブに挿入され、自動的にマウントされるものとします。各コマンドの実行が終了すると、CD が取り出されます。

---

## CD イメージを用いたサブネット上での ブートサーバーの作成

Solaris ソフトウェアをネットワークからインストールするためには、インストールサーバーを作成する必要があります。ブートサーバーは必ずしも設定する必要はありません。ブートサーバーには、ネットワーク上でシステムをブートするために必要なブートソフトウェアが含まれています。一方、インストールサーバーは Solaris ソフトウェアのインストールを行います。

- DHCP を使用してインストールパラメータを設定している場合、あるいは、インストールサーバーとクライアントが同じサブネット上にある場合、ブートサーバーは必要ありません。161 ページの「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を追加する」へ進みます。
- インストールサーバーとクライアントが同じサブネット上に存在せず、DHCP を使用していない場合は、サブネットごとに異なるブートサーバーを作成する必要があります。インストールサーバーはサブネットごとに作成できます。しかし、インストールサーバーはより多くのディスク容量を必要とします。

### ▼ CD イメージを使用してサブネット上にブートサーバーを作成する方法

1. サブネット内でブートサーバーとして使用するシステムにログインし、スーパーユーザーになります。  
システムに CD-ROM ドライブが存在するか、あるいはリモートの Solaris 9 ディスクイメージ (通常、インストールサーバーに存在する) にアクセスできなければなりません。ネームサービスを使用する場合は、システムがすでにそのネームサービスに登録されていなければなりません。ネームサービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。

2. インストールサーバーから **Solaris SOFTWARE 1 of 2 CD** イメージをマウントします。

```
# mount -F nfs -o ro server_name:path /mnt
```

*server\_name:path* インストールサーバー名とディスクイメージの絶対パスを指定します。

3. ブートイメージ用のディレクトリを作成します。

```
# mkdir -p boot_dir_path
```

*boot\_dir\_path* ブートソフトウェアをコピーするディレクトリを指定します。

4. Solaris SOFTWARE 1 of 2 CD イメージの **Tools** ディレクトリへ移動します。

```
# cd /mnt/Solaris_9/Tools
```

5. ブートソフトウェアをブートサーバーにコピーします。

```
# ./setup_install_server -b boot_dir_path
```

-b システムをブートサーバーとして設定することを示します。

boot\_dir\_path ブートソフトウェアをコピーするディレクトリを指定します。

---

注 -setup\_install\_server コマンドは、イメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、df -k1 コマンドを使用します。

---

6. ルート (/) ディレクトリへ移動します。

```
# cd /
```

7. インストールイメージをマウント解除します。

```
# umount /mnt
```

以上の操作で、ネットワークからインストールするためにシステムを設定する用意が整いました。161 ページの「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を追加する」を参照してください。

#### 例 15-5 CD メディアを使用してサブネット上にブートサーバーを作成する

次の例は、ブートサーバーをサブネット上に作成する方法を示したものです。次のコマンドを実行すると、Solaris SOFTWARE 1 of 2 CD (SPARC 版) イメージのブートソフトウェアがシステムのローカルディスク (/export/install/boot) にコピーされます。

```
# mount -F nfs -o ro crystal:/export/install/boot /mnt
# mkdir -p /export/install/boot
# cd /mnt/Solaris_9/Tools
# ./setup_install_server -b /export/install/boot
# cd /
# umount /mnt
```

この例では、コマンドを実行する前に各ディスクがドライブに挿入され、自動的にマウントされるものとします。コマンドの実行が終了すると、ディスクが取り出されません。



---

## 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を追加する

インストールサーバーとブートサーバー (必要な場合) を作成した後、ネットワークからインストールする各システムを設定する必要があります。インストールする各システムは、次の情報にアクセスできる必要があります。

- インストールサーバー
- ブートサーバー (必要な場合)
- sysidcfg ファイル (sysidcfg ファイルを使用してシステム構成情報を事前設定する場合)
- ネームサーバー (ネームサービスを使用してシステム構成情報を事前設定する場合)
- プロファイルサーバー上の JumpStart ディレクトリにあるプロファイル (カスタム JumpStart インストールを使用する場合)

インストールサーバーとクライアントの設定には、`add_install_client` を使用して以下の作業を実行してください。次の手順の例も参照してください。

- DHCP を使用してインストールパラメータを設定する場合は、例 15-6 を参照してください。
- インストールサーバーとクライアントが同じサブネット上に存在する場合は、例 15-7 を参照してください。
- インストールサーバーとクライアントが同じサブネット上に存在せず、DHCP を使用していない場合は、例 15-8 を参照してください。
- x86 クライアントのインストールパラメータの設定に DHCP を使用する場合は、例 15-9 を参照してください。
- x86 ベースのシステムのインストール中に特定のシリアルポートに出力を表示したい場合は、例 15-10 を参照してください。
- x86 クライアントのインストール時に、特定のネットワークインタフェースを使用するように設定したい場合は、例 15-11 を参照してください。

このコマンドで使用できるオプションについての詳細は、`add_install_client (1M)` のマニュアルページを参照してください。

## ▼ add\_install\_client を実行してネットワークからインストールするシステムを追加する方法 (CD)

ブートサーバーが存在する場合は、インストールサーバーのインストールイメージを共有しているか確認してください。「インストールサーバーを作成する方法」の手順6を参照してください。

1. インストールサーバーまたはブートサーバー上でスーパーユーザーになります。
2. **NIS**、**NIS+**、**DNS**、または **LDAP** のネームサービスを使用する場合、インストールするシステムについての次のような情報がネームサービスに追加されていることを確認します。

- ホスト名
- IP アドレス
- Ethernet アドレス

ネームサービスの詳細は、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

3. インストールサーバー上の **Solaris 9 CD** イメージの **Tools** ディレクトリに移動します。

```
# cd /install_dir_path/Solaris_9/Tools
```

*install\_dir\_path* Tools ディレクトリのパスを指定します。

4. ネットワークでインストールするためのシステムを設定します。

```
# ./add_install_client [-d] [-s install_server:install_dir_path] \  
[-c jumpstart_server:jumpstart_dir_path] [-p sysid_server:path] \  
[-t boot_image_path] [-b "network_boot_variable=value"] \  
[-e ethernet_address] client_name platform_group
```

-d

クライアントが DHCP を使用してネットワークインストールパラメータを取得することを指定します。-d だけを使用した場合、add\_install\_client コマンドは、同じクラスに属するすべてのクライアントシステム (たとえばすべての SPARC クライアントマシン) のインストール情報を設定します。特定のクライアントのインストール情報を設定する場合は、-d と -e を併用します。

x86 クライアントの場合、このオプションを指定し、PXE ネットワークブートを使用してネットワークからシステムをブートしてください。

	DHCP を使用した特定のクラスのインストールの詳細は、76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」を参照してください。
<code>-s install_server:install_dir_path</code>	<p>インストールサーバーの名前とパスを指定します。</p> <ul style="list-style-type: none"> <li>■ <code>install_server</code> は、インストールサーバーのホスト名です。</li> <li>■ <code>install_dir_path</code> は、Solaris 9 CD イメージの絶対パスです。</li> </ul>
<code>-c jumpstart_server:jumpstart_dir_path</code>	<p>カスタム JumpStart インストールの JumpStart ディレクトリを指定します。<code>jumpstart_server</code> には JumpStart ディレクトリがあるサーバーのホスト名を指定します。<code>jumpstart_dir_path</code> は、JumpStart ディレクトリへの絶対パスです。</p>
<code>-p sysid_server:path</code>	<p>システムの構成情報を事前設定するための <code>sysidcfg</code> ファイルのパスを指定します。<code>sysid_server</code> は、このファイルを持っているサーバーの有効なホスト名または IP アドレスです。<code>path</code> は、<code>sysidcfg</code> ファイルを含むディレクトリの絶対パスです。</p>
<code>-t boot_image_path</code>	<p>Solaris 9 ネットインストールイメージ、CD、または DVD 上の <code>Tools</code> ディレクトリにあるブートイメージ以外のブートイメージを使用したい場合、別のブートイメージへのパスを指定します。</p>
<code>-b "boot-property=value"</code>	<p><b>x86 システムのみ:</b> ネットワークからクライアントをブートするとき使用するブートプロパティ変数の値を設定できます。<code>-b</code> は、<code>-e</code> オプションと組み合わせて使用してください。</p> <p>ブートプロパティの詳細は、<code>eeprom (1M)</code> のマニュアルページを参照してください。</p>
<code>-e ethernet_address</code>	<p>インストールするクライアントの Ethernet アドレスを指定します。このオ</p>

プションを指定すると、特定のクライアントのインストール情報を設定できます。

DHCP を使用した特定のクライアントのインストールの詳細は、76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」を参照してください。

*client\_name*

ネットワークからインストールするシステムのホスト名を指定します。この名前はインストールサーバーのホスト名ではありません。

*platform\_group*

インストール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。プラットフォームグループの詳細は、付録 C を参照してください。

#### 例 15-6 SPARC: DHCP を使用している場合の SPARC インストールサーバー上の SPARC インストールクライアントの追加方法 (CD)

次の例に、DHCP を使用してネットワーク上でインストールパラメータを設定しているときに、インストールクライアントを追加する方法を示します。インストールクライアントは、*basil* という名前の Ultra 5 システムです。ファイルシステム `/export/home/cdsparc/Solaris_9/Tools` には、`add_install_client` コマンドが入っています。

DHCP を使用してネットワークインストール用のインストールパラメータを設定する方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP サービスを使用した Solaris ネットワークインストールのサポート (作業マップ)」を参照してください。

```
sparc_install_server# cd /export/home/cdsparc/Solaris_9/Tools
sparc_install_server# ./add_install_client -d basil sun4u
```

#### 例 15-7 サーバーと同じサブネット上にあるインストールクライアントの追加方法 (CD)

次の例に、インストールサーバーと同じサブネット上にあるインストールクライアントを追加する方法を示します。インストールクライアントは、*basil* という名前の Ultra 5 システムです。ファイルシステム `/export/home/cdsparc/Solaris_9/Tools` には、`add_install_client` コマンドが入っています。

```
install_server# cd /export/home/cdsparc/Solaris_9/Tools
install_server# ./add_install_client basil sun4u
```

例 15-8 ブートサーバーへのインストールクライアントの追加方法 (CD)

次の例に、ブートサーバーにインストールクライアントを追加する方法を示します。インストールクライアントは、`rose` という名前の Ultra 5 システムです。次のコマンドはブートサーバー上で実行します。`-s` オプションには、`/export/home/cdsparc` に Solaris 9 CD イメージを持っている `rosemary` という名前のインストールサーバーを指定しています。

```
boot_server# cd /export/home/cdsparc/Solaris_9/Tools
boot_server# ./add_install_client -s rosemary:/export/home/cdsparc rose sun4u
```

例 15-9 x86: DHCP を使用している場合の x86 インストールサーバー上の x86 インストールクライアントの追加方法 (CD)

次の例に、DHCP を使用してネットワークのインストールパラメータを設定している場合に、x86 インストールクライアントをインストールサーバーに追加する方法を示します。`-d` オプションは、クライアントが構成に DHCP プロトコルを使用することを指定するために使用されます。PXE ネットワークブートを使用する場合には、DHCP プロトコルを使用する必要があります。DHCP クラス名 `SUNW.i86pc` は、このコマンドが単一のクライアントだけでなく、ネットワーク上のすべての Solaris x86 ブートクライアントに適用されることを意味します。`-s` オプションを使用して、`rosemary` という名前のインストールサーバーからクライアントをインストールすることを指定します。このサーバーは、`/export/home/cdx86` に Solaris SOFTWARE 1 of 2 CD (x86 版) イメージを保持しています。

DHCP を使用してネットワークインストール用のインストールパラメータを設定する方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP サービスを使用した Solaris ネットワークインストールのサポート (作業マップ)」を参照してください。

```
install_server# cd /export/boot/Solaris_9/Tools
install_server# ./add_install_client -d -s rosemary:/export/home/cdx86 SUNW.\
i86pc i86pc
```

例 15-10 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するコンソールを指定する方法を示しています。この例では、インストールクライアントは次のように設定されます。

- `-d` オプションが指定されているので、クライアントは DHCP を使用してインストールパラメータを設定します。
- `-e` オプションが指定されているので、Ethernet アドレスが `00:07:e9:04:4a:bf` と一致するクライアントに対してのみ、インストールが行われます。
- 1 番目と 2 番目の `-b` オプションにより、インストールプログラムの入出力デバイスがシリアルポート `ttya` に設定されます。

```
install_server# cd /export/boot/Solaris_9/Tools
install_server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
```

例 15-10 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)  
(続き)

```
-b "input-device=ttya" -b "output-device=ttya" i86pc
```

-b オプションで使用可能なブートプロパティ変数の詳細は、`eeprom(1M)` のマニュアルページを参照してください。

例 15-11 x86: ネットワークインストール時に使用するブートデバイスを指定 (CD)

次の例は、x86 インストールクライアントをインストールサーバーに追加して、インストール時に使用するブートデバイスを指定する方法を示しています。インストールクライアントの設定時にブートデバイスを指定すると、インストール時、Device Configuration Assistant (デバイス構成用補助) はこの情報の入力プロンプトを表示しません。

この例では、インストールクライアントは次のように設定されます。

- -d オプションが指定されているので、クライアントは DHCP を使用してインストールパラメータを設定します。
- -e オプションが指定されているので、Ethernet アドレスが 00:07:e9:04:4a:bf と一致するクライアントに対してのみ、インストールが行われます。
- 1 番目と 2 番目の -b オプションにより、インストールプログラムの入出力デバイスがシリアルポート `ttya` に設定されます。
- 3 番目の -b オプションにより、インストールプログラムがインストール時に特定のブートデバイスを使用するように設定されます。

---

注 - ブートデバイスパスの値は、使用するハードウェアによって異なります。

---

- `i86pc` プラットフォーム名から、クライアントが x86 システムであることがわかります。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

-b オプションで使用可能なブートプロパティ変数の詳細は、`eeprom(1M)` のマニュアルページを参照してください。

## 第 16 章

# ネットワークインストールの準備 (コマンドリファレンス)

この章では、ネットワークインストールを設定するときに使用できるコマンドを示します。

表 16-1 ネットワークインストールコマンド

コマンド	プラットフォーム	説明
<code>add_install_client</code>	すべて	システムに関するネットワークインストール情報を、ネットワークからインストールサーバーまたはブートサーバーに追加するコマンド。詳細は、 <code>add_install_client(1M)</code> のマニュアルページを参照
<code>setup_install_server</code>	すべて	Solaris 9 DVD または CD をインストールサーバーのローカルディスクにコピーするか、ブートソフトウェアをブートサーバーにコピーするスクリプト。詳細は、 <code>setup_install_server(1M)</code> のマニュアルページを参照
<code>add_to_install_server</code> (CDメディアのみ)	すべて	CD 上のプロダクトツリー内の追加パッケージを、既存のインストールサーバー上のローカルディスクにコピーするスクリプト。詳細は、 <code>add_to_install_server(1M)</code> のマニュアルページを参照
<code>modify_install_server</code> (CDメディアのみ)	すべて	Solaris Web Start ユーザーインタフェースソフトウェアを、既存のインストールサーバーまたはブートサーバー上の Solaris 9 CD イメージに追加するスクリプト。ユーザーが Solaris Web Start によってシステムをブートし、Solaris 9 ソフトウェアをネットワークからインストールできるようにする。詳細は、 <code>modify_install_server(1M)</code> のマニュアルページを参照。RAM が 64M バイト以下のシステムは、Solaris Web Start プログラムを使用してインストールできません。

表 16-1 ネットワークインストールコマンド (続き)

コマンド	プラットフォーム	説明
mount	すべて	ファイルシステムをマウントできるようにし、マウントされたファイルシステム (Solaris DVD または Solaris SOFTWARE と Solaris LANGUAGES CD 上のファイルシステムを含む) を表示するコマンド。詳細は、mount (1M) のマニュアルページを参照
showmount -e	すべて	リモートホスト上の共有ファイルシステムすべてをリスト表示するコマンド。詳細は、showmount (1M) のマニュアルページを参照
uname -i	すべて	システムのプラットフォーム名 SUNW,SPARCstation-5、i86pc などを調べるコマンド。Solaris ソフトウェアをインストールするには、システムのプラットフォーム名が必要になることがあります。詳細は、uname (1) のマニュアルページを参照
patchadd -C <i>net_install_image</i>	すべて	<p>setup_install_server で作成した DVD または CD のネットインストールイメージ上にあるミニルート (Solaris_9/Tools/Boot) にあるファイルにパッチを追加するコマンド。これにより、Solaris インストールコマンドと他のミニルート固有のコマンドにパッチを適用できる。net_install_image はネットワークインストールイメージの絶対パス名である。詳細は、patchadd (1M) のマニュアルページを参照</p> <p>注意 - patchadd -C を使用する前には必ず、パッチの README を読むか、ご購入先におたずねください。</p>
reset	SPARC	システムをリセットし、マシンを再起動するための Open Boot PROM コマンド。また、ブート時に入出力割り込みに関するエラーメッセージが表示された場合は、L1 または STOP キーと A キーを同時に押し、その後 ok または >PROM プロンプトで reset と入力する。
banner	SPARC	モデル名、Ethernet アドレス、インストールされているメモリーなどのシステム情報を表示する Open Boot PROM コマンド。ok (>PROM) プロンプトでのみ使用可能。



## 第 17 章

---

# Solaris Web Start の使用 (作業)

---

この章では、Solaris DVD または Solaris INSTALLATION CD に含まれている Solaris Web Start を使用して、Solaris ソフトウェアのインストールやアップグレードを行う方法について説明します。

---

注 - Solaris オペレーティング環境をインストールするシステムまたはドメインに直接接続された DVD-ROM ドライブまたは CD-ROM ドライブがない場合は、別のマシンに接続された DVD-ROM ドライブまたは CD-ROM ドライブを使用できます。詳細は、付録 F を参照してください。

---

この章の内容は、次のとおりです。

- 169 ページの「Solaris Web Start の実行方法」
- 170 ページの「SPARC: Solaris Web Start によるインストールまたはアップグレード」
- 175 ページの「x86: Solaris Web Start によるインストールまたはアップグレード」
- 182 ページの「Solaris Web Start によるインストールまたはアップグレード後に必要な作業」

---

## Solaris Web Start の実行方法

Solaris Web Start は、GUI または CLI を使って実行できます。x86 システムの場合、Solaris Web Start プログラムに Device Configuration Assistant (デバイス構成用補助) が含まれています。

- GUI - ローカルまたはリモートの DVD-ROM ドライブか CD-ROM ドライブ、またはネットワーク接続、およびビデオアダプタ、キーボード、モニター、十分なメモリーが必要です。

- CLI – ローカルまたはリモートの DVD-ROM ドライブか CD-ROM ドライブ、またはネットワーク接続、およびキーボードとモニターが必要です。Solaris Web Start インストール CLI は、tip コマンドを使って実行できます。詳細は、tip(1) のマニュアルページを参照してください。

Solaris Web Start は、システムにビデオアダプタがあることを検出すると自動的に GUI を使用し、検出できないと CLI を使用します。操作手順や設定内容は、GUI も CLI もほとんど同じです。

---

## SPARC: Solaris Web Start によるインストールまたはアップグレード

Solaris DVD (SPARC 版) または Solaris INSTALLATION CD (SPARC 版) に含まれている Solaris Web Start プログラムを使用して、Solaris オペレーティング環境をインストールまたはアップグレードできます。

---

注 – Solaris INSTALLATION CD からインストールする場合は、47 ページの「Solaris INSTALLATION CD を使用するための要件」で特殊な要件を確認してください。

---

以下のものが揃っているか確認します。

- DVD からインストールする場合は、Solaris DVD (SPARC 版) を使用してください。
- CD からインストールする場合は、以下の CD を使用してください。
  - Solaris INSTALLATION CD (SPARC 版)
  - Solaris SOFTWARE 1 of 2 CD (SPARC 版)
  - Solaris SOFTWARE 2 of 2 CD (SPARC 版) – 必要に応じて、インストールプログラムがこの CD をドライブに挿入するように要求します。
  - Solaris LANGUAGES CD (SPARC 版) – 特定地域の言語をサポートするために必要な場合、インストールプログラムがこの CD をドライブに挿入するように要求します。

# SPARC: 作業マップ: Solaris Web Start インストールの実行

表 17-1 作業マップ: Solaris Web Start インストールの実行

作業	説明	参照先
システム要件を確認する	Solaris Web Start を使用してインストールまたはアップグレードをするための要件をシステムが満たしていることを確認します。	43 ページの「メモリー要件」
必要な情報を収集する	チェックリストに従ってワークシートに記入しながら、Solaris ソフトウェアのインストールに必要なすべての情報を収集します。	第 6 章
(省略可能) システム構成情報を事前設定する	sysidcfg ファイルやネームサービスを使って、システムのインストール情報 (locale など) を事前設定することができます。これによって、インストール時にこの情報を入力しなくて済みます。	第 7 章
(アップグレードのみ) システムをアップグレードするための準備をする	システムのバックアップをとります。	『Solaris のシステム管理 (基本編)』
(省略可能) ネットワーク上でインストールするためのシステムを設定する	リモートの DVD または CD のネットインストール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。	第 15 章
インストールまたはアップグレードを行う	システムをブートし、プロンプトに従って Solaris ソフトウェアをインストールまたはアップグレードします。	171 ページの「SPARC: Solaris Web Start 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」
(アップグレードのみ) アップグレード後に必要な作業を行う	アップグレードで失われたローカル変更がある場合は、それを復元します。	182 ページの「アップグレード後にローカル変更を復元する方法」

## ▼ SPARC: Solaris Web Start を使用してインストールまたはアップグレードする方法

1. **DVD-ROM** ドライブ、**CD-ROM** ドライブ、ネットイメージのどれを使用してソフトウェアをインストールするかを決定します。
  - DVD-ROM ドライブか CD-ROM ドライブを使用する場合は、Solaris DVD (SPARC 版) または Solaris INSTALLATION CD (SPARC 版) を挿入します。
  - ネットインストールイメージを使用する場合は、インストールメディアが置かれているディレクトリに移動します。必要に応じ、ネットワーク管理者に場所を尋ねてください。ネットワークからのインストール方法の詳細は、第 14 章

たは第 15 章を参照してください。

## 2. 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。

- ローカル DVD またはローカル CD からブートする場合は、次のコマンドを入力します。

```
ok boot cdrom [- nowin]
```

- 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場合は、次のコマンドを入力します。

```
ok boot net [- nowin]
```

`nowin` Solaris Web Start を CLI モードで実行することを指定します。

Solaris Web Start インストールが始まります。

## 3. システム構成の質問に答えます。

- すべてのシステム構成情報が事前設定されている場合は、構成情報の入力はありません。
- システム構成情報の事前設定を行っていない場合は、52 ページの「インストール用のワークシート」または 58 ページの「アップグレード用のワークシート」を参照して、システム構成の質問に答えます。

GUI を使用している場合は、システム構成情報の確認が終わると、Solaris Web Start の Kiosk が起動し、「ようこそ (Welcome)」画面が表示されます。システムのメモリーが十分でない場合は、Kiosk は表示されません。



図 17-1 Solaris Web Start Kiosk メニュー

Kiosk メニューのリンクは、どれでもクリックできます。

---

注 - Kiosk によって画面が隠れてしまうことがあります。隠れている画面を表示するには、Kiosk メニューの「Kiosk を奥へ (Send Kiosk to Background)」をクリックします。

---

「インストーラ・オプション (Installer Questions)」画面が表示されます。

4. システムのレポートとディスクの取り出しを自動的に行うかどうかを決定します。  
「次へ (NEXT)」をクリックします。  
「媒体の指定 (Specify Media)」画面が表示されます。
  5. インストールに使用するメディアを指定します。
    - フラッシュアーカイブをインストールする場合には、作業を続行します。
    - フラッシュアーカイブをインストールしない場合には、手順 6 に進みます。
- a. プロンプトに従って次の情報を入力します。

選択された媒体	プロンプト
DVD または CD	フラッシュアーカイブを含むディスクを挿入してください。
ネットワークファイルシステム	フラッシュアーカイブを含むネットワークファイルシステムのパスを指定してください。アーカイブのファイル名も指定できます。
HTTP	フラッシュアーカイブにアクセスするための URL とプロキシ情報を指定してください。
FTP	FTP サーバーと、フラッシュアーカイブのパスを指定します。FTP サーバーへのアクセスに必要なユーザーおよびパスワードの情報を指定します。FTP サーバーへのアクセスに必要なプロキシ情報を指定します。
ローカルテープ	フラッシュアーカイブを含むローカルテープデバイスとテープ上の位置を指定してください。

DVD、CD、または NFS サーバーに格納されたアーカイブのインストールを選択した場合は、「フラッシュアーカイブの選択 (Select Flash Archives)」画面が表示されます。

- b. ディスクまたは NFS サーバーに格納されたアーカイブについては、「フラッシュアーカイブの選択 (Select Flash Archives)」画面で、インストールするフラッシュアーカイブを選択します (1 つでも複数でも可)。
  - c. 「フラッシュアーカイブの一覧 (Flash Archives Summary)」画面で選択されたアーカイブを確認し、「次へ (Next)」をクリックします。
  - d. 「追加するフラッシュアーカイブ (Additional Flash Archives)」画面で、別のアーカイブを含む媒体を指定して、フラッシュアーカイブの追加インストールを実行できます。アーカイブをそれ以上インストールしない場合は、「なし - アーカイブの選択が完了しました (None)」を選択し、「次へ (Next)」をクリックしてインストールを続けます。手順 7 へ進みます。
6. 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。
- システムがアップグレード可能かどうかが判定されます。このためには、Solaris ルート(/) ファイルシステムがすでに存在していなければなりません。Solaris INSTALLATION CD を使用してアップグレードする場合は、512M バイトのスライズが必要です。アップグレードが可能であることが確認されると、システムのアップグレードが開始されます。
7. 画面の指示に従って、Solaris ソフトウェアをインストールします。フラッシュアーカイブをインストールしていない場合、ソフトウェアの追加インストールを行うことも可能です。
- Solaris Web Start インストールプログラムが Solaris ソフトウェアのインストールを終了すると、システムは自動的にリブートされるか、あるいはシステムのリブートを求める画面が表示されます。

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

8. アップグレードを行った場合は、保存されなかったローカル変更があればそれを復元します。詳細は、**182** ページの「アップグレード後にローカル変更を復元する方法」を参照してください。

---

## x86: Solaris Web Start によるインストールまたはアップグレード

Solaris DVD または Solaris INSTALLATIONCD に含まれている Solaris Web Start を使用して、Solaris オペレーティング環境を x86 システムにインストールできます。

---

注 - Solaris INSTALLATION CD からインストールする場合は、47 ページの「Solaris INSTALLATION CD を使用するための要件」で特殊な要件を確認してください。

---

次のメディアを用意してください。

- DVD からインストールする場合は、Solaris DVD (x86 版) を使用してください。
- CD メディアからインストールする場合は、以下の CD を使用してください。
  - Solaris INSTALLATION CD (x86 版)
  - Solaris SOFTWARE 1 of 2 CD (x86 版)
  - Solaris SOFTWARE 2 of 2 CD (x86 版) - 必要に応じて、インストールプログラムがこの CD を挿入するように要求します。
  - Solaris LANGUAGES CD (x86 版) - 特定地域の言語をサポートするために必要な場合、インストールプログラムがこの CD を挿入するように要求します。

## x86: 作業マップ: Solaris Web Start インストールの実行

表 17-2 x86: 作業マップ: Solaris Web Start インストールの実行

作業	説明	参照先
システム要件を確認する	Solaris Web Start を使用してインストールまたはアップグレードをするための要件をシステムが満たしていることを確認します。	43 ページの「メモリー要件」
必要な情報を収集する	チェックリストに従ってワークシートに記入しながら、Solaris ソフトウェアのインストールに必要なすべての情報を収集します。	第 6 章
(アップグレードのみ) システムをアップグレードするための準備をする	システムのバックアップをとります。	『Solaris のシステム管理 (基本編)』
(省略可能) ネットワーク上でインストールするためのシステムを設定する	リモートの Solaris DVD (x86 版) または Solaris SOFTWARE (x86 版) のネットインストールイメージからシステムをインストールするには、インストールサーバーまたはブートサーバーからブートとインストールを行うようにシステムを設定する必要があります。	第 15 章
インストールまたはアップグレードを行う	システムをブートし、プロンプトに従って Solaris ソフトウェアをインストールまたはアップグレードします。	176 ページの「x86: Solaris Web Start 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」
(アップグレードのみ) アップグレード後に必要な作業を行う	アップグレードで失われたローカル変更がある場合は、それを復元します。	182 ページの「アップグレード後にローカル変更を復元する方法」

### ▼ x86: Solaris Web Start を使用してインストールまたはアップグレードする方法

1. **DVD-ROM** ドライブ、**CD-ROM** ドライブ、ネットインストールイメージのどれを使用してソフトウェアをインストールするかを決定します。
  - DVD-ROM ドライブまたは CD-ROM ドライブを使用する場合は、Solaris DVD (x86 版) または Solaris INSTALLATION CD (x86 版) を挿入します。
  - ネットインストールイメージを使用する場合は、インストールメディアが置かれているディレクトリに移動します。必要に応じて、ネットワーク管理者に場所をたずねてください。ネットワークからのインストール方法の詳細は、第 14 章または第 15 章を参照してください。
2. システムのブート方法を決定します。



- Solaris DVD または Solaris INSTALLATION 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 が使用でき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、システムのコモンドプロンプトに Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) を挿入します。

---

注 - 付録 I の手順に従って Solaris DVD (x86 版) または Solaris SOFTWARE 2 of 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. システムを停止して電源を切り、再び電源を入れてシステムをブートします。メモリーテストとハードウェア検出が実行されます。画面が再表示されます。
4. 「Solaris Device Configuration Assistant」画面が表示されたら **F2** キーを押します。  
「Bus Enumeration」画面に次のメッセージが表示されます。  

```
Determining bus types and gathering hardware configuration data ...
```

「Scanning Devices」画面が表示されます。システムデバイスのスキャンが行われます。スキャンが終了すると、「Identified Devices」画面が表示されます。
5. 変更する必要があるか確認します。
  - 変更しない場合は、「Continue」を選択します。
  - 変更する場合は、「changes」を選択します。「Loading」画面が表示され、システムをブートするためにロードされるドライバの情報が表示されます。数秒後に「Boot Solaris」画面が表示されます。
6. 「Boot Solaris」画面で、「DVD」、「CD」、「Net」、または「Disk」を選択し、「Continue」を選択します。  
デフォルトのブートディスクが、システムのインストールまたはアップグレードに必要な条件を満たしているかどうかを検査されます。

---

注 - Solaris INSTALLATIONCD を使用してインストールを行う場合、いくつかの要件を満たしている必要があります。これらの要件を満たすことができない場合、DVD または ネットインストールイメージから Solaris Web Start プログラムを使用するか、Solaris suninstall プログラムまたはカスタム JumpStart を使用してインストールしてください。要件は以下のとおりです。

- デフォルトブートディスクの BIOS および SCSI ドライバが、LBA (Logical Block Addressing) をサポートしている必要があります。
  - デフォルトブートディスクに Solaris fdisk パーティションが存在しない場合、パーティションを作成するようにメッセージが表示されます。fdisk を実行して Solaris fdisk パーティションを作成するかどうかを尋ねられます。「はい(Yes)」と応答すると、fdisk ユーザーインターフェイスが表示され、ディスク上に Solaris fdisk パーティションを手動で追加できます。「いいえ(No)」と応答すると、インストーラが終了します。
  - デフォルトブートディスクに x86BOOT パーティションが存在しない場合は、Solaris Web Start プログラムが Solaris fdisk パーティションの一部を使用して作成してくれます。Solaris fdisk パーティション上にデータが存在していた場合、それらはすべて破棄されます。このパーティションを手動で作成することはできません。ディスクをフォーマットするかという問いに「はい(Yes)」と応答すると、Solaris fdisk パーティションからパーティションが作成されます。「いいえ(No)」と応答した場合、別のインストール方法を選択する必要があります。
  - システム上の既存の Service パーティションを保持する場合、Solaris Web Start プログラムによりデフォルトブートディスク上の fdisk パーティションをカスタマイズするかどうか尋ねられたら、「デフォルト」を選択します。
  - パーティションの作成、変更、削除は、後でフォーマットパネルを使用して行うことができます。ただし、Solaris fdisk パーティションと x86BOOT パーティションが作成されると、ディスクは凍結され、ディスクの編集は不可能になります。複数のディスクが存在する場合、それらのディスクを fdisk パネルで編集できます。
- 

Solaris Web Start プログラムは、デフォルトブートディスクの各要件を確認し、見つからなかった構成情報を入力するようにユーザーに要求します。

7. システム構成の質問に答えます。
  - すべてのシステム構成情報が事前設定されている場合は、手順 8 へ進みます。
  - システム構成情報の事前設定を行っていない場合は、52 ページの「インストール用のワークシート」または 58 ページの「アップグレード用のワークシート」を参照して、システム構成の質問に答えます。数秒後に「The Solaris Installation Program」画面が表示されます。
8. 「The Solaris Installation Program」画面で「Continue」を選択します。  
「kdmconfig - Introduction」画面が表示されます。

9. GUI を実行するだけの十分なメモリーがあるかどうかを確認します。
  - メモリーが十分な場合は「Continue」を選択します。
  - メモリーが十分でない場合は「Bypass」を選択し、CLI を表示させます。「kdmconfig - View and Edit Window System Configuration」画面が表示されま  
す。

kdmconfig ユーティリティは、ビデオドライバを検出できないと 640x480 VGA  
ドライバを選択します。640x480 VGA ドライバでは、Solaris Web Start GUI を表  
示できないため、Solaris Web Start CLI が使用されます。Solaris Web Start GUI を  
使用したい場合は、kdmconfig ユーティリティを使って適切なドライバを選択す  
る必要があります。
10. (省略可能) 「kdmconfig – View and Edit Window System Configuration」画面の  
構成情報を確認し、必要な場合は変更します。
11. 構成情報の確認が終わったら、「No changes needed – Test/Save and Exit」を選  
択し、「Continue」を選択します。

「kdmconfig Window System Configuration Test」画面が表示されます。
12. 「Continue」を選択します。

画面が再表示され、「kdmconfig Window System Configuration Test palette and  
pattern」画面が表示されます。
13. ポインタを移動でき、パレット上に示されている色が正しく表示されていることを  
確認します。
  - 色が正しく表示されていない場合は、「No」をクリックします。可能な場合、  
キーボードのいずれかのキーを押すか、「kdmconfig Window System  
Configuration Test」画面が自動的に終了するのを待ちます。そして、一連の色  
が正しく表示され、ポインタを正しく移動できるまで、手順 10 から手順 13 を  
繰り返します。
  - 一連の色が正しく表示されている場合は、「Yes」をクリックします。

「ようこそ (Welcome)」画面が表示されます。システムのメモリーが十分でない場  
合は、Kiosk は表示されません。



図 17-2 Solaris Web Start Kiosk メニュー

Kiosk メニューのリンクは、どれでもクリックできます。

---

注 - Kiosk によって画面が隠れてしまうことがあります。隠れている画面を表示するには、Kiosk メニューの「Kiosk を奥へ (Send Kiosk to Background)」をクリックします。

---

14. 構成に関する残りの質問が表示される場合は、それらに答えます。  
「インストーラ・オプション (Installer Questions)」画面が表示されます。
15. システムのリポートとディスクの取り出しを自動的に行うかどうかを決定します。  
「次へ (NEXT)」をクリックします。

---

注 - PXE を使用してネットワークからブートする場合、手動リポートを選択します。システムがその起動時にネットワークからブートしないように設定する必要があります。ネットワークブートを無効にするには、リポート時にシステムの BIOS 設定ツールかネットワークアダプタの構成設定ツールを使用してください。

---

「媒体の指定 (Specify Media)」画面が表示されます。

16. インストールに使用するメディアを指定します。

- フラッシュアーカイブをインストールする場合には、作業を続行します。
  - フラッシュアーカイブをインストールしない場合には、手順 17 に進みます。
- a. プロンプトに従って次の情報を入力します。

選択された媒体	プロンプト
DVD または CD	フラッシュアーカイブを含むディスクを挿入してください。
ネットワークファイルシステム	フラッシュアーカイブを含むネットワークファイルシステムのパスを指定してください。アーカイブのファイル名も指定できます。
HTTP	フラッシュアーカイブにアクセスするための URL とプロキシ情報を指定してください。
FTP	FTP サーバーと、フラッシュアーカイブのパスを指定します。FTP サーバーへのアクセスに必要となるユーザーおよびパスワードの情報を指定します。FTP サーバーへのアクセスに必要となるプロキシ情報を指定します。
ローカルテープ	フラッシュアーカイブを含むローカルテープデバイスとテープ上の位置を指定してください。

DVD、CD、または NFS サーバーに格納されたアーカイブのインストールを選択した場合は、「フラッシュアーカイブの選択 (Select Flash Archives)」画面が表示されます。

- b. ディスクまたは NFS サーバーに格納されたアーカイブについては、「フラッシュアーカイブの選択 (Select Flash Archives)」画面で、インストールするフラッシュアーカイブを選択します (1 つでも複数でも可)。
- c. 「フラッシュアーカイブの一覧 (Flash Archives Summary)」画面で選択されたアーカイブを確認し、「次へ (Next)」をクリックします。
- d. 「追加するフラッシュアーカイブ (Additional Flash Archives)」画面で、別のアーカイブを含む媒体を指定して、フラッシュアーカイブの追加インストールを実行できます。アーカイブをそれ以上インストールしない場合は、「なし - アーカイブの選択が完了しました (None)」を選択し、「次へ (Next)」をクリックしてインストールを続けます。インストールが完了します。

Solaris Web Start インストールプログラムが Solaris ソフトウェアのインストールを終了すると、システムは自動的にリブートされるか、あるいはシステムのリブートを求める画面が表示されます。

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

17. 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。

システムがアップグレード可能かどうか判定されます。このためには、Solaris ルート(/) ファイルシステムがすでに存在していなければなりません。Solaris INSTALLATION CD を使用してアップグレードする場合は、512M バイトのスライスは必要です。アップグレードが可能であることが確認されると、システムのアップグレードが開始されます。

---

注 - インストールの開始前にシステムに Service パーティションを復元した場合、Solaris オペレーティング環境をアップグレードできない可能性があります。

---

18. 画面に示される手順に従って、**Solaris** ソフトウェアと追加ソフトウェア (必要な場合) をシステムにインストールします。

Solaris Web Start インストールプログラムが Solaris ソフトウェアのインストールを終了すると、システムは自動的にリブートされるか、あるいはシステムのリブートを求める画面が表示されます。

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

19. アップグレードを行った場合は、保存されなかったローカル変更があればそれを復元します。詳細は、182 ページの「アップグレード後にローカル変更を復元する方法」を参照してください。

---

## Solaris Web Start によるインストールまたはアップグレード後に必要な作業

Solaris オペレーティング環境のインストールまたはアップグレードを終了した後、必要に応じて次の作業を行う必要があります。

### ▼ アップグレード後にローカル変更を復元する方法

アップグレードを行うと、Solaris Web Start は既存システムに対して行なったローカルなソフトウェア変更と新しい Solaris ソフトウェアをマージします。ただし場合によっては、マージが行われなかったことがあります。したがって、システムのアップグレードが終了したら、保存されなかったローカル変更を復元しなければならない場合があります。



---

注意 - 保存されなかったローカル変更を復元しないと、システムがブートされないことがあります。

---

1. 次のファイルの内容を確認して、**Solaris Web Start** によって保存されなかったローカル変更を復元する必要があるかどうかを判断します。

```
/a/var/sadm/system/data/upgrade_cleanup
```

2. 保存されなかったローカル変更があれば、復元します。

3. システムをリブートします。

```
# reboot
```





## 第 18 章

---

# Solaris suninstall プログラムの使用 (作業)

---

この章では、Solaris SOFTWARE 1 of 2 CD に含まれている Solaris suninstall プログラムを使用して Solaris ソフトウェアをインストールまたはアップグレードする方法について説明します。

- 185 ページの「Solaris suninstall プログラム」
- 186 ページの「SPARC: Solaris suninstall プログラムによるインストールまたはアップグレード」
- 189 ページの「x86: Solaris suninstall プログラムによるインストールまたはアップグレード」
- 196 ページの「Solaris suninstall プログラムによるアップグレード後に必要な作業」

---

## Solaris suninstall プログラム

Solaris suninstall プログラムは CLI で実行できます。x86 システムの場合、Solaris suninstall プログラムに Device Configuration Assistant (デバイス構成用補助) が含まれています。Solaris suninstall プログラムには、ローカルまたはリモートの CD-ROM ドライブまたはネットワーク接続、キーボード、およびモニターが必要です。Solaris suninstall プログラムは `tip` コマンドで実行できます。詳細は、`tip (1)` のマニュアルページを参照してください。

---

注 - テキストを操作するとき、Control-U を使用するとページを上に移動し、Control-D を使用するとページを下に移動します。

---

---

# SPARC: Solaris suninstall プログラムによるインストールまたはアップグレード

Solaris SOFTWARE 1 of 2 CD に含まれている Solaris suninstall プログラムを使用して、Solaris オペレーティング環境を SPARC システムにインストールまたはアップグレードすることができます。Solaris suninstall プログラムでは、追加ソフトウェアをインストールすることはできません。

次の CD を用意してください。

- Solaris SOFTWARE 1 of 2 CD (SPARC 版)
- Solaris SOFTWARE 2 of 2 CD (SPARC 版) – 必要に応じて、インストールプログラムがこの CD をドライブに挿入するように要求します。
- Solaris LANGUAGES CD (SPARC 版) – 特定地域の言語をサポートするために必要な場合、インストールプログラムがこの CD をドライブに挿入するように要求します。

## SPARC: 作業マップ: Solaris suninstall プログラムインストールの実行

表 18-1 SPARC: 作業マップ: Solaris suninstall プログラムインストールの実行

作業	説明	参照先
必要な情報を収集する	チェックリストに従ってワークシートに記入しながら、Solaris ソフトウェアのインストールに必要なすべての情報を収集します。	第 6 章
(アップグレードのみ) システムをアップグレードするための準備をする	システムのバックアップをとります。	『Solaris のシステム管理 (基本編)』
(省略可能) システム構成情報を事前設定する	sysidcfg ファイルまたはネームサービスを使用して、システムのインストール情報 (locale など) を事前設定します。システム情報をあらかじめ設定すると、インストール時に情報入力を求めるプロンプトは表示されません。	第 7 章

表 18-1 SPARC: 作業マップ: Solaris suninstall プログラムインストールの実行 (続き)

作業	説明	参照先
(省略可能) ネットワーク上でインストールするためのシステムを設定する	リモートの Solaris SOFTWARE (SPARC 版) イメージからシステムをインストールする場合、インストールサーバーまたはブートサーバーからブートおよび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。	第 15 章
インストールまたはアップグレードを行う	システムをブートし、プロンプトに従って Solaris ソフトウェアをインストールまたはアップグレードします。	187 ページの「SPARC: Solaris suninstall 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には」
(アップグレードのみ) アップグレード後に必要な作業を行う	アップグレードで失われたローカル変更がある場合は、それを復元します。	196 ページの「Solaris suninstall プログラムによるアップグレード後に必要な作業」

## ▼ SPARC: Solaris suninstall 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には

1. **CD-ROM** ドライブとネットイメージのどちらを使用してソフトウェアをインストールするかを決定します。

- CD-ROM ドライブを使用する場合、Solaris SOFTWARE 1 of 2 CD (SPARC 版) を挿入します。
- ネットインストールイメージを使用する場合、次の例のように、インストールメディアが入っているディレクトリに移動します。必要に応じて、ネットワーク管理者に場所をたずねてください。次にコマンド例を示します。

```
※ cd /net/install-svr/export/Solaris_9/sparc
```

2. 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。
- ローカル CD からブートする場合は、次のコマンドを入力します。

```
ok boot cdrom
```

- 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場合は、次のように入力します。

```
ok boot net
```

Solaris suninstall プログラムが始まります。

3. システム構成の質問に答えます。

すべてのシステム構成情報が事前設定されている場合は、構成情報の入力はありません。

システム構成情報の事前設定を行っていない場合は、52 ページの「インストール用のワークシート」または 58 ページの「アップグレード用のワークシート」を参照して、システム構成の質問に答えます。

4. 画面に示される手順に従って、**Solaris** ソフトウェアをシステムにインストールします。

- フラッシュアーカイブをインストールしない場合には、インストールを完了します。アップグレードを実行している場合、またはソフトウェアを追加インストールする場合には、手順 5 に進みます。

Solaris ソフトウェアのインストールが終了すると、システムは自動的にリブートするか、または手動でリブートするようにプロンプトが表示されます。

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

- フラッシュアーカイブをインストールする場合には、作業を続行します。

a. 「フラッシュアーカイブの検索方法 (**Flash Archive Retrieval Method**)」画面で、フラッシュアーカイブのロケーションを選択します。

選択された媒体に応じて、Solaris suninstall プログラムインストールが続行するためのプロンプトが表示されます。

b. プロンプトに従って次の情報を入力します。

選択された媒体	プロンプト
HTTP	フラッシュアーカイブのアクセスに必要な URL 情報とプロキシ情報を指定します。
FTP	FTP サーバーと、フラッシュアーカイブのパスを指定します。FTP サーバーへのアクセスに必要なユーザーおよびパスワードの情報を指定します。FTP サーバーへのアクセスに必要なプロキシ情報を指定します。
ネットワークファイルシステム	フラッシュアーカイブが置かれているネットワークファイルシステムのパスを指定します。アーカイブのファイル名も指定できます。
ローカルファイル	フラッシュアーカイブが置かれているローカルファイルシステムのパスを指定します。
ローカルテープ	フラッシュアーカイブが置かれているローカルテープドライブとテープ上の位置を指定します。

選択された媒体	プロンプト
ローカルデバイス	フラッシュアーカイブが置かれているローカルデバイス、パス、ファイルシステムの種類を指定します。

- c. 「フラッシュアーカイブの選択 (**Flash Archive Selection**)」画面で、「新規 (**New**)」を選択すると、フラッシュアーカイブを追加インストールできます。アーカイブをそれ以上インストールしない場合は、インストールを完了させるために「継続 (**Continue**)」を選択します。

Solaris ソフトウェアのインストールが終了すると、システムは自動的にリブートするか、または手動でリブートするようにプロンプトが表示されます。

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
5. アップグレードを行った場合は、保存されなかったローカル変更があればそれを復元します。詳細は、196 ページの「**Solaris suninstall** プログラムによるアップグレード後に必要な作業」を参照してください。
6. (省略可能) 追加のソフトウェアをインストールする方法は、『Solaris のシステム管理 (基本編)』を参照してください。

## x86: Solaris suninstall プログラムによるインストールまたはアップグレード

Solaris SOFTWARE 1 of 2 CD に含まれている Solaris suninstall プログラムを使用して、Solaris オペレーティング環境を x86 システムにインストールまたはアップグレードすることができます。Solaris suninstall プログラムでは、追加ソフトウェアをインストールすることはできません。

次の CD を用意してください。

- Solaris SOFTWARE 1 of 2 CD (x86 版)
- Solaris SOFTWARE 2 of 2 CD (x86 版) – 必要に応じて、インストールプログラムがこの CD を挿入するように要求します。
- Solaris LANGUAGES CD (x86 版) – 特定地域の言語をサポートするために必要な場合、インストールプログラムがこの CD を挿入するように要求します。

## x86: 作業マップ: suninstall プログラムを実行する

表 18-2 x86: 作業マップ: 対話型インストールを実行する

作業	説明	参照先
必要な情報を収集する	チェックリストに従ってワークシートに記入しながら、Solaris ソフトウェアのインストールに必要なすべての情報を収集します。	第 6 章
(アップグレードのみ) システムをアップグレードするための準備をする	システムのバックアップをとります。	『Solaris のシステム管理 (基本編)』
(省略可能) システム構成情報を事前設定する	sysidcfg ファイルやネームサービスを使って、システムのインストール情報(locale など)を事前設定することができます。これによって、インストール時にこの情報を入力するように求められなくなります。	第 7 章
(省略可能) ネットワーク上でインストールするためのシステムを設定する	リモートの 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。	第 15 章
インストールまたはアップグレードを行う	システムをブートし、プロンプトに従って Solaris ソフトウェアをインストールまたはアップグレードします。	190 ページの「x86: Solaris suninstall 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には」
(アップグレードのみ) アップグレード後に必要な作業を行う	アップグレードで失われたローカル変更がある場合は、それを復元します。	196 ページの「Solaris suninstall プログラムによるアップグレード後に必要な作業」

### ▼ x86: Solaris suninstall 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には

1. **CD-ROM** ドライブとネットイメージのどちらを使用してソフトウェアをインストールするかを決定します。
  - CD-ROM ドライブを使用する場合、Solaris SOFTWARE 1 of 2 CD (x86 版) を挿入します。
  - ネットイメージを使用する場合、インストールメディアが置かれているディレクトリへ移動します。必要に応じて、ネットワーク管理者に場所をたずねてください。次にコマンド例を示します。

```

% cd /net/install-svr/export/Solaris_9/x86

```
2. システムのブート方法を決定します。

- Solaris INSTALLATION CD からブートする場合は、CD を挿入します。システムの BIOS は、CD からのブートをサポートするものでなければなりません。
- 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、システムのコモドールディスクドライブに Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) を挿入します。

---

**x86** のみ – 付録 I の手順に従って Solaris DVD (x86 版) または Solaris SOFTWARE 2 of 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. システムをシャットダウンし電源を切ってから、また入れ直してシステムをブートします。

メモリーテストとハードウェア検出が実行されます。画面が再表示されます。

- Solaris SOFTWARE 1 of 2 CD (x86 版) および Solaris SOFTWARE 2 of 2 CD (x86 版) を使用する場合は、次のメッセージが表示されます。

```
SunOS - x86 Platform Edition Primary Boot Subsystem, vsn 2.0
```

続いて、次のようなメッセージが表示されます。

```
SunOS Secondary Boot version 3.00
```

```
Solaris x86 Platform Edition Booting System
```

```
Running Configuration Assistant...
```

- PXE ネットワークブートを使用してネットワーク上でブートする場合は、次のメッセージが表示されます。

```
Solaris network boot...
```

続いて、次のようなメッセージが表示されます。

```
SunOS Secondary Boot version 3.00
```

```
Solaris x86 Platform Edition Booting System
```

```
Running Configuration Assistant...
```

- Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) を使用する場合は、次のような情報が表示されます。

```
Solaris Boot Sector                               Version 1
```

続いて、次のようなメッセージが表示されます。

```
Solaris for x86 - FCS DCB                         Version 1.242
```

```
loading /solaris/boot.bin
```

画面が再表示され、次のようなメッセージが表示されます。

```
SunOS Secondary Boot version 3.00
```

```
Solaris x86 Platform Edition Booting System
```

```
Running Configuration Assistant...
```

4. 「**Solaris Device Configuration Assistant**」画面が表示されたら、「**Continue**」を選択します。

「**Bus Enumeration**」画面に次のメッセージが表示されます。

```
Determining bus types and gathering hardware configuration data ...
```

「**Scanning Devices**」画面が表示されます。システムデバイスのスキャンが行われます。スキャンが終了すると、「**Identified Devices**」画面が表示されます。

5. 変更する必要があるか確認します。

- 変更しない場合は、「**Continue**」を選択します。
- 変更する場合は、「**changes**」を選択します。

「**Loading**」画面に、システムをブートするためにロードされたドライバに関するメッセージが表示されます。数秒後に「**Boot Solaris**」画面が表示されます。

6. 「**Boot Solaris**」画面で「**CD**」、「**Net**」、または「**Disk**」を選択し、「**Continue**」を選択します。

デフォルトのブートディスクが、システムのインストールまたはアップグレードに必要な条件を満たしているかどうかを検査されます。



---

注 - システムに既存の Service パーティションを保持する場合、「Create Solaris fdisk Partition」画面で適切なオプションを選択します。Service パーティションの保持については、48 ページの「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」を参照してください。

インストールの開始前にシステムに Service パーティションを復元した場合、Solaris オペレーティング環境をアップグレードできない可能性があります。

---

7. システム構成の質問に答えます。
  - すべてのシステム構成情報をあらかじめ構成してある場合は、手順 8 へ進みます。
  - システム構成情報の事前設定を行っていない場合は、52 ページの「インストール用のワークシート」または 58 ページの「アップグレード用のワークシート」を参照して、システム構成の質問に答えます。数秒後に「The Solaris Installation Program」画面が表示されます。
8. 「Continue」を選択します。  
「kdmconfig - Introduction」画面が表示されます。
9. 「Continue」を選択します。  
「kdmconfig - View and Edit Window System Configuration」画面が表示されます。
10. (省略可能) 「kdmconfig - View and Edit Window System Configuration」画面の構成情報を確認し、必要な場合は変更します。
11. 構成情報の確認が終わったら、「No changes needed - Test/Save and Exit」を選択し、「Continue」を選択します。  
「kdmconfig Window System Configuration Test」画面が表示されます。
12. 「Continue」を選択します。  
画面が再表示され、「kdmconfig Window System Configuration Test palette and pattern」画面が表示されます。
13. ポインタを移動でき、パレット上に示されている色が正しく表示されていることを確認します。
  - 色が正しく表示されていない場合は、「No」をクリックします。可能な場合、キーボードのいずれかのキーを押すか、「kdmconfig Window System Configuration Test」画面が自動的に終了するのを待ちます。そして、一連の色が正しく表示され、ポインタを正しく移動できるまで、手順 10 から手順 13 を繰り返します。
  - 一連の色が正しく表示されている場合は、「Yes」をクリックします。Solaris suninstall プログラムが始まります。

14. システム構成に関する残りの質問が表示される場合は、それらに答えます。
- すべてのシステム構成情報が事前設定されている場合は、構成情報の入力は求められません。
  - システム構成情報の事前設定を行っていない場合は、52 ページの「インストール用のワークシート」または 58 ページの「アップグレード用のワークシート」を参照して、システム構成の質問に答えます。
15. システムのリポートとディスクの取り出しを自動的に行うかを決定します。

---

注 - PXE を使用してネットワークからブートする場合、手動リポートを選択します。システムがその起動時にネットワークからブートしないように設定する必要があります。ネットワークブートを無効にするには、リポート時にシステムの BIOS 設定ツールかネットワークアダプタの構成設定ツールを使用してください。

---

16. 画面に示される手順に従って、**Solaris** ソフトウェアをシステムにインストールします。
- フラッシュアーカイブをインストールしない場合には、インストールを完了します。アップグレードを実行している場合、またはソフトウェアを追加インストールする場合には、手順 17 に進みます。
- Solaris ソフトウェアのインストールが終了すると、システムは自動的にリポートするか、または手動でリポートするようにプロンプトが表示されます。
- 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
- フラッシュアーカイブをインストールする場合には、作業を続行します。
- a. 「フラッシュアーカイブの検索方法 (Flash Archive Retrieval Method)」画面で、フラッシュアーカイブのロケーションを選択します。
- 選択された媒体に応じて、Solaris suninstall プログラムインストールが続行するためのプロンプトが表示されます。
- b. プロンプトに従って次の情報を入力します。

選択された媒体	プロンプト
HTTP	フラッシュアーカイブのアクセスに必要な URL 情報とプロキシ情報を指定します。

選択された媒体	プロンプト
FTP	FTP サーバーと、フラッシュアーカイブのパスを指定します。FTP サーバーへのアクセスに必要なユーザーおよびパスワードの情報を指定します。FTP サーバーへのアクセスに必要なプロキシ情報を指定します。
ネットワークファイルシステム	フラッシュアーカイブが置かれているネットワークファイルシステムのパスを指定します。アーカイブのファイル名も指定できます。
ローカルファイル	フラッシュアーカイブが置かれているローカルファイルシステムのパスを指定します。
ローカルテープ	フラッシュアーカイブが置かれているローカルテープデバイスとテープ上の位置を指定します。
ローカルデバイス	フラッシュアーカイブが置かれているローカルデバイス、パス、ファイルシステムの種類を指定します。

- c. 「フラッシュアーカイブの選択 (**Flash Archive Selection**)」画面で、「新規 (**New**)」を選択すると、フラッシュアーカイブを追加インストールできます。アーカイブをそれ以上インストールしない場合は、インストールを完了させるために「継続 (**Continue**)」を選択します。

Solaris ソフトウェアのインストールが終了すると、システムは自動的にリブートするか、または手動でリブートするようにプロンプトが表示されます。

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

17. アップグレードを行った場合は、保存されなかったローカル変更があればそれを復元します。詳細は、196 ページの「**Solaris suninstall** プログラムによるアップグレード後に必要な作業」を参照してください。
18. (省略可能) 追加のソフトウェアをインストールする方法は、『Solaris のシステム管理 (基本編)』を参照してください。

---

## Solaris suninstall プログラムによる アップグレード後に必要な作業

アップグレードを行うと、Solaris suninstall プログラムは、既存システムに対して行ったローカルなソフトウェア変更と新しい Solaris ソフトウェアをマージします。ただし場合によっては、マージが行われないことがあります。したがって、システムのアップグレードが終了したら、保存されなかったローカル変更を復元しなければならない場合があります。



---

注意 - 保存されなかったローカル変更を復元しないと、システムがブートされないことがあります。

---

### ▼ アップグレード後にローカル変更を復元する方法

1. 次のファイルの内容を確認して、**Solaris suninstall** プログラムによって保存されなかったローカル変更を復元する必要があるかどうかを判断します。

```
/a/var/sadm/system/data/upgrade_cleanup
```

2. 保存されなかったローカル変更があれば、復元します。
3. リブートします。

```
# reboot
```

## 第 19 章

# フラッシュインストール機能 (トピック)

---

以下の章では、フラッシュアーカイブを作成する方法と、フラッシュアーカイブを使用して複数のシステムに Solaris オペレーティング環境をインストールする方法について説明します。

第 20 章	フラッシュ (Flash) アーカイブの作成とインストールの概要と計画について説明します。
第 21 章	フラッシュアーカイブの作成手順を説明します。
第 22 章	フラッシュアーカイブを使用してシステムをインストールする手順を説明します。また、アーカイブの管理方法についても説明します。
第 23 章	<code>flar create</code> コマンドの構文およびオプションについて説明します。また、フラッシュアーカイブ情報のキーワードについても説明します。



## 第 20 章

---

# フラッシュ (概要と計画)

---

この章では、フラッシュインストール機能の概要を述べるとともに、使用している環境におけるフラッシュインストールの計画に必要な情報を記載します。

- 199 ページの「フラッシュの概要」
- 204 ページの「フラッシュインストールの計画」

---

## フラッシュの概要

フラッシュインストール機能を利用すると、マスターシステムと呼ばれるシステム上の Solaris オペレーティング環境を、単一の参照用インストールイメージとして使用できます。そのインストールをクローンシステムと呼ばれる多数のシステムに複製できます。システムの全ファイルを上書きするフラッシュ初期インストール、または2つのシステムイメージの差分のみを更新する更新用フラッシュアーカイブでクローンシステムを複製できます。差分更新では、指定されたファイルだけが変更されます。また、更新の対象は、以前のマスターイメージとの一貫性を保持するソフトウェアを含むシステムだけになります。

## 初期インストールでクローンシステムにインストールする

初期インストールでは、Solaris Web Start、Solaris suninstall プログラム、カスタム JumpStart、Solaris Live Upgrade、または WAN ブートのいずれかを使用して、フラッシュアーカイブをマスターシステムにインストールできます。すべてのファイルが上書きされます。フラッシュインストール処理は、5つの部分に分けられます。

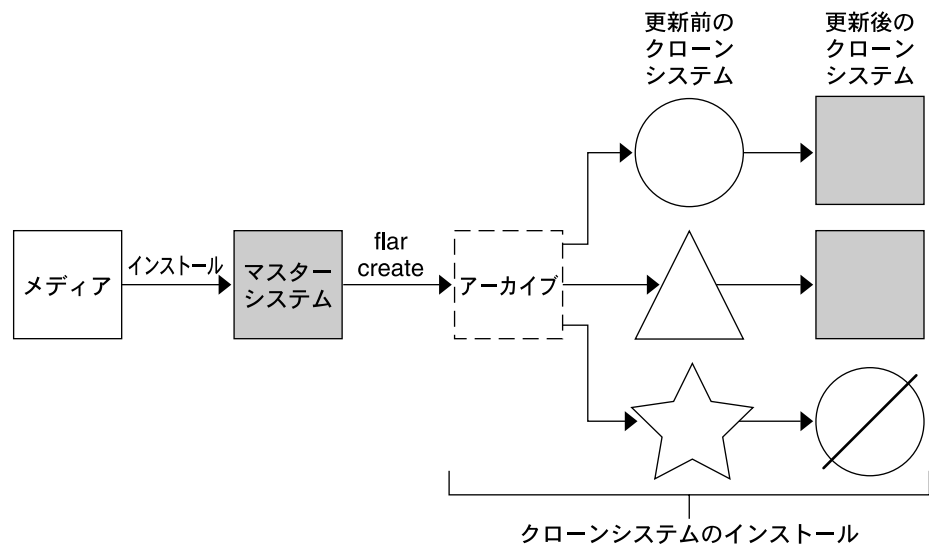
1. マスターシステムをインストールします。システムを1台選択し、Solaris の任意のインストール方法で Solaris オペレーティング環境およびその他のソフトウェアをインストールします。

2. (省略可能) カスタムスクリプトを準備して、インストールの前または後にクローンシステムの再構成またはカスタマイズを実行します。
3. フラッシュアーカイブを作成します。削除された不要なファイル以外は、フラッシュアーカイブにはマスターシステム上のすべてのファイルのコピーが含まれます。
4. フラッシュアーカイブをクローンシステムへインストールします。マスターシステムとクローンシステムとで、カーネルアーキテクチャーを同じにする必要があります。  
システムにフラッシュアーカイブをインストールすると、アーカイブのすべてのファイルがそのシステムにコピーされます。新しくインストールされたシステムのインストール構成は、オリジナルのマスターシステムと同じになります (このため、クローンシステムと呼ばれます)。スクリプトを使用してカスタマイズすることも可能です。
5. (省略可能) マスターイメージのコピーを保存します。差分アーカイブを作成する場合、マスターイメージが利用可能であり、クローンシステムにインストールされたイメージと同一でなければなりません。

手順については、214 ページの「マスターシステムへのインストール」を参照してください。

図 20-1 に、初期インストールでのクローンシステムのインストール方法を示します。すべてのファイルが上書きされます。





- 任意のオペレーティング環境が稼動しているシステム
- △ オペレーティング環境の存在しないシステム
- ☆ 異なるアーキテクチャのシステム
- ⊗ 更新の失敗

図 20-1 Solaris フラッシュの初期インストール

## 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更新する

保持しているクローンシステムを更新する場合、更新前のマスターイメージと更新後のマスターイメージとの差異のみを含む差分アーカイブを作成できます。差分アーカイブを使用してクローンシステムを更新する場合、差分アーカイブ内のファイルだけが変更されます。カスタム JumpStart インストールまたは Solaris Live Upgrade においてのみ、フラッシュ差分アーカイブのインストールを選択できます。更新では、次の 5 つのプロセスを実行します。

1. マスターシステムで変更の準備を行います。変更を加える前に、マスターシステムで元のアーカイブの複製を作成する必要があります。

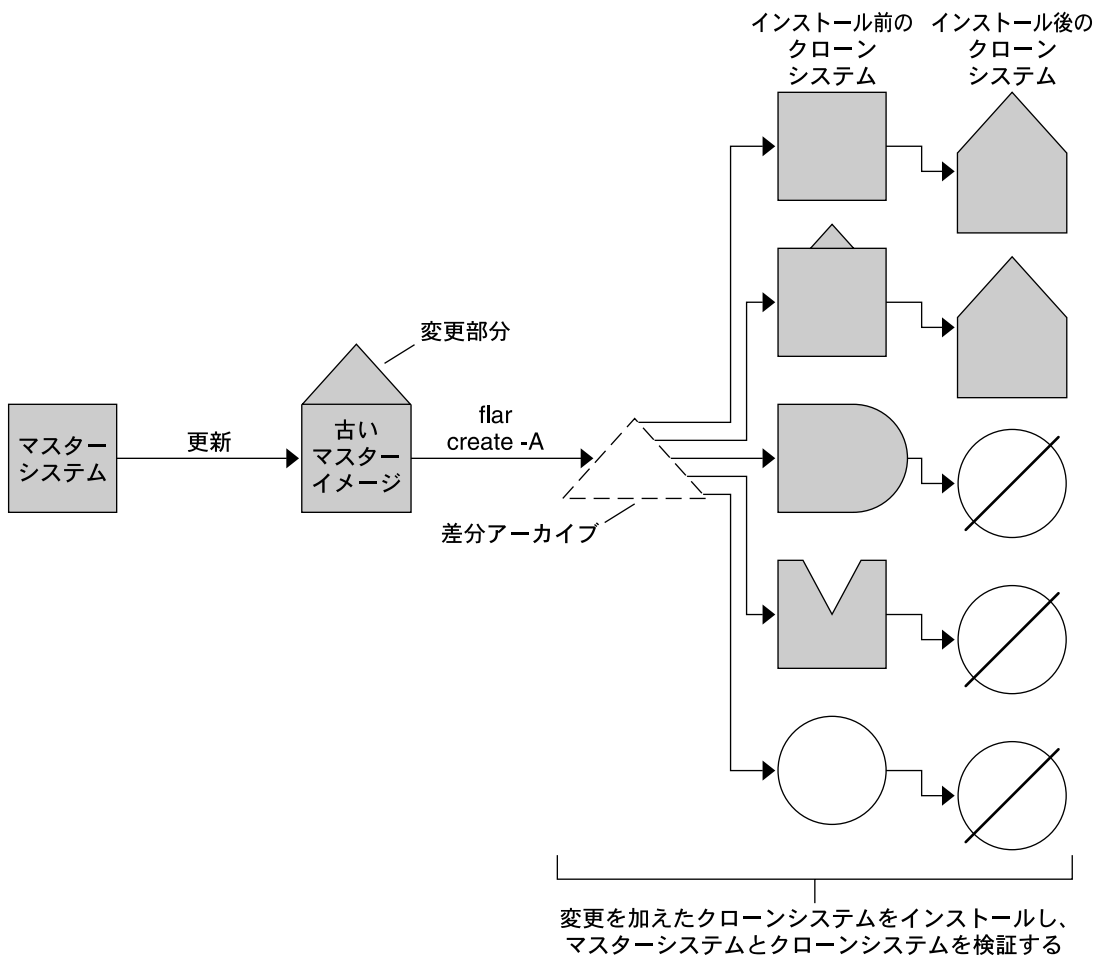
---

注 - マスターシステムで元のアーカイブの複製を作成していない場合、2つのシステムイメージの相違部分により、大規模な差分アーカイブが生成されることがあります。このため、差分アーカイブのインストールにかなりの時間がかかります。この場合、フルアーカイブを使用して初期インストールを実行してください。

---

2. (省略可能) カスタムスクリプトを準備して、インストールの前または後にクローンシステムの再構成またはカスタマイズを実行します。
3. 保存済みの更新前マスターイメージのコピーが存在するディレクトリをマウントします。この2番目のイメージは、2つのシステムイメージの比較に使用します。以下の方法で、イメージにアクセスします。
  - Solaris Live Upgrade ブート環境からマウントする
  - NFS 経由でクローンシステムからマウントする
  - `ufsrestore` コマンドを使用して、バックアップから復元する
4. `flar create` コマンドで `-A` オプションを指定して、差分アーカイブを作成します。
5. カスタム `JumpStart` を使用して、クローンシステムに差分アーカイブをインストールします。あるいは、`Solaris Live Upgrade` を使用して、非アクティブブート環境に差分アーカイブをインストールします。

図 20-2 に、差分アーカイブの作成およびインストール手順を示します。更新により、マスターイメージにいくらかの変更が加えられます。変更には、いくつかのファイルの追加、再構成、または削除のような単純なものもあれば、パッチの適用のような複雑なものもあります。更新されたマスターイメージは、更新前のマスターイメージと比較されます。2つのイメージの相違部分が、差分アーカイブになります。このアーカイブは、更新前のマスターイメージを現在使用している他のクローンシステムの更新にも使用できます。クローンシステムがすでに変更してあるか、更新前のマスターイメージを稼動していない場合、更新は失敗します。クローンシステムに多数の変更を加える場合には、いつでも初期インストールを実行できます。



- マスターと全く同一の複製
- ◒ わずかな変更を含むマスターの複製
- ◐ 付加されたファイルを含むマスターの複製
- ◑ いくつかのファイルを除いたマスターの複製
- 異なるマスターから作成されたものあるいは別にインストールされたもの
- ⊗ 更新の失敗

図 20-2 Solaris フラッシュによる更新

---

## フラッシュインストールの計画

フラッシュアーカイブの作成とインストールを行う前に、システムに Solaris オペレーティング環境をどのようにインストールするかについていくつか決定しておくべきことがあります。最初にシステムをインストールする場合、アーカイブ全体をインストールする必要があります。これが初期インストールとなります。アーカイブを使用してインストールを実行した後で、差分アーカイブを使用してシステムを更新できます。差分アーカイブでは、2つのアーカイブの相違部分のみがインストールされます。

## マスターシステムへの初期インストールの設計

フラッシュインストールの最初の作業は、各クローンシステムで行いたい構成と同じ構成でマスターシステムにインストールすることです。アーカイブは、任意の Solaris のインストール方法でマスターシステムにインストールできます。インストールは、Solaris オペレーティング環境のサブセットであっても完全なインストールであってもかまいません。このインストールの完了後、ソフトウェアの追加または削除、構成ファイルの変更を行います。

マスターシステムとクローンシステムとで、カーネルアーキテクチャを同じにする必要があります。たとえば、sun4u アーキテクチャのマスターシステムから作成されたアーカイブを使用できるのは、sun4u アーキテクチャのクローンをインストールする場合だけです。

各クローンシステムと全く同じ構成で、マスターシステムをインストールする必要があります。マスターシステムへのインストールの設計は、次の要素で決まります。

- クローンシステムにインストールするソフトウェア
- マスターシステムとクローンシステムに接続されている周辺装置
- マスターシステムとクローンシステムのアーキテクチャ

その後、システム上のソフトウェアをカスタマイズできます。208 ページの「アーカイブのファイルとディレクトリのカスタマイズ」を参照してください。マスターシステムは、クローンシステムの更新のために変更および設定が可能です。207 ページの「フラッシュアーカイブの作成の計画」を参照してください。

## マスターシステムのカスタマイズ

Solaris の任意のインストール方法でマスターシステムに Solaris オペレーティング環境をインストールした後、必要に応じてソフトウェアの追加や削除、およびシステム構成情報の変更を行います。

- ソフトウェアの削除 - クローンシステムにインストールする必要がないソフトウェアを削除することができます。マスターシステムにインストールされているソフトウェアのリストを表示するには、Solaris Product Registry を使用します。手順

の詳細は、『Solaris のシステム管理 (基本編)』を参照してください。

- ソフトウェアの追加 - Solaris リリースに含まれるソフトウェアをインストールできます。また、Solaris オペレーティング環境の一部として提供されていないソフトウェアを追加することも可能です。マスターシステムにインストールされたすべてのソフトウェアはフラッシュアーカイブに含まれ、クローンシステムにインストールされます。
- 構成ファイルの変更 - マスターシステム上の構成ファイルを変更できます。たとえば、`/etc/inet/inetd.conf` ファイルを変更することにより、システムが実行するデーモンを制限できます。加えられた変更はすべてフラッシュアーカイブの一部として保存され、クローンシステムにインストールされます。

アーカイブの作成時に、さらにカスタマイズを行うことができます。たとえば、アーカイブに含める必要のない大規模なデータファイルを除外できます。この概要については、208 ページの「アーカイブのファイルとディレクトリのカスタマイズ」を参照してください。

クローンシステムを更新する場合、差分アーカイブを作成できます。207 ページの「フラッシュアーカイブの作成の計画」を参照してください。

## SPARC および x86 システム用のアーカイブの作成

フラッシュインストールを使用して SPARC システムと x86 システムの両方に Solaris ソフトウェアをインストールしたい場合は、それぞれのプラットフォーム用のフラッシュアーカイブを個別に作成する必要があります。SPARC システムにインストールするには、SPARC マスターシステムから作成されたフラッシュアーカイブを使用します。x86 マスターシステムにインストールするには、x86 マスターシステムから作成されたフラッシュアーカイブを使用します。

## SPARC: マスターシステムに存在しない周辺装置のサポート

マスターシステムにインストールするドライバの選択は、マスターシステムとクローンシステムの両方に取り付けられた周辺装置、およびインストールされたソフトウェアグループのタイプによって異なります。クローンシステムが、マスターシステムとは異なる周辺装置を保持している場合、アーカイブを作成する前に、マスターシステムに適切なドライバをインストールする必要があります。

---

注 - Solaris ソフトウェアをインストールする場合、いくつかのハードウェア固有のドライバは、システム上でサポートするハードウェアが検出された場合のみインストールされることがあります。

---

コアシステムサポート (Core) ソフトウェアグループ、エンドユーザーシステムサポート (End User) ソフトウェアグループ、開発者システムサポート (Developer) ソフトウェアグループ、または全体ディストリビューション (Entire) ソフトウェアグループ

をマスターシステムにインストールした場合、マスターシステムがサポートするのはインストール時にマスターシステムに接続されていた周辺装置だけです。マスターシステムに周辺装置が接続されていない場合でも、全体ディストリビューションとOEMサポート(Entire plus OEM) ソフトウェアグループをインストールすることにより、周辺装置のサポートをマスターシステムにインストールできます。システムに存在するハードウェアに関係なく、すべてのドライバがインストールされます。また、必要な周辺装置をサポートするパッケージを選択してインストールすることも可能です。

全体ディストリビューションとOEMサポート (Entire plus OEM) ソフトウェアグループ以外のソフトウェアグループをマスターシステムにインストールする場合は、クローンシステムにインストールする必要がある適切なドライバすべては保持していない可能性があります。たとえば、GX CG6 フレームバッファを持つマスターシステムに全体ディストリビューション (Entire) ソフトウェアグループをインストールした場合、GX CG6 フレームバッファードライバだけがインストールされます。インストールするクローンシステムすべてが GX CG6 フレームバッファを保持するか、フレームバッファを一切保持しない場合、この状況で問題は発生しません。

Elite 3D など、他のフレームバッファがインストールされたクローンシステムにアーカイブをインストールする場合、クローンシステムはフレームバッファサポートを保持しません。このため、全体ディストリビューションとOEMサポート (Entire plus OEM) ソフトウェアグループをインストールしてすべてのフレームバッファを含むアーカイブを作成するか、適切なフレームバッファードライバパッケージをマスターシステムに追加して該当するフレームバッファードライバをインストールする必要があります。

以下のいずれかの方法で、マスターシステムとは異なる周辺装置のサポートをクローンシステムにインストールできます。

- 全体ディストリビューションとOEMサポート(Entire plus OEM) ソフトウェアグループをインストールする – 全体ディストリビューションとOEMサポート(Entire plus OEM) ソフトウェアグループは、利用できる最大のソフトウェアグループです。このグループには、Solaris オペレーティング環境のすべてのパッケージが含まれています。マスターシステムにこのグループをインストールすると、マスターシステムには Solaris リリースに含まれるすべてのドライバが含まれることになります。全体ディストリビューションと OEM サポート (Entire plus OEM) ソフトウェアグループをインストールしたマスターシステムから作成されるフラッシュアーカイブは、インストールした Solaris オペレーティング環境のリリースがサポートする周辺装置を持つクローンシステムで機能します。

全体ディストリビューションと OEM サポート (Entire plus OEM) ソフトウェアグループをマスターシステムにインストールすると、ほかの周辺装置構成との互換性を保つことができます。しかし、全体ディストリビューションと OEM サポート (Entire plus OEM) ソフトウェアグループには、1G バイトを超えるディスク容量が必要です。クローンシステムの中には、全体ディストリビューションと OEM サポート (Entire plus OEM) ソフトウェアグループのインストールに必要な容量を確保できないものもあるでしょう。

- パッケージを選択してインストールする: マスターシステムにインストールする際に、マスターシステムとクローンシステムに必要なパッケージを追加インストールできます。特定のパッケージを選択することにより、マスターシステムまたはク

ローンシステムに存在することが明らかな周辺装置のサポートだけをインストールすることができます。

## フラッシュアーカイブの作成の計画

マスターシステムからのアーカイブの作成または2つのシステムイメージからの差分アーカイブの作成が可能です。差分アーカイブでは、2つのイメージの相違部分のみがインストールされます。

### 初期インストール用フラッシュアーカイブの作成を計画する

マスターシステムへのインストールの完了後、フラッシュインストールの次の作業として、フラッシュアーカイブを作成します。マスターシステム上のファイルは、さまざまな識別情報と共にフラッシュアーカイブにコピーされます。フラッシュアーカイブは、マルチユーザーモードまたはシングルユーザーモードでマスターシステムが稼働している間に作成できます。フラッシュアーカイブは、次のメディアのうちの1つからブートした後でも作成できます。

- Solaris DVD
- Solaris SOFTWARE 1 of 2 CD
- Solaris SOFTWARE と Solaris LANGUAGES CD のイメージ

### 更新用フラッシュ差分アーカイブの作成を計画する

保持しているクローンシステムを更新する場合、更新前のマスターイメージと更新後のマスターイメージとの変更点のみを含む差分アーカイブを作成できます。クローンにインストールされた元のソフトウェアが稼働しているシステムをイメージの1つにできます。これを使用しない場合は、保存された更新前のマスターイメージのコピーにアクセスする必要があります。このイメージに該当する変更を適用して、更新を行います。アクセスするもう1つのイメージは、比較に使用します。2つのイメージの相違部分が、差分アーカイブになります。カスタム JumpStart インストール方法を使用して、フラッシュ差分アーカイブをインストールします。あるいは、Solaris Live Upgrade を使用して、非アクティブブート環境に差分アーカイブをインストールします。差分アーカイブを使用してクローンシステムを更新すると、差分アーカイブ内に存在するファイルだけがクローンシステム上で変更されます。インストールの前または後に、スクリプトを使用してアーカイブをカスタマイズできます。再構成を行う場合、これは特に有効です。

次のいずれかの方法でアクセス可能にするため、初期インストールの実行後に更新前のマスターイメージを保存する必要があります。

- `lumount` コマンドを使用してディレクトリにマウントされる、Solaris Live Upgrade ブート環境。Solaris Live Upgrade ブート環境の詳細は、第 33 章を参照してください。

- ルートアクセス権を使用して NFS 経由でマウントされるクローンシステム
- `ufsdump` コマンドを使用して復元可能なシステムのバックアップ

手順については、225 ページの「更新されたマスターイメージを使用してフラッシュ差分アーカイブを作成する方法」を参照してください。

## アーカイブのファイルとディレクトリのカスタマイズ

フラッシュアーカイブの作成時に、マスターシステムからコピーされるファイルやディレクトリの一部分をアーカイブから除外することができます。アーカイブからディレクトリを除外する際に、そのディレクトリ内にある特定のファイルやサブディレクトリを除外せずにアーカイブに含めることも可能です。たとえば、`/a/aa/bb/c` にあるすべてのファイルやディレクトリを除外するアーカイブを作成できます。このアーカイブを作るときに、`bb` サブディレクトリの内容だけを除外せずにアーカイブに含めることができます。これにより、アーカイブに含まれるのは `bb` サブディレクトリの内容だけになります。



注意 - `flar create` のファイル除外オプションは、注意して使用してください。一部のディレクトリを除外する際、気づかずにシステム構成ファイルなどの他のファイルがアーカイブに残ってしまう場合があります。この場合、システムの整合性が損なわれるため、インストールが失敗してしまいます。ディレクトリやファイルの除外は、大規模なデータファイルなど、システムを破綻させることなく容易に削除可能なデータに対して行うのが最善です。

次の表に、ファイルやディレクトリの除外および追加用の `flar create` コマンドのオプションを示します。

指定方法	除外用のオプション	追加用のオプション
ディレクトリまたはファイルの名前を指定します。	<code>-x exclude_dir/filename</code>	<code>-y include_dir/filename</code>
対象のファイルやディレクトリのリストを含むファイルを使用します。	<code>-x list_filename</code> <code>-z list_filename</code>	<code>-f list_filename</code> <code>-z list_filename</code>

これらのオプションの詳細は、表 23-7 を参照してください。

アーカイブのカスタマイズ例については、221 ページの「例 — 初期インストール用アーカイブの作成」を参照してください。



## スクリプトを使用したアーカイブのカスタマイズ

マスターシステムへのソフトウェアのインストールが完了したら、作成、インストール、ポストインストール、および初回再起動時に特別なスクリプトを実行できます。これらのスクリプトを使用して、以下を実行できます。

- クローンシステム上でアプリケーションを構成します。一部の複雑でない構成に対して、カスタム JumpStart スクリプトを使用できます。より複雑な構成の場合、マスターシステム上で、またはインストールの前か後にクローンシステム上で、特別な構成ファイル処理が必要な場合があります。
- クローンシステム上でのローカルなカスタマイズを保護します。ローカルのプリインストールおよびポストインストールスクリプトは、クローン上に存在し、ローカルのカスタマイズがフラッシュソフトウェアにより上書きされないように保護する働きをします。
- クローン不可のホスト依存データを識別して、アーカイブをホスト非依存にできます。ホスト非依存にするには、この種のデータを変更するか、アーカイブから除外します。ホスト依存データの例は、ログファイルです。
- 作成時に、アーカイブ内でソフトウェアの整合性を確認します。
- クローンシステム上のインストールを検査します。

## カスタムスクリプト作成のガイドライン

再起動スクリプト以外のスクリプトを作成する場合、スクリプトがオペレーティング環境やシステムを破壊してしまうことがないように、以下のガイドラインに従ってください。これらのガイドラインに従うことで、Solaris Live Upgrade を使用して、オペレーティング環境インストール用の新規ブート環境が作成可能になります。新規ブート環境は、現行システムの稼働中にアーカイブを使用してインストールできます。

---

注 - これらのガイドラインは、デーモンの実行やルート (/) ファイルシステムへの他のタイプの変更を実行可能にする再起動スクリプトに適用されるものではありません。

---

- スクリプトは、稼働中のシステムに影響を与えてはいけません。現在稼働中のオペレーティング環境が、フラッシュアーカイブのインストール時に稼働しているオペレーティング環境ではない場合もあります。
- スクリプトで、いかなるデーモンプロセスも開始または停止させないでください。
- スクリプトで、オペレーティング環境に依存するコマンド ( ps, truss, uname など) の出力に依存しないでください。これらのコマンドは、稼働中のシステムに関する情報をレポートします。
- スクリプトで、いかなるシグナルも送信しないでください。送信した場合、現在稼働中のプロセスが影響を受けます。
- スクリプトでは、シェルスクリプトの実行を円滑にする標準の UNIX コマンド (expr, cp, ls など) を使用できます。

Solaris Live Upgrade の概要については、第 33 章を参照してください。

## フラッシュアーカイブのセクション

フラッシュアーカイブには、次のセクションが含まれます。一部のセクションを使用して、アーカイブの識別やカスタマイズ、およびインストールの状態情報の表示を実行できます。各セクションの詳細は、第 23 章を参照してください。

表 20-1 フラッシュアーカイブのセクション

セクション名	情報提供のみ	説明
Cookie	X	最初のセクションには、ファイルをフラッシュアーカイブとして識別する cookie が含まれます。
識別		2 番目のセクションには、アーカイブについての識別情報を値に持つキーワードが含まれます。一部の識別情報は、アーカイブソフトウェアにより提供されます。その他の特定の識別情報は、flar create コマンドのオプションを使用して追加できます。
ユーザー定義		このセクションは、識別セクションの次に存在します。これらのセクションを定義および挿入して、アーカイブをカスタマイズできます。フラッシュアーカイブは、ユーザーが定義したセクションは処理しません。たとえば、このセクションには、アーカイブの説明や、アプリケーションの整合性をチェックするスクリプトなどを含めることができます。
マニフェスト	X	このセクションは、フラッシュ差分アーカイブに対して生成され、クローンシステムの検証に使用されます。マニフェストセクションには、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あり、内部形式でファイルをリスト表示します。スクリプトの記述には使用できません。
配置前、配置後、再起動	X	このセクションには、オペレーティング環境イメージのインストール前または後にフラッシュソフトウェアが使用する内部情報が含まれます。指定したスクリプトはすべて、このセクションに含まれます。
サマリー		このセクションには、アーカイブ作成に関するメッセージが含まれます。セクションには、配置前および配置後スクリプトのアクティビティも記録できます。このセクションに出力を送信するスクリプトを記述することにより、このセクション内でインストールの成功を確認できます。
アーカイブファイル	X	アーカイブファイルセクションには、マスターシステムから収集されたファイルが含まれます。

## 初期インストールでのアーカイブの作成時期

アーカイブは、システムができるだけ静的な状態である時に作成してください。

マスターシステムにソフトウェアをインストールした後でソフトウェアを構成する前にアーカイブを作成します。たとえば、Solaris ボリュームマネージャは、ファイルシステム外部にメタ情報を格納します。起動ディスクをカプセル化およびミラー化している場合、アーカイブは作成できません。アーカイブ作成ではファイルシステム外部のメタ情報にアクセスできないため、構成前にアーカイブを作成する必要があります。

## フラッシュアーカイブの保存先

フラッシュアーカイブを作成した後、そのアーカイブをマスターシステムのハードディスクまたはテープに保存できます。保存後は、任意のファイルシステムまたは媒体へそのアーカイブをコピーできます。

- ネットワークファイルシステム (NFS) サーバー
- HTTP または HTTPS サーバー
- FTP サーバー
- テープ
- CD、DVD
- フロッピーディスク
- インストールしたいクローンシステムのローカルドライブ

## アーカイブの圧縮

フラッシュアーカイブを作成する際は、`compress` (1) ユーティリティを使用して、そのアーカイブが圧縮ファイルとして保存されるように指定することができます。圧縮されたアーカイブはディスク容量が少なく済み、ネットワークを介してアーカイブをインストールする場合の負荷も減ります。

## フラッシュアーカイブのインストールの計画

フラッシュインストールの最後の作業は、クローンシステムへのフラッシュアーカイブのインストールです。

クローンシステムへのフラッシュアーカイブのインストールには、次のインストール方法を使用できます。

Solaris DVD または Solaris INSTALLATION CD に含まれている Solaris Web Start では、以下の媒体に格納されているフラッシュアーカイブのインストールが可能です。

- ディスク (DVD または CD)
- ネットワークファイルシステム (NFS) サーバー
- HTTP サーバー
- FTP サーバー
- ローカルテープ

インストール方法の詳細は、170 ページの「SPARC: Solaris Web Start によるインストールまたはアップグレード」または 175 ページの「x86: Solaris Web Start によるインストールまたはアップグレード」を参照してください。

Solaris SOFTWARE 1 of 2 CD に含まれている Solaris suninstall プログラムでは、以下の媒体に格納されているフラッシュアーカイブのインストールが可能です。

- ネットワークファイルシステム (NFS) サーバー
- HTTP サーバー
- FTP サーバー
- ローカルファイル
- ローカルテープ
- CD を含むローカルデバイス

インストール方法の詳細は、186 ページの「SPARC: Solaris suninstall プログラムによるインストールまたはアップグレード」または 189 ページの「x86: Solaris suninstall プログラムによるインストールまたはアップグレード」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、以下の媒体に格納されているフラッシュアーカイブのインストールが可能です。

- ネットワークファイルシステム (NFS) サーバー
- HTTP または HTTPS サーバー
- FTP サーバー
- ローカルテープ
- ローカルデバイス (DVD または CD)
- ローカルファイル

インストール方法の詳細は、315 ページの「カスタム JumpStart インストールを使用して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。

Solaris Live Upgrade を利用すると、次の場所に保存されているフラッシュアーカイブをインストールできます。

- HTTP サーバー
- FTP サーバー
- ネットワークファイルシステム (NFS) サーバー
- ローカルファイル
- ローカルテープ
- ローカルデバイス (DVD または CD)

インストール方法の詳細は、459 ページの「ブート環境へのフラッシュアーカイブのインストール」を参照してください。

## 第 21 章

# フラッシュアーカイブの作成 (作業)

この章では、フラッシュアーカイブを作成する手順を説明します。手順には、マスターシステムにインストールし、その後そのマスターシステムからフラッシュアーカイブを作成する方法が含まれます。クローンシステムにアーカイブをインストール済みの場合、差分アーカイブを作成することも可能です。差分アーカイブの作成時に、更新前のマスターイメージと更新されたイメージの 2 つが比較されます。差分アーカイブでは、2 つのイメージの相違部分のみがインストールされます。さらに、アーカイブの再構成またはカスタマイズを実行するスクリプトの作成手順も示します。

- 213 ページの「作業マップ: フラッシュアーカイブを作成する」
- 214 ページの「マスターシステムへのインストール」
- 215 ページの「カスタムスクリプトの作成」
- 220 ページの「フラッシュアーカイブの作成」

## 作業マップ: フラッシュアーカイブを作成する

表 21-1 作業マップ: 初期インストールでインストールするフラッシュアーカイブを作成する

作業	説明	参照先
マスターマシンに選択したソフトウェア構成をインストールする	ニーズに合った構成を決定し、Solaris の任意のインストール方法を使用してマスターマシンにインストールする。	215 ページの「初期インストールでマスターシステムにインストールする方法」

表 21-1 作業マップ: 初期インストールでインストールするフラッシュアーカイブを作成する (続き)

作業	説明	参照先
(省略可能) カスタマイズスクリプトを作成する	次の作業を行うスクリプトの作成が必要かどうかを判断します。 <ul style="list-style-type: none"> <li>■ アーカイブをカスタマイズまたは再構成する</li> <li>■ クローンシステム上のローカルな変更を保護する</li> </ul>	215 ページの「カスタムスクリプトの作成」
フラッシュアーカイブを作成する	flar create コマンドを使用してアーカイブを作成する。	220 ページの「初期インストール用フラッシュアーカイブの作成方法」
(省略可能) アーカイブのコピーを保存する	将来、比較に使用するためにアーカイブのコピーを保存します。差分アーカイブを使用して、クローンシステムの更新が可能になります。	220 ページの「初期インストール用フラッシュアーカイブの作成方法」

表 21-2 作業マップ: フラッシュアーカイブを作成してクローンシステムを更新する

作業	説明	参照先
マスターイメージの準備を行う	パッケージを追加または削除したり、パッチをインストールして、更新前のマスターイメージに変更を加えます。	214 ページの「マスターシステムへのインストール」
(省略可能) カスタマイズスクリプトを作成する	次の作業を行うスクリプトの作成が必要かどうかを判断します。 <ul style="list-style-type: none"> <li>■ アーカイブをカスタマイズまたは再構成する</li> <li>■ クローンシステム上のローカルな変更を保護する</li> </ul>	215 ページの「カスタムスクリプトの作成」
フラッシュ差分アーカイブを作成する	<ol style="list-style-type: none"> <li>1. 更新前のマスターイメージをマウントします。</li> <li>2. flar create コマンドを使用して2つのイメージを比較し、差分アーカイブを作成します。</li> </ol>	225 ページの「更新されたマスターイメージを使用してフラッシュ差分アーカイブを作成する方法」

## マスターシステムへのインストール

マスターシステムには、クローンシステムで行いたいソフトウェア構成と同じ構成でインストールします。クローンシステムに、初期インストールによりシステム上のすべてのファイルを上書きすることも、2つのイメージの相違部分のみを更新することも可能です。初期インストールでは、任意の Solaris インストール方法を使用してマスターシステムに Solaris オペレーティング環境をインストールします。

アーカイブをクローンシステムにインストール済みの場合、差分アーカイブを使用してシステムの変更部分を更新できます。元のイメージへの変更は、パッチのインストールやパッケージの追加および削除により実行されます。差分アーカイブでは、アーカイブ内で指定されたファイルだけが上書きされます。元のマスターイメージを更新して、差分アーカイブを作成する手順については、225 ページの「更新されたマスターイメージを使用してフラッシュ差分アーカイブを作成する方法」を参照してください。

## ▼ 初期インストールでマスターシステムにインストールする方法

1. インストールしたいシステム構成を決定します。
2. **Solaris** のインストール方法のいずれかを使用して、マスターシステムに **Solaris** オペレーティング環境をインストールします。インストール方法の選択については、第 3 章を参照してください。
3. マスターシステムを次のいずれかの方法でカスタマイズします。

- ソフトウェアを削除する
- ソフトウェアを追加する
- 構成ファイルを変更する
- クローンシステム上にある周辺装置のサポートを追加する

カスタムスクリプトを作成するか、`flar create` コマンドを使用してアーカイブを作成できます。

- カスタムスクリプトの作成方法については、215 ページの「カスタムスクリプトの作成」を参照してください。
- アーカイブの作成方法については、220 ページの「フラッシュアーカイブの作成」を参照してください。

---

## カスタムスクリプトの作成

スクリプトを使用して、アーカイブをカスタマイズできます。カスタムスクリプトは、次の目的で使用します。

- 作成前スクリプトは、作成時にアーカイブの妥当性検査を行い、後でカスタマイズできるようにアーカイブ (特に差分アーカイブ) の準備を行います。このスクリプトを使用して、アーカイブ内にユーザー定義セクションを作成することもできます。
- 配置前スクリプトは、インストール時にアーカイブの妥当性検査を行い、後でカスタマイズできるようにアーカイブの準備を行います。

- 配置後スクリプトは、クローンシステム上で新規システムイメージの再構成を行います。
- 再起動スクリプトは、システムの再起動後に最終的な再構成処理を実行します。

スクリプト作成のガイドラインについては、209 ページの「カスタムスクリプト作成のガイドライン」を参照してください。

## ▼ 作成前スクリプトの作成方法

このスクリプトは、アーカイブの作成時に実行されます。このスクリプトには、さまざまな使用方法があります。

- ソフトウェアの内容および整合性を検証する。整合性に問題がある場合、スクリプトはアーカイブの作成に失敗する。
  - クローンシステム上での将来のカスタマイズに備えてプロダクトの準備を行う。
  - アーカイブの作成時に、他のインストールスクリプトを動的に登録する。
  - フラッシュ作成サマリーファイルにメッセージを追加する。メッセージは短いものでなければならず、スクリプトが起動および終了したこと、およびその結果のみを記録するものでなければならない。結果は、サマリーセクション内で確認できる。
1. 作成前スクリプトを作成します。209 ページの「カスタムスクリプト作成のガイドライン」に記されているガイドラインに従います。
  2. スクリプトを `/etc/flash/precreation` ディレクトリに保存します。

### 例 21-1 作成前スクリプトからの抜粋

以下の例は、作成前スクリプトからの抜粋です。

- サマリーセクションに開始時刻を記録するには、次の例を使用します。

```
echo "MyApp precreation script started">> $FLASHDIR/summary
```

- ソフトウェアの整合性をチェックするには、`flcheck` コマンドを使用します。このコマンドは、コマンド行では使用できません。このコマンドの構文は次のとおりです。

```
flcheck software component files and directories ...| -
```

たとえば、ファイルおよびディレクトリの妥当性検査を実行するには、次の例を使用します。

```
flcheck software component files and directories
If Not in selection - refuse creation
```

```
echo "Myapp Integrity Damage">>$FLASHDIR/summary
```

また、予期しない新規ファイルおよびディレクトリを保持し、アーカイブの作成が失敗しないようにするには、次の例を使用します。



例 21-1 作成前スクリプトからの抜粋 (続き)

```
flcheck software component files and directories
If Not in selection include by force
flinclude software component
```

- 配置スクリプトおよびデータを登録するには、次の例を使用します。

- 次のディレクトリにスクリプトをコピーします。

```
cp predeployment script /etc/flash/predeployment
```

- または、アーカイブの作成時にスクリプトを動的に登録するためには、次のディレクトリにスクリプトをコピーします。

```
cp predeployment script $FLASHDIR/predeployment
```

- ユーザー定義セクションにアプリケーション固有のデータを保存するには、次の例を使用します。

```
cp custom section $FLASHDIR/custom_sections/MyApp
```

- サマリーセクションにインストールの成功を記録するには、次の例を使用します。

```
echo "product one flash preparation started.">>$FLASH_DIR/summary
...
echo "product one flash preparation finished successfully">>$FLASH_DIR/summary
```

例 21-2 作成前スクリプト

```
#!/bin/sh
echo "Test precreation script started">> $FLASH_DIR/summary
cat /opt/TestApp/critical_file_list | flcheck -
if [ $? != 0 ]; then
    echo "Test precreation script failure">> $FLASH_DIR/summary
    exit 1
fi
echo "Test precreation script started">> $FLASH_DIR/summary
/opt/TestApplication/license_cloning
$FLASH_DIR/predeployment/.TestApplicationLicenceTransfer \
$FLASH_DIR/custom_sections/TestApplicationLicenceCounter
echo "Test precreation script finished">> $FLASH_DIR/summary
exit 0
```

## 作成前スクリプトを使用してユーザー定義アーカイブセクションを作成する

作成前スクリプトを使用して、アーカイブ内にユーザー定義セクションを作成し、アプリケーションに関する特定の情報を提供できます。このセクションの目的は、アーカイブの保守です。スクリプトは、`$FLASH_DIR/sections` ディレクトリに配置する必要があります。フラッシュアーカイブは、ユーザー定義セクションを処理しません。たとえば、このセクションには、アーカイブの説明や、アプリケーションの整合性をチェックするスクリプトなどを含めることができます。

ユーザー定義セクションでは、次の書式を使用する必要があります。

- 行単位で作成する必要がある
- 復帰改行 (ASCII 0x0a) 文字で終了させる必要がある
- 個々の行の長さに制限はない
- base64 またはそれに類似したアルゴリズムを使用してバイナリデータにコード化する

## ▼ 配置前スクリプトの作成方法

このスクリプトは、アーカイブのインストール前に実行されます。スクリプトの目的がアーカイブの妥当性検査である場合、このスクリプトはアーカイブ内に保持されます。スクリプトの目的がクローンシステムのローカル構成を保存することである場合、このスクリプトはクローンシステム上に保持されます。このスクリプトは、将来のカスタマイズに必要なローカルデータの分析および収集も実行可能です。たとえば、クライアント固有の情報を、アーカイブから抽出されるファイルで上書きされる前に保存できます。この情報は、抽出後の最終段階で使用できます。

1. 配置前スクリプトを作成します。209 ページの「カスタムスクリプト作成のガイドライン」に記されているガイドラインに従います。
2. スクリプトを次のディレクトリのいずれかに保存します。
  - アーカイブの妥当性検査を実行するには、`/etc/flash/predeployment` ディレクトリに保存します。
  - 作成前スクリプトを使用して参照を行う場合、`$FLASH_DIR/preinstall` ディレクトリに保存します。
  - クローンシステム上の構成を保持する場合、JumpStart プロファイル内の `local_customization` キーワードに、クローンシステムに保存されているスクリプトへのパスを指定します。

### 例 21-3 配置前スクリプト

```
#!/bin/sh
$FLASH_DIR/TestApplication/check_hardware
if [ $? != 0 ]; then
    echo Unsupported hardware
    exit 1
fi
$FLASH_DIR/TestApplication/check_licence_key
if [ $? != 0 ]; then
    echo No license for this host
    exit 1
fi
$FLASH_DIR/TestApplication/deploy_license_key \
    $FLASH_DIR/TestApplication/.TestApplicationLicenceTransfer
$FLASH_DIR/TestApplication/save_data_files $FLASH_DIR/flash

exit 0
```

## ▼ 配置後スクリプトの作成方法

このスクリプトは、アーカイブ内またはクローンシステムのローカルディレクトリに格納され、インストール後に実行されます。スクリプトにより、クローンシステム上で新規システムイメージが再構成されます。スクリプトがアーカイブ内に格納されている場合、変更はすべてのクローンシステムに影響を及ぼします。スクリプトがクローンシステムのローカルディレクトリに格納されている場合、変更はそのクローンシステムにのみ影響します。たとえば、配置前スクリプトにより保存されたクライアント固有の情報をクローン環境に適用して、インストールを完了できます。

配置後スクリプトを使用して、アーカイブのインストール後にファイルを整理することも可能です。たとえば、`/var/adm` 内のログファイルなどを削除できます。

---

注 - すべてのログファイルが、クリーンアップ用のスクリプトを必要とするわけではありません。`/var/tmp` 内のログファイルはアーカイブの作成時に自動的に削除されます。

---

1. 配置後スクリプトを作成します。209 ページの「カスタムスクリプト作成のガイドライン」に記されているガイドラインに従います。
2. スクリプトを次のディレクトリのいずれかに保存します。
  - すべてのクローンシステムを操作するには、スクリプトを `/etc/flash/postdeployment` ディレクトリに保存します。
  - ローカルのクローンシステムのみを操作する場合は、JumpStart プロファイル内の `local_customization` キーワードに、クローンシステムに保存されているスクリプトへのパスを指定します。

### 例 21-4 配置後スクリプト

```
#!/bin/sh
$FLASH_DIR/TestApplication/clone_reconfiguration
$FLASH_DIR/TestApplication/restore_data $FLASH_DIR/flash
```

## ▼ 再起動スクリプトの作成方法

このスクリプトはアーカイブ内に格納され、システムの再起動後に実行されます。このスクリプトにより、システムの再構成後の最終構成がすべて実行されます。

クローンシステムにフラッシュアーカイブをインストールしたあと、ホスト固有のファイルがいくつか削除され、クローンマシン用として作成し直されます。インストールプログラムは、`sys-unconfig(1M)` コマンドと `sysidtool(1M)` プログラムを使用して、ホスト固有のネットワーク構成ファイルを削除して再作成します。再作成されるファイルは、`/etc/hosts`、`/etc/defaultrouter`、`/etc/defaultdomain` などです。再起動スクリプトを使用することで、任意の最終的な再構成処理を実行できます。

1. 再起動スクリプトを作成します。
2. スクリプトを `/etc/flash/reboot` ディレクトリに保存します。

例 21-5 再起動スクリプトの作成

```
#!/bin/sh
$FLASH_DIR/TestApplication/finalize_license
```

---

## フラッシュアーカイブの作成

クローンシステムのファイルすべてを上書きする初期インストール用アーカイブを作成することも、指定された変更部分のみを上書きする差分アーカイブを作成することもできます。差分アーカイブの詳細は、207 ページの「更新用フラッシュ差分アーカイブの作成を計画する」を参照してください。

### ▼ 初期インストール用フラッシュアーカイブの作成方法

マスターシステムにインストールした後で、他のシステムへのインストールに使用するフラッシュアーカイブを作成します。

1. マスターシステムをブートし、できるだけ静的な状態で稼働させます。  
可能であれば、システムをシングルユーザーモードで実行してください。これが不可能な場合、アーカイブしたいアプリケーションおよび大量のオペレーティングシステムリソースを必要とするアプリケーションを停止します。  
フラッシュアーカイブの作成は、マルチユーザーモードまたはシングルユーザーモードで稼働しているマスターシステム上で、あるいは次に示すメディアの1つからブートさせたシステム上で行うことができます。
  - Solaris DVD
  - Solaris SOFTWARE 1 of 2 CD
  - Solaris SOFTWARE のイメージ。CD メディアを使用している場合、イメージには必要に応じて Solaris LANGUAGES CD が含まれます。
2. `flar create` コマンドを使用して、アーカイブを作成します。

```
# flar create -n name options path/filename
```

**name**        アーカイブに指定する名前です。指定する *name* は、`content_name` キーワードの値になります。

**options**      オプションの詳細は、246 ページの「`flar create`」を参照してください。

*path* アーカイブファイルを保存するディレクトリへのパスです。パスを指定しない場合、`flar create` によりアーカイブファイルを現在のディレクトリに保存します。

*filename* アーカイブファイルの名前です。

- アーカイブの作成が正常に完了すると、`flar create` コマンドは終了コード 0 を返します。
- アーカイブの作成が失敗すると、`flar create` コマンドは 0 以外の終了コードを返します。

3. アーカイブのコピーを作成して保存します。将来、クローンシステムを差分アーカイブで更新するときに、このコピーを使用できます。

## 例 — 初期インストール用アーカイブの作成

ファイルシステムを複製する際、全く同じようにコピーすることも、カスタマイズしてディレクトリやファイルのいくつかを除外することもできます。異なるオプションを使用して、同じ結果を得ることも可能です。使用する環境に最も適したオプションを使用してください。

次の例に示すファイルシステムは、分かりやすくするために大幅に簡略化されています。これらの例では、マスターシステムのファイル構造は、`/var`、`/usr`、または `/opt` などのファイルシステム名の代わりに、次のようなファイル構造を使用します。

```
/aaa/bbb/ccc/ddd  
/aaa/bbb/fff  
/aaa/eee  
/ggg
```



---

注意 - `flar create` のファイル除外オプションを指定する際は、注意が必要です。一部のディレクトリを除外する際、気づかずにシステム構成ファイルなどの他のファイルがアーカイブに残ってしまう場合があります。この場合、システムの整合性が損なわれるため、インストールが失敗してしまいます。ディレクトリやファイルの除外は、大規模なデータファイルなど、システムを破綻させることなく容易に削除可能なデータに対して行うのが最善です。

---

### 例 21-6 同一の複製アーカイブの作成

この例では、アーカイブの名前は `archive1` です。これは、マスターシステムそのものがコピーされ、その後圧縮されます。アーカイブはマスターシステムと全く同一な複製で、`archive1.flar` に格納されます。

```
# flar create -n archive1 -c archive1.flar
```

アーカイブのファイル構造を確認するには、以下のように入力します。

例 21-6 同一の複製アーカイブの作成 (続き)

```
# flar info -l archive1.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/bbb/fff
aaa/eee
aaa/eee
ggg
```

例 21-7 アーカイブから除外するファイルとディレクトリおよびアーカイブに含めるファイルとディレクトリを指定してアーカイブを作成する

この例では、アーカイブの名前は archive2 です。これは、マスターシステムからコピーされますが、同一のコピーではありません。/aaa ディレクトリ以下の内容はアーカイブから除外されますが、/aaa/bbb/ccc の内容はアーカイブに含まれます。

```
# flar create -n archive2 -x /aaa -y /aaa/bbb/ccc archive2.flar
```

アーカイブのファイル構造を確認するには、以下のように入力します。除外されたディレクトリ (/aaa) であっても、コピーされたファイル (/aaa/bbb/ccc) が下位に存在する場合、そのディレクトリも表示されていますが、実際にアーカイブに含まれているのはコピーされたファイル (/aaa/bbb/ccc) だけです。

```
# flar info -l aaa
aaa
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/bbb
ggg
```

例 21-8 アーカイブから除外するファイルとディレクトリおよびアーカイブに含めるファイルとディレクトリをリストで指定してアーカイブを作成する

この例では、アーカイブの名前は archive5 です。これは、マスターシステムからコピーされますが、同一のコピーではありません。

exclude ファイルには、次のリストが含まれています。

```
/aaa
```

include ファイルには、次のリストが含まれています。

```
/aaa/bbb/ccc
```

/aaa ディレクトリ以下の内容はアーカイブから除外されますが、/aaa/bbb/ccc の内容はアーカイブに含まれます。

```
# flar create -n archive5 -X exclude -f include archive5.flar
```

例 21-8 アーカイブから除外するファイルとディレクトリおよびアーカイブに含めるファイルとディレクトリをリストで指定してアーカイブを作成する (続き)

アーカイブのファイル構造を確認するには、以下のように入力します。除外されたディレクトリ (/aaa) であっても、コピーされたファイル (/aaa/bbb/cc) が下位に存在する場合、そのディレクトリも表示されていますが、実際にアーカイブに含まれているのはコピーされたファイル (/aaa/bbb/cc) だけです。

```
# flar info -l archive5.flar
aaa
aaa/bbb/cc
aaa/bbb/cc/dd
aaa/bbb
ggg
```

例 21-9 アーカイブから除外するファイルとディレクトリをリストで指定し、アーカイブに含めるディレクトリを直接指定してアーカイブを作成する

オプション `-x`、`-y`、`-X`、および `-f` を組み合わせて使用できます。この例では、オプション `-x` および `-y` が組み合わせて使用されています。アーカイブの名前は、`archive5` です。これは、マスターシステムからコピーされますが、同一のコピーではありません。

`exclude` ファイルには、次のリストが含まれています。

```
/aaa
```

`-y` オプションにより、ディレクトリ `/aaa/bbb/cc` がアーカイブに含まれます。次のコマンドにより、アーカイブが生成されます。

```
# flar create -n archive5 -X exclude -y /aaa/bbb/cc archive5.flar
```

アーカイブのファイル構造を確認するには、以下のように入力します。除外されたディレクトリ (/aaa) であっても、コピーされたファイル (/aaa/bbb/cc) が下位に存在する場合、そのディレクトリも表示されていますが、実際にアーカイブに含まれているのはコピーされたファイル (/aaa/bbb/cc) だけです。

```
# flar info -l archive5.flar
aaa
aaa/bbb
aaa/bbb/cc
aaa/bbb/cc/dd
ggg
```

例 21-10 `-z` オプションを使用して、アーカイブから除外するファイルとディレクトリおよびアーカイブに含めるディレクトリをリストで指定してアーカイブを作成する

この例では、アーカイブの名前は `archive3` です。これは、マスターシステムからコピーされますが、同一のコピーではありません。選択するファイルおよびディレクトリが、`filter1` ファイルに含まれます。ファイル内では、ディレクトリにプラス記号 (+) またはマイナス (-) 記号が付けられ、アーカイブから除外するかアーカイブに含め

例 21-10 -z オプションを使用して、アーカイブから除外するファイルとディレクトリおよびアーカイブに含めるディレクトリをリストで指定してアーカイブを作成する (続き)

るかが示されます。この例では、ディレクトリ /aaa に除外されることを示すマイナス記号が、サブディレクトリ /aaa/bbb/ccc にアーカイブに含めることを示すプラス記号が付けられています。filter1 ファイルには、次のリストが含まれています。

```
- /aaa  
+ /aaa/bbb/ccc
```

次のコマンドにより、アーカイブが生成されます。

```
# flar create -n archive3 -z filter1 archive3.flar
```

アーカイブのファイル構造を確認するには、以下のコマンドを入力します。除外されたディレクトリ (/aaa) であっても、コピーされたファイル (/aaa/bbb/ccc) が下位に存在する場合、そのディレクトリも表示されていますが、実際にアーカイブに含まれているのはコピーされたファイル (/aaa/bbb/ccc) だけです。

```
# flar info -l archive3.flar  
aaa  
aaa/bbb  
aaa/bbb/ccc  
aaa/bbb/ccc/ddd  
ggg
```

例 21-11 代替ブート環境からのアーカイブの作成

この例では、アーカイブの名前は archive4 です。これは、マスターシステムそのものがコピーされ、その後圧縮されます。アーカイブはマスターシステムの全く同一な複製で、archive4.flar に格納されます。-R オプションは、別のディレクトリツリーからアーカイブを作成する場合に使用します。

```
# flar create -n archive4 -c -R /x/yy/zz archive4.flar
```

例 21-12 アーカイブを作成して、アーカイブについて記述するキーワードを追加する

この例では、アーカイブの名前は archive3 です。このアーカイブは、マスターシステムそのものがコピーされ、その後圧縮されます。オプションで、アーカイブ識別セクションに説明を追加できます。この説明により、後でアーカイブを識別するのが容易になります。キーワード、およびその値と書式については、241 ページの「フラッシュのキーワード」を参照してください。

```
# flar create -n archive3 -i 20000131221409 -m pumbaa \  
-e "Solaris 8 Print Server" -a "Mighty Matt" -U "Internal Finance" \  
-T server archive3.flar
```

アーカイブの作成後に、詳細な説明を含む識別セクションにアクセスできます。識別セクションの例を、以下に示します。

```
section_begin=identification  
files_archived_method=cpio
```



例 21-12 アーカイブを作成して、アーカイブについて記述するキーワードを追加する  
(続き)

```
files_compressed_method=compress
files_archived_size=259323342
files_unarchived_size=591238111
creation_date=20000131221409
creation_master=pumbaa
content_name=Finance Print Server
content_type=server
content_description=Solaris 8 Print Server
content_author=Mighty Matt
content_architectures=sun4u,sun4m
creation_node=pumbaa
creation_hardware_class=sun4u
creation_platform=SUNW,Sun-Fire
creation_processor=sparc
creation_release=5.9
creation_os_name=SunOS
creation_os_version=s81_49
x-department=Internal Finance
```

## ▼ 更新されたマスターイメージを使用してフラッシュ差分アーカイブを作成する方法

差分アーカイブを作成するには、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2つが必要です。1つ目のイメージは、変更が加えられていないマスターイメージです。このイメージはどこかに保存されていてアクセスできる必要があります。2つ目のイメージは、マイナーチェンジにより変更が加えられたマスターイメージです。このイメージのデフォルトの格納場所はルート (/) ファイルシステムですが、他の場所に格納された場合でもアクセスが可能です。2つのイメージが用意できたら、2つのイメージの差分だけを含む差分アーカイブを作成します。次に、更新前のマスターイメージを使用してインストールされたクローンシステムに差分アーカイブをインストールできます。

1. マスターシステムで変更の準備を行います。変更を加える前に、マスターシステムで元のアーカイブの複製を作成する必要があります。

---

注 - 更新前のマスターイメージのコピーが変更されないよう保護し、後でこのイメージをマウントできるようにしておく必要があります。

---

2. 次のいずれかの変更に合わせて、更新前のマスターイメージを更新します。
  - パッケージを削除する
  - パッケージまたはパッチを追加する



ドの値になります。名前は、アーカイブ識別セクションに記載されます。

`-A unchanged_master_image_dir`

新規システムイメージを `unchanged_master_image_dir` 引数で指定されたイメージと比較して、差分アーカイブを作成します。デフォルトでは、新しいシステムイメージはルート (/) です。-R オプションを使用してデフォルトを変更できます。  
`unchanged_master_image_dir` は、UFS、NFS、または `lumount` を使用して格納またはマウントされた更新前システムイメージの格納先ディレクトリです。

内容選択用のオプションを使用して、一部のファイルを含めることも除外することもできます。オプションの一覧は、246 ページの「`flar create`」を参照してください。

*options*

オプションの詳細は、246 ページの「`flar create`」を参照してください。

*path*

アーカイブファイルを保存するディレクトリへのパスを指定します。パスを指定しない場合、`flar create` によりアーカイブファイルを現在のディレクトリに保存します。

*filename*

アーカイブファイル名を指定します。

- 差分アーカイブの作成が正常に完了すると、`flar create` コマンドは終了コード 0 を返します。
- 差分アーカイブの作成が失敗すると、`flar create` コマンドは 0 以外の終了コードを返します。

アーカイブのインストール手順については、315 ページの「カスタム JumpStart インストールを使用して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。

## 例 — 差分アーカイブの作成

例 21-13 マスターシステム上の新規マスターイメージを使用して差分アーカイブを作成する

次の例では、更新前のマスターイメージのディレクトリの名前は `unchanged_master1` です。変更を含む新しいマスターイメージは、ルート (/) ディレクトリにおかれているものとします。新しいマスターイメージが更新前のマスターイメージと比較され、その後、生成された差分アーカイブが圧縮されます。差分アーカイブは、`diffarchive1.flar` ファイルに格納されます。アーカイブには、インストール時に削除、変更、または追加されるファイルが含まれます。

```
# flar create -n diffarchive1 -A /a/unchanged_master1 -c diffarchive1.flar
```

例 21-14 非アクティブブート環境に格納されたイメージを使用して差分アーカイブを作成する

次の例では、更新前のマスターイメージ `unchanged_master1` が非アクティブブート環境に格納されており、ブート環境をマウントすることによりアクセス可能になります。新しいマスターイメージは、ルート (`/`) ディレクトリにおかれているものとします。新しいマスターイメージが更新前のマスターイメージと比較され、その後、生成された差分アーカイブが圧縮されます。アーカイブは、`diffarchive4.flar` に格納されます。アーカイブには、インストール時に削除、変更、または追加されるファイルが含まれます。

```
# lumount unchanged_master1 /a
# flar create -n diffarchive4 -A /a -c diffarchive4.flar
```

## ▼ Solaris Live Upgrade を使用してフラッシュ差分アーカイブを作成する方法

システムの更新を管理するには、Solaris Live Upgrade を使用してオペレーティング環境のコピーを作成し、新しいブート環境を作成します。このコピーを、マイナーチェンジが加えられたマスターシステムと比較できます。こうして作成された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できます。

Solaris Live Upgrade の詳細は、第 32 章を参照してください。

1. **lucreate** コマンドを実行して、変更されていないマスターシステムから新しいブート環境を作成します。  
この新しいブート環境はマスターシステムの正確なコピーであり、差分アーカイブの作成に使用できます。

2. 2 つのブート環境の状態をチェックします。

```
# lustatus copy_BE
```

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
master_BE	yes	yes	yes	no	-
copy_BE	yes	no	no	yes	-

3. 次のどれかの操作を行い、マスターイメージを更新します。
  - パッケージを削除する
  - パッケージまたはパッチを追加する
  - 構成ファイルを変更する
  - クローンシステム上にある周辺装置のサポートを追加する
4. (省略可能) カスタムスクリプトを作成します。215 ページの「カスタムスクリプトの作成」を参照してください。

5. 差分アーカイブを作成します。

- a. 新しく作成されたブート環境をマウントします。

```
# luumount BE_name /a
```

- b. マスターシステムとブート環境を比較して、差分アーカイブを作成します。

```
# flarcreate -n archive_name -A new_BE_dir\  
options path/filename
```

*archive\_name* アーカイブに付ける名前を指定します。

*-A new\_BE\_dir* 新しいシステムイメージと *new\_BE\_dir* 引数で指定されたイメージを比較して、差分アーカイブを作成します。

*options* オプションの一覧は、246 ページの「flar create」を参照してください。

*path* アーカイブファイルを保存するディレクトリへのパスを指定します。パスを指定しない場合、アーカイブファイルは現在のディレクトリに保存されます。

*filename* アーカイブファイル名を指定します。

- c. 新しいブート環境をマウント解除します。

```
# luumount copy_BE
```

flar create コマンドは終了コードを返します。

- 作成に成功した場合、返される終了コードは 0 です。
- 障害が発生した場合、返される終了コードは 0 以外です。

6. JumpStart プロファイルを使用してフラッシュ差分アーカイブをインストールします。

インストールするクローンシステムが元のマスターシステムの複製でない場合、インストールに失敗します。

次のプロファイル例では、デバイス *clt1d0s0* に差分アーカイブ *test.diff* がインストールされます。

```
JumpStart profile  
-----  
install_type flash_update  
archive_location http server /rw/test.diff  
root_device clt1d0s0
```

例 21-15 Solaris Live Upgrade による差分アーカイブの作成

この例では、現在のブート環境の名前は *master\_BE* です。新しいブート環境の名前は *copy\_BE* です。ルート (/) ファイルシステムと /usr ファイルシステムは、それぞれ *s0* と *s3* に配置されています。lustatus コマンドにより、新しいブート環境のコピーが完了したことが報告されます。マスターシステムに SUNWman パッケージを

例 21-15 Solaris Live Upgrade による差分アーカイブの作成 (続き)

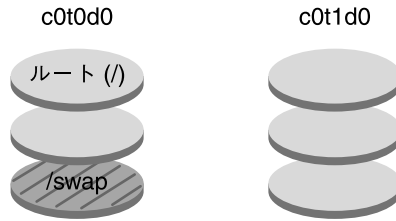
追加します。SUNWman パッケージの追加によってマスターシステムが更新されたあと、flarcreate コマンドにより、変更されたマスターと変更されていない新しいブート環境の比較が行われ、差分アーカイブが作成されます。

```
# lucreate -c master_BE
-m /:/dev/dsk/c0t1d0s0:ufs -m /usr:/dev/dsk/c0t1d0s3:ufs \
-n copy_BE
# lustatus
# pkgadd SUNWman
# lumount copy_BE /a
# flarcreate -n test.diff -c -A /a /net/server/export/test.diff
# lumount copy_BE
```

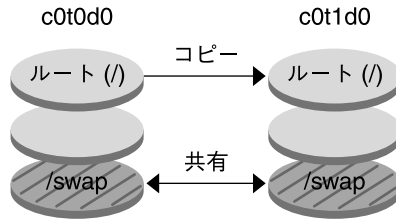
クローンシステムに差分アーカイブをインストールします。アーカイブのインストール手順については、315 ページの「カスタム JumpStart インストールを使用してフラッシュアーカイブをインストールする方法」を参照してください。

図 21-1 は、lucreate コマンドで新しいブート環境が作成される様子を示しています。

2つの物理ディスクからなる  
元のマスターシステム



新しいブート環境の作成後の  
マスターシステム



```
コマンド: # lucreate  
          -m /:/dev/dsk/c0t1d0s0:ufs \  
          -n second_disk
```

図 21-1 Solaris Live Upgrade による新しいブート環境の作成





## 第 22 章

---

# フラッシュアーカイブのインストールと管理 (作業)

---

この章では、フラッシュアーカイブをクローンシステムにインストールする手順およびアーカイブを管理する手順を説明します。

- 233 ページの「フラッシュアーカイブをインストールする方法」
- 234 ページの「フラッシュアーカイブの管理」

---

## フラッシュアーカイブをインストールする方法

フラッシュアーカイブは、任意の Solaris インストール方法で、初期インストールできません。フラッシュ差分アーカイブをインストールするには、カスタム JumpStart を使用する必要があります。

- 初期インストールでフラッシュアーカイブをインストールするには、以下を参照してください。
  - Solaris Live Upgrade – 459 ページの「ブート環境へのフラッシュアーカイブのインストール」を参照してください。
  - Solaris Web Start プログラム – 170 ページの「SPARC: Solaris Web Start によるインストールまたはアップグレード」または 175 ページの「x86: Solaris Web Start によるインストールまたはアップグレード」を参照してください。
  - Solaris suninstall プログラム – 186 ページの「SPARC: Solaris suninstall プログラムによるインストールまたはアップグレード」または 189 ページの「x86: Solaris suninstall プログラムによるインストールまたはアップグレード」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 – 274 ページの「プロファイルの作成」および 315 ページの「カスタム JumpStart インストールを使用して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。
  - WAN ブートインストール – 第 41 章を参照してください。

- フラッシュ差分アーカイブの更新方法については、以下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 – 274 ページの「プロファイルの作成」および 315 ページの「カスタム JumpStart インストールを使用して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。
  - Solaris Live Upgrade – 462 ページの「プロファイルを使用した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。

---

## フラッシュアーカイブの管理

`flar` コマンドは、アーカイブの管理に使用します。1 つのアーカイブを複数のセクションに分割できます。これらのセクションに対し変更、追加、または削除を実行してから結合して、アーカイブを作成できます。アーカイブに関する情報を取得することもできます。



---

注意 – アーカイブセクションは変更しないようにしてください。変更するとアーカイブの整合性に悪影響が及びます。

---

## フラッシュアーカイブの分割

アーカイブを複数のセクションに分割し、その一部を変更したり、新たなセクションを追加したり、セクションを削除したりできます。セクションの変更後に、セクションを結合して新規アーカイブを作成する必要があります。たとえば、ユーザー定義セクションの追加や、識別セクションの変更を実行できます。アーカイブセクションは変更しないようにしてください。変更するとアーカイブの整合性に悪影響が及びます。

`flar split` コマンドは、フラッシュアーカイブを複数のセクションに分割します。`flar` コマンドは、各セクションを、現在のディレクトリまたは指定されたディレクトリ内の別個のファイルにコピーします。ファイルには、セクション名からとった名前が付けられます。たとえば、Cookie セクションは、`cookie` と名付けられたファイルに保存されます。`flar split` コマンドでは、1 つのセクションだけを保存するように指定できます。このコマンドの構文は次のとおりです。

```
flar split[-d dir] [-u section] [-f archive] [-S section] [-t [-p posn] [-b blocksize]]  
filename
```

---

-d <i>dir</i>	コピーするセクションを、現在のディレクトリからではなく <i>dir</i> から取得します。
-u <i>section</i>	<ul style="list-style-type: none"> <li>■ このオプションを使用した場合、<code>flar</code> は Cookie セクション、識別セクション、アーカイブセクション、および <i>section</i> セクションをコピーします。1つのセクション名を指定することも、あるいは空白で区切って複数のセクション名を指定することも可能です。</li> <li>■ このオプションを使用しない場合、<code>flar</code> は Cookie、識別、およびアーカイブセクションのみをコピーします。</li> </ul>
-f <i>archive</i>	アーカイブセクションを、 <i>archive</i> という名前のファイルに入れるのではなく、 <i>archive</i> というディレクトリに抽出します。
-S <i>section</i>	アーカイブから <i>section</i> と名付けられたセクションだけをコピーします。これは、ユーザー定義セクションです。

---

#### 例 22-1 アーカイブの分割

次の例では、`archive1.flar` が 3 つのファイルに分割されます。

- `cookie` – アーカイブの最初の行。アーカイブフォーマットのバージョンを示します。この識別子は変更しないでください。
- `identification` – 識別セクションのコピー。すべてのキーワードと値のペアを保持します。
- `archive – cpio` アーカイブ本体。このファイルは圧縮可能です。

```
# flar split archive1.flar
```

アーカイブの分割後に、識別セクションの変更またはユーザー定義セクションの追加を実行できます。その後、セクションを結合してアーカイブを再作成できます。

## フラッシュアーカイブの結合

アーカイブをセクションに分割した後で、セクションを結合して新規アーカイブを作成できます。

`flar combine` コマンドは、個別のセクションからフラッシュアーカイブを作成します。各セクションは、セクション名を名前に持つ個々のファイル内にあると見なされます。少なくとも、次の 3 つのファイルが存在する必要があります。

- `cookie` セクション (`cookie`)
- 識別セクション (`identification`)

- アーカイブファイルセクション (archive)

セクションを結合する際、以下の点に留意してください。

- `archive` がディレクトリである場合、`flar` は、結合されたアーカイブに含める前に、`cpio` を使用してディレクトリをアーカイブします。
- 識別セクションにアーカイブの圧縮が指定されている場合、`flar` は新しく結合されたアーカイブのコンテンツを圧縮します。
- どのセクションについても検証は何も行われません。特に、識別セクション内のどのフィールドについても、検証や更新は行われません。

```
flar combine [-d dir] [-u section] [-t [-p posn]] [-b blocksize]] filename
```

`-d dir`          結合するセクションを、現在のディレクトリからではなく `dir` から取得します。

`-u section`

- このオプションを使用した場合、`flar` は Cookie セクション、識別セクション、アーカイブセクション、および `section` セクションをコピーします。1つのセクション名を指定することも、あるいは空白で区切って複数のセクション名を指定することも可能です。
- このオプションを使用しない場合、`flar` は Cookie、識別、およびアーカイブセクションのみをコピーします。

#### 例 22-2 フラッシュアーカイブの結合

この例では、Cookie セクション、識別セクション、アーカイブセクションが結合して、完成したアーカイブになります。アーカイブ名は `newarchive.flar` です。

```
# flar combine newarchive.flar
```

#### 例 22-3 フラッシュアーカイブの結合とユーザー定義セクションの追加

この例では、Cookie セクション、識別セクション、アーカイブファイルセクション、およびユーザー定義セクションが結合して、完成したアーカイブになります。アーカイブ名は `newarchive.flar` です。ユーザー定義セクションの内容は、現在のディレクトリ内の `user_defined` という名前のファイルの中にあります。

```
# flar combine -u user_defined newarchive.flar
```

## アーカイブからの情報の抽出

`flar info` コマンドを使用して、作成済みのアーカイブに関する情報を取得します。このコマンドの構文は次のとおりです。

```
flar info [-l] [-k keyword] [-t [-p posn]] [-b blocksize]] filename
```

`-k keyword`      キーワード `keyword` の値のみを返します。

-1                   アーカイブセクション内のすべてのファイルをリスト表示します。

例 22-4 アーカイブセクション内のファイルの一覧表示

この例では、アーカイブ `archive3.flar` のファイル構造がチェックされます。

```
# flar info -l archive3.flar
aaa
aaa/bbb
aaa/bbb/ccc
aaa/bbb/ccc/ddd
aaa/eee
```



## 第 23 章

# フラッシュ (リファレンス)

この章では、フラッシュのセクション、キーワード、およびキーワード値について説明します。また、`flar create` コマンドのオプションについても説明します。

- 239 ページの「フラッシュアーカイブセクションの説明」
- 241 ページの「フラッシュのキーワード」
- 246 ページの「フラッシュ `flar create` コマンド」

## フラッシュアーカイブセクションの説明

各フラッシュアーカイブは、セクションにグループ化されます。一部のセクションは、フラッシュソフトウェアにより生成されるため、ユーザーが入力する必要はありません。一部のセクションは入力が必要であったり、情報追加が可能であったりします。以下の表で、各セクションについて説明します。

表 23-1 フラッシュアーカイブのセクション

セクション名	説明	アーカイブに必要	ユーザー入力が必要
Cookie セクション	最初のセクションには、ファイルをフラッシュアーカイブとして識別する <code>cookie</code> が含まれます。配置処理では、識別および検証に <code>cookie</code> を使用します。アーカイブを有効にするために、 <code>cookie</code> が必要です。	必須	省略可能

表 23-1 フラッシュアーカイブのセクション (続き)

セクション名	説明	アーカイブに必要	ユーザー入力が必要
識別セクション	<p>2 番目のセクションには、アーカイブを識別する情報を値とするキーワードが含まれます。ソフトウェアは、以下に示すような情報を生成します。</p> <ul style="list-style-type: none"> <li>■ アーカイブの ID 番号</li> <li>■ アーカイブのメソッド (cpio など)</li> <li>■ 作成日 (デフォルト情報)</li> </ul> <p>フラッシュアーカイブの名前はユーザーが指定する必要があります。アーカイブについて指定できる情報には、次のものが含まれます。</p> <ul style="list-style-type: none"> <li>■ アーカイブの作者</li> <li>■ アーカイブの作成日</li> <li>■ アーカイブを作成したマスターシステムの名前</li> </ul> <p>アーカイブに関するキーワードのリストについては、242 ページの「識別セクションのキーワード」を参照してください。</p>	必須	内容は、ユーザーとソフトウェアの両方で生成されません。
マニフェストセクション	<p>クローンシステムの検証に使用されるフラッシュアーカイブのセクション。マニフェストセクションには、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。インストールされるファイルが、このリストセットと一致しない場合、インストールは失敗します。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。</p> <p>flar create -M オプションを使用して差分アーカイブを作成することにより、このセクションを省略できますが、アーカイブの検証が行われなくなるため、このセクションを省略することを推奨しません。</p>	省略可能	省略可能
■ 配置前 ■ 配置後 ■ 再起動	<p>このセクションには、オペレーティング環境イメージのインストール前または後にフラッシュソフトウェアが使用する内部情報が含まれます。指定したすべてのカスタマイズスクリプトが、このセクションに保存されます。</p>	必須	省略可能
サマリーセクション	<p>このセクションには、アーカイブ作成に関するメッセージが含まれます。また、配置前スクリプトのアクティビティが記録されます。</p>	必須	内容は、ユーザーとソフトウェアの両方で生成されません。
ユーザー定義セクション	<p>このセクションは、識別セクションの次に存在します。アーカイブには、ゼロ以上のユーザー定義セクションを含めることができます。これらのセクションは、アーカイブの抽出処理では処理されません。これらのセクションは別個に取り出され、内容の説明に使用されます。</p>	省略可能	必須



表 23-1 フラッシュアーカイブのセクション (続き)

セクション名	説明	アーカイブに必要	ユーザー入力が必要
アーカイブファイルセクション	アーカイブファイルセクションには、マスターシステムから収集されたファイルの内容がバイナリデータで書きこまれます。このセクションは、 <code>section_begin=archive</code> で始まりますが、セクションの終了境界を示すものではありません。	必須	省略可能

## フラッシュのキーワード

フラッシュのキーワードは、カスタム JumpStart のキーワードと類似しています。どちらのキーワードも、インストール要素を定義します。各キーワードは、フラッシュソフトウェアによるクローンシステムへのソフトウェアのインストール方法の一部分を制御するコマンドです。

以下のガイドラインに従って、キーワードおよび値の書式を指定します。

- キーワードと値は1つの等号で区切られ、1行に1つのキーワードと1つの値を指定します。
- キーワードは、大文字と小文字を区別しません。
- 個々の行の長さに制限はありません。

### 一般的なキーワード

フラッシュアーカイブの各セクションは、`section_begin` と `section_end` キーワードで定義されます。たとえば、アーカイブファイルセクションには、値の異なる `section_begin` キーワードが含まれます。ユーザー定義アーカイブセクションは、`section_begin` と `section_end` キーワードで区切られます。各キーワードには、そのセクションに適した値が指定されます。`section_begin` と `section_end` キーワードの値を以下の表に示します。

表 23-2 `section_begin` と `section_end` キーワードの値

アーカイブのセクション	<code>section_begin</code> と <code>section_end</code> キーワードの値
Cookie セクション	<code>cookie</code> — このセクションは、 <code>section_begin</code> と <code>section_end</code> キーワードで区切られません。
識別セクション	<code>identification</code>
ユーザー定義セクション	<code>section_name</code> 。 <code>section_name</code> キーワードの一例は、 <code>X-user_section_1</code> です。

表 23-2 section\_begin と section\_end キーワードの値 (続き)

アーカイブのセクション	section_begin と section_end キーワードの値
アーカイブファイルセクション	archive

## 識別セクションのキーワード

以下の表に、識別セクションで使用されるキーワードと、これらに定義可能な値を示します。

各セクションでは、表 23-3 に示したキーワードがセクションの区切りに使用されません。

表 23-3 識別セクションのキーワード: 一般的なキーワード

キーワード	値の定義	値	必須
section_begin	これらのキーワードは、アーカイブ内のセクションの区切りに使用されます。使用は、識別セクションに限定されません。これらのキーワードの詳細は、241 ページの「一般的なキーワード」を参照してください。	テキスト	必須
section_end			

以下のキーワードは、識別セクションで使用され、アーカイブファイルセクションの内容を説明します。

表 23-4 識別セクションのキーワード: アーカイブファイルセクションの内容

キーワード	値の定義	値	必須
archive_id (省略可能)	このキーワードは、アーカイブの内容を独自の方法で説明します。この値は、インストールソフトウェアにより、アーカイブインストール時のアーカイブの内容確認にのみ使用されます。キーワードが存在しない場合、整合性チェックは行われません。  たとえば、archive_id キーワードを FLASH-ARcHive-2.0 に設定できます。	テキスト	省略可能

表 23-4 識別セクションのキーワード: アーカイブファイルセクションの内容 (続き)

キーワード	値の定義	値	必須
<code>files_archived_method</code>	<p>このキーワードは、ファイルセクションで使用されるアーカイブ方法を記述します。</p> <ul style="list-style-type: none"> <li>■ このキーワードが存在する場合、値は <code>cpio</code> になります。</li> <li>■ このキーワードが存在しない場合、ファイルセクションは、ASCII ヘッダーをもつ CPIO フォーマットと見なされます。このフォーマットは、<code>cpio -c</code> オプションです。</li> </ul> <p><code>files_compressed_method</code> が存在する場合、そこで指定された圧縮方法がこの <code>files_archived_method</code> キーワードで指定されたアーカイブ方法で作成されるアーカイブファイルに適用されます。</p>	テキスト	省略可能
<code>files_archived_size</code>	<p>このキーワード値は、アーカイブファイルセクションのサイズをバイトで表現した値です。</p>	数値	省略可能
<code>files_compress_method</code>	<p>このキーワードは、ファイルセクションで使用される圧縮アルゴリズムを示します。</p> <ul style="list-style-type: none"> <li>■ キーワードが存在する場合、次のいずれかの値を指定できます。 <ul style="list-style-type: none"> <li>■ <code>none</code> - アーカイブファイルセクションは圧縮されません。</li> <li>■ <code>compress</code> - アーカイブファイルセクションは、<code>compress</code> コマンドを使用して圧縮されます。</li> </ul> </li> <li>■ このキーワードが存在しない場合、アーカイブファイルセクションは非圧縮と判断されます。</li> </ul> <p>このキーワードで指定された圧縮方法は、<code>files_archived_method</code> キーワードで指定されたアーカイブ方法で作成されるアーカイブファイルに適用されます。</p>	テキスト	省略可能
<code>files_unarchived_size</code>	<p>このキーワードは、抽出されたアーカイブの累積サイズをバイトで定義します。この値は、ファイルシステムのサイズ検証に使用されます。</p>	数値	省略可能

以下のキーワードは、アーカイブ全体に関する情報を提供します。通常、これらのキーワードは、アーカイブの選択および管理を支援する目的で使用されます。これらのキーワードはすべてオプションであり、各アーカイブの識別を容易にするために使用されます。これらのキーワードは、`flar create` コマンドのオプションの中で使用されます。具体例は、例 21-12 を参照してください。

表 23-5 識別セクションのキーワード: ユーザーによるアーカイブの記述

キーワード	値の定義	値	必須
creation_date	<p>このキーワード値は、アーカイブの作成時刻を表すタイムスタンプのテキストです。</p> <ul style="list-style-type: none"> <li>■ flar create コマンドに <code>-i</code> オプションを使用して、日付を作成できます。</li> <li>■ flar create コマンドを使用して作成日付を指定しない場合、デフォルトの日付がグリニッジ標準時 (GMT) で設定されます。</li> <li>■ 値は、ISO-8601 に準拠し、時間指示子なしの完全基本カレンダー形式で指定する必要があります (ISO-8601, §5.4.1(a))。書式は、<code>CCYYMMDDhhmmss</code> です。たとえば、<code>20000131221409</code> は、2000 年 1 月 31 日、22 時 14 分 09 秒を示します。</li> </ul>	テキスト	省略可能
creation_master	<p>このキーワード値には、アーカイブの作成に使用したマスターシステムの名前を指定します。flar create <code>-m</code> オプションを使用して、この値を指定できます。値を指定しない場合、<code>uname -n</code> コマンドから値が取得されます。</p>	テキスト	省略可能
content_name	<p>このキーワードにより、アーカイブが識別されます。この値は、flar create <code>-n</code> オプションから生成されます。この値を作成する際、以下のガイドラインに従ってください。</p> <ul style="list-style-type: none"> <li>■ 256 文字を超える長さの記述名を指定することはできません。</li> <li>■ 説明には、アーカイブの機能および目的を記述する必要があります。</li> </ul>	テキスト	必須
content_type	<p>このキーワード値により、アーカイブのカテゴリが指定されます。値の生成には、flar create <code>-T</code> オプションを使用します。</p>	テキスト	省略可能
content_description	<p>このキーワード値は、アーカイブの内容を示します。このキーワードの値の長さに制限はありません。この値の生成には、flar create <code>-E</code> オプションを使用します。</p>	テキスト	省略可能
content_author	<p>このキーワード値は、アーカイブの作成者を示します。この値の作成には、flar create <code>-a</code> オプションを使用します。作成者のフルネームと電子メールアドレスを含めることをお勧めします。</p>	テキスト	省略可能

表 23-5 識別セクションのキーワード: ユーザーによるアーカイブの記述 (続き)

キーワード	値の定義	値	必須
content_architectures	<p>このキーワード値は、アーカイブがサポートするカーネルアーキテクチャのリストで、コンマで区切られています。</p> <ul style="list-style-type: none"> <li>■ このキーワードが存在する場合、インストールソフトウェアは、アーカイブがサポートするアーキテクチャのリストを参照して、クローンシステムのカーネルアーキテクチャを検証します。アーカイブがクローンシステムのカーネルアーキテクチャをサポートしていない場合、インストールは失敗します。</li> <li>■ キーワードが存在しない場合、インストールソフトウェアはクローンシステムのアーキテクチャを検証しません。</li> </ul>	テキストのリスト	省略可能

以下のキーワードも、アーカイブ全体の説明に使われます。デフォルトでは、値は、フラッシュアーカイブの作成時に `uname` で指定されます。ルートディレクトリが / ではないフラッシュアーカイブを作成すると、アーカイブソフトウェアにより、これらのキーワードに対して文字列 `UNKNOWN` が挿入されます。例外は、`creation_node`、`creation_release`、および `creation_os_name` キーワードです。

- `creation_node` では、ソフトウェアは `nodename` ファイルの内容を使用します。
- `creation_release` および `creation_os_name` では、ソフトウェアは `root` ディレクトリの `/var/sadm/system/admin/INST_RELEASE` の内容を使用しようとしています。ソフトウェアがこのファイルの読み込みに失敗した場合、値 `UNKNOWN` が割り当てられます。

これらのキーワードの値は設定方法に関係なく、上書きできません。

表 23-6 識別セクションのキーワード: ソフトウェアによるアーカイブの記述

キーワード	値
<code>creation_node</code>	<code>uname -n</code> から返される値
<code>creation_hardware_class</code>	<code>uname -m</code> から返される値
<code>creation_platform</code>	<code>uname -i</code> から返される値
<code>creation_processor</code>	<code>uname -p</code> から返される値
<code>creation_release</code>	<code>uname -r</code> から返される値
<code>creation_os_name</code>	<code>uname -s</code> から返される値
<code>creation_os_version</code>	<code>uname -v</code> から返される値

## ユーザー定義セクションのキーワード

フラッシュアーカイブで定義されるキーワードに加えて、他のキーワードを定義できます。フラッシュアーカイブはユーザー定義のキーワードを無視しますが、識別セクションを処理しユーザー定義のキーワードを使用するスクリプトまたはプログラムを提供できます。ユーザー定義キーワードを作成する際、以下の書式に従います。

- キーワード名を *x* で始めます。
- 改行、等号、NULL 文字以外の任意の文字を使用してキーワードを作成します。
- ユーザー定義キーワードの命名規則には、定義済みのキーワードに対して使用される、下線で区切る記述方法を推奨します。他によく使われる命名規則としては、Java パッケージの命名に類似したフェデレーテッド規則があります。

たとえば、`x-department` はユーザー定義のキーワードとして有効です。

オプションを使用して、識別セクションにユーザー定義キーワードを含める具体例は、例 21-12 を参照してください。

---

## フラッシュ `flar create` コマンド

フラッシュ `flar create` コマンドを使用して、フラッシュアーカイブを作成できます。

### `flar create`

`flar create` コマンドは、マスターシステムからフラッシュアーカイブを作成するために使用します。このコマンドは、マスターシステムがマルチユーザーモードまたはシングルユーザーモードで稼働している時に使用できます。また、`flar create` は、マスターシステムが Solaris DVD または Solaris SOFTWARE 1 of 2 CD からブートしている時、あるいは Solaris SOFTWARE および Solaris LANGUAGES CD のネットイメージからブートしている時にも使用できます。フラッシュアーカイブを作成する時は、マスターシステムはできるだけ静的な状態にしておく必要があります。このコマンドの構文は次のとおりです。

```
flar create -n archive_name [-R root] [-A unchanged_master_image_dir] [-S]
[-M] [-H] [-I] [-c] [-x exclude_dir/filename] [-y include_dir/filename] [-z
list_filename] [-X list_filename] [-t [-p posn] [-b blocksize] [-i date] [-m
master] [-u section ... [-d dir]] [-f [list_filename| -] [-F]] [-U key=val
...] [-a author] [-e descr| -E descr_file] [-T type] path/filename
```

このコマンド行の *path* は、アーカイブファイルを保存するディレクトリです。*filename* は、アーカイブファイルの名前です。パスを指定しない場合、`flar create` によりアーカイブファイルを現在のディレクトリに保存します。

表 23-7 flar create のコマンド行オプション

オプション	説明
必須オプション	
-n <i>archive_name</i>	このオプションの値は、アーカイブの名前です。指定する <i>archive_name</i> は、 <i>content_name</i> キーワードの値になります。
圧縮用のオプション	
-c	compress (1) を使用してアーカイブを圧縮します。
ディレクトリとサイズを指定するオプション	
-R <i>root</i>	<i>root</i> のファイルシステムツリーからアーカイブを作成します。このオプションを指定しない場合、flar create は / ファイルシステムからアーカイブを作成します。
-S	アーカイブにサイズ情報を含めません。
-H	ハッシュ識別子を生成しません。
差分アーカイブを作成するオプション	
-A <i>unchanged_master_image_dir</i>	<p>新しいシステムイメージを <i>unchanged_master_image_dir</i> 引数で指定されたイメージと比較して、差分アーカイブを作成します。デフォルトでは、新しいシステムイメージはルート (/) です。-R オプションを使用してデフォルトを変更できます。<i>unchanged_master_image_dir</i> は、UFS、NFS、または lumount を使用して更新前のマスターシステムイメージが格納またはマウントされるディレクトリです。</p> <p>次に説明されている内容を選択するオプションを使用して、差分アーカイブに対するファイル選択を変更できます。</p>
-M	<p>マニフェストファイルを除外します。このオプションを使用すると、差分アーカイブに対する検証は行われません。差分アーカイブの作成時に、flar create はシステム内の変更されないファイル、変更されるファイル、アーカイブから削除されるファイルの長いリストを作成します。このリストは、アーカイブのマニフェストセクションに保存されます。差分アーカイブの配置時に、ソフトウェアはこのリストを使用してファイルごとにチェックを行い、クローンシステムの整合性が維持されることを保証します。このオプションを使用すると、このチェックが省略され、差分アーカイブ内のマニフェストセクションが使用するスペースを節約できます。ただし、インストール時に時間とディスク容量が節約できることと、整合性のチェックが実行されない影響とを比較考慮する必要があります。検証が実行されなくなるため、このオプションの使用は推奨しません。</p>

表 23-7 flar create のコマンド行オプション (続き)

オプション	説明
内容を選択するオプション	
<p>注意 - flar create のファイル除外オプションは、注意して使用してください。一部のディレクトリを除外する際、気づかずにシステム構成ファイルなどの他のファイルがアーカイブに残ってしまう場合があります。この場合、システムの整合性が損なわれるため、インストールが失敗してしまいます。ディレクトリやファイルの除外は、大規模なデータファイルなど、システムを破綻させることなく容易に削除可能なデータに対して行うのが最善です。</p>	
-y <i>include_dir/filename</i>	<p>コマンド行で指定されたファイルおよびディレクトリをアーカイブに追加します。ディレクトリは除外するが、個々のサブディレクトリやファイルを含める場合、このオプションを使用します。</p>
<p><i>include_dir/filename</i> には、含めるサブディレクトリやファイルの名前を指定します。</p>	
-f <i>list_filename</i>	<p>リストからファイルおよびディレクトリをアーカイブに追加します。</p>
<p><i>list_filename</i> は、リストを含むファイルのフルパスです。-F が指定されていない限り、ファイルの内容がファイルリストに追加されます。</p>	
<ul style="list-style-type: none"> <li>■ <i>list_filename</i> ファイルでは、1 ファイルを 1 行で表す必要があります。</li> <li>■ -R <i>root</i> を使用してファイルシステムを指定する場合、各ファイルのパスは代替 <i>root</i> ディレクトリからの相対パスにするか、絶対パスにする必要があります。</li> <li>■ <i>filename</i> が「-」の場合、flar create は標準入力からファイルのリストを読み取ります。「-」を指定すると、アーカイブのサイズは計算されません。</li> </ul>	
-F	<p>このオプションを指定すると、-f <i>list_filename</i> に示されるファイルだけを使って、アーカイブを作成します。このオプションを使用することで、-f <i>list_filename</i> は、アーカイブに含まれるファイルの絶対リストになります。</p>
-x <i>exclude_dir/filename</i>	<p>アーカイブからファイルおよびディレクトリを除外します。除外するファイルは、コマンド行で指定します。このオプションのインスタンスを複数使用して、複数のファイルまたはディレクトリを除外できます。</p>
<p><i>exclude_dir/filename</i> には、除外するディレクトリやファイルの名前を指定します。</p>	



表 23-7 flar create のコマンド行オプション (続き)

オプション	説明
-x <i>list_filename</i>	<p>アーカイブからリスト内のファイルまたはディレクトリを除外します。</p> <p><i>list_filename</i> は、リストを含むファイルのフルパスです。</p> <ul style="list-style-type: none"> <li>■ <i>list_filename</i> ファイルでは、1 ファイルを 1 行で表す必要があります。</li> <li>■ -R <i>root</i> を使用してファイルシステムを指定する場合、各ファイルのパスは代替 <i>root</i> ディレクトリからの相対パスにするか、絶対パスにする必要があります。</li> <li>■ <i>list_filename</i> が「-」の場合、<code>flar create</code> は標準入力からファイルのリストを読み取ります。「-」を指定すると、アーカイブのサイズは計算されません。</li> </ul>
-z <i>list_filename</i>	<p>アーカイブに対し、リスト内のファイルまたはディレクトリを除外または含めます。リスト内の各ファイルまたはディレクトリには、プラス「+」またはマイナス「-」記号を付けます。プラスはファイルやディレクトリを含めることを、マイナスはファイルやディレクトリを除外することを示します。</p> <p><i>list_filename</i> は、リストを含むファイルのフルパスです。</p> <ul style="list-style-type: none"> <li>■ <i>list_filename</i> ファイルでは、1 ファイルを 1 行で表す必要があります。</li> <li>■ -R <i>root</i> を使用してファイルシステムを指定する場合、各ファイルのパスは代替 <i>root</i> ディレクトリからの相対パスにするか、絶対パスにする必要があります。</li> </ul>
-I	<p>整合性チェックを無効にします。重要なシステムファイルをアーカイブから除外してしまうことを防ぐために、<code>flar create</code> は整合性チェックを実行します。このチェックにより、システムパッケージデータベースに登録されたすべてのファイルが検査され、そのいずれかが除外される場合、アーカイブの作成が停止します。このオプションを指定すると、この整合性チェックが無効になります。このため、-I オプションの使用は推奨しません。</p>
ユーザー定義のセクションで使用されるオプション	
-u <i>section</i>	<p><i>section</i> をユーザー定義のセクションとして含めます。ユーザー定義のセクションを複数含めるには、セクション名を空白で区切ったリストにします。</p>
-d <i>dir</i>	<p><i>dir</i> から、-u で指定したセクションを取り出します。</p>
テープアーカイブで使用されるオプション	
-t	<p>テープデバイス上にアーカイブを作成します。<i>filename</i> 引数は、テープデバイスの名前です。</p>

表 23-7 flar create のコマンド行オプション (続き)

オプション	説明
-p <i>posn</i>	-t オプションとの併用でのみ使用できます。flar create がアーカイブを格納するための、テープデバイス上の位置を指定します。このオプションを指定しない場合、flar create はテープの現在の位置にアーカイブを配置します。
-b <i>blocksize</i>	flar create がアーカイブの作成時に使用するブロックサイズを指定します。ブロックサイズを指定しない場合、flar create はデフォルトのブロックサイズ 64k を使用します。
アーカイブ識別のためのオプション	
以下のキーワードおよび値は、アーカイブの識別セクションに記載されます。	
-U <i>key=val</i>	識別セクションに、ユーザー定義のキーワードと値を含めます。
-i <i>date</i>	<i>date</i> は、creation_date キーワードの値として使用されます。 <i>date</i> を指定しない場合、flar create は現在のシステム日時を使用します。
-m <i>master</i>	<i>master</i> は、アーカイブを作成したマスターシステムの名前として使用されます。 <i>master</i> は、creation_master キーワードの値です。 <i>master</i> を指定しない場合、flar create は uname -n が出力するシステム名を使用します。
-e <i>descr</i>	<i>descr</i> は、content_description キーワードの値として使用されます。-E オプションを使用する場合はこのオプションを使用できません。
-E <i>descr_file</i>	<i>descr_file</i> ファイルから content_description キーワードの値を取得します。-e オプションを使用する場合はこのオプションを使用できません。
-a <i>author</i>	<i>author</i> は、識別セクション内の作成者名として使用されます。 <i>author</i> は content_author キーワードの値です。作成者を指定しない場合、flar create は 識別セクションの content_author キーワードを含めません。
-T <i>type</i>	<i>type</i> は、content_type キーワードの値として使用されます。 <i>type</i> はユーザー定義のキーワードです。タイプを指定しない場合、flar create は content_type キーワードを含めません。

## 第 24 章

# カスタム JumpStart インストール (トピック)

以下の章では、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 25 章	カスタム JumpStart インストールの紹介と概要を示します。
第 26 章	各サイトでカスタム JumpStart インストールを使用して Solaris 9 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。
第 27 章	カスタム JumpStart インストールツールで実行するスクリプトを作成するときに使用できるオプション機能について説明します。
第 28 章	独自のカスタムルールとカスタムプロンプトキーワードを作成するための情報と手順を説明します。
第 29 章	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 9 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してください。
第 30 章	カスタム JumpStart を使用して SPARC システムと x86 システムに Solaris ソフトウェアをインストールするための設定と実際のインストールを行う例を示します。
第 31 章	rules ファイル、プロファイル、begin スクリプト、および finish スクリプトで使用できるキーワードと値を示します。



## 第 25 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。

- 253 ページの「カスタム JumpStart とは」
- 254 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。それらの選択にもとづいて処理が行われます。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。

以下に、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、システムは次のパラメータを指定して設定する必要があるとします。

- 100 台の新しいシステムに Solaris をインストールする。
- 100 台の新しい SPARC システムのうち 70 台はエンジニアリング部が所有し、そのシステムは「開発者」Solaris オペレーティング環境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なければならない。
- 残りの 30 台は x86 ベースのシステムで、マーケティング部が所有し、そのシステムは「エンドユーザー」Solaris オペレーティング環境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なければならない。

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グループ用のルールで、もう1つはマーケティンググループ用のルールです。ルールごとに、エンジニアリンググループとマーケティンググループを区別するために、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、エンジニアリンググループ用のルールでは、エンジニアリンググループ用に作成した `eng_profile` というプロファイルへのリンクを追加します。マーケティンググループ用のルールでは、マーケティンググループ用に作成した `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が必要です。
- 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を使用します。

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ると、カスタム JumpStart インストールを開始できます。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 25-1 は、ネットワークに接続されていないスタンドアロン型のシステムでカスタム JumpStart インストールがどのように行われるかを示しています。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、システムのコピーディスクドライブに挿入されているコピーディスク上の rules ファイルにアクセスし、rule 2 をシステムに対応付けます。rule 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。

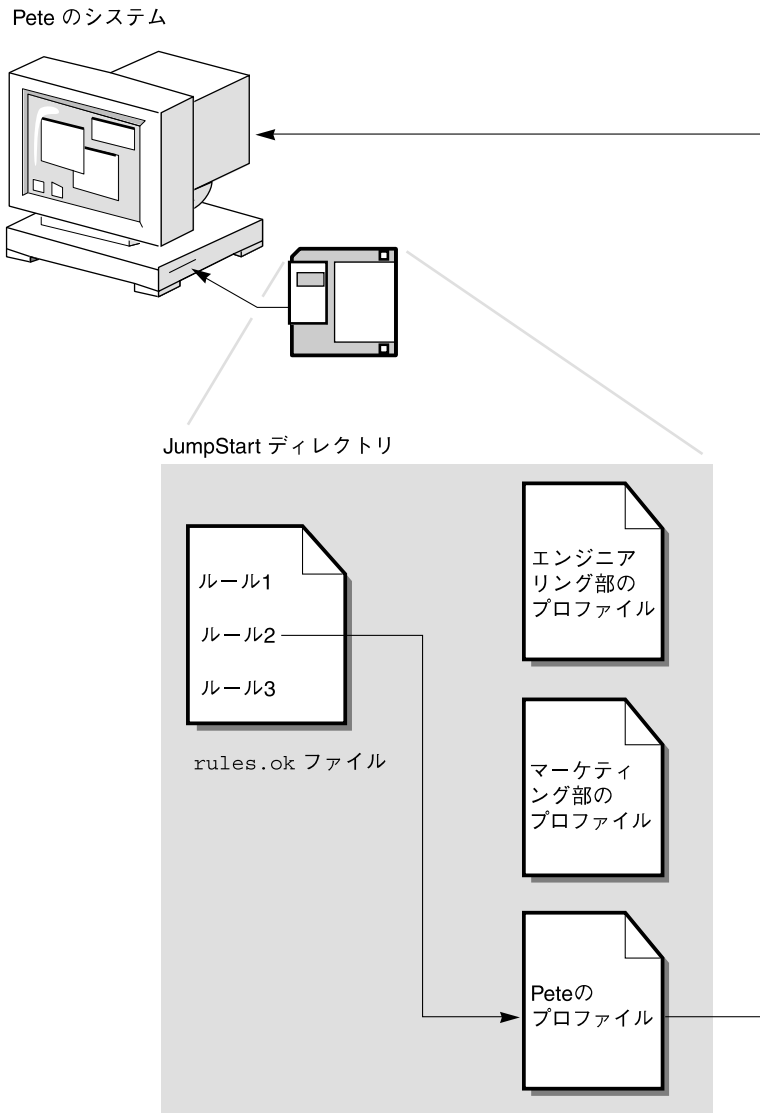


図 25-1 カスタム JumpStart インストールの動作: ネットワークに接続されていないシステムの例

図 25-2 は、ネットワーク上の複数のシステムが存在する場合にカスタム JumpStart インストールがどのように行われるかを示しています。すで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 rule 1 に対応付けます。rule 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。

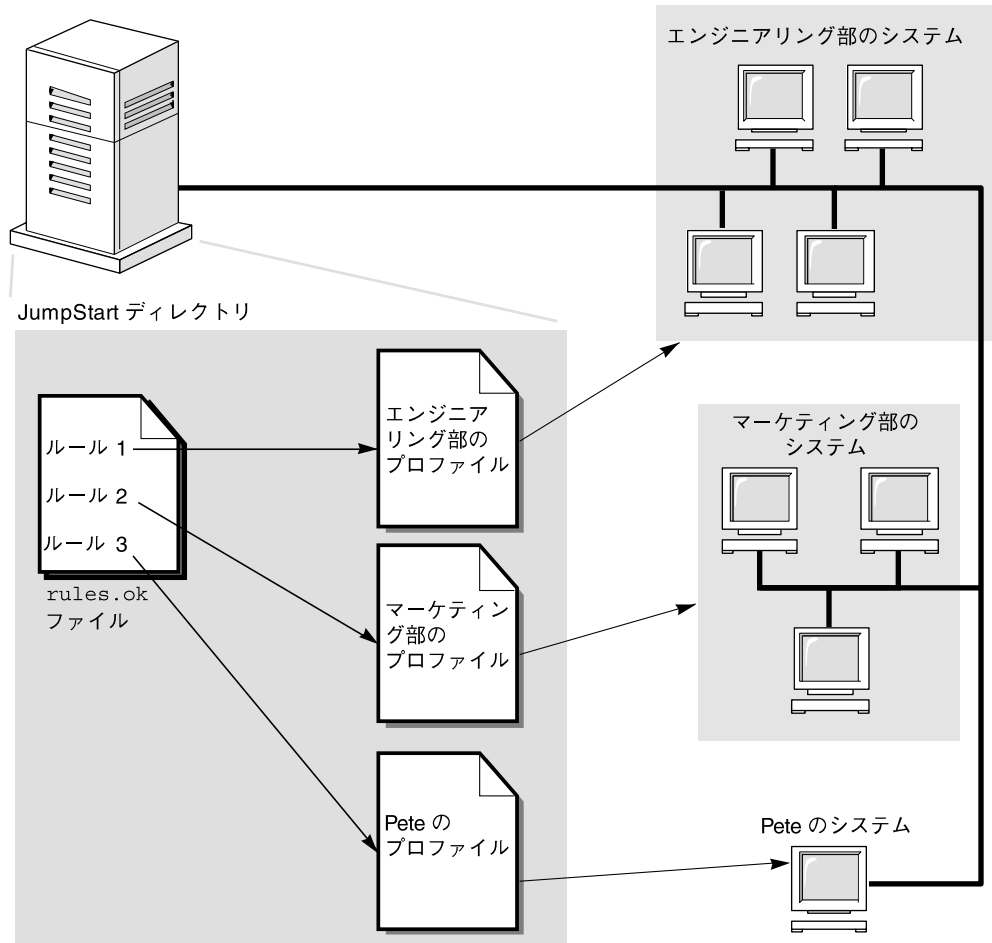


図 25-2 カスタム JumpStart インストールの動作: ネットワークに接続されているシステムの例

図 25-3 は、JumpStart プログラムがカスタム JumpStart ファイルを検索する順番を示しています。

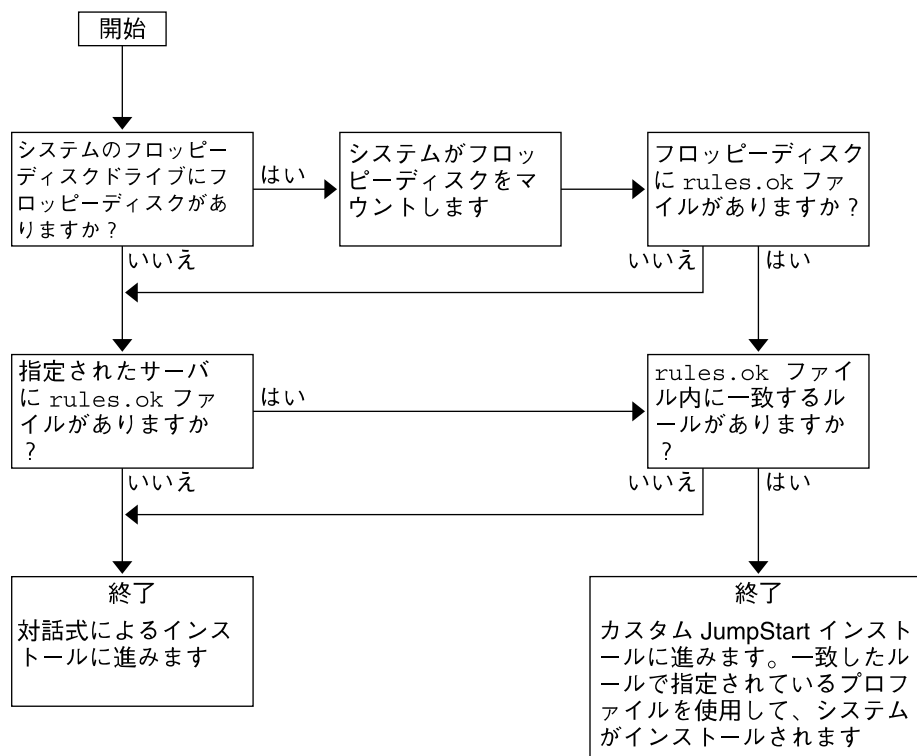


図 25-3 カスタム JumpStart インストール実行時の流れ

## 第 26 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 9 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。

- 260 ページの「作業マップ: カスタム JumpStart インストールの準備」
- 261 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」
- 265 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」
- 270 ページの「rules ファイルの作成」
- 274 ページの「プロファイルの作成」
- 284 ページの「プロファイルのテスト」
- 288 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 26-1 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する	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、begin スクリプト、finish スクリプトなどの作成が容易になります。	第 8 章
JumpStart ディレクトリを作成する	<p>サーバーの場合</p> <p>ネットワーク接続されたシステムに対してカスタム JumpStart インストールを実行する場合は、プロファイルサーバーを作成する必要があります。プロファイルサーバーには、カスタム JumpStart ファイル用の JumpStart ディレクトリが入っています。</p> <p>フロッピーディスクの場合</p> <p>ネットワークに接続されていないシステムに対してカスタム JumpStart インストールを実行する場合は、プロファイルフロッピーディスクを作成する必要があります。プロファイルフロッピーディスクには、カスタム JumpStart ファイルが入っています。</p>	<p>261 ページの「ネットワーク上のシステム用のプロファイルサーバーの作成」</p> <p>265 ページの「スタンドアロンシステム用のプロファイルフロッピーディスクの作成」</p>
ルールを rules ファイルに追加する	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後で、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。	270 ページの「rules ファイルの作成」

表 26-1 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する	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が、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。	274 ページの「プロファイルの作成」
(省略可能) プロファイル进行测试する	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、プロファイル进行测试できます。	284 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する	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 プログラムがプロファイルを使用してインストールするシステムを一致させるために使用されます。rules ファイルの妥当性を検査するには、 <code>check</code> スクリプトを使用する必要があります。	288 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 ディレクトリを持つサーバーは、「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、あるいは全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、異なるプラットフォームのシステムにもカスタム JumpStart ファイルを提供できます。たとえば、x86 サーバーは、SPARC システムと x86 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。詳細は、263 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を実行していると仮定しています。ボリュームマネージャを使用せずにメディアを管理する方法については、『Solaris のシステム管理 (基本編)』を参照してください。

---

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、ルートファイルシステムに *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. **/etc/dfs/dfstab** ファイルを編集します。次のエントリを追加してください。

```
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** と入力して **Enter** 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 へ進みます。
- コピーする場合は、次の表を利用して次に行う作業を決定します。

サンプルのコピー元	手順
プラットフォームに対応した Solaris DVD または Solaris SOFTWARE 1 of 2 CD	サーバーの CD-ROM ドライブに、Solaris DVD または Solaris SOFTWARE 1 of 2 CD を挿入します。 ボリュームマネージャが自動的に CD をマウントします。
ローカルディスク上で使用されているプラットフォーム用の Solaris DVD または Solaris SOFTWARE 1 of 2 CD	Solaris DVD または Solaris SOFTWARE 1 of 2 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>

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の **JumpStart** ディレクトリにコピーします。

```
# cp -r media_path/Solaris_9/Misc/jumpstart_sample/* jumpstart_dir_path
```

*media\_path* ディスク上の CD、DVD、またはイメージのパス

*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_9/Misc/jumpstart_sample/* /jumpstart
```

- x86 システムの場合:

```
cp -r /cdrom/cdrom0/s2/Solaris_9/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。  
詳細は、263 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方法

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。

- `add_install_client` 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、`add_install_client` コマンドの `-c` オプションを使用します。手順の詳細は 161 ページの「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を追加する」を参照してください。
- `boot` コマンド – 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。詳細は、299 ページの「圧縮された構成ファイルの作成」を参照してください。

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。詳細は、SPARC システムについては 316 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 を、x86 システムについては 320 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」の手順 8 をそれぞれ参照してください。
- `/etc/bootparams` ファイル – `/etc/bootparams` ファイルでワイルドカードを使用します。`etc/bootparams` ファイルにワイルドカードを追加するには、以下の手順に従ってください。

---

注 – 以下の手順は、`JumpStart` ディレクトリをフロッピーディスクに保存する場合と、システムのブート時にプロファイルサーバーの場所を指定する場合は不要です。

---

この手順は、ネットワークインストール情報が `/etc/bootparams` ファイルに格納されている場合だけ有効です。ネットワークインストール情報は、次の場所に格納することもできます。

- ネームサービスデータベース – ネットワークインストール情報がネームサービスの `bootparams` データベースに格納されている場合は、手順 3 に示すエントリを `bootparams` データベースに追加する必要があります。
  - DHCP サーバー – ネットワークインストール情報が DHCP サーバーに格納されている場合は、`boot` コマンドを使用してカスタム `JumpStart` が DHCP サーバーを使用するように指定する必要があります。詳細は、SPARC システムについては 316 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 を、x86 システムについては 320 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」の手順 8 をそれぞれ参照してください。
1. インストールサーバーまたはブートサーバーにスーパーユーザーとしてログインします。
  2. テキストエディタを使用して `/etc/bootparams` を開きます。
  3. 次のエントリを追加します。



```
* install_config=server:jumpstart_dir_path
*
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 というサーバーにある /jumpstart ディレクトリにアクセスできるようにします。
* install_config=sherlock:/jumpstart
```



---

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

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

このエラーメッセージの詳細は、648 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 ディレクトリを含む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呼ばれます。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、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。

## ▼ 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を実行していると仮定しています。ボリュームマネージャを使用せずにフロッピーディスク、CD、およびDVDを管理する方法の詳細は、『Solaris のシステム管理 (基本編)』を参照してください。

---

1. フロッピーディスクドライブを持つ **SPARC** システムに、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。  
システム上のファイル `/etc/mnttab` の内容を調べ、次のようなエントリがないか確認します。

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

- このようなエントリが存在する場合は、手順 7 へ進みます。
- エントリが存在しない場合は、次の手順へ進みます。

5. フロッピーディスクをフォーマットします。



---

注意 - この手順によって、フロッピーディスク上のデータはすべて消去されます。

---

```
# fdformat -U
```

6. フロッピーディスクに **UFS** ファイルシステムを作成します。

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

7. サンプルのカスタム **JumpStart** ファイルを **JumpStart** ディレクトリにコピーするかどうかを決定します。
  - コピーしない場合は、手順 10 へ進みます。
  - コピーする場合は、次の表を利用して次に行う作業を決定します。

サンプルのコピー元	手順
Solaris DVD (SPARC 版) または Solaris SOFTWARE 1 of 2 CD (SPARC 版)	サーバーの CD-ROM ドライブに Solaris DVD (SPARC 版) または Solaris SOFTWARE 1 of 2 CD (SPARC 版) を挿入します。  ボリュームマネージャが自動的に CD をマウントします。
ローカルディスク上の SPARC 版 Solaris 9 ソフトウェアの Solaris DVD (SPARC 版) または Solaris SOFTWARE 1 of 2 CD (SPARC 版) イメージ	Solaris DVD (SPARC 版) または Solaris SOFTWARE 1 of 2 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>

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

```
# cp -r media_path/Solaris_9/Misc/jumpstart_sample/* jumpstart_dir_path
```

*media\_path*                    ディスク上の CD、DVD、またはイメージのパス

*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 of 2 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_9_sparc/s0/Solaris_9/Misc/jumpstart_sample/* /floppy/scrap
```

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

```
# eject floppy
```

これで、プロファイルフロッピーディスクの作成が完了しました。rules ファイルを更新して、プロファイルフロッピーディスクにプロファイルを作成して、カスタム JumpStart インストールを実行できます。処理を続けるには、270 ページの「rules ファイルの作成」へ進んでください。

## ▼ x86: 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を実行していると仮定しています。ボリュームマネージャを使用せずにフロッピーディスク、CD、およびDVDを管理する方法の詳細は、『Solarisのシステム管理(基本編)』を参照してください。

---

1. フロッピーディスクドライブを持つ **x86** システムにスーパーユーザーとしてログインします。
2. **Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版)** 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 - 付録 I の手順に従って Solaris DVD (x86 版) または Solaris SOFTWARE 2 of 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 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にコピーします。

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

ここで *boot\_image* は、Solaris Device Configuration Assistant (デバイス構成用補助) のイメージをコピーするファイル名を示します。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 Device Configuration Assistant (デバイス構成用補助) (x86 版)** 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。




---

注意 - この手順によって、フロッピーディスク上のデータはすべて消去されます。

---

```
# fdformat -d -U
```

11. **Solaris 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
```

ここで *boot\_image* は、Solaris Device Configuration Assistant (デバイス構成用補助) のイメージをコピーするファイル名を示します。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 15 へ進みます。
  - コピーする場合は、次の表を利用して次に行う作業を決定します。

サンプルのコピー元	手順
Solaris DVD (x86 版) または Solaris SOFTWARE 1 of 2 CD (x86 版)	サーバーの CD-ROM ドライブに、Solaris DVD (x86 版) または Solaris SOFTWARE 1 of 2 CD (x86 版) を挿入します。 ボリュームマネージャが自動的に CD をマウントします。
ローカルディスク上の Solaris DVD (x86 版) または Solaris SOFTWARE 1 of 2 CD (x86 版) のイメージ	Solaris DVD (x86 版) または Solaris SOFTWARE 1 of 2 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

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** ディレクトリにコピーします。

```
# cp -r media_path/Solaris_9/Misc/jumpstart_sample/* jumpstart_dir_path
```

*media\_path*                    ディスク上の CD、DVD、またはイメージのパス

*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 of 2 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_9_x86/s2/Solaris_9/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。

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。処理を続けるには、270 ページの「rules ファイルの作成」へ進んでください。

---

## rules ファイルの作成

rules ファイルは、Solaris オペレーティング環境のインストール先となる各システムグループのルールを記述し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基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 - 265 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 261 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」の手順にしたがって JumpStart ディレクトリを設定した場合は、JumpStart ディレクトリに rules ファイルのサンプルがあります。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 ファイルは、次のように作成する必要があります。

- ファイルには rules という名前を割り当てる。
- ファイルには 1 つ以上のルールを含める。

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  
行の # 記号の後に含まれるテキストは、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続けるには、Return キーを押す直前に、バックslash 文字 (\) を含めます。

## ▼ rules ファイルを作成する方法

1. テキストエディタを使用して **rules** というテキストファイルを作成するか、あるいは作成済みの **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** ソフトウェアのインストール先となるシステムグループごとに 1 つのルールを **rules** ファイルに追加します。  
rules ファイルのキーワードと値の一覧は、335 ページの「ルールキーワードと値の説明」を参照してください。  
rules ファイル内のルールは、次の構文に従っている必要があります。

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

表 26-2 ルールの構文要素

要素	説明
!	ルールキーワードの前で使用し、否定を示す記号
<i>rule_keyword</i>	これは、ホスト名 ( <i>hostname</i> ) やメモリーサイズ ( <i>memsize</i> ) 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。 <i>rule_keyword</i> 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る。ルールキーワードの一覧は、335 ページの「ルールキーワードと値の説明」を参照
<i>rule_value</i>	対応するルールキーワードに特定のシステム属性を与える値。ルール値の一覧は、335 ページの「ルールキーワードと値の説明」を参照
&&	ルールキーワードとルール値のペアを同じルールで結合する (論理積をとる) ときに使用する記号。カスタム <b>JumpStart</b> 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ない
<i>begin</i>	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。 <i>begin</i> 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る。 <i>begin</i> スクリプトはすべて、 <b>JumpStart</b> ディレクトリになければならない。  <i>begin</i> スクリプトの作成方法の詳細は、291 ページの「 <i>begin</i> スクリプトの作成」を参照
<i>profile</i>	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る。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る。すべてのプロファイルは <b>JumpStart</b> ディレクトリになければならない  注 - プロファイルフィールドについては、別の使用方法もあります。詳細は、306 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 292 ページの「 <i>begin</i> スクリプトによる動的プロファイルの作成」を参照してください。
<i>finish</i>	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。 <i>finish</i> 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る。 <i>finish</i> スクリプトはすべて、 <b>JumpStart</b> ディレクトリになければならない  <i>finish</i> スクリプトの作成方法の詳細は、293 ページの「 <i>finish</i> スクリプトの作成」を参照

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 (-) (*begin* および *finish* 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対応付けることができた場合、JumpStart プログラムは **profile** 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ている Solaris ソフトウェアをインストールします。

例 26-1 rule ファイル

```
# rule keywords and rule values      begin script      profile      finish script
# -----
hostname eng-1                       -               basic_prof   -
network 192.43.34.0 && !model \
'SUNW,SPARCstation-20'              -               net_prof     -
model SUNW,SPARCstation-LX          -               lx_prof      complete
network 193.144.2.0 && karch i86pc  setup           x86_prof     done
memsize 64-128 && arch i386         -               prog_prof    -
any -                                -               generic_prof -
```

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

- |          |  |
|----------|--|
| hostname | このルールは、システムのホスト名が <code>eng-1</code> の場合に一致します。<br><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>192.43.34.0</code> にあって、 <code>SPARCstation™ 20 (SUNW, SPARCstation-20)</code> では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、271 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 <code>SPARCstation LX</code> である場合に一致します。 <code>lx_prof</code> プロファイルと <code>complete</code> 終了スクリプトは、このルールに一致するシステムに Solaris ソフトウェアをインストールするために使用されます。   |
| network  | このルールは、システムが <code>193.144.2.0</code> にあって、 <code>x86</code> ベースの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>x86_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。                                |

例 26-1 rule ファイル (続き)

- memsize このルールは、システムに 64 ~ 128M バイトのメモリーがあって、x86 システムである場合に一致します。prog\_prof 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。generic\_prof 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せん。any 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示した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 つまたは複数のプロファイルキーワードとそれらの値から構成されます。各プロファイルキーワードは、JumpStart プログラムがどのようにしてシステムに Solaris ソフトウェアをインストールするかを制御するコマンドです。たとえば、次のプロファイルキーワードと値は、JumpStart プログラムがシステムをサーバーとしてインクルードすることを指定しています。

```
system_type server
```

---

注 - 261 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」または 265 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」の手順を使用して JumpStart ディレクトリを作成した場合、プロファイルのサンプルが JumpStart ディレクトリにあります。

---

## 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

プロファイルには以下の要素を含めることができます。

- コメント  
行の # 記号の後に含まれるテキストは、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

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、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 - プロファイルの名前は、システムに Solaris ソフトウェアをインストールするためにそのプロファイルをどのように使用するかを示すものにしてください。(例: `basic_install`、`eng_profile`、`user_profile`)。

---

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。  
プロファイルのキーワードと値の一覧は、340 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. **root** がプロ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。
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に関する情報については、284 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。340 ページの「プロファイルキーワードと値」には、プロファイルキーワードと値の説明を示してあります。

### 例 26-2 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
fileSYS                any 512 swap # specify size of /swap
cluster                SUNWCprog
package                SUNWman delete
cluster                SUNWCacc
```

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

install_type	install_type キーワードは、すべてのプロファイルに必要です。
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	標準のマニュアルページは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ため、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。ただし、System Accounting ユーティリティ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 26-3 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
```

例 26-3 ファイルシステムのインストール場所の指定 (続き)

```
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 26-4 x86: fdisk キーワードの使用方法

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

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

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

**fdisk** タイプ **DOSOS16 (04 16 進)** の **fdisk** パーティションはすべて、**c0t0d0** ディスクから削除されます。

**fdisk** **Solaris fdisk** パーティションが、**c0t0d0** ディスク上の最大の連続空き領域に作成されます。

**cluster** 全体ディストリビューションソフトウェアグループ (SUNWCall) がシステムにインストールされます。

**cluster** システムアカウントユーティリティ (SUNWCacc) は、システムにインストールされません。

例 26-5 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
```

例 26-5 アップグレード用ディスク容量の再配置 (続き)

```
layout_constraint      c0t3d0s4 changeable
layout_constraint      c0t3d0s5 movable
package                SUNWbcp delete
package                SUNWxwman add
cluster                SUNWCacc add
locale                 de
```

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

install_type	このプロファイルは、ディスク容量を再配置することによってシステムをアップグレードします。この例では、システム上のファイルシステムのいくつかにアップグレード用の容量が十分でないため、ディスク領域を再配置する必要があります。
root_device	c0t3d0s2 のルートファイルシステムがアップグレードされます。
backup_media	リモートシステム timber が、ディスク容量の再配置中のデータのバックアップに使用されます。backup_media キーワード値の詳細は、347 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以下のことを実施できることを示します。 <ul style="list-style-type: none"><li>■ 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。</li><li>■ 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。</li></ul>
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	ドイツ語ローカライズパッケージがシステムにインストールされます。

例 26-6 HTTP サーバーからフラッシュアーカイブを取得する

次のプロファイル例は、カスタム JumpStart プログラムが HTTP サーバーからフラッシュアーカイブを取り出すことを示しています。

例 26-6 HTTP サーバーからフラッシュアーカイブを取得する (続き)

```
# profile keywords      profile values
# -----
install_type           flash_install
archive_location       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、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	フラッシュアーカイブは、HTTP サーバーから取得されません。
partitioning	ファイルシステムスライスは、filesys キーワード (explicit 値) によって指定します。ルート (/) のサイズはフラッシュアーカイブのサイズに基づいて決定されます。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 26-7 HTTPS サーバーからフラッシュアーカイブを取得する

次のプロファイル例は、カスタム JumpStart プログラムが HTTPS サーバーからフラッシュアーカイブを取り出すことを示しています。

```
# 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、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	圧縮されたフラッシュアーカイブが HTTPS サーバーから取得されます。

例 26-7 HTTPS サーバーから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 (/) のサイズはフラッシュアーカイブのサイズに基づいて決定されます。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 26-8 NFS サーバーからフラッシュ差分アーカイブを取得する

次のプロファイル例は、カスタム JumpStart プログラムが NFS サーバーからフラッシュアーカイブを取り出すことを示しています。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。

install\_type            このプロファイルは、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        フラッシュアーカイブは、NFS サーバーから取得されません。

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

例 26-9 空のブート環境の作成

次の例では、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、プロファイルが記述されています。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
```



例 26-9 空のブート環境の作成 (続き)

```
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 <code>explicit</code> であるため、ファイルシステムスライスは、 <code>filesys</code> キーワードによって指定します。ルート (/) のサイズは、値 <code>auto</code> 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 <code>c0t0d0s0</code> にインストールされます。 <code>swap</code> も同様に、必要なサイズに設定された上で <code>c0t3d0s1</code> 上にインストールされます。 <code>usr</code> は、値 <code>any</code> 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 ( <code>SUNWCall</code> ) 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 <code>c0t1d0</code> 上に設定されます。ルート (/)、スワップ、および <code>/export</code> のファイルシステムがそれぞれ作成されますが、空のままです。この 2 番目のブート環境には、あとでフラッシュアーカイブをインストールできます。その後、このブート環境をアクティブにして、オペレーティング環境にすることができます。

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。

- キーワード値の説明については、340 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、第 33 章を参照してください。
- フラッシュアーカイブの基本的な使用方法については、第 20 章を参照してください。

例 26-10 ルートファイルシステムをミラー化するための RAID-1 ボリュームの作成

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

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

例 26-10 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)

```
cluster          SUNWCXall
fileys           mirror:d30 c0t1d0s0 c0t0d0s0 /
fileys           c0t0d0s3 512 swap
metadb           c0t0d0s4 size 8192 count 4
metadb           c0t1d0s4 size 8192 count 4
```

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

- cluster 「全体ディストリビューション+OEM」ソフトウェアグループ (SUNWCXall) がシステムにインストールされます。
- fileys ルート (/) ファイルシステムが作成され、スライス c0t1d0s0 および c0t0d0s0 にミラー化されます。c0t1d0s0 および c0t0d0s0 をミラー化する RAID-1 ボリュームに d30 という名前が付けられます。カスタム JumpStart プログラムにより、2つのサブミラーに名前が割り当てられません。
- fileys swap ファイルシステムが作成され、スライス c0t0d0s3 にミラー化されます。サイズは 512M バイトです。
- metadb 4つの状態データベースの複製 (metadb) がスライス c0t0d0s4 にインストールされます。サイズは 8192 ブロック (4M バイト) です。
- metadb 4つの状態データベースの複製 (metadb) がスライス c0t1d0s4 にインストールされます。サイズは 8192 ブロック (4M バイト) です。
- 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は、第 10 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に関するガイドラインと要件については、第 11 章を参照してください。
  - キーワード値の説明については、359 ページの「fileys プロファイルキーワード (ミラー化されたファイルシステムの作成)」および 365 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 26-11 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
fileys                 mirror:d100 c0t1d0s0 c0t0d0s0 200 /
fileys                 c0t1d0s5 500 /var
fileys                 c0t0d0s5 500
```

例 26-11 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          mirror c0t0d0s1 512 swap
metadb          c0t0d0s3 size 8192 count 5
filesys         mirror c0t1d0s4 c0t0d0s4 2000 /usr
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は、第 10 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に関するガイドラインと要件については、第 11 章を参照してください。
- キーワード値の説明については、359 ページの「filesys プロファイルキーワード (ミラー化されたファイルシステムの作成)」および 365 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。

---

## プロファイルのテスト

プロファイルを作成した後で、`pfinstall(1M)` コマンドを使用し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` が生成するインストール出力を調べることによって、意図するとおりにプロファイルが動作するかを簡単に調べることができます。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、以下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については、301 ページの「ディスク構成ファイルの作成」と 302 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。

---

### ▼ 一時的な Solaris 9 環境を作成してプロファイル进行测试する方法

特定の Solaris リリースで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 環境内にあるプロファイル进行测试する必要があります。たとえば、Solaris 9 初期インストールプロファイル进行测试する場合は、Solaris 9 を実行しているシステムに対して `pfinstall` コマンドを実行する必要があります。

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

- 旧バージョンの Solaris ソフトウェアを使用しているシステム上で Solaris 9 のアップグレードプロファイル进行测试する場合。

- Solaris 9 の初期インストールプロファイル进行测试するための Solaris 9 システムをまだインストールしていない場合。
1. 次に示すメディアのいずれか 1 つのイメージからシステムをブートします。
    - Solaris DVD (SPARC 版)
    - Solaris DVD (x86 版)
    - Solaris SOFTWARE 1 of 2 CD (SPARC 版)
    - Solaris SOFTWARE 1 of 2 CD (x86 版)

---

注 - アップグレードプロファイル进行测试したい場合は、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します。
  - Solaris DVD を使用している場合は、次のプロンプトに対して ! を入力します。

```
Solaris Web Start will assist you in installing software for Solaris.  
<Press ENTER to continue> {"!" exits}
```
  - Solaris SOFTWARE 1 of 2 CD を使用している場合は、Solaris 9 オペレーティング環境をインストールするプログラムとして Solaris suninstall プログラムを選択します。表示される最初の画面で「Exit」を選択します。
4. このシェルから **pfinstall** コマンドを実行します。**pfinstall** コマンドの使用方法については、285 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

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。システムが、プロファイルを作成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**) のシステムであることを確認してください。

アップグレードプロファイル进行测试する場合、アップグレードする実際のシステム上のプロファイル进行测试してください。
2. 次の表に基づいて、次に行うことを決定します。

状態	手順
初期インストールプロファイル进行测试する (Solaris 9 ソフトウェアが動作しているシステムがある場合)	そのシステムで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する (初期インストールプロファイル进行测试するための Solaris 9 が動作しているシステムがない場合)	一時的な Solaris 9 環境を作成し、プロファイル进行测试します。詳細は、284 ページの「一時的な Solaris 9 環境を作成してプロファイル进行测试する方法」を参照してください。続いて、手順 3 へ進みます。

- 一時的なマウントポイントを作成します。

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

- テストしたいプロファイル (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する (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する	<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する	<code>mount -F pcfs /dev/diskette /tmp/mnt</code>

- システムのメモリーサイズを指定してプロファイル进行测试するには、**SYS\_MEMSIZE** に使用するメモリーサイズを **M** バイトで設定します。

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

- 手順 4 でディレクトリを

- 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
```

- 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 9 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、301 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`) で -d *disk\_config\_file* オプションを使用することはできません。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 9 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を使用して Solaris SOFTWARE 1 of 2 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 DVD または Solaris SOFTWARE 1 of 2 CD イメージからブートした場合は、-c オプションは不要です。DVD または CD イメージは、ブート処理の一環として /cdrom 上にマウントされます。

---

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

## プロファイルテストの例

次の例は、`pfinstall` を使用して `basic_prof` というプロファイル进行测试する方法を示しています。このテストは、Solaris 9 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を使用しているため、Solaris SOFTWARE 1 of 2 CD (SPARC 版) または Solaris SOFTWARE 1 of 2 CD (x86 版) イメージへのパスが指定されています。

例 26-12 Solaris 9 システムによる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 9 システム上の `basic_prof` というプロファイル进行测试する方法を示しています。`535_test` ディスク構成ファイルがテストされ、システムメモリーが 64M バイトあるか確認されます。また、この例で `pfinstall` は、`/export/install` ディレクトリにある Solaris SOFTWARE 1 of 2 CD (SPARC 版) または Solaris SOFTWARE 1 of 2 CD (x86 版) のイメージを使用します。

例 26-13 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 26-3 に `check` スクリプトの動作を示します。



表 26-3 check スクリプトを使用した場合の動作

手順	説明
1	rules ファイルの構文検査を行う。  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する( <i>begin</i> 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 (-) であることもある)。
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る。
3	エラーがなければ、check スクリプトは rules ファイルから rules.ok ファイルを作成する。コメントと空白行をすべて削除し、すべてのルールを保持し、終わりに次のコメント行を追加する。  # version=2 checksum=num

## ▼ rules ファイルの妥当性を検査する方法

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

---

注 - check スクリプトは、Solaris DVD または Solaris SOFTWARE 1 of 2 CD の Solaris\_9/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 Solaris 9 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 DVD、または Solaris SOFTWARE 1 of 2 CD 上のイメージです。

システムが以前のバージョンの Solaris を実行している場合、このオプションを使用して、最新バージョンの check スクリプトを実行します。

**-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 ファイルの有効性と各プロファイルの有効性の検査結果が表示されます。エラーが検出されないと、The custom JumpStart configuration is ok と表示されます。

4. **root** が **rules.ok** 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** ファイルの検証が完了したら、第 27 章でオプションのカスタム **JumpStart** 機能について参照してください。カスタム **JumpStart** インストールの実行については、第 29 章を参照してください。

## 第 27 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。

- 291 ページの「begin スクリプトの作成」
- 293 ページの「finish スクリプトの作成」
- 299 ページの「圧縮された構成ファイルの作成」
- 301 ページの「ディスク構成ファイルの作成」
- 306 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、異なるプラットフォームタイプ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 システムと x86 システムの両方に、カスタム JumpStart ファイルを提供できます。

---

---

## begin スクリプトの作成

「begin スクリプト」とは、ユーザーが定義する Bourne シェルスクリプトで、rules ファイル内で指定し、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。begin 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。

begin スクリプトは、次のような場合に使用してください。

- 動的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

## begin スクリプトに関する注意事項

- スクリプト内には、初期インストールまたはアップグレードインストール時に /a に対して行われるファイルシステムのマウントを妨げるような記述をしないでください。JumpStart プロファイルが /a にファイルシステムをマウントできない場合は、エラーが発生してインストールは失敗します。
- インストール時に、begin スクリプトの出力は /tmp/begin.log に格納されます。インストール完了後に、このログファイルは /var/sadm/system/logs/begin.log に出力先が変更されます。
- root が begin スクリプトを所有していて、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。
- begin スクリプト内では、カスタム JumpStart 環境変数を使用できます。環境変数の一覧は、371 ページの「カスタム JumpStart の環境変数」を参照してください。
- begin スクリプトは、JumpStart ディレクトリに保存してください。

## begin スクリプトによる動的プロファイルの作成

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

動的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。

- プロファイルの代わりに、プロファイルフィールドに等号 (=) を指定します。
- インストール対象のシステムに依存する動的プロファイルを作成する begin 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、begin 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動的プロファイルを作成します。

次に、毎回同じ動的プロファイルを作成する begin 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動的プロファイルを生成する begin スクリプトも作成できます。

### 例 27-1 動的プロファイルを作成する begin スクリプト

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

例 27-1 動的プロファイルを作成する begin スクリプト (続き)

```
echo "package          SUNWxman    delete">> ${SI_PROFILE}
```

たとえば、begin スクリプトは、デフォルトで /tmp/install.input に設定されている動的プロファイル名に SI\_PROFILE 環境変数を使用する必要があります。

---

注 – begin スクリプトを動的プロファイルの作成に使用する場合は、スクリプトにエラーがないことを確認してください。動的プロファイルは、begin スクリプトが実行されてから作成されます。このため、check スクリプトで動的プロファイルの妥当性は検査されません。

---

## finish スクリプトの作成

「finish スクリプト」とは、ユーザーが定義する Bourne シェルスクリプトで、rules ファイル内で指定し、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。finish 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。

finish 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

## finish スクリプトに関する注意事項

- Solaris suninstall 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。したがって、/a にマウントされている新しくインストールされたファイルシステムの階層にファイルを追加、変更、または削除するように finish スクリプトを作成できます。
- インストール時に、finish スクリプトの出力は /tmp/finish.log に格納されます。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 に出力先が変更されます。
- finish スクリプトの所有者は root、アクセス権は 644 にしてください。
- finish スクリプト内では、カスタム JumpStart 環境変数を使用できます。環境変数の一覧は、371 ページの「カスタム JumpStart の環境変数」を参照してください。

- finish スクリプトは、JumpStart ディレクトリに保存してください。

## ▼ finish スクリプトでファイルを追加する方法

finish 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。

---

1. インストールされた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したいファイルごとに、次の行を **finish** スクリプトに挿入します。

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

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、finish 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
```

## finish スクリプトによるパッケージとパッチの追加

finish 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。finish 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。

`pkgadd(1M)` コマンドや `patchadd(1M)` コマンドを finish スクリプトで使用するときには、`-R` オプションを使用して、`/a` をルートパスとして指定する必要があります。

- 例 27-2 はパッケージを追加する finish スクリプトの例を示します。
- 例 27-3 はパッチを追加する finish スクリプトの例を示します。

例 27-2 finish スクリプトでパッケージを追加する

```
#!/bin/sh

BASE=/a
```

例 27-2 finish スクリプトでパッケージを追加する (続き)

```
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}
```

以下に、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
```

- 下の `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 27-3 finish スクリプトでパッチを追加する

```
#!/bin/sh

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

# The location of the patches to add to the system after it's installed.
```

例 27-3 finish スクリプトでパッチを追加する (続き)

```
# 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"
if
```

---

注 - 以前は finish 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。しかし、一部のパッケージやパッチには -R オプションが正しく動作しないものがあります。このような状況では、chroot コマンドを実行する前に /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。

/etc/mnttab ダミーファイルを作成するには、次の行を finish スクリプトに追加します。

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

---



## finish スクリプトによるルート環境のカスタマイズ

finish スクリプトを使って、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できます。たとえば、例 27-4 の finish 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 27-4 finish 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
}
```

## finish 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 ソフトウェアがシステムにインストールされると、そのシステムはリブートします。ブートプロセス終了前に、システムはルートパスワードを入力するように求められます。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 finish スクリプトは、auto\_install\_sample ディレクトリに保存されます。この finish 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。例 27-5 に set\_root\_pw を示します。

---

注 - finish スクリプトを使用してシステムのルートパスワードを設定した場合は、finish スクリプト内の暗号化されたパスワードからルートパスワードを解読されないように対策を講じてください。

---

例 27-5 finish スクリプトによるシステムのルートパスワードの設定

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

例 27-5 finish スクリプトによるシステムのルートパスワードの設定 (続き)

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

echo "setting password for root"

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

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

以下に、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ルート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 ファイルにあるルート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ルート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.$$
```

例 27-5 finish スクリプトによるシステムのルートパスワードの設定 (続き)

## finish スクリプトを使用して Web Start インストールプログラムでソフトウェアをインストールする

finish スクリプトを使用すると、Solaris オペレーティング環境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。一部のソフトウェアプログラムは、インストール時に情報入力を求めるプロンプトを表示する Solaris Web Start プログラムでインストールできます。自動インストールを行うには、`-nodisplay` または `-noconsole` オプションを指定して Solaris Web Start プログラムを実行します。

表 27-1 Solaris Web Start 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> オプションによってインストールが変更された場合を除き、デフォルトの製品インストールを使用してください。
<code>-noconsole</code>	対話形式のテキストコンソールデバイスを介さずにインストールを実行します。UNIX スクリプトで <code>-nodisplay</code> と共に使用すると便利です。

詳細は、`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、`boot` コマンドに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。圧縮された構成ファイルの形式は、次のいずれかにしてください。

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

## ▼ 圧縮された構成ファイルを作成する方法

1. プロファイルサーバー上の **JumpStart** ディレクトリに移動します。

```
# cd jumpstart_dir_path
```

2. 圧縮ツールを使って、すべてのカスタム **JumpStart** 構成ファイルを 1 つのファイルに圧縮します。

---

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

---

圧縮された構成ファイルには、次のファイルが含まれている必要があります。

- プロファイル
- rules
- rules.ok

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

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

## 圧縮された構成ファイルの例

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

例 27-6 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または複数ディスク構成ファイルを作成する方法について説明します。ディスク構成ファイルを作成すると、単一のシステムから `pfinstall(1M)` を使用して、異なるディスク構成に対してプロファイルをテストできます。

### ▼ SPARC: ディスク構成ファイルを作成する方法

1. テストするディスクを持つ **SPARC** システムを決定します。
2. スーパーユーザーになります。
3. 単一ディスク構成ファイルを作成するには、`prtvtoc(1M)` コマンドの出力をファイルに保存します。

```
# 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 の形式にしてください。  
  
disk_config_file         ディスク構成ファイルの名前
```

4. 複数のディスクに **Solaris** ソフトウェアをインストールするテストを
  - 実行しない場合は、ここで作業は終了です。
  - 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のディスク構成ファイルにおいて、ディスクデバイス名のターゲット番号がそれぞれ
  - 固有である場合は、ここで作業は終了です。
  - 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 システムで、単一ディスク構成ファイル 04\_test を作成します。

例 27-7 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用してプロファイルをテストする方法の詳細は、284 ページの「プロファイルのテスト」を参照してください。

## ▼ x86: ディスク構成ファイルを作成する方法

1. テストするディスクを持つ x86 システムを決定します。
2. スーパーユーザーになります。

3. 単一ディスク構成ファイルを作成するには、**fdisk(1M)** コマンドの出力をファイルに保存します。

```
# 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 *cwtxdyp0* または *cxdyp0* の形式に  
してください。

4. **prtvtoc(1M)** コマンドの出力をディスク構成ファイルに追加します。

```
# 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* の形式にしてください。

*disk\_config*                        ディスク構成ファイルの名前

5. 複数のディスクに **Solaris** ソフトウェアをインストールするテストを

- 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
```

6. 前の手順で作成した複数のディスク構成ファイルにおいて、ディスクデバイス名のターゲット番号がそれぞれ

- 固有である場合は、ここで作業は終了です。
- 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 27-8 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:  PCIIXOS
* 4:  DOSOS16
* 5:  EXTDOS
* 6:  DOSBIG
* 86: DOSDATA
* 98: OTHEROS
* 99: UNIXOS
* 130: SUNIXOS
*
* Id  Act  Bhead Bsect   Bcyl  Ehead  Esect   Ecyl  Rsect  Numsect
130  128  44    3      0     46    30     1001  1410   2050140
```

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

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

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

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



例 27-8 x86: ディスク構成ファイルの作成 (続き)

```
*      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
*           Sector  Count  Sector  Mount Directory
* 2         5    01      1410  2045910 2047319
* 7         6    00      4230  2043090 2047319 /space
* 8         1    01         0    1410    1409
* 9         9    01      1410    2820   422987
```

これで x86 システム用のディスク構成ファイルの作成が終了しました。ディスク構成ファイルを使用してプロファイル进行测试する方法の詳細は、284 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

`begin` スクリプトと `finish` 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 (-) が指定されているときには、`begin` スクリプトと `finish` スクリプトが、プロファイルや Solaris `suninstall` プログラムの代わりに、システムのインストール方法を制御します。

たとえば次のルールに一致した場合は、`begin` スクリプト `x_install.beg` と `finish` 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 28 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。

- 307 ページの「プローブキーワード」
- 308 ページの「custom\_probes ファイルの作成」
- 311 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、begin スクリプトおよび finish 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を入手できるのは、installed ルールキーワードに続いて SI\_INSTALLED を使用する場合だけです。

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 begin および finish 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の一覧は、373 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

335 ページの「ルールキーワードと値の説明」と 373 ページの「プローブキーワードと値」で説明したルールおよびプローブキーワードでは不十分な場合は、`custom_probes` ファイルを作成して、独自のカスタムルールまたはプローブキーワードを定義できます。

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

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

## custom\_probes ファイルの構文

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

---

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

`rules` ファイル内で対応するカスタムルールキーワードを使用する場合、引数はこのキーワードの後から解釈され、次の `&&` または `begin` スクリプトのどちらか最初にきた方の前で解釈が終了します。

---

`custom_probes` ファイルは、次の要件を満たす必要があります。

- `custom_probes` という名前を使用する
- 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 の関数名の構文

プローブ関数名は、probe\_ で始める必要があります。比較関数名は、cmp\_ で始める必要があります。

probe\_ で始まる関数は、新しいプローブキーワードを定義します。たとえば、関数 probe\_tcx は、新しいプローブキーワード tcx を定義します。cmp\_ で始まる関数は、新しいルールキーワードを定義します。たとえば、cmp\_tcx は、新しいルールキーワード tcx を定義します。

### ▼ custom\_probes ファイルを作成する方法

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

---

注 - 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、引数はこのキーワードの後から解釈され、次の && または begin スクリプトのどちらか最初に来た方の前で解釈が終了します。

---

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 DVD または Solaris SOFTWARE 1 of 2 CD 上の  
/Solaris\_9/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 28-1 custom\_probes ファイル

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

#
# COMPARISON FUNCTIONS
#
cmp_tcx() {
    probe_tcx

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

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

---

注 - プローブキーワードは、rules ファイルの先頭または先頭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前に読み取られて実行されるようにするためです。

---

例 28-2 rules ファイルで使用されるカスタムプローブキーワード

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

---

## custom\_probes ファイルの妥当性検査

プロファイル、rules、および custom\_probes ファイルを使用するには、これらのファイルの設定が正しいかどうかを check スクリプトを実行して検査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 28-1 は、check スクリプトの実行内容を説明しています。

表 28-1 check スクリプトを使用した場合の動作

手順	説明
1	check は、custom_probes ファイルを検索します。
2	この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白行を削除し、すべての Bourne シェルコマンド、変数、およびアルゴリズムを保存し、次に check は次のコメント行をファイルの最後に追加します。  # version=2 checksum=num

### ▼ custom\_probes ファイルを妥当性検査する方法

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

---

注 - check スクリプトは、Solaris DVD または Solaris SOFTWARE 1 of 2 CD の Solaris\_9/Misc/jumpstart\_sample ディレクトリにあります。

---

2. **JumpStart** ディレクトリに移動します。
3. check スクリプトを実行して、**rules** および **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、プラットフォームの Solaris 9 ソフトウェアイメージの check スクリプトを使用して、custom\_probes ファイルを妥当性検査します。path は、ローカルディスクか、あるいはマウントされた Solaris DVD または Solaris SOFTWARE 1 of 2 CD 上のイメージです。

システムが以前のバージョンの Solaris を実行している場合、このオプションを使用して、最新バージョンの check スクリプトを実行します。

`-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` ファイルと `custom_probes` ファイル、および各プロファイルの妥当性が表示されます。エラーが検出されないと、「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。

- 実行可能な場合は、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 29 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 9 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してください。

- 316 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」
- 320 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」

---

## SPARC: カスタム JumpStart インストールの実行

### 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

カスタム JumpStart インストールの際に、JumpStart プログラムはインストール対象のシステムを `rules.ok` ファイル内のルールに対応付けます。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。

表 29-1 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する	ハードウェアのマニュアルで、Solaris 9 環境におけるシステムサポートを確認します。	Solaris 9 Sun ハードウェアマニュアル
Solaris 9 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する	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 5 章
(省略可能) システム構成情報を事前設定する	sysidcfg ファイルまたはネームサービスを使用してシステムのインストール情報をあらかじめ設定できます。システム情報をあらかじめ設定すると、インストール時に情報入力を求めるプロンプトは表示されません。	第 7 章
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する	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 26 章
(省略可能) カスタム JumpStart のオプション機能を準備する	begin スクリプト、finish 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 27 章と第 28 章
(省略可能) ネットワーク上でインストールするためのシステムを設定する	リモートの Solaris 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。	第 15 章
(省略可能) フラッシュアーカイブのインストール準備をする	フラッシュアーカイブをインストールする前に、特定の手順を実行する必要があります。	315 ページの「カスタム JumpStart インストールを使用して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行う	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。	316 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」

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

カスタム JumpStart を使用して、初期インストール用の完全アーカイブまたはアップグレード用の差分アーカイブを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。

- 完全アーカイブまたは差分アーカイブについては、第 20 章を参照してください。
  - 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する手順については、462 ページの「プロファイルを使用した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。
1. 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 ファイルの作成手順の詳細は、第 26 章を参照してください。
  2. 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。フラッシュアーカイブのプロファイル例については、276 ページの「プロファイルの例」を参照してください。

表 31-2 に記載されているカスタム JumpStart のプロファイルキーワードのリストの中で、フラッシュアーカイブをインストールする時に有効なキーワードは、以下の X 印の付いたものだけです。

表 29-2 フラッシュアーカイブのキーワード

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

- 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、フラッシュアーカイブへのパスを追加します。
- `archive_location` プロファイルキーワードの詳細は、342 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。
- 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) クローンシステムにフラッシュアーカイブを追加インストールしたい場合は、インストールしたいアーカイブごとに **archive\_location** 行を指定してください。
3. インストールサーバーで、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。
- 手順の詳細は、128 ページの「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を追加する」または 161 ページの「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を追加する」を参照してください。
4. 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。
- 詳細は、316 ページの「SPARC: カスタム JumpStart プログラムを使用してインストールまたはアップグレードする方法」を参照してください。

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

1. システムがネットワークに接続されている場合は、**Ethernet** コネクタまたはそれに類似したネットワークアダプタがシステムに装着されていることを確認します。
2. **tip (1)** ラインで接続されているシステムをインストールする場合は、ウィンドウ表示が横 **80** 桁、縦 **24** 行以上あることを確認します。

`tip` ウィンドウの現在の大きさを調べるには、`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 9** ソフトウェアをインストールする場合は、ドライブに **Solaris DVD (SPARC 版)** または **Solaris SOFTWARE 1 of 2 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、システムのコ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せん。
- 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 プロンプトで次のコマンドを入力します。

```
ok boot cdrom:net - install [url:ask] [dhcp] [nowin]
```

cdrom CD または DVD からブートすることを指定します。

システムの EEPROM が旧式のものである場合は、cdrom を sd (0,6,2) に置き換えてシステムの CD-ROM または DVD-ROM ドライブからブートしてください。

net 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意味します。

url カスタム JumpStart ファイルの場所を指定します。以下にあるファイルを URL で指定できます。

- ローカルハードディスク

```
file://jumpstart_dir_path/compressed_config_file
```

- ネットワークファイルシステム (NFS) サーバー

```
nfs://server_name:IP_address/jumpstart_dir_path/compressed_config_file
```

- HTTP サーバー

```
http://server_name:IP_address/jumpstart_dir_path/  
compressed_config_file&proxy_info
```

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。

```
http://131.141.2.32/jumpstart/config.tar
```

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。

```
http://www.shadow.com/jumpstart/  
config.tar&proxy=131.141.6.151
```

ask 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。

Return キーを押してこのプロンプトへの入力を行わないと、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、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。Return キーを押してこのプロンプトへの入力を行わないと、Solaris suninstall プログラムが開始されます。

**dhcp** DHCP サーバーから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を取得します。

dhcp 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。

**nowin** X Window を起動しません。カスタム JumpStart インストールの実行には、X Window は必要ありません。このため、このオプションを指定すると、インストール時間を短縮できます。

---

**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

---

## x86: カスタム JumpStart インストールの実行

### 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

カスタム JumpStart インストールの際に、JumpStart プログラムはインストール対象のシステムを `rules.ok` ファイル内のルールに対応付けます。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。

表 29-3 x86: 作業マップ: カスタム JumpStart インストールのためのシステムの設定

作業	説明	参照先
既存のオペレーティングシステムとユーザーデータを保存する必要があるかを決定する	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 9 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。	354 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する	ハードウェアのマニュアルで、Solaris 9 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 9 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する	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 5 章

表 29-3 x86: 作業マップ: カスタム JumpStart インストールのためのシステムの設定 (続き)

作業	説明	参照先
(省略可能) システム構成情報を事前設定する	<code>sysidcfg</code> ファイルまたはネームサービスを使用してシステムのインストール情報をあらかじめ設定できます。システム情報をあらかじめ設定すると、インストール時に情報入力を求めるプロンプトは表示されません。	第 7 章
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する	<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 26 章
(省略可能) カスタム JumpStart のオプション機能を準備する	<code>begin</code> スクリプト、 <code>finish</code> スクリプトなどの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の準備を行います。	第 27 章と第 28 章
(省略可能) ネットワーク上でインストールするためのシステムを設定する	リモートの Solaris 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。	第 15 章
インストールまたはアップグレードを行う	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。	320 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」

## ▼ 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

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。  
tip ウィンドウの現在の大きさを調べるには、`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、システムのコモドロン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 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 **DVD-ROM** または **CD-ROM** ドライブを使用して **Solaris 9** ソフトウェアをインストールする場合は、ドライブに **Solaris DVD (x86 版)** または **Solaris SOFTWARE 1 of 2 CD (x86 版)** を挿入します。
5. システムのブート方法を決定します。
  - Solaris DVD または Solaris INSTALLATION 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 が使用でき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 Device Configuration Assistant (デバイス構成用補助) (x86 版) を挿入します。

---

**x86** のみ - 付録 I の手順に従って Solaris DVD (x86 版) または Solaris SOFTWARE 2 of 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。

---

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

Enter the number of your choice followed by the <ENTER> key.

If you enter anything else, or if you wait for 30 seconds, 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 suninstall 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。

---

- 2 を入力し、Enter キーを押します。
- カスタム JumpStart 構成ファイルの場所を指定するには、次のコマンドを入力します。

**b install** [*url:ask*] [*dhcp*] [*nowin*]

*url*        カスタム JumpStart ファイルの場所を指定します。以下にあるファイルを URL で指定できます。

- ローカルハードディスク

`file://jumpstart_dir_path/compressed_config_file`

- ネットワークファイルシステム (NFS) サーバー

`nfs://server_name:IP_address/jumpstart_dir_path/compressed_config_file`

- HTTP サーバー

`http://server_name:IP_address/jumpstart_dir_path/compressed_config_file&proxy_info`

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。

`http://131.141.2.32/jumpstart/config.tar`

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。

`http://www.shadow.com/jumpstart/  
config.tar&proxy=131.141.6.151`

*ask*        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。

Return キーを押してこのプロンプトへの入力を行わないと、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、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。Return キーを押してこのプロンプトへの入力を行わないと、Solaris `suninstall` プログラムが開始されます。

`dhcp` DHCP サーバーから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を取得します。

`dhcp` 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。

`nowin` X Window を起動しません。カスタム JumpStart インストールの実行には、X Window は必要ありません。このため、このオプションを指定すると、インストール時間を短縮できます。

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。

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`



## 第 30 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 を使用して SPARC システムと x86 システムに Solaris ソフトウェアをインストールするための設定と実際のインストールを行う例を示します。

- 325 ページの「サイトの設定例」
- 327 ページの「インストールサーバーの作成」
- 328 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」
- 329 ページの「JumpStart ディレクトリの作成」
- 329 ページの「JumpStart ディレクトリの共有」
- 329 ページの「SPARC: エンジニアリンググループのプロファイルの作成」
- 330 ページの「x86: マーケティンググループのプロファイルの作成」
- 330 ページの「rules ファイルの更新」
- 331 ページの「rules ファイルの妥当性を検査する」
- 332 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」
- 332 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」
- 333 ページの「SPARC: エンジニアリングシステムのブートと Solaris 9 ソフトウェアのインストール」
- 334 ページの「x86: マーケティングシステムのブートと Solaris 9 ソフトウェアのインストール」

---

## サイトの設定例

この例で使用するサイトの設定を図 30-1 に示します。

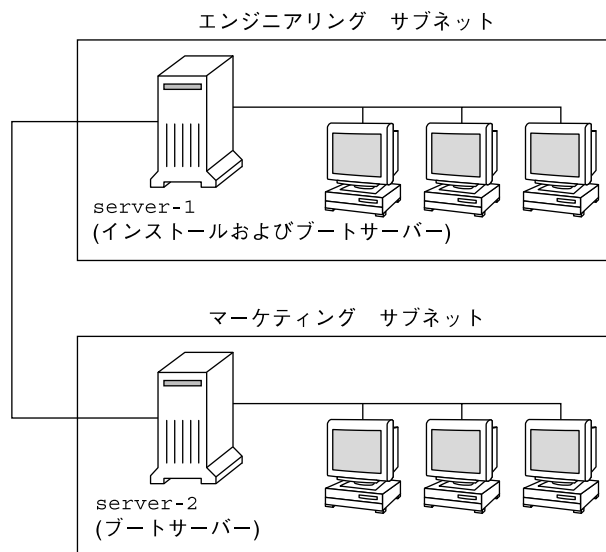


図 30-1 サイトの設定例

この例におけるサイトの状況は次のとおりです。

- SPARC: エンジニアリンググループは、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、ソフトウェア開発用に SPARCstation™ システムを使用しています。
- x86: マーケティンググループも、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 9 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

これらのグループは Solaris 9 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。これらのイメージは、Solaris SOFTWARE CD と Solaris LANGUAGES CD、または Solaris DVD からコピーしてください。

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。これらの例では、`sparc_9` および `x86_9` ディレクトリが該当します。

### 例 30-1 SPARC: Solaris 9 CD のコピー

`server-1` の CD-ROM ドライブに Solaris SOFTWARE 1 of 2 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_9
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/sparc_9
```

`server-1` の CD-ROM ドライブに Solaris SOFTWARE 2 of 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_9/Tools
server-1# ./add_to_install_server /export/install/sparc_9
```

`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_9/Tools
server-1# ./add_to_install_server /export/install/sparc_9
```

### 例 30-2 x86: Solaris 9 CD のコピー

`server-1` の CD-ROM ドライブに Solaris SOFTWARE 1 of 2 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_9
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/x86_9
```

`server-1` の CD-ROM ドライブに Solaris SOFTWARE 2 of 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_9/Tools
server-1# ./add_to_install_server /export/install/x86_9
```

`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。

例 30-2 x86: Solaris 9 CD のコピー (続き)

```
server-1# cd /CD_mount_point/Solaris_9/Tools
server-1# ./add_to_install_server /export/install/x86_9
```

例 30-3 SPARC: Solaris 9 DVD のコピー

server-1 の DVD-ROM ドライブに Solaris 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_9
server-1# cd /DVD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/sparc_9
```

例 30-4 x86: Solaris 9 DVD のコピー

server-1 の DVD-ROM ドライブに Solaris 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_9
server-1# cd /DVD_mount_point/Solaris_9/Tools
server-1# ./setup_install_server /export/install/x86_9
```

---

## x86: マーケティングシステム用のブートサーバーの作成

システムは、異なるサブネット上のインストールサーバーからはブートできません。したがって、server-2 をマーケティンググループのサブネット上のブートサーバーにします。setup\_install\_server(1M) コマンドを使用して Solaris DVD (x86 版) または Solaris SOFTWARE 1 of 2 CD (x86 版) から server-2 ローカルディスクの /export/boot ディレクトリへブートソフトウェアをコピーします。

server-2 の CD-ROM ドライブに Solaris SOFTWARE 1 of 2 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_9/Tools
server-2# ./setup_install_server -b /export/boot
```

server-2 の DVD-ROM ドライブに Solaris 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_9/Tools
server-2# ./setup_install_server -b /export/boot
```

setup\_install\_server コマンドの -b 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ことを指定します。



---

## 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終了したところで、次は `server-1` に JumpStart ディレクトリを作成します (ネットワーク上の任意のシステムを利用できます)。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 DVD イメージまたは Solaris SOFTWARE 1 of 2 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_9/Solaris_9/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が `rules` ファイルやプロファイルにアクセスできるようにするために、`/jumpstart` ディレクトリを共有します。このためには、`/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、コマンド行で `shareall` コマンドを使用します。

```
server-1# shareall
```

---

## SPARC: エンジニアリンググループのプロファイルの作成

エンジニアリングシステムの場合、`eng_prof` ファイルを `/jumpstart` ディレクトリに作成します。`eng_prof` ファイルのエントリは次のようになっています。エンジニアリンググループのシステムにインストールされる Solaris 9 ソフトウェアを定義しています。

```
install_type  initial_install1
system_type   standalone2
partitioning  default3
cluster       SUNWCprog4
fileysys      any 512 swap5
```

1. アップグレードではなく、初期インストールによりインストールするよう指定します。

2. エンジニアリングシステムが、スタンドアロンシステムであることを指定します。
3. 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、エンジニアリングシステムに Solaris をインストールするよう指定します。
4. 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するように指定します。
5. エンジニアリンググループの各システムが、512M バイトのスワップ領域を持つよう指定します。

---

## x86: マーケティンググループのプロファイルの作成

マーケティングシステムの場合、marketing\_prof ファイルを /jumpstart ディレクトリに作成します。marketing\_prof ファイルのエントリは次のようになっています。マーケティンググループのシステムにインストールされる Solaris 9 ソフトウェアを定義しています。

```
install_type  initial_install1
system_type   standalone2
partitioning  default3
cluster       SUNWCuser4
package       SUNWaudio5
```

1. アップグレードではなく、初期インストールによりインストールするよう指定します。
2. マーケティングシステムが、スタンドアロンであることを指定します。
3. 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 をインストールするよう指定します。
4. 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するように指定します。
5. 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するよう指定します。

---

## 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、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、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、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と、255.222.44.0 ネットワーク上のシステムが marketing\_prof プロファイルを使用してインストールされることを示しています。

---

注 - サンプルのルールでは、どのシステムが eng\_prof または marketing\_prof を使用してインストールされるかをネットワークアドレスを使用して識別できます。この他にも、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リストについては、表 31-1 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を適切に設定した後は、check スクリプトを実行して、これらのファイルを確認します。

```
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 ディレクトリおよび必要なファイルを設定した後は、インストールサーバー (server-1) で `add_install_client` コマンドを使用して、インストールサーバーから Solaris 9 ソフトウェアをインストールでき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、エンジニアリンググループ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_9/Solaris_9/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
```

`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。

<code>-c</code>	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。
<code>host-eng1</code>	エンジニアリンググループのシステム名
<code>host-eng2</code>	エンジニアリンググループの別のシステム名
<code>sun4u</code>	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アーキテクチャを指定する。Ultra 5 システムのプラットフォームグループ

---

## 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。このコマンドで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 9 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_9/Tools
server-2# ./add_install_client -s server-1:/export/install/x86_9 \
-c server-1:/jumpstart host-mkt1 i86pc
server-2# ./add_install_client -s server-1:/export/install/x86_9 \
-c server-1:/jumpstart host-mkt2 i86pc
server-2# ./add_install_client -d -s server-1:/export/install/x86_9 \
-c server-1:/jumpstart SUNW.i86pc i86pc
```

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 ネットワークインストールパラメータの取得にクライアントが DHCP を使用することを指定します。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 9 software へのパス (/export/install/x86\_9)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。
- host-mkt1 マーケティンググループのシステム名
- host-mkt2 マーケティンググループの別のシステム名
- SUNW.i86pc すべての Solaris x86 クライアントの DHCP クラス名。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したい場合は、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 9 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後は、各システムの ok (PROM) プロンプトで次の boot コマンドを使用して、エンジニアリングシステムをブートできます。

```
ok boot net - install
```

システムは自動的に Solaris オペレーティング環境をエンジニアリンググループのシステムにインストールします。

---

## x86: マーケティングシステムのブートと Solaris 9 ソフトウェアのインストール

次に示す方法のどれか 1 つを使用してシステムをブートできます。

- Solaris SOFTWARE 1 of 2 CD (x86 版)
- Solaris 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 Device Configuration Assistant (デバイス構成用補助) (x86 版)

Solaris 9 はマーケティンググループのシステムに自動的にインストールされます。

## 第 31 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 begin/finish 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。

- 335 ページの「ルールキーワードと値の説明」
- 340 ページの「プロファイルキーワードと値」
- 371 ページの「カスタム JumpStart の環境変数」
- 373 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 31-1 で、rules ファイルで使用できるルールキーワードとルール値について説明します。rules ファイルの作成方法についての詳細は、270 ページの「rules ファイルの作成」を参照してください。

表 31-1 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する。
arch	<i>processor_type</i> <i>processor_type</i> に有効な値: <ul style="list-style-type: none"><li>■ SPARC: sparc</li><li>■ x86: i386</li></ul>	システムのプロセッサタイプを照合する。 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る。

表 31-1 ルールキーワードとルール値の説明 (続き)

キーワード	値	説明
disksize	<p><i>actual_disk_name</i> <i>size_range</i></p> <p><i>actual_disk_name</i> - cxyzdz (c0t3d0 または c0d0 など) という形式のディスク名、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る。</p> <ul style="list-style-type: none"> <li>■ 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 でインストールされた新しい SPARC ベースのシステム) を持つディスク</li> <li>■ c0t3d0s0 ディスク (存在する場合)</li> <li>■ 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)</li> </ul> <p><i>size_range</i> - ディスクのサイズ。M バイト単位の範囲 (x-x) で指定する必要がある。</p>	<p>システムのディスクの名前とサイズを照合する (M バイト単位)。</p> <p>例:</p> <pre>disksize c0t3d0 250-300</pre> <p>この例では、JumpStart プログラムは c0t3d0 というシステムディスクの照合を試みる。このディスクは 250M から 300M バイトの情報を保持できる。</p> <p>例:</p> <pre>disksize rootdisk 750-1000</pre> <p>この例では、JumpStart プログラムはまず事前にインストールされたブートイメージが入ったシステムディスクを照合する。続いて c0t3d0s0 ディスク (存在する場合) を照合し、最後に、750M から 1G バイトの情報を保持できるディスクの中で利用可能な最初のディスクを照合する。</p> <p>注 - <i>size_range</i> を計算するときは、1M バイトが 1,048,576 バイトであることに注意してください。「535M バイト」ディスクと明記されているディスクでも、ディスク空間が 510M バイトしかない場合があります。535,000,000/1,048,576=510 により、JumpStart は「535M バイト」ディスクを実際には 510M バイトのディスクと見なします。したがって、この「535M バイト」ディスクは「530-550」の <i>size_range</i> には一致しません。</p>
domainname	<i>actual_domain_name</i>	<p>システムのドメイン名を照合する。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する。</p> <p>システムがインストール済みの場合、domainname コマンドによりシステムのドメイン名を表示できる。</p>
hostaddress	<i>actual_IP_address</i>	システムの IP アドレスを照合する。
hostname	<i>actual_host_name</i>	<p>システムのホスト名を照合する。</p> <p>システムがインストール済みの場合、uname -n コマンドによりシステムのホスト名を表示できる。</p>



表 31-1 ルールキーワードとルール値の説明 (続き)

キーワード	値	説明
installed	<p><i>slice version</i></p> <p><i>slice</i> — <i>cwtxdysz</i> 形式 (たとえば、<i>c0t3d0s5</i>) のディスクスライス名、または <i>any</i> か <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する (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る。</p> <ul style="list-style-type: none"> <li>■ 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 でインストールされた新しい SPARC ベースのシステム) を持つディスク</li> <li>■ <i>c0t3d0s0</i> ディスク (存在する場合)</li> <li>■ 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)</li> </ul> <p><i>version</i> — バージョン名、あるいは <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する。<i>upgrade</i> を使用すると、アップグレード可能な Solaris 2.6 以降の互換リリースのどれとでも照合する。</p> <p>Solaris リリースは検出されたが、バージョンを判断できないという場合は、返されるバージョンは <i>SystemV</i> になる。</p>	<p>Solaris ソフトウェアの特定バージョンに対応するルートファイルシステムが存在するディスクを照合する。</p> <p>例:</p> <pre>installed c0t3d0s1 Solaris_9</pre> <p>この例では、<i>c0t3d0s1</i> に Solaris 9 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る。</p>
karch	<p><i>actual_platform_group</i></p> <p>有効な値は、<i>sun4m</i>、<i>sun4u</i>、<i>i86pc</i>、<i>prep</i>。システムおよび対応するプラットフォームグループの一覧は、『Solaris 9 ハードウェアマニュアル』に記載されている。</p>	<p>システムのプラットフォームグループを照合する。</p> <p>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る。</p>
memsize	<p><i>physical_mem</i></p> <p>値は M バイト単位の範囲 (<i>x-x</i>) または 1 つの M バイト値で指定する。</p>	<p>システムの物理メモリーサイズを照合する (M バイト単位)。</p> <p>例:</p> <pre>memsize 64-128</pre> <p>この例では、64M から 128M バイトの物理メモリーサイズを持つシステムと照合している。</p> <p>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る。</p>

表 31-1 ルールキーワードとルール値の説明 (続き)

キーワード	値	説明
model	<i>actual_platform_name</i>	<p>システムのプラットフォーム名を照合する。有効なプラットフォーム名については、『Solaris 9 Sun ハードウェアマニュアル』を参照してください。</p> <p>インストール済みのシステムのプラットフォーム名を見つけるには、<code>uname -i</code> コマンドか <code>prtconf</code> コマンド (5 行目) の出力を使用する。</p> <p>注 - <i>actual_platform_name</i> にスペースが含まれている場合は、スペースを下線 (<u>  </u>) で置き換える必要があります。</p> <p>例:</p> <p>SUNW,Sun_4_50</p>
network	<i>network_num</i>	<p>システムのネットワーク番号を照合する。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する。</p> <p>例:</p> <p>network 193.144.2.8</p> <p>この例では、IP アドレスが 193.144.2.8 のシステムを照合する (サブネットマスクが 255.255.255.0 の場合)</p>
osname	Solaris_x	<p>システムにすでにインストールされている Solaris のバージョンを照合する。</p> <p>例:</p> <p>osname Solaris_9</p> <p>この例では、JumpStart プログラムは Solaris 9 オペレーティング環境がすでにインストールされているシステムを照合する。</p>

表 31-1 ルールキーワードとルール値の説明 (続き)

キーワード	値	説明
probe	<i>probe_keyword</i>	<p>有効なプローブキーワードまたは有効なカスタムプローブキーワード</p> <p>例:</p> <pre>probe disks</pre> <p>この例は、システムのディスクサイズ (M バイト) をカーネルプローブ順に返します (例: SPARC システムでは 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。</p> <p>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。このキーワードは、値を返すだけです。したがって、probe ルールキーワードで、begin スクリプト、プロファイル、および finish スクリプトは指定できません。</p> <p>プローブキーワードについては、第 28 章を参照。</p>

表 31-1 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
totaldisk	<p><i>size_range</i></p> <p>値は M バイト単位の範囲 (<i>x-x</i>) で指定する必要があります。</p>	<p>システムのディスク空間の全体量 (M バイト単位) を照合する。ディスク空間の全体量には、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る。</p> <p>例:</p> <pre>totaldisk 300-500</pre> <p>この例では、全体として 300M~500M バイトのディスク空間を持つシステムと照合している。</p> <p>注 - <i>size_range</i> を計算するときは、1M バイトが 1,048,576 バイトであることに注意してください。「535M バイト」ディスクと明記されているディスクでも、ディスク空間が 510M バイトしかない場合があります。535,000,000/1,048,576=510 により、JumpStart は「535M バイト」ディスクを実際には 510M バイトのディスクと見なします。したがって、この「535M バイト」ディスクは「530-550」の <i>size_range</i> には一致しません。</p>

## プロファイルキーワードと値

次の節では、プロファイルで使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。プロファイル作成の詳細は、274 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リスト

表 31-2 を使用すれば、どのキーワードがユーザーのインストールに適しているかを簡単に決定できます。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 31-2 プロファイルキーワード

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 (フラッシュアーカイブをインストールする場合)	X	X			
backup_media					X
boot_device	X	X	X		
bootenv createbe	X	X	X		
client_arch			X		
client_root			X		
client_swap			X		
cluster (ソフトウェアグループを追加する場合)	X	X	X		
cluster (クラスタを追加または削除する場合)	X	X	X	X	X
dontuse	X	X	X		
fdisk (x86 のみ)	X	X	X		
filesystem (リモートファイルシステムをマウントする場合)		X	X		
filesystem (ローカルファイルシステムを作成する場合)	X	X	X		
filesystem (ミラーファイルシステムを作成する場合)	X	X	X		
forced_deployment (フラッシュ差分アーカイブをインストールする場合)	X	X			
geo	X	X	X	X	X
install_type	X	X	X	X	X
isa_bits	X	X	X	X	X
layout_constraint					X

表 31-2 プロファイルキーワード (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
local_customization (フラッシュアーカイブをインストールする場合)	X	X			
locale	X	X	X	X	X
metadb (状態データベースの複製を作成する場合)	X	X	X		
no_master_check (フラッシュ差分アーカイブをインストールする場合)	X	X			
no_content_check (フラッシュ差分アーカイブをインストールする場合)	X	X			
num_clients			X		
package	X	X	X	X	X
partitioning	X	X	X		
root_device	X	X	X	X	X
system_type	X	X	X		
usedisk	X	X	X		

## 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` と `location` の値は、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。

- 343 ページの「ネットワークファイルシステム (NFS) サーバー」
- 343 ページの「HTTP または HTTPS サーバー」
- 344 ページの「FTP サーバー」
- 345 ページの「ローカルテープ」
- 346 ページの「ローカルデバイス」

■ 347 ページの「ローカルファイル」

## ネットワークファイルシステム (NFS) サーバー

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 `$HOST` を含めると、フラッシュインストールユーティリティは `$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。

*filename* 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、フラッシュユーティリティがアーカイブのマウントを試みる最大回数です。

例:

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

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

## HTTP または HTTPS サーバー

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
```

*server\_name* アーカイブを格納するサーバーの名前です。*server\_name* 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。

*port* オプションのポートです。ポートを指定しない場合、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。

*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 `$HOST` を含めると、フラッシュインストールユーティリティは `$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。

*filename* フラッシュアーカイブファイルの名前です。

*optional\_keywords* 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 31-3 *optional\_keywords* 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>auth basic user_name password</i>	<p>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。</p> <p>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。</p>
<i>timeout min</i>	<p><i>timeout</i> キーワードには、HTTP サーバーからのデータ受信を待機する最長の時間を分単位で指定できます。この時間に達すると、接続が切断されて、再接続が行われ、タイムアウトが発生した地点から再開されます。<i>timeout</i> 値として 0 (ゼロ) を指定すると、何も起きないため再接続されません。</p> <p>タイムアウトによる再接続が発生すると、フラッシュインストールユーティリティ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。</p>
<i>proxy host:port</i>	<p><i>proxy</i> キーワードを使用して、プロキシホストとプロキシポートを指定できます。プロキシホストを使用すると、ファイアウォール越しにフラッシュアーカイブを取得できます。<i>proxy</i> キーワードを指定する場合は、プロキシポートを指定する必要があります。</p>

例:

```
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

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。*server\_name* 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。



<i>port</i>	オプションのポートです。ポートを指定しない場合、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FTPポート番号 21 を使用します。
<i>path</i>	指定したサーバーから取得するアーカイブの場所です。このパスに \$HOST を含めると、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>	フラッシュアーカイブファイルの名前です。
<i>optional_keywords</i>	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 31-4 archive\_location ftp で使用可能なオプションのキーワード

キーワード	値の定義
<i>timeout min</i>	<p>timeout キーワードには、FTP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、接続が切断されて、再接続が行われ、タイムアウトが発生した地点から再開されます。timeout 値として 0 (ゼロ) を指定すると、何も起きないため再接続されません。</p> <p>タイムアウトによる再接続が発生すると、フラッシュインストールユーティリティ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。</p>
<i>proxy host:port</i>	<p>proxy キーワードを使用して、プロキシホストとプロキシポートを指定できます。プロキシホストを使用すると、ファイアウォール越しにフラッシュアーカイブを取得できます。proxy キーワードを指定する場合は、プロキシポートを指定する必要があります。</p>

例:

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

## ローカルテープ

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

```
archive_location local_tape device position
```

*device* 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、フラッシュインストールユーティリティ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、フラッシュインストールユーティリティはパスに /dev/rmt/ を加えます。

*position* アーカイブを保存したテープドライブ上の位置を示します。位置が指定されないと、フラッシュインストールユーティリティ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:

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

```
archive_location local_tape 0n 5
```

## ローカルデバイス

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 CD-ROM など) に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>	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せん。正規のパスでないデバイス名を指定すると、フラッシュインストールユーティリティはパスに <code>/dev/dsk/</code> を加えます。
<i>path</i>	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 <code>\$HOST</code> を含めると、フラッシュインストールユーティリティは <code>\$HOST</code> をインストール先のクローンシステムの名前に置き換えます。
<i>filename</i>	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、フラッシュインストールユーティリティは、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、フラッシュインストールユーティリティは HSFS ファイルシステムのマウントを試みます。

例:

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

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

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

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

## ローカルファイル

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、CD-ROM 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、CD-ROM または NFS ベースのミニルートに格納した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 INSTALLATION CD または Solaris 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 INSTALLATION CD または Solaris 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。

*filename* フラッシュアーカイブファイルの名前です。

例:

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

## backup\_media プロファイルキーワード

```
backup_media type path
```

---

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

---

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

有効な <i>type</i> 値	有効な <i>path</i> 値	仕様
local_tape	/dev/rmt/ <i>n</i>	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する。 <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る。 <i>n</i> はテープドライブの番号
local_diskette	/dev/rdisketten	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する。 <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る。 <i>n</i> はフロッピーディスクドライブの番号  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い
local_filesystem	/dev/dsk/cwtxdysz /file_system	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する。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ない。 <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わない。たとえば、/dev/dsk/cwtxdysz 内の tx は必須ではない。あるいは、 <i>path</i> 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への絶対パスでもかまわない
remote_filesystem	host:/file_system	リモートシステムの NFS ファイルシステムを指定する。 <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らない。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る
remote_system	user@host:/directory	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する。アップグレードされるシステムは、リモートシステムの .rhosts 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らない。 <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らない。ユーザーログイン ( <i>user</i> ) を指定しないと、スーパーユーザーとしてログインされる

例:

```

backup_media local_tape /dev/rmt/0

backup_media local_diskette /dev/rdiskette1

backup_media local_filesystem /dev/dsk/c0t3d0s4

backup_media local_filesystem /export

```

```
backup_media remote_filesystem system1:/export/temp
```

```
backup_media remote_system user1@system1:/export/temp
```

## boot\_device プロファイルキーワード

`boot_device device eeprom`

`boot_device` には、JumpStart がルート (/) ファイルシステムをインストールするデバイスと、システムのブートデバイスを指定します。

`boot_device` キーワードをプロファイルに指定しない場合、インストール中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* - 以下の値のいずれかを使用して、ブートデバイスにするデバイスを指定します。

- SPARC: `cwtxdysz` または `cxdysz` - JumpStart プログラムがルート (/) ファイルシステムを格納するディスクスライス。たとえば、`c0t0d0s0`。
- x86: `cwtxdy` または `cxd y` - JumpStart プログラムがルート (/) ファイルシステムを格納するディスク。たとえば、`c0d0`。
- `existing` - JumpStart プログラムは、システムの既存のブートデバイスにルート (/) ファイルシステムを格納します。
- `any` - JumpStart プログラムは、ルート (/) ファイルシステムを配置する場所を選択します。システムの既存のブートデバイスを使用する場合がありますが、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。

SPARC: システムの EEPROM を、指定したブートデバイスに変更または保存する場合に選択します。

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自動的に指定したブートデバイスからブートするように、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。

---

注 - SPARC: SPARC システムでは、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。

---

例:

```
boot_device c0t0d0s2 update
```

---

注 - `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 (指定した場合) に一致する必要があります。

---

## 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、オペレーティング環境のインストール時に、空の非アクティブブート環境を作成できます。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。ファイルシステムは作成されますが、中身は空のままです。現在のブート環境からファイルシステムがコピーされることはありません。新しいブート環境に名前が付けられ、ファイルシステムの追加の準備が整います。この空のブート環境にフラッシュアーカイブをインストールできます。アーカイブをインストールした後でこのブート環境をアクティブにし、このブート環境からシステムを稼働できます。*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環境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、`/dev/dsk/cwt.xdysz` という形式で表されます。
- *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す
  - `swap`: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す

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。

- プロファイルの例については、例 26-9を参照
- 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、第 32 章を参照
- フラッシュアーカイブの基本的な使用方法については、第 19 章を参照

## 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。`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` に有効な値は、`sun4m`、`sun4u`、および `i86pc` です。プラットフォーム名と各種のシステムの詳細は、『Solaris 9 ハードウェアマニュアル』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せん。

---

## client\_root プロファイルキーワード

`client_root root_size`

`client_root` は、各クライアント用に割り当てるルート領域の大きさ (`root_size`、M バイト単位) を定義します。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`

プロファイル内の `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領域の大きさ (`swap_size`、M バイト単位) を定義します。`client_swap` を指定しない場合、32M バイトのスワップ領域がデフォルトで割り当てられます。

例:

```
client_swap 64
```

この例は、各ディスククライアントが 64M バイトのスワップ領域を持つことを定義します。

---

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

---

## スワップサイズを決定する方法

プロファイルがスワップ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 31-5 に、カスタム JumpStart インストール中にスワップのサイズがどのように決定されるかを示します。

表 31-5 スワップのサイズの決定

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

JumpStart プログラムはスワップのサイズがスワップ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、他のファイルシステムを配置した後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スワップに割り当てて、可能であれば、表 31-5 に示す量を割り当てます。

---

注 - 物理メモリーとスワップ領域の合計は、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。各ソフトウェアグループの `group_name` 名は次のとおりです。



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

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。

## 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
```

注 - `cluster` (クラスタの追加または削除) は、初期インストールオプションとアップグレードオプションの両方で使用できます。

`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または削除するかを指定します。

`cluster_name` は `SUNWCname` 形式で指定します。インストールが終了したシステムで `Admintool` を起動し、「ブラウズ」メニューから「ソフトウェア」を選択すると、クラスタの詳細情報とクラスタ名を表示できます。

`add_delete_switch` は、オプション `add` または `delete` を示します。このオプションを使用すると、指定したクラスタを追加または削除できます。`add_delete_switch` を指定しないと、デフォルトによって `add` が使用されます。

次に、アップグレード時に `cluster` (クラスタの追加または削除) を使用した場合の動作を示します。

- 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。

## 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 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。

---

**x86 のみ** - 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。369 ページの「システムのルートディスクを決定する方法」で説明しているように、これは JumpStart プログラムで決定されます。
- 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 fdisk パーティションの別名 (データ DOS 用に拡張または予約されている fdisk パーティションの別名ではない)。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 (有効な値は 1 から 255 までの整数)

---

**x86** のみ - 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 (有効な値は 01 から FF までの 16 進数)

---

**x86** のみ - 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 - 次の中から 1 つを使用します。

- `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とインストールは失敗します。値 `maxfree` を指定できるのは、`type` が `solaris` または `dosprimary` の場合だけです。

---

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

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

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

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

例:

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

`server`:- 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ます)

`path` - 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)

`server_address` - `server:path` で指定するサーバーの IP アドレス。ネットワーク上で実行されているネームサービスがない場合、この値 `server_address` を使用して、サーバーのホスト名と IP アドレスを登録している `/etc/hosts` ファイルを生成できます。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 (-) を指定してください。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がありません。

`mount_pt_name` - 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

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

---

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

---

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

*filesystem slice size [file\_system optional\_parameters]*

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

*slice* - 次の中から 1 つを指定します。

- *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*。
- *rootdisk.sn* - システムのルートディスク値が含まれる変数。369 ページの「システムのルートディスクを決定する方法」に示されているように、この値は JumpStart プログラムが決定します。拡張子 *sn* は、ディスク上の特定のスライスを示します。

*size* - 次の中から 1 つを使用します。

- *num* - ファイルシステムのサイズを *num* (M バイト単位) で設定します。
- *existing* - 既存のファイルシステムの現在のサイズを使用します。

---

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

---

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

- `all` – 指定した `slice` は、そのファイルシステム用にディスク全体を使用します。この値を指定すると、指定したディスク上にはほかのファイルシステムは存在できません。
- `free` – ディスク上の残りの未使用領域をファイルシステム用に使用します。

---

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

---

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

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

- `mount_pt_name` – ファイルシステムのマウントポイント名 (例: `/var`)。
- `swap` – 指定した `slice` をスワップとして使用します。
- `overlap` – 指定した `slice` をディスク領域 (VTOC 値は `V_BACKUP`) を表すものとして定義します。デフォルトでは、スライス 2 はディスク全体を表すオーバーラップスライスです。

---

注 – `size` に `existing`、`all`、または `start:size` を指定した場合だけ `overlap` を指定できます。

---

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

`optional_parameters` – 次のいずれか 1 つを使用します。

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

---

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

---

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

---

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

---

## filesystem プロファイルキーワード (ミラー化されたファイルシステムの作成)

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

JumpStart プログラムでは、filesystem 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、filesystem mirror を繰り返し指定できます。

---

注 - filesystem 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。

---

*name* - 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。ミラー名の付け方については、106 ページの「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。

*slice* - 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、ミラー化するファイルシステムを配置します。スライスの値は、*cwtxdysz* 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。ミラー化するスライスは最大 2 つまで指定できます。

*size* - 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。

*file\_system* - この値は、ミラー化するファイルシステムを示します。カスタム JumpStart プログラムは、指定されたスライスからミラーを作成し、このミラーを指定されたファイルシステムへマウントします。ルート(/)、/usr、/var などの重要なファイルシステムに加えて、swap 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)。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)。

インストール時に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、第 10 章を参照してください。

## forced\_deployment プロファイルキーワード (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 locale

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。locale に指定できる値は、次のとおりです。

値	説明
N_Africa	北アフリカ。エジプトを含む
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、パナマを含む
N_America	北アメリカ。カナダ、アメリカ合衆国を含む
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、ベネズエラを含む
Asia	アジア。日本、韓国、中華人民共和国、台湾、タイを含む
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む
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、スイスを含む



値	説明
E_Europe	東ヨーロッパ。アルバニア、ボスニア、ブルガリア、クロアチア、エストニア、ラトビア、リトアニア、マケドニア、ルーマニア、ロシア、セルビア、スロヴェニア、トルコを含む
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、スウェーデンを含む
S_Europe	南ヨーロッパ。ギリシャ、イタリア、ポルトガル、スペインを含む
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、オランダを含む
M_East	中近東。イスラエルを含む

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、付録 D 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_swith`

`install_type` は、システムにおいて、(既存の Solaris オペレーティング環境を) 消去して、新しい Solaris オペレーティング環境をインストールするか、既存の Solaris オペレーティング環境をアップグレードするか、あるいは、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。

- `initial_install` - Solaris オペレーティング環境を初期インストールすることを指定します。
- `upgrade` - Solaris オペレーティング環境をアップグレードすることを指定します。
- `flash_install` - すべてのファイルを上書きする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。
- `flash_update` - 指定したファイルだけを上書きする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。

---

## isa\_bits プロファイルキーワード

`isa_bits bit_switch`

`isa_bits` は、64 ビットまたは 32 ビットのどちらの Solaris 9 パッケージをインストールするかを指定します。

`bit_switch` は、オプション 64 または 32 を表します。これは、64 ビットまたは 32 ビットのどちらの Solaris 9 パッケージをインストールするかを指定するために使用します。このキーワードをプロファイルに設定しない場合、JumpStart プログラムは次のようにパッケージをインストールします。

- UltraSPARC™ システムの場合は 64 ビットパッケージ
- それ以外のシステムの場合は 32 ビットパッケージ

---

注 - `isa_bits` キーワードを使用する場合は、Solaris SOFTWARE 1 of 2 CD または Solaris DVD 上の `Solaris_9/Misc/jumpstart_sample` ディレクトリの最新の `check` スクリプトを使用する必要があります。

---

## layout\_constraint プロファイルキーワード

`layout_constraint slice constraint [minimum_size]`

---

注 - `layout_constraint` 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。

---

`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。

`layout_constraint` キーワードを指定しないと、次のように配置されます。

- 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` とマークされません。

- 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と、次のように配置されます。

- 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。

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 `minimum_size` 値を変更できます。

---

注 - 自動配置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。

---

*slice* - これは、制約を指定するファイルシステムのディスクスライスです。システムのディスクスライスは `cwt.xdysz` または `cxdysz` 形式で指定する必要があります。

*constraint* - 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。`minimum_size` 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、`minimum_size` 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。`minimum_size` 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。

- `movable` - 自動配置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。
- `available` - 自動配置は、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、`/etc/vfstab` 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せん。

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

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

例:

```
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 (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へのパスです。配置前および配置後スクリプトの詳細は、215 ページの「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せん。

---

`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。有効なロケールの値については、付録 D を参照してください。

`local` 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。

## 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 ポリユーモマネージャの状態データベースの複製 (`metadb`) 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。

<code>slice</code>	カスタム <code>JumpStart</code> 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。 <code>slice</code> の値の形式は、 <code>cwtzdxysz</code> です。
<code>size size-in-blocks</code>	<code>size</code> 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。 <code>size</code> を指定しないと、カスタム <code>JumpStart</code> 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。
<code>count number-of-replicas</code>	プロファイル内にオプションの <code>count</code> 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。 <code>count</code> を指定しないと、カスタム <code>JumpStart</code> プログラムは、デフォルトにより、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 ポリユーモマネージャの状態データベースの複製を作成する方法の詳細は、106 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 (フラッシュアーカイブのインストール)

`no_content_check`

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にある場合を除き、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。

---

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

`no_master_check`

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にある場合を除き、このキーワードの使用は避けてください。フラッシュ差分アーカイブのインストールについては、315 ページの「カスタム `JumpStart` インストールを使用して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 `swap` ファイルシステムにディスク空間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_delete_switch]`

---

注 - `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せん。

---

`package` は、システムにインストールするソフトウェアグループにパッケージを追加または削除するかを指定します。

`package_name` は、`SUNWname` の形式で指定する必要があります。パッケージとその名前の詳細を表示するには、インストール済みシステムで `pkginfo -l` コマンドを使用します。

`add_delete_switch` は、`add` または `delete` オプションを表します。このオプションは、指定のパッケージを追加または削除するかを指定します。`add_delete_switch` を指定しないと、デフォルトによって `add` が使用されます。

---

注 - 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。

---

アップグレードに `package` を使用すると、次の処理が行われます。

- すでにシステム上にあるすべてのパッケージが自動的にアップグレードされます。
- `package_name add` を指定したが、`package_name` がシステムにインストールされていない場合は、そのパッケージがインストールされます。
- `package_name delete` を指定したが、`package_name` がシステムにインストールされていた場合、アップグレードが開始される前にそのパッケージは削除されます。
- `package_name delete` を指定したが、`package_name` がシステムにインストールされていない場合、インストールするように指定したクラスタの一部にそのパッケージが含まれていると、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。

`type` - 次の中から 1 つを指定します。

- `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。

---

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。

## root\_device プロファイルキーワード

`root_device slice`

---

注 - **root\_device** 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。

---

**root\_device** は、システムのルートディスクを指定します。詳細は、369 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。

システムをアップグレードする場合、**root\_device** は指定されるルート (/) ファイルシステムおよびその `/etc/vfstab` ファイルでマウントされるファイルシステムがアップグレードされることを示します。システム上で複数のルートファイルシステムがアップグレードできる場合は、**root\_device** を指定する必要があります。**slice** は、`cwtxdysz` または `cxdysz` 形式で指定してください。



例:

```
root_device c0t0d0s2
```

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。

## 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 31-6 に、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ています。

---

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 31-6 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 root_device キーワードが指定されている場合、JumpStart プログラムは rootdisk をルートデバイスに設定します。
2	プロファイル内で、rootdisk が設定されていなくて、boot_device キーワードが指定されている場合、JumpStart プログラムは rootdisk をブートデバイスに設定します。
3	プロファイル内で rootdisk が設定されていなくて、filesys cwtxdysz size / エントリが指定されている場合、JumpStart プログラムは rootdisk をエントリで指定されたディスクに設定します。
4	プロファイル内で、rootdisk が設定されていなくて、rootdisk.sn 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で、(カーネルのプローブ順で) 指定し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 rootdisk を設定します。

表 31-6 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
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、(カーネルのブロード順で) 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、インストールする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。

プロファイルで `usedisk` を指定すると、JumpStart プログラムは `usedisk` キーワードで指定したディスクだけを使用します。

---

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

---

## カスタム JumpStart の環境変数

begin および finish スクリプトには、環境変数を使用できます。たとえば、begin 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、システムにすで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ついての情報は、installed キーワードの使用後 SI\_INSTALLED によってしか入手できません。

表 31-7 は、これらの変数とその値を説明しています。

表 31-7 インストール環境変数

環境変数	値
CHECK_INPUT	JumpStart ディレクトリ内の rules ファイルへのパス。ファイルは /tmp/install_config/rules にマウントされます。
HOME	インストール時のスーパーユーザーのホームディレクトリ (/tmp/root)
PATH	インストール時のシェル検索パス (/sbin:/usr/sbin/install.d:/usr:/usr/sbin:/usr/bin )
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します。
SI_BEGIN	begin 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_CONFIG_DIR	JumpStart ディレクトリへのパス。ディレクトリは /tmp/install_config にマウントされます。  注 - SI_CONFIG_DIR 変数は、WAN ブートを利用したカスタム JumpStart インストールでは使用できません。
SI_CONFIG_FILE	JumpStart ディレクトリ内の rules ファイルへのパス。ファイルは /tmp/install_config/rules にマウントされます。
SI_CONFIG_PROG	rules ファイル

表 31-7 インストール環境変数 (続き)

環境変数	値
SI_CUSTOM_PROBES_FILE	custom_probes.ok ファイル。ここでは、独自のルールキーワードやプローブキーワードを定義できます。custom_probes.ok ファイルを作成して使用すると、デフォルトのルールキーワードのリストを拡張できます (335 ページの「ルールキーワードと値の説明」を参照)。また、このファイルを使用すると、デフォルトのプローブキーワードのリストも拡張できます (373 ページの「プローブキーワードと値」を参照)。
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します。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 369 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)。
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します。
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します。
SI_FINISH	finish 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_HOSTID	インストールクライアントの Ethernet 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します。
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します。SI_INST_OS と SI_INST_VER は、SI_INSTALLED の値を決定するために使用します。
SI_INST_OS	オペレーティングシステムの名前。SI_INST_OS と SI_INST_VER は、SI_INSTALLED の値を決定するために使用します。
SI_INST_VER	オペレーティングシステムのバージョン。SI_INST_OS と SI_INST_VER は、SI_INSTALLED の値を決定するために使用します。
SI_KARCH	インストールクライアントのカーネルアーキテクチャー。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します。
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します。
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します。
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します。

表 31-7 インストール環境変数 (続き)

環境変数	値
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 <code>disksize</code> キーワードを使用して照合する場合に設定します。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 369 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)。
SI_OSNAME	Solaris 9 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。この変数は、たとえば、Solaris DVD または Solaris SOFTWARE 1 of 2 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もとづいてシステムに 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に指定できます。
SI_PROFILE	マウントされた JumpStart ディレクトリ内のプロファイルへのパス。このパスは <code>/tmp/install_config/profile_name</code> です。派生プロファイルを作成する場合は、SI_PROFILE に <code>/tmp/install.input</code> ファイルが設定されます。
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 <code>disksize</code> または <code>installed</code> キーワードを rootdisk に設定した場合に設定します。
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 <code>disksize</code> または <code>installed</code> キーワードを rootdisk に設定した場合に設定します。
SI_SYS_STATE	<code>/a/etc/.sysIDtool.state</code> ファイル。finish スクリプトにこのファイルを指定すれば、システムのリブート時に <code>sysidroot</code> プログラムからスーパーユーザーのパスワードを入力するように求めるプロンプトを表示しないようにすることができます。
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 <code>totaldisk</code> キーワードを使用して照合する場合に設定します。
SHELL	インストール時のデフォルトシェル ( <code>/sbin/sh</code> )
TERM	インストールクライアントの端末タイプ
TZ	NIS または NIS+ ネームサービスに指定されているデフォルトの時間帯

## プローブキーワードと値

表 31-8 は、各ルールキーワードと対応するプローブキーワードを説明しています。

---

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 31-8 プロープキーワードの説明

ルールキーワード	対応するプロープキーワード	プロープキーワードの説明
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) をカーネルプロープ順 (c0t3d0s0、c0t3d0s1、c0t4d0s0) で返します。SI_DISKLIST、SI_DISKIZES、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、あるいは空白を返して、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 オペレーティング環境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 になる。
karch	karch	システムのプラットフォームグループ (i86pc、sun4m、sun4 など) を返し、SI_KARCH を設定します。プラットフォーム名のリストについては、『Solaris 9 ハードウェアマニュアル』を参照してください。
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については、『Solaris 9 ハードウェアマニュアル』を参照してください。
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 オペレーティング環境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 になる。
	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 31-8 プロープキーワードの説明 (続き)

ルールキーワード	対応するプロープ キーワード	プロープキーワードの説明
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 を返します。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。





## 第 32 章

# Solaris Live Upgrade (トピック)

---

以下の章では、Solaris Live Upgrade を使って非アクティブ環境を作成し、ブート環境をアップグレードする方法について説明します。このブート環境は、アクティブブート環境に切り替えることができます。

第 33 章	Solaris Live Upgrade の概要情報を提供します。
第 34 章	ブート環境の作成に必要な情報を提供します。
第 35 章	Solaris Live Upgrade のインストール方法、メニューの使用方法、ブート環境の作成方法について説明します。
第 36 章	オペレーティングシステムのアップグレード方法、ブート環境にフラッシュアーカイブをインストールする方法、ブート環境をアクティブにする方法、アップグレード時に障害が発生した場合のスムーズな修復方法について説明します。
第 37 章	ブート環境の管理と状態使用の表示方法について説明します。
第 38 章	Solaris Live Upgrade の実例を提示します。
第 39 章	Solaris Live Upgrade コマンドの一覧を提示します。



## 第 33 章

# Solaris Live Upgrade (概要)

---

この章では、Solaris Live Upgrade の実行手順について説明します。

---

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。

---

---

## Solaris Live Upgrade の紹介

Solaris Live Upgrade を使用すると、稼動中のシステムを停止することなくシステムをアップグレードできます。現在のブート環境を動作させたまま、ブート環境のコピーを作成し、それをアップグレードできます。アップグレードする代わりに、フラッシュアーカイブをブート環境にインストールすることもできます。環境をアップグレードしても、アーカイブをインストールしても、元のシステム構成は影響を受けずに支障なく機能します。準備ができたところでシステムをリブートすると、新しいブート環境がアクティブになります。障害が発生した場合は、リブートするだけで元のブート環境に戻ることができます。このように切り替えが可能なので、テストや評価処理のためにサービスを停止する必要がなくなります。

Solaris Live Upgrade を使用すると、現在動作しているシステムに影響を与えずに、ブート環境のコピーを作成して、次のような作業を行うことができます。

- システムのアップグレード
- 現在のブート環境のディスク構成を、新しいブート環境のディスク構成（ファイルシステムのタイプ、サイズ、および配置）に変更
- 異なるイメージを持つ複数のブート環境の保守。たとえば、現在のパッチを持つブート環境を作成すると同時に、Update リリースを持つ別のブート環境を作成できます。

Solaris Live Upgrade を使用するには、システム管理についての基礎的な事柄を理解しておく必要があります。ファイルシステムの管理、マウント、ブート、スワップの管理など、システム管理作業に関する基本的な情報については、『Solaris のシステム管理 (基本編)』を参照してください。

---

## Solaris Live Upgrade の実行手順

次に、現在のブート環境のコピーを作成してこのコピーをアップグレードし、アクティブなブート環境になるように切り替える作業の概要を示します。

### ブート環境の作成

ブート環境を作成すると、クリティカルなファイルシステムをアクティブなブート環境から新しいブート環境にコピーできます。必要であれば、ディスクを編成し直して、ファイルシステムをカスタマイズし、クリティカルなファイルシステムを新しいブート環境にコピーします。

### ファイルシステムのタイプ

Solaris Live Upgrade では、クリティカルファイルシステムと共有可能ファイルシステムの2種類のファイルシステムを区別します。クリティカルなファイルシステムとは、Solaris オペレーティング環境に必須であり、アクティブなブート環境と非アクティブなブート環境の `vfstab` において別々のマウントポイントを持つファイルシステムのことです。たとえば、ルート (`/`)、`/usr`、`/var`、または `/opt` がクリティカルな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常に、ソースブート環境から非アクティブなブート環境にコピーされます。クリティカルなファイルシステムのことを「共有不可能」と呼ぶこともあります。共有可能なファイルシステムとは、`/export` のように、アクティブなブート環境と非アクティブなブート環境の両方の `vfstab` において同じマウントポイントを持つユーザー定義ファイルのことです。したがって、アクティブなブート環境内の共有ファイルを更新すると、非アクティブなブート環境のデータも更新されます。新しいブート環境を作成するとき、共有可能なファイルシステムはデフォルトで共有されます。しかし、コピー先のスライスを指定した場合、そのファイルシステムは (共有されずに) コピーされます。共有可能なファイルシステムについての詳細は、407 ページの「共有可能なファイルシステムのスライスを選択するための指針」を参照してください。

スワップは、特殊な共有可能ファイルシステムです。他の共有可能なファイルシステムと同様に、すべてのスワップスライスはデフォルトで共有されます。しかし、スワップ用のディレクトリを指定した場合、スワップスライスは (共有されずに) コピーされます。スワップを再構成する手順については、次の節を参照してください。

- 「ブート環境を作成する (キャラクターインターフェース)」の手順 9

- 430 ページの「ブート環境を作成しスワップを再構成する (コマンド行インタフェース)」

Solaris Live Upgrade では、ファイルシステム上に RAID-1 ボリューム (ミラー) を持つブート環境を作成できます。この概要については、384 ページの「ミラー化されたファイルシステムを持つブート環境の作成」を参照してください。

## ファイルシステムのコピー

新しいブート環境を作成するには、まず、クリティカルなファイルシステムをコピーできる未使用のスライスが存在することを確認します。スライスが使用できないかあるいは最小限の要件を満たしていない場合は、新しいスライスをフォーマットする必要があります。

スライスを定義した後、ファイルシステムをディレクトリにコピーする前に、新しいブート環境上のファイルシステムを再構成できます。ファイルシステムを分割およびマージすることによって `vfstab` を簡単に編集でき、ファイルシステムを再構成することができます。ファイルシステムは、同じマウントポイントを指定して親ディレクトリにマージすることも、異なるマウントポイントを指定して親ディレクトリから分割することも可能です。

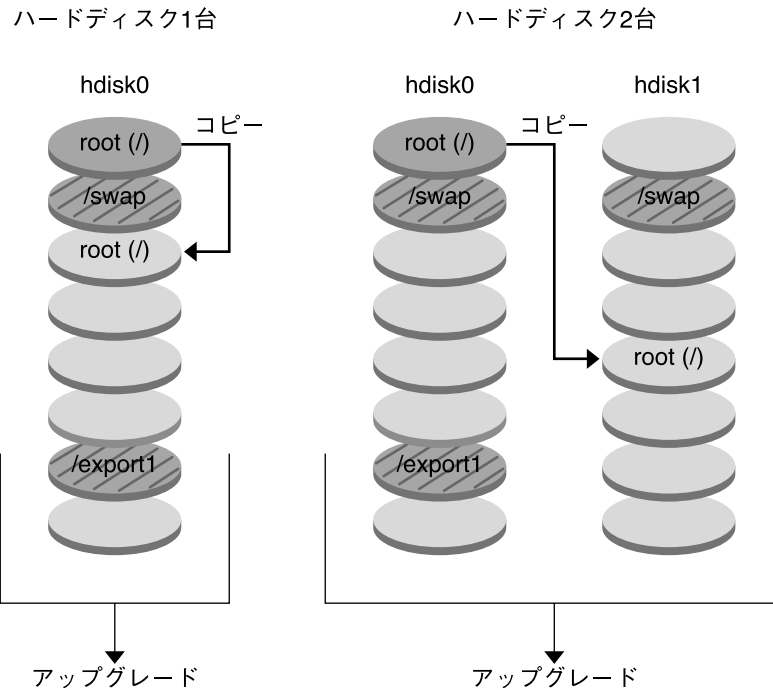
非アクティブブート環境でファイルシステムを構成した後、自動コピーを開始します。クリティカルなファイルシステムは、指定された宛先ディレクトリにコピーされます。共有可能なファイルシステムは (それらの一部をコピーするように指定しない限り)、コピーされずに共有されます。ファイルシステムをアクティブなブート環境から非アクティブなブート環境にコピーする時、ファイルは新しいディレクトリにコピーされるので、アクティブなブート環境は変更されません。

- ファイルシステムを分割およびマージする手順については、次の節を参照してください。
  - 「ブート環境を作成する (キャラクタインタフェース)」 の手順 7 または手順 8
  - 418 ページの「新しいブート環境の作成」
  - 428 ページの「ブート環境を作成しファイルシステムを分割する (コマンド行インタフェース)」
- ミラー化されたファイルシステムを持つブート環境の作成については、384 ページの「ミラー化されたファイルシステムを持つブート環境の作成」を参照してください。

次の図に、さまざまな方法で新しいブート環境を作成する例を示します。

図 33-1 に、クリティカルなファイルシステムのルート (/) を同じまたは別のディスク上のスライスにコピーして、新しいブート環境を作成する方法を示します。アクティブなブート環境は、既存のスライス上にルート (/) を持っています。新しいブート環境は、新しいスライス上にルート (/) の複製を持っています。ファイルシステム `/swap` および `/export/home` はアクティブなブート環境と非アクティブなブート環境で共有されます。

ブート環境の作成 - ルート(/) からルート(/) へのコピー



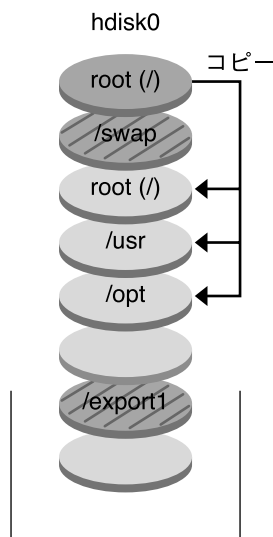
- 現在のリリース X  
クリティカルなファイルシステムのルート (/)
- 非アクティブなリリース X  
クリティカルなファイルシステムのルート (/)
- 共有ファイルシステム

図 33-1 非アクティブなブート環境の作成 - ルート (/) からルート (/) へのコピー

図 33-2 に、クリティカルなファイルシステムを同じまたは別のディスク上の複数のスライスに分割およびコピーして、新しいブート環境を作成する方法を示します。アクティブなブート環境は、既存のスライス上にルート (/) を持っています。このスライスでは、ルート (/) 内に、/usr、/var、および /opt ディレクトリがあります。新しいブート環境では、ルート (/) は分割され、/usr と /opt は別のスライスに配置されています。ファイルシステム /swap と /export/home は両方のブート環境で共有されます。

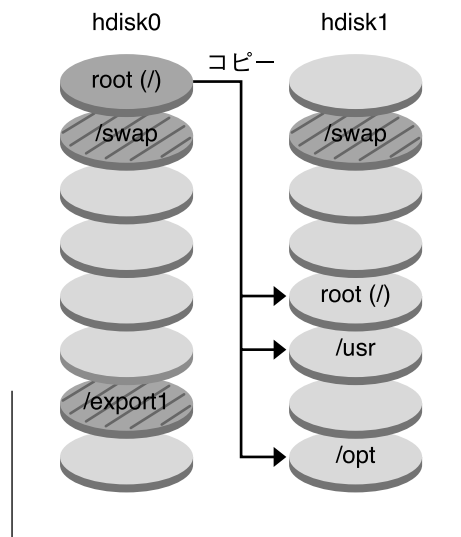
## ブート環境の作成 - ファイルシステムの分割

ハードディスク1台



アップグレード

ハードディスク2台



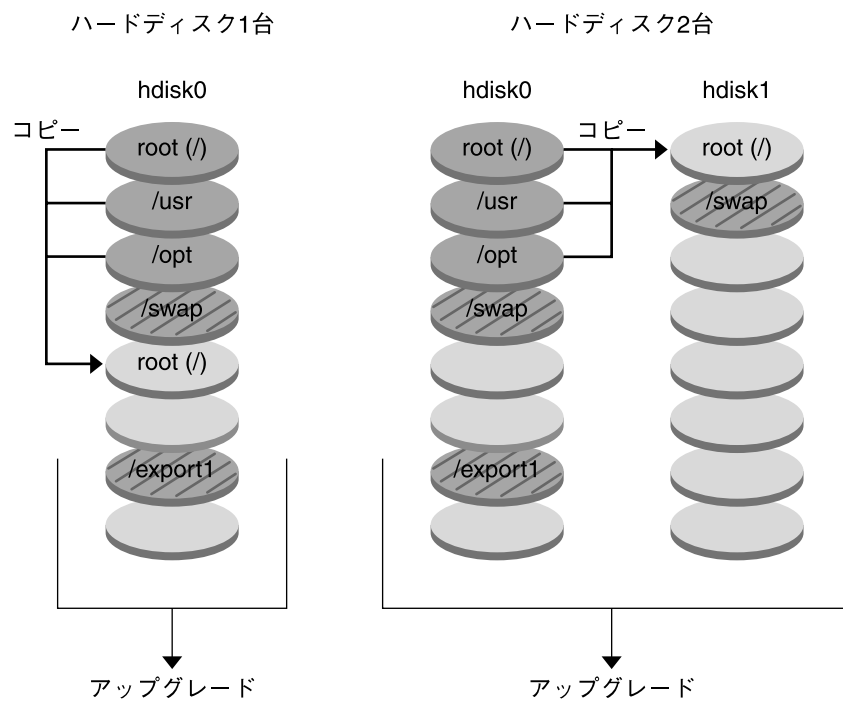
アップグレード

- 現在のリリース X  
クリティカルなファイルシステムのルート(/)
- 非アクティブなリリース X  
クリティカルなファイルシステムのルート (/) /usr /opt
- ▨ 共有ファイルシステム

図 33-2 非アクティブなブート環境の作成 - ファイルシステムの分割

図 33-3 に、クリティカルなファイルシステムをマージし、同じまたは別のディスク上のスライスにコピーして、新しいブート環境を作成する方法を示します。アクティブなブート環境では、ルート (/)、/usr、/var、および /opt がそれぞれ別のスライス上にあります。新しいブート環境では、/usr と /opt はルート (/) と同一のスライス上にマージされます。ファイルシステム /swap と /export/home は両方のブート環境で共有されます。

## ブート環境の作成 - ファイルシステムのマージ



- 現在のリリース X  
クリティカルなファイルシステムのルート (/) /usr /opt
- 非アクティブなリリース Y  
クリティカルなファイルシステムのルート (/)
- ▣ 共有ファイルシステム

図 33-3 非アクティブなブート環境の作成 - ファイルシステムのマージ

## ミラー化されたファイルシステムを持つブート環境の作成

Solaris Live Upgrade は Solaris ボリュームマネージャテクノロジーを使って、ミラー化されたファイルシステムを持つブート環境を作成できます。Solaris ボリュームマネージャでは、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では、連結、ストライプ、その他の複雑な構成が可能です。Solaris Live Upgrade では、これらの作業の一部を実行できます。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。



ボリュームを使用すると、複数のディスクにまたがるディスクスライスをグループ化して、オペレーティング環境で単一のディスクとして扱われるようになります。Solaris Live Upgrade で作成できるのは、RAID-1 ボリューム (ミラー) 内に単一スライスの連結を持つルート (/) ファイルシステムのブート環境だけです。これは、ブート用のスライスを 1 つだけ選択するようにブート PROM が制限されているためです。

ブート環境を作成するとき、Solaris Live Upgrade を使って次の作業を行うことができます。

- 単一スライスの連結 (サブミラー) を RAID-1 ボリューム (ミラー) から切り離す。必要な場合は、内容を保持して新しいブート環境の内容にすることができます。内容はコピーされないため、新しいブート環境を短時間で作成できます。ミラーから切り離されたサブミラーは、元のミラーの一部ではなくなります。サブミラーに対する読み取りや書き込みがミラーを介して実行されることはなくなります。
- ミラーを含んだブート環境を作成する
- 新しく作成したミラーに単一スライスの連結を 3 つまで接続する

Solaris Live Upgrade のミラー化機能を使用するには、状態データベースを少なくとも 1 つ作成し、状態データベースの複製を少なくとも 3 つ作成する必要があります。状態データベースでは、Solaris ボリュームマネージャ構成の状態に関する情報がディスクに保存されます。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。状態データベースのコピーを作成することで、単一点障害によるデータ損失を防ぐことができます。状態データベースの作成手順については、『Solaris ボリュームマネージャの管理』の「状態データベース (概要)」を参照してください。

lucreate コマンドの -m オプションを使って、新しいブート環境に対してミラーの作成、サブミラーの切り離し、およびサブミラーの接続を行うことができます。

- 手順については、438 ページの「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。
- インストール時のミラー化の概要は、第 10 章を参照してください。
- Solaris Live Upgrade では使用できない Solaris ボリュームマネージャの複雑な構成については、『Solaris ボリュームマネージャの管理』の「記憶装置管理の概念」を参照してください。

表 33-1 Solaris Live Upgrade で使用される Solaris ボリュームマネージャの要素

用語	説明
状態データベース	状態データベースでは、Solaris ボリュームマネージャ構成の状態に関する情報がディスクに保存されます。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。

表 33-1 Solaris Live Upgrade で使用される Solaris ボリュームマネージャの要素 (続き)

用語	説明
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。

表 33-2 に、Solaris Live Upgrade で管理できるコンポーネントを示します。

表 33-2 ボリュームクラス

用語	説明
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 ボリュームはミラーと呼ばれることもあります。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せん。
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。
ミラー	RAID-1 ボリューム。「RAID-1 ボリューム」を参照してください。
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。
サブミラー	「RAID-0 ボリューム」を参照。

図 33-4 は、2 つの物理ディスク上に作成された RAID-1 ボリューム (ミラー) を持つ新しいブート環境を示しています。この新しいブート環境とミラーは、次のコマンドで作成されたものです。

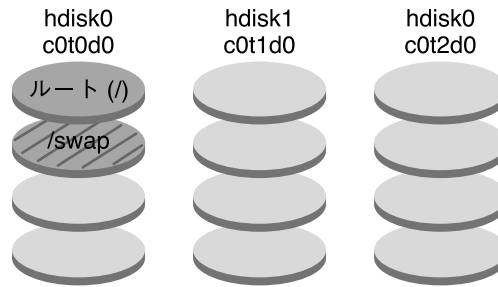
```
# lucreate -n second_disk -m /:/dev/md/dsk/d30:mirror,ufs \
-m /:c0t1d0s0,d31:attach -m /:c0t2d0s0,d32:attach \
-m -:c0t1d0s1:swap -m -:c0t2d0s1:swap
```

このコマンドは、次のような処理を実行します。

- 新しいブート環境 `second_disk` を作成する。
- ミラー `d30` を作成し、UFS ファイルシステムを構成する。
- 各物理ディスクのスライス 0 に単一デバイスの連結を作成する。これらの連結に `d31` および `d32` という名前を付ける。
- これら 2 つの連結をミラー `d30` に追加する。
- ルート (`/`) ファイルシステムをミラーにコピーする。
- 各物理ディスクのスライス 1 に、スワップ用のファイルシステムを構成する。

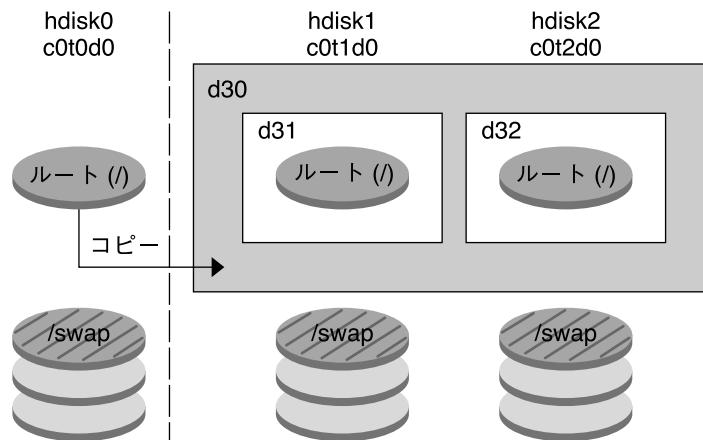
### ミラーを使った新しいブート環境の作成

3つの物理ディスクからなる元のシステム



```
コマンド: lucreate -n second_disk -m /:/dev/md/dsk/d30:mirror,ufs \
-m /:/c0t1d0s0,d31:attach -m /:/c0t2d0s0,d32:attach \
-m -:/c0t1d0s1:swap -m -:/c0t2d0s1:swap
```

新しいブート環境 second\_disk



- 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 33-4 ブート環境の作成とミラーの作成

図 33-5 は、RAID-1 ボリューム (ミラー) を持つ新しいブート環境を示しています。この新しいブート環境とミラーは、次のコマンドで作成されたものです。

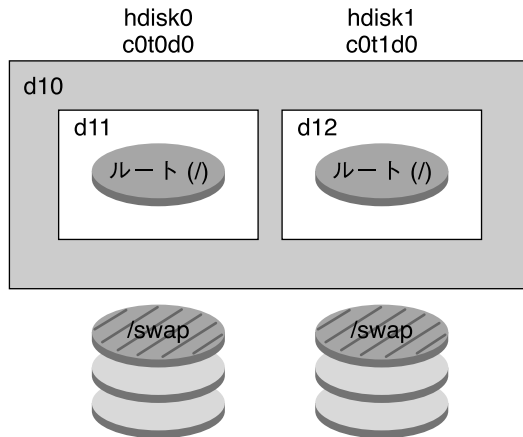
```
# lucreate -n second_disk -m /:/dev/md/dsk/d20:ufs,mirror \
-m /:/dev/dsk/c0t1d0s0:detach,attach,preserve
```

このコマンドは、次のような処理を実行します。

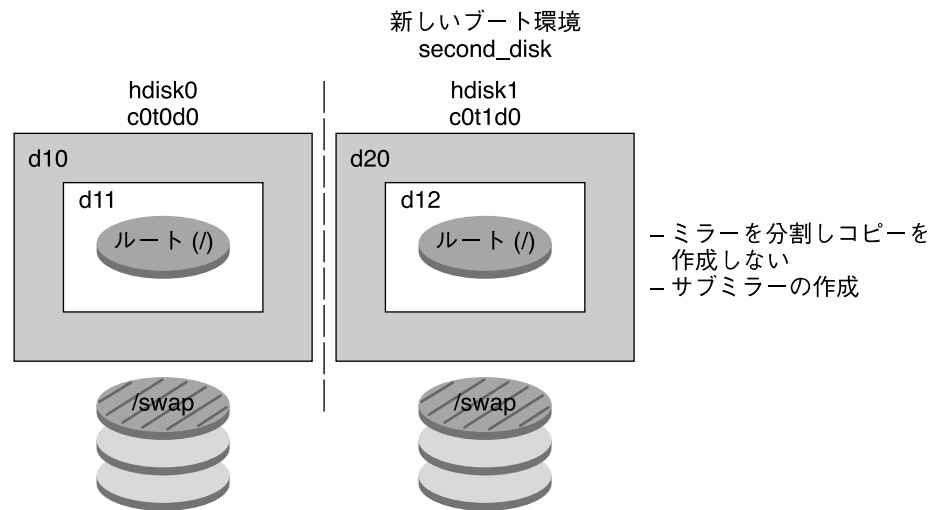
- 新しいブート環境 `second_disk` を作成する。
- ミラー `d10` を分割し、連結 `d12` を切り離す。
- 連結 `d12` の内容を保持する。ファイルシステムのコピーは行われない。
- 新しいミラー `d20` を作成する。これで、`d10` および `d20` という 2 つの 1 面ミラーが作成される。
- 連結 `d12` をミラー `d20` に接続する。

新しいブート環境の作成と既存のサブミラーの使用

2つの物理ディスクからなる元のシステム



コマンド: `lucreate -n second_disk -m /:/dev/md/dsk/d20:ufs,mirror \`  
`-m /:/dev/dsk/c0t1d0s0:detach,attach,preserve`



- d10 – RAID-1 ボリューム (ミラー)
- d11 – 単一スライスの連結 (サブミラー)
- d12 – 単一スライスの連結 (サブミラー)
- d20 – 新しい RAID-1 ボリューム (ミラー)

図 33-5 ブート環境の作成と既存のサブミラーの使用

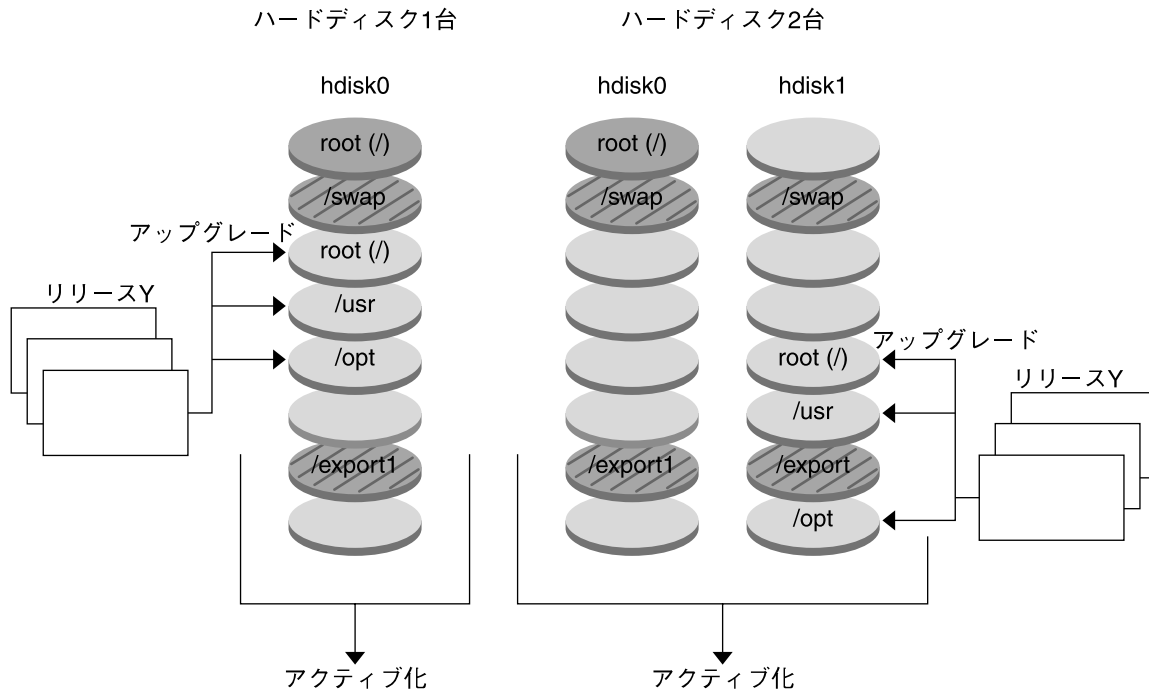
## ブート環境のアップグレード

ブート環境の作成が完了したら、そのブート環境をアップグレードできます。アップグレード作業の過程で、ブート環境の任意のファイルシステムに RAID-1 ボリューム (ミラー) を持たせることができます。アップグレードを行なっても、アクティブなブート環境内のファイルには影響ありません。準備ができたところでこの新しいブート環境をアクティブ化し、このブート環境を現行のブート環境とします。

- ブート環境のアップグレード手順については、第 36 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を持つブート環境のアップグレード例については、494 ページの「RAID 1 ボリューム (ミラー) の一方を切り離してアップグレードする例 (コマンド行インタフェース)」を参照してください。

図 33-6 に、非アクティブなブート環境のアップグレードの例を示します。

## ブート環境のアップグレード



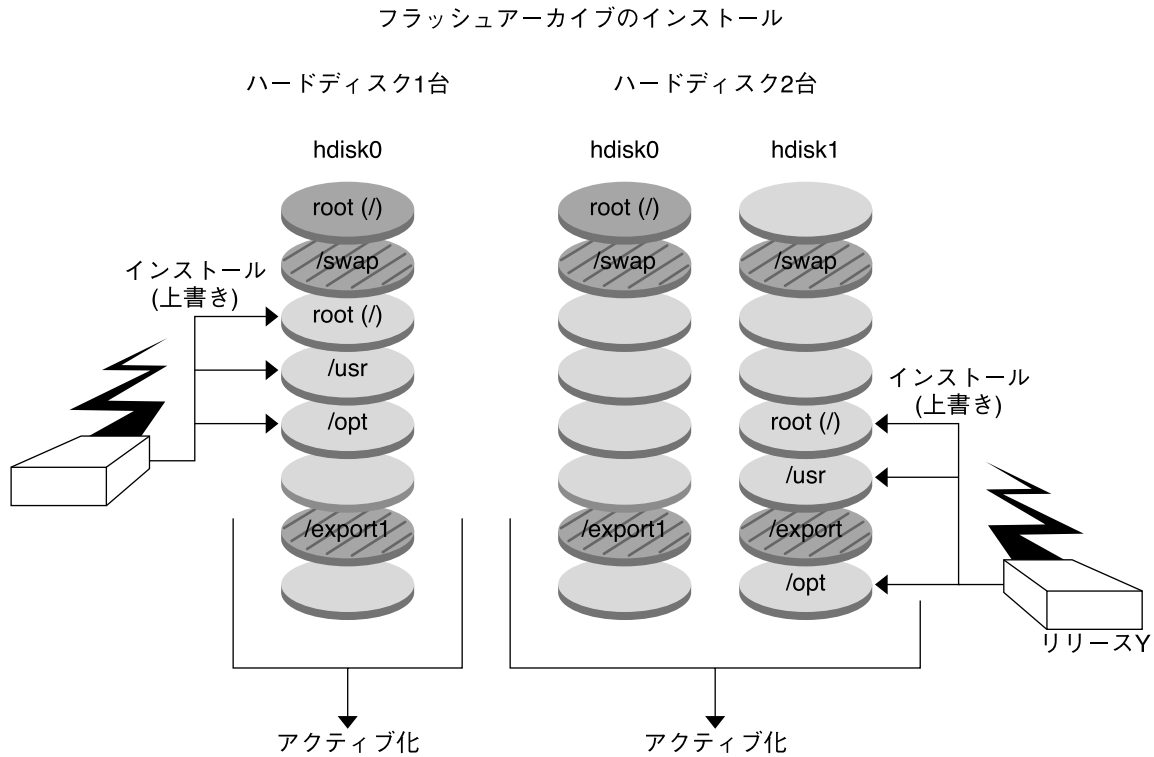
- 現在のリリース X  
クリティカルなファイルシステムのルート (/)
- 非アクティブなリリース Y  
クリティカルなファイルシステムのルート (/) /usr /opt
- ▨ 共有ファイルシステム

図 33-6 非アクティブなブート環境のアップグレード

アップグレードする代わりに、フラッシュアーカイブをブート環境にインストールすることもできます。フラッシュインストール機能を利用して、あるシステム上に Solaris オペレーティング環境の単一の参照用インストールを作成します。このシステムはマスターシステムと呼ばれます。続いて、クローンシステムと呼ばれる多数のシステム上にこのインストールを複製できます。この場合、非アクティブなブート環境はクローンシステムです。フラッシュアーカイブをシステムにインストールするとき、初期インストールの場合と同じように、アーカイブは既存のブート環境にあるすべてのファイルを置き換えます。

フラッシュアーカイブのインストール手順については、459 ページの「ブート環境へのフラッシュアーカイブのインストール」を参照してください。

図 33-7 に、非アクティブなブート環境におけるフラッシュアーカイブのインストールを示します。



- 現在のリリース X  
クリティカルなファイルシステムのルート (/)
- 非アクティブなリリース Y  
クリティカルなファイルシステムのルート (/) /usr /opt
- ▨ 共有ファイルシステム

図 33-7 フラッシュアーカイブのインストール

## ブート環境のアクティブ化

新しいブート環境に切り換えてアクティブにする準備ができれば、新しいブート環境をアクティブにしてリブートするだけです。新たに作成したブート環境を初めて起動するとき、ブート環境間でファイルの同期がとられます。ここでいう「同期」とは、前にアクティブであったブート環境のあるシステムファイルやディレクトリを、ブー

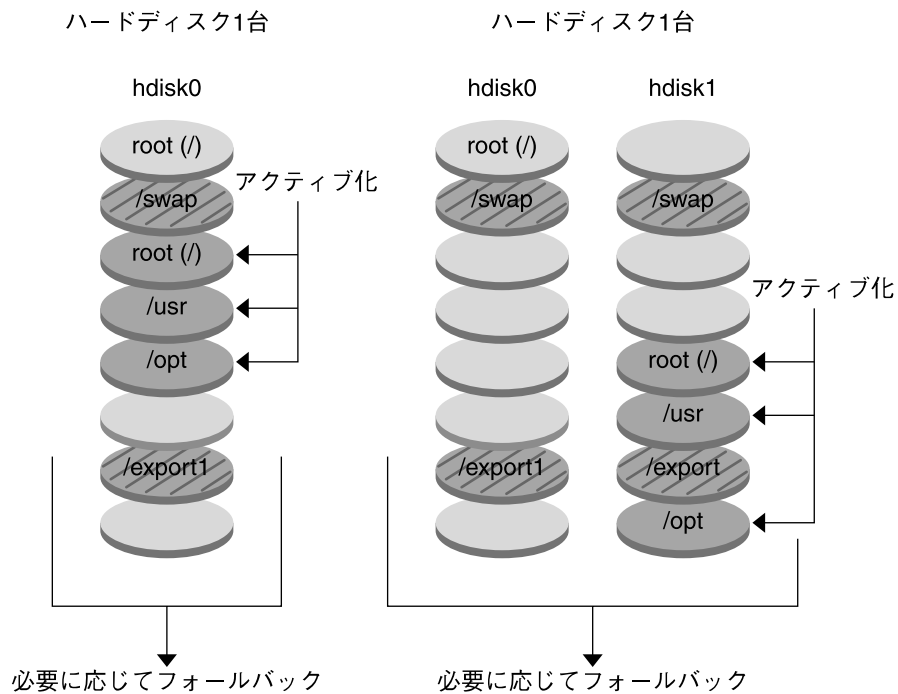


トされているブート環境にコピーすることです。システムをリブートすると、非アクティブなブート環境にインストールした構成がアクティブになります。この時点で、元のブート環境は非アクティブブート環境となります。

- ブート環境をアクティブにする手順については、464 ページの「ブート環境のアクティブ化」を参照してください。
- アクティブなブート環境と非アクティブなブート環境の同期については、409 ページの「ブート環境間でのファイルの同期」を参照してください。

図 33-8 に、リブート後に非アクティブなブート環境からアクティブなブート環境に切り替わる様子を示します。

### ブート環境のアクティブ化



- 現在のリリース Y  
クリティカルなファイルシステムのルート (/) /usr /opt
- 非アクティブなリリース X  
クリティカルなファイルシステムのルート (/)
- ▨ 共有ファイルシステム

図 33-8 非アクティブなブート環境のアクティブ化

## 元のブート環境へのフォールバック

問題が発生する場合は、アクティブ化とリブートを行なって元のブート環境にすぐにフォールバックできます。フォールバックを実行する必要があるのは、次のような場合です。

- 新しいブート環境がブートできない場合
- ブートできるが部分的にしか稼動しない場合
- ブート結果に満足できない場合

元のブート環境をバックアップして復元するよりも、フォールバックの方がはるかに時間がかかりません。ブートに失敗した新しいブート環境は保存されるので、障害を解析できます。フォールバックを実行できるのは、`luactivate` を使用して新しいブート環境をアクティブにしたブート環境だけです。

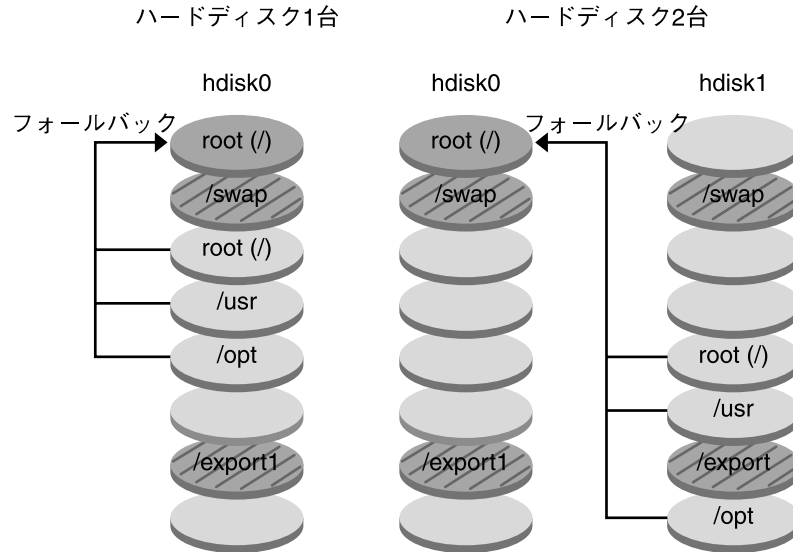
以前のブート環境にフォールバックするには、次の手順に従います。

- 新しいブート環境は正常にブートしたが、結果に満足できない場合、`luactivate` コマンドに以前のブート環境の名前を指定して実行し、リブートします。
- 新しいブート環境がブートしない場合、フォールバックしたいブート環境をシングルユーザーモードでブートし、`luactivate` コマンドを実行し、リブートします。
- シングルユーザーモードでブートできない場合は、次のいずれかの操作を行います。
  - DVD/CD メディアまたはネットインストールイメージからブートする
  - フォールバックしたいブート環境上のルート (/) ファイルシステムをマウントする
  - `luactivate` コマンドを実行し、リブートする

フォールバックの手順については、468 ページの「問題の解決: 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)」を参照してください。

図 33-9 に、リブートしてフォールバックしたときにブート環境が切り替わる様子を示します。

## 元のブート環境へのフォールバック



- 現在のリリース X  
クリティカルなファイルシステムのルート (/)
- 非アクティブなリリース Y  
クリティカルなファイルシステムのルート (/) /usr /opt
- ▨ 共有ファイルシステム

図 33-9 元のブート環境へのフォールバック

## ブート環境の保守

ブート環境のステータス確認、名前変更、削除など、さまざまな保守作業も行うことができます。保守作業の手順については、第 37 章を参照してください。



## 第 34 章

---

# Solaris Live Upgrade (計画)

---

この章では、Solaris Live Upgrade のインストールと使用を開始する前に考慮すべき指針と要件を説明します。57 ページの「アップグレード用のチェックリスト」でアップグレード全般に関する情報も確認してください。この章の内容は次のとおりです。

- 397 ページの「Solaris Live Upgrade の要件」
- 400 ページの「Solaris Live Upgrade によるパッケージやパッチの管理」
- 402 ページの「lucreate コマンドを用いたファイルシステムの作成のための指針」
- 403 ページの「ファイルシステムのスライスを選択するための指針」
- 408 ページの「新しいブート環境の内容のカスタマイズ」
- 409 ページの「ブート環境間でのファイルの同期」
- 411 ページの「リモートシステムからの Solaris Live Upgrade の使用」

---

## Solaris Live Upgrade の要件

### Solaris Live Upgrade のシステム要件

Solaris Live Upgrade は Solaris 9 ソフトウェアに含まれています。Solaris Live Upgrade を使用してアップグレードする場合は、現在のオペレーティング環境に Solaris Live Upgrade パッケージをインストールする必要があります。ブート環境は、マシンにインストールされている Solaris Live Upgrade パッケージと同じリリースの Solaris オペレーティング環境にアップグレードできます。たとえば、現在 Solaris 8 オペレーティング環境を実行していて Solaris 9 Live Upgrade パッケージをインストールした場合、ブート環境を Solaris 9 マーケティングまたは Update リリースにアップグレードできます。

表 34-1 では、Solaris Live Upgrade でサポートされるリリースの一覧を示しています。

表 34-1 サポートされる Solaris リリース

プラットフォーム	アップグレード前のリリース	アップグレード後のリリース
SPARC システム	Solaris 2.6、Solaris 7、または Solaris 8 オペレーティング環境	Solaris 8 オペレーティング環境
SPARC システム	Solaris 2.6、Solaris 7、または Solaris 8 オペレーティング環境	Solaris 9 オペレーティング環境
x86 システム	Solaris 7 オペレーティング環境	Solaris 8 オペレーティング環境
x86 システム	Solaris 7、または Solaris 8 オペレーティング環境	Solaris 9 オペレーティング環境

---

注 - Solaris 7 オペレーティング環境にはアップグレードできません。

---

## Solaris Live Upgrade のインストール

Solaris Live Upgrade パッケージのインストールには、以下を使用します。

- pkgadd コマンド。Solaris Live Upgrade パッケージは SUNW1ur と SUNW1uu です。この順序でインストールする必要があります。
- Solaris DVD、Solaris SOFTWARE 2 of 2 CD、またはネットインストールイメージ上にあるインストーラ。

---

注 - Solaris 2.6、Solaris 7、または Solaris 8 を実行している場合、Solaris Live Upgrade インストーラを実行できないことがあります。これらのリリースには、Java™ 2 Runtime Environment の実行に必要なパッチのセットが含まれていません。Solaris Live Upgrade インストーラを実行してパッケージをインストールするには、Java 2 Runtime Environment の推奨パッチクラスタが必要です。Solaris Live Upgrade パッケージをインストールするには、pkgadd コマンドを使用します。または、<http://sunsolve.sun.com> から入手可能な Java 2 Runtime Environment 推奨パッチクラスタをインストールします。

---

Solaris Live Upgrade ソフトウェアのインストール方法については、415 ページの「Solaris Live Upgrade のインストール」を参照してください。

## Solaris Live Upgrade のディスク容量の要件

アップグレードの一般的なディスク容量の要件に従います。第 5 章を参照してください。

ブート環境の作成に必要なファイルシステムのサイズを見積もるには、新しいブート環境の作成を開始してください。サイズが計算されたところで、処理を中断できません。

新しいブート環境上のディスクをブートデバイスとして使用する必要があります。システムの中には、ブートデバイスとして機能するディスクを限定するものがあります。ブート制限が適用されるかどうかを確認するには、各システムのマニュアルを参照してください。

新しいブート環境を作成する前に、ディスクの準備が必要になることもあります。ディスクが正しくフォーマットされていることを次のように確認します。

- スライスがファイルシステムをコピーできるだけの十分な大きさであることを確認します。
- ブート環境間でコピーするのではなく、共有したいディレクトリが入っているファイルシステムを確認します。ディレクトリを共有したい場合、そのディレクトリを固有のスライスに配置して新しいブート環境を作成する必要があります。こうすることにより、ディレクトリは、将来のブート環境と共有可能なファイルシステムになります。異なるファイルシステムを作成して共有する方法についての詳細は、407 ページの「共有可能なファイルシステムのスライスを選択するための指針」を参照してください。

## RAID-1 ボリューム (ミラー) を作成する場合の Solaris Live Upgrade の要件

Solaris Live Upgrade は Solaris ボリュームマネージャテクノロジーを使って、ミラー化されたファイルシステム (RAID-1 ボリューム) を持つブート環境を作成します。

Solaris Live Upgrade のミラー化機能を使用するには、状態データベースを 1 つ以上、状態データベースの複製を 3 つ以上作成する必要があります。状態データベースでは、Solaris ボリュームマネージャ構成の状態に関する情報がディスクに保存されます。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。状態データベースのコピーを作成することで、単一点障害によるデータ損失を防ぐことができます。状態データベースの作成手順については、『Solaris ボリュームマネージャの管理』の「状態データベース (概要)」を参照してください。

Solaris Live Upgrade には、Solaris ボリュームマネージャの機能の一部だけが実装されています。Solaris Live Upgrade では、ルート (/) ファイルシステムに単一スライスの連結を持つ RAID-1 ボリューム (ミラー) だけがサポートされます。ミラーは最大 3 つの連結で構成されます。ミラー化されたファイルシステムを作成するためのガイドラインについては、404 ページの「ミラー化されたファイルシステムのスライスを選択するための指針」を参照してください。

---

## Solaris Live Upgrade によるパッケージ やパッチの管理

以下の各節では、Solaris Live Upgrade に必要なパッケージを示し、推奨パッチ情報を提供します。Solaris Live Upgrade を使ってパッケージおよびパッチを追加する方法については、401 ページの「パッケージとパッチによるシステムのアップグレード」を参照してください。



---

注意 - パッケージやパッチのアップグレード、追加、削除を行うためには、そのパッケージまたはパッチが SVR4 Advanced Packaging Guideline に準拠していなければなりません。サンのパッケージはこのガイドラインに準拠していますが、サードパーティベンダーのパッケージがこれに準拠しているとは限りません。非準拠のパッケージを追加しようとすると、アップグレード時にパッケージ追加ソフトウェアの障害が発生するか、アクティブブート環境が改変されてしまう可能性があります。

Solaris Live Upgrade によるパッケージの追加や削除については、luupgrade (1M) のマニュアルページを参照してください。パッケージ化の要件については、付録 G を参照してください。

---

### 必要なパッケージ

次の表は、Solaris Live Upgrade を使用する上で必要なパッケージを示しています。この表で、現在のオペレーティング環境に必要なパッケージを確認してください。使用しているリリースに必要なパッケージがインストールされていない場合は、pkgadd コマンドを使用してパッケージを追加してください。

表 34-2 Solaris Live Upgrade に必要なパッケージ

Solaris 2.6 リリース	Solaris 7 リリース	Solaris 8 リリース
SUNWadmap	SUNWadmap	SUNWadmap
SUNWadmcc	SUNWadmcc	SUNWadmcc
SUNWjvrt	SUNWjvrt	SUNWj2rt
SUNWlibC	SUNWlibC	SUNWlibC
SUNWadmfw		SUNWbzip
SUNWmfrun		
SUNWloc		



システム上のパッケージを確認するには、次のコマンドを入力します。

```
% pkginfo [package_name]
```

## パッケージとパッチによるシステムのアップグレード

Solaris Live Upgrade を使ってパッチやパッケージをシステムに追加することができます。Solaris Live Upgrade を使ってパッチをマシンに追加すると、システムのダウンタイムはリブートの時間だけですみます。luupgrade コマンドまたはフラッシュアーカイブを使って、パッチやパッケージをブート環境に追加できます。

- パッチをブート環境に直接追加する場合は、新しいブート環境を作成してから、`-t` オプションを指定して `luupgrade` コマンドを実行します。パッケージをブート環境に追加する場合は、`luupgrade` コマンドに `-p` オプションを指定します。詳細は、`luupgrade (1M)` のマニュアルページを参照してください。
- Solaris Live Upgrade を使ってフラッシュアーカイブをインストールすることもできます。アーカイブには、新しいパッケージやパッチがすでに追加されているブート環境の完全なコピーが格納されています。この完全なブート環境または単一の参照用システムをマスターシステムと呼びます。フラッシュアーカイブを作成するプロセスは、マスターシステムの作成から始まります。マスターシステムを作成したら、インストールしたいパッチやパッケージを追加します。そして、マスターシステムのフラッシュアーカイブを作成します。次に Solaris Live Upgrade を使ってこのアーカイブを新しいブート環境にインストールします。ブート環境は、必要に応じて何回でもコピーしたり、変更したり、配布したりできます。フラッシュアーカイブの作成方法については、第 21 章を参照してください。Solaris Live Upgrade を使ってフラッシュアーカイブをインストールする方法については、459 ページの「ブート環境へのフラッシュアーカイブのインストール」を参照してください。



---

注意 – Solaris Live Upgrade でパッケージやパッチのアップグレードや追加、削除を行うためには、パッケージやパッチが SVR4 パッケージ化ガイドラインに準拠していなければなりません。サンのパッケージはこのガイドラインに準拠していますが、サードパーティベンダーのパッケージがこれに準拠しているとは限りません。非準拠のパッケージを追加しようとすると、パッケージ追加ソフトウェアの障害が発生するか、アクティブブート環境が改変されてしまう可能性があります。

Solaris Live Upgrade によるパッケージの追加や削除については、`luupgrade (1M)` のマニュアルページを参照してください。パッケージ化の要件については、付録 G を参照してください。

---

## Solaris Live Upgrade の実行に必要なパッチ

Solaris Live Upgrade を正常に実行するためには、OS のバージョンごとに限定されたパッチリビジョンセットをインストールする必要があります。これらのパッチリビジョンセットは、Live Upgrade のインストールまたは実行の前にインストールしてください。 <http://sunsolve.sun.com> で最新のパッチリストを確認してください。SunSolve の Web サイトで、72099 番のドキュメントを検索してください。

---

## lucreate コマンドを用いたファイルシステムの作成のための指針

lucreate コマンドに `-m` オプションを指定することにより、新しいブート環境に作成するファイルシステムとその数を指定できます。作成するファイルシステムの数だけ、このオプションを繰り返し指定する必要があります。たとえば、`-m` オプションを 1 回だけ使用した場合、すべてのファイルシステムが指定の場所に格納されます。元のブート環境のすべてのファイルシステムが、`-m` で指定されたファイルシステムにマージされます。`-m` オプションを 2 回使用すると、ファイルシステムが 2 つ作成されます。`-m` オプションを使ってファイルシステムを作成する場合、次の項目に留意してください。

- `-m` オプションを 1 個指定して、新しいブート環境のルート (`/`) ファイルシステムを作成する必要があります。`-m` オプションを指定しないで `lucreate` を実行すると、「Configuration」メニューが表示されます。「Configuration」メニューでは、新しいマウントポイントにファイルを変更して新しいブート環境をカスタマイズできます。
- 現在のブート環境にクリティカルファイルシステムがある場合、このファイルシステムは、`-m` オプションで指定しなくても新しく作成されたファイルシステムの上位 2 番目のファイルシステムにマージされます。
- 新しいブート環境には、`-m` オプションで指定されたファイルシステムだけが作成されます。現在のブート環境に複数のファイルシステムがあり、新しいブート環境にも同じ数のファイルシステムを作成したい場合は、ファイルシステムごとに 1 個ずつ `-m` オプションを指定します。たとえば、ルート (`/`) ファイルシステム、`/opt` ファイルシステム、`/var` ファイルシステムがある場合、これらを新しいブート環境に作成するには、それぞれに `-m` オプションを 1 個ずつ指定します。
- マウントポイントが重複しないようにしてください。たとえば、ルート (`/`) ファイルシステムを 2 つ作成することはできません。

---

## ファイルシステムのスライスを選択するための指針

ブート環境のファイルシステムを作成する場合の規則は、Solaris オペレーティング環境のファイルシステムを作成する場合と同じです。Solaris Live Upgrade では、クリティカルファイルシステムに無効な構成を作成できてしまいます。たとえば、lucreate コマンドを用いて、ルート (/) と /kernel を別々のファイルシステムに作成することも可能です。しかし、これは、無効な分割方式です。

ディスクスライスを作成するときは、スライスがオーバーラップしないように注意してください。スライスのオーバーラップがあると、新しいブート環境を作成したつもりでも、アクティブにした後ブートすることができません。こうしたオーバーラップは、ファイルシステムの破損の原因となります。

Solaris Live Upgrade を正しく機能させるには、アクティブブート環境の vfstab ファイルの内容が有効で、少なくとも ルート (/) のエントリが含まれている必要があります。

## ルート (/) ファイルシステムのスライスを選択するための指針

非アクティブブート環境を作成する場合は、ルート (/) ファイルシステムがコピーされるスライスを確認する必要があります。ルート (/) ファイルシステムのスライスを選択する場合は、次の項目に留意してください。スライスは、次の条件を満たしていなければなりません。

- システムをブートできるスライスである
- 推奨されている最小サイズ以上である
- sun4m システムの場合、ルート (/) ファイルシステムのサイズは最大で 2G バイトである
- アクティブなルート (/) ファイルシステムとは異なる物理ディスクでも同じディスクでもかまわない
- Veritas Volume Manager のボリュームを選択することはできるが、これらのボリュームはサポートされない

## ミラー化されたファイルシステムのスライスを 選択するための指針

新しく作成するブート環境には、物理ディスクスライス、Solaris ボリュームマネージャのボリューム、および Veritas Volume Manager のボリュームを自由に組み合わせで使用できます。新しいブート環境にコピーされるクリティカルファイルシステムには、次のような種類があります。

- 物理スライス
- RAID-1 ボリューム (ミラー) に含まれる単一スライスの連結。ルート (/) ファイルシステムが置かれているスライスは、RAID-1 ボリュームでもよい。
- RAID-0 ボリュームに含まれる単一スライスの連結。ルート (/) ファイルシステムが置かれているスライスは、RAID-0 ボリュームでもよい。

新しいブート環境を作成する際、`lucreate - m` コマンドは、次の 3 種類のデバイスを認識します。

- 物理スライス (`/dev/dsk/cwtxdysz`)
- Solaris ボリュームマネージャのボリューム (`/dev/md/dsk/dnum`)
- Veritas Volume Manager のボリューム (`/dev/vx/dsk/volume_name`)

---

注 – Veritas VxVm のアップグレードで問題が生じる場合は、656 ページの「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」を参照してください。

---

## ミラー化されたファイルシステムを作成するための一般的な指針

### ボリュームの指定

ミラーまたはサブミラーを指定することも、`lucreate` コマンドで空いているボリュームを自動的に選択することもできます。

- ボリュームを指定するには、デバイス名の後ろにコンマで区切ってサブミラーまたはミラーの名前を追加します。次の例では、サブミラー `d1` が指定されています。

```
-m /:/dev/dsk/c0t0d0s0,d1:attach
```

- 空いているボリュームを自動的に選択するには、サブミラーやミラーの指定を省略します。次の例では、サブミラーは指定されていないので、空いているボリュームが `lucreate` によって選択されます。

```
-m /:/dev/dsk/c0t0d0s0:attach
```

## ボリューム名の省略形

物理ディスクスライスや Solaris ボリュームマネージャ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。

命名の要件とガイドラインについては、『Solaris ボリュームマネージャの管理』の「Solaris ボリュームマネージャコンポーネントの概要」を参照してください。

## ボリュームのステータスの確認

ミラーやサブミラーが保守を必要としている場合や使用中である場合、コンポーネントを切り離すことはできません。新しいブート環境を作成して `detach` キーワードを使用する前に、`metastat` コマンドを実行してください。`metastat` コマンドは、ミラーが再同期の処理中かどうか、または使用中かどうかを確認します。詳細は、`metastat (1M)` のマニュアルページを参照してください。

## ボリュームの切り離しとミラーの再同期

`detach` キーワードを使ってサブミラーを切り離す場合、`lucreate` コマンドは、デバイスが再同期の処理中かどうかを確認します。デバイスが再同期中である場合、サブミラーを切り離すことはできず、エラーメッセージが返されます。

再同期処理とは、次のような問題の後で、あるサブミラーから別のサブミラーにデータをコピーする処理のことです。

- サブミラーの障害
- システムのクラッシュ
- オフラインであったサブミラーがオンラインに復帰
- 新しいサブミラーの追加

再同期処理の詳細は、『Solaris ボリュームマネージャの管理』の「RAID 1 ボリューム (ミラー) の再同期」を参照してください。

## Solaris ボリュームマネージャのコマンドの使用法

非アクティブなブート環境のボリュームを操作するには、Solaris ボリュームマネージャのコマンドではなく `lucreate` コマンドを使用します。Solaris ボリュームマネージャソフトウェアにはブート環境に関する考慮はありませんが、`lucreate` コマンドでは、ブート環境を誤って破棄しないように確認が行われます。たとえば、`lucreate` では、Solaris ボリュームマネージャのボリュームの上書きや削除が防止されます。

ただし、Solaris ボリュームマネージャを使って複雑な連結、ストライプ、ミラーなどを作成した場合、その操作には Solaris ボリュームマネージャを使用する必要があります。Solaris Live Upgrade では、これらのコンポーネントを認識して使用できます。Solaris ボリュームマネージャのコマンドでボリュームコンポーネントを作成、変更、または破棄する前に、`lustatus` コマンドまたは `lufslist` コマンドを実行してください。これらのコマンドを使用すると、Solaris Live Upgrade ブート環境で使用されているファイルシステムがどの Solaris ボリュームマネージャボリュームに置かれているかを確認できます。

## スワップファイルシステムのスライスを選択するための指針

### 新しいブート環境のスワップの構成

`lucreate` コマンドの `-m` オプションを使って、3 通りの方法でスワップスライスを構成できます。

- スワップスライスを指定しないと、現在のブート環境のスワップスライスが新しいブート環境用に構成されます。
- スワップスライスを 1 つ以上指定すると、それらのスワップスライスだけが新しいブート環境で使用されます。この場合指定したスワップスライスは 2 つのブート環境の間で共有されません。
- スワップスライスを共有すると同時に、新しいスライスを追加することもできます。

3 通りのスワップ構成の例を次に示します。現在のブート環境では、ルート (/) ファイルシステムが `c0t0d0s0` 上に構成されています。スワップファイルシステムは `c0t0d0s1` 上に構成されています。

- 次の例では、スワップスライスを指定していません。新しいブート環境では、ルート (/) ファイルシステムが `c0t1d0s0` 上に置かれます。`c0t0d0s1` 上のスワップが、現在のブート環境と新しいブート環境の間で共有されます。

```
# lucreate -n be2 -m /:c0t1d0s0:ufs
```

- 次の例では、スワップスライスを指定しています。新しいブート環境では、ルート (/) ファイルシステムが `c0t1d0s0` 上に置かれます。新しいスワップファイルシステムが `c0t1d0s1` 上に作成されます。現在のブート環境と新しいブート環境の間でスワップスライスは共有されません。

```
# lucreate -n be2 -m /:c0t1d0s0:ufs -m -:c0t1d0s1:swap
```

- 次の例では、スワップスライスを 1 つ追加すると同時に、別のスワップスライスを 2 つのブート環境で共有しています。新しいブート環境では、ルート (/) ファイルシステムが `c0t1d0s0` 上に置かれます。新しいスワップスライスが `c0t1d0s1` 上に作成されます。`c0t0d0s1` 上のスワップスライスが、現在のブート環境と新しいブート環境の間で共有されます。

```
# lucreate -n be2 -m /:c0t1d0s0:ufs -m -:shared:swap -m -:c0t1d0s1:swap
```

## スワップの使用中に起きるブート環境の作成失敗

現在のブート環境以外のブート環境によってスワップスライスが使用されている場合、ブート環境の作成は失敗します。ブート環境が `-s` オプションを使って作成された場合は、代替ソースブート環境がスワップスライスを使用している可能性があります。ほかのブート環境が使用している可能性はありません。

## 共有可能なファイルシステムのスライスを選択するための指針

Solaris Live Upgrade は、スライスの内容を指定した新しいブート環境のスライスにコピーします。容量とコピーにかかる時間を節約したい場合は、そのスライス上に複数のブート環境で共有できるだけの大きなファイルシステムを用意することもできます。ルート(/)、/var のようなオペレーティング環境に欠かせないクリティカルファイルシステムは必ずコピーしてください。/home などの非クリティカルファイルシステムは、複数のブート環境で共有できます。共有可能なファイルシステムは、ユーザーによって定義され、アクティブブート環境と新しいブート環境の両方の個々のスワップスライス上に存在していなければなりません。必要に応じて、複数の方法でディスクを再構成できます。

- 新しいブート環境を作成する前にディスクスライスを作成し直し、そのスライス上に共有可能なファイルシステムを配置することができます。たとえば、ルート(/)、/var、/home がすべて同じスライス上にある場合、ディスクを再構成して /home を固有のスライスに配置できます。デフォルトの設定では、新しいブート環境を作成すると、/home はアクティブブート環境と新しいブート環境で共有されます。
- ディレクトリを共有したい場合、そのディレクトリを固有のスライスに配置する必要があります。こうすることにより、ディレクトリは、そのほかのブート環境と共有可能なファイルシステムになります。lucreate -m コマンドを実行すると、新しいブート環境が作成され、ディレクトリを固有のスライスに配置することができます。しかし、この新しいファイルシステムはまだ元のブート環境と共有できません。再度 lucreate -m コマンドを実行して、もう 1 つ別のブート環境を作成する必要があります。この 2 つの新しいブート環境では、ディレクトリを共有できません。

たとえば、Solaris 8 から Solaris 9 にアップグレードし、両方のオペレーティング環境で /home を共有したい場合は、lucreate -m コマンドを実行して、/home 固有のスライス上に独立したファイルシステムとして配置し、Solaris 8 リリースを作成します。次に、再度 lucreate -m コマンドを実行し、そのブート環境を複製します。さらに、この複製ブート環境を Solaris 9 リリースへアップグレードします。/home は Solaris 8 リリースと Solaris 9 リリース間で共有されます。

共有可能なファイルシステムおよびクリティカルファイルシステムの概要については、380 ページの「ファイルシステムのタイプ」を参照してください。

---

## 新しいブート環境の内容のカスタマイズ

新しいブート環境を作成するときに、ディレクトリやファイルの一部を新しいブート環境へのコピーから除外できます。ディレクトリを除外した場合は、そのディレクトリ内にある特定のファイルやサブディレクトリが含まれるようにすることが可能です。これらのサブディレクトリは、新しいブート環境にコピーされます。たとえば、`/etc/mail` にあるすべてのファイルとディレクトリを除外するが、`/etc/mail/staff` にあるすべてのファイルとディレクトリは含まれるように指定できます。次のコマンドでは、`staff` サブディレクトリが新しいブート環境にコピーされます。

```
# lucreate -n second_disk -x /etc/mail -y /etc/mail/staff
```



---

注意 - ファイル除外オプションは、注意して使用してください。システムに必要なファイルやディレクトリは削除しないでください。

---

次の表に、`lucreate` コマンドでディレクトリやファイルを除外または追加するためのオプションを示します。

指定方法	除外用のオプション	追加用のオプション
ディレクトリまたはファイルの名前を指定します。	<code>-x exclude_dir</code>	<code>-y include_dir</code>
対象のファイルやディレクトリのリストを含むファイルを使用します。	<code>-f list_filename</code> <code>-z list_filename</code>	<code>-Y list_filename</code> <code>-z list_filename</code>

ブート環境の作成時にディレクトリやファイルをカスタマイズする例については、444ページの「ブート環境の作成と内容のカスタマイズ (コマンド行インタフェース)」を参照してください。



---

## ブート環境間でのファイルの同期

新しいブート環境に切り替えてアクティブにする準備ができれば、新しいブート環境をアクティブにしてリポートするだけです。新たに作成したブート環境を初めて起動するとき、ブート環境間でファイルの同期がとられます。ここでいう「同期」とは、前にアクティブであったブート環境にあるシステムファイルやディレクトリを、ブートされているブート環境にコピーすることです。変更されているファイルやディレクトリがコピーされます。

### /etc/lu/synclist へのファイルの追加

Solaris Live Upgrade では、変更されているクリティカルなファイルがチェックされます。クリティカルなファイルの内容が2つのブート環境で異なっている場合、そのファイルはアクティブなブート環境から新しいブート環境にコピーされます。ファイルの同期は、新しいブート環境の作成後に /etc/passwd や /etc/group などのクリティカルなファイルが変更された場合のために用意されています。

/etc/lu/synclist ファイルには、同期するディレクトリやファイルのリストが記述されています。アクティブなブート環境から新しいブート環境にほかのファイルをコピーしたい場合もあるでしょう。必要に応じて、ディレクトリやファイルを /etc/lu/synclist に追加できます。

/etc/lu/synclist のリストにないファイルを追加すると、システムをブートできなくなる場合があります。同期処理では、ファイルのコピーとディレクトリの作成だけが行われます。ファイルやディレクトリの削除は行われません。

次の /etc/lu/synclist ファイルの例は、このシステムで同期される標準のディレクトリとファイルを示しています。

```
/var/mail                OVERWRITE
/var/spool/mqueue        OVERWRITE
/var/spool/cron/crontabs OVERWRITE
/var/dhcp                OVERWRITE
/etc/passwd              OVERWRITE
/etc/shadow              OVERWRITE
/etc/opasswd             OVERWRITE
/etc/oshadow            OVERWRITE
/etc/group               OVERWRITE
/etc/pwhist              OVERWRITE
/etc/default/passwd     OVERWRITE
/etc/dfs                 OVERWRITE
/var/log/syslog          APPEND
/var/adm/messages       APPEND
```

次のディレクトリやファイルなどは、synclist ファイルに追加してもよいでしょう。

/var/yp	OVERWRITE
/etc/mail	OVERWRITE
/etc/resolv.conf	OVERWRITE
/etc/domainname	OVERWRITE

synclist ファイルのエントリは、ファイルまたはディレクトリです。2 番目のフィールドは、ブート環境をアクティブ化するときに行われる更新の方法を示します。ファイルの更新には 3 通りの方法があります。

- **OVERWRITE** — 新しいブート環境のファイルの内容を、アクティブなブート環境のファイルの内容で上書きします。2 番目のフィールドに動作を指定しない場合は、**OVERWRITE** がデフォルトの動作となります。エントリがディレクトリである場合は、サブディレクトリもすべてコピーされます。すべてのファイルが上書きされます。新しいブート環境では、ファイルの日付、モード、および所有者は前のブート環境のものと同じになります。
- **APPEND** — 新しいブート環境のファイルの末尾に、アクティブなブート環境のファイルの内容を追加します。この処理によってファイル内のエントリが重複することがあります。ディレクトリには **APPEND** 動作を指定することはできません。新しいブート環境では、ファイルの日付、モード、および所有者は前のブート環境のものと同じになります。
- **PREPEND** — 新しいブート環境のファイルの先頭に、アクティブなブート環境のファイルの内容を追加します。この処理によってファイル内のエントリが重複することがあります。ディレクトリには **PREPEND** 動作を指定することはできません。新しいブート環境では、ファイルの日付、モード、および所有者は前のブート環境のものと同じになります。

## ブート環境間での強制的な同期

新しく作成したブート環境で初めてブートする時に、Solaris Live Upgrade は新しいブート環境と以前のアクティブブート環境の同期をとります。この最初のブートと同期の後には、要求しない限り Solaris Live Upgrade で同期は行われません。

- **CUI** を使って強制的に同期を行うには、プロンプトに対して **yes** と入力します。
- **CLI** を使って強制的に同期を行うには、`luactivate` コマンドの `-s` オプションを使用します。

複数のバージョンの Solaris オペレーティング環境を使用していると、強制的に同期を行いたい場合があります。email や passwd/group などのファイルに加えた変更を、アクティブにするブート環境に反映させたい場合があります。強制的に同期を実行すると、Solaris Live Upgrade は、同期をとるファイルの間に矛盾がないかチェックします。新しいブート環境がブートされ、矛盾が検出されると、警告が出されます。この場合、ファイルの同期は行われません。このような場合でも、アクティブ化は正常に終了します。新しいブート環境とアクティブなブート環境の両方で同じファイルに変更を加えると、矛盾が発生することがあります。たとえば、元のブート環境で /etc/passwd ファイルに変更を加えます。そして、新しいブート環境で /etc/passwd ファイルに別の変更を加えた場合などです。このような場合、同期処理では、どちらのファイルをコピーするべきか判断できません。



---

注意 - 以前のアクティブブート環境で発生した変更によりユーザーが気付いていない場合や、それらの変更を制御できない場合もあるため、このオプションを使用する際には十分注意してください。たとえば、現在のブート環境で Solaris 9 ソフトウェアを実行しているとします。この場合、Solaris 7 リリースをブートし、強制的な同期を実行すると、7 リリースのファイルは変更されることがあります。ファイルはオペレーティング環境のリリースに依存しているため、Solaris 7 リリースのブートは失敗することがあります。Solaris 9 のファイルと Solaris 7 のファイルは互換性があるとは限らないからです。

---

---

## リモートシステムからの Solaris Live Upgrade の使用

キャラクタインタフェースを tip 回線などを介してリモートで表示する場合は、必要に応じて TERM 環境変数を VT220 に設定してください。また、共通デスクトップ環境 (CDE) を使用する場合は、TERM 変数の値を xterm ではなく dtterm に設定してください。



## 第 35 章

---

# Solaris Live Upgrade によるブート環境の作成 (作業)

---

この章では、Solaris Live Upgrade のインストール、メニューの使用、およびブート環境の作成について説明します。この章の内容は次のとおりです。

- 413 ページの「Solaris Live Upgrade インタフェースの概要」
- 414 ページの「Solaris Live Upgrade メニューの使用」
- 415 ページの「作業マップ: Solaris Live Upgrade のインストールとブート環境の作成」
- 415 ページの「Solaris Live Upgrade のインストール」
- 417 ページの「Solaris Live Upgrade の起動と停止 (キャラクタインタフェース)」
- 418 ページの「新しいブート環境の作成」

---

## Solaris Live Upgrade インタフェースの概要

Solaris Live Upgrade は、キャラクタユーザーインタフェース (CUI) を介して使用することもコマンド行インタフェース (CLI) で使用することも可能です。以下の説明では、CUI と CLI の両方の場合について手順を示してあります。CUI では、Solaris Live Upgrade の一部の機能にアクセスできません。ここで説明する CLI の手順は、Solaris Live Upgrade コマンドの基本的な使い方を示すものです。コマンド一覧については、第 39 章を参照してください。また、これらのコマンドで使用するオプションの詳細は、関連する各マニュアルページを参照してください。

日本語環境では CUI および CLI のメッセージは、翻訳されていません。

## Solaris Live Upgrade メニューの使用

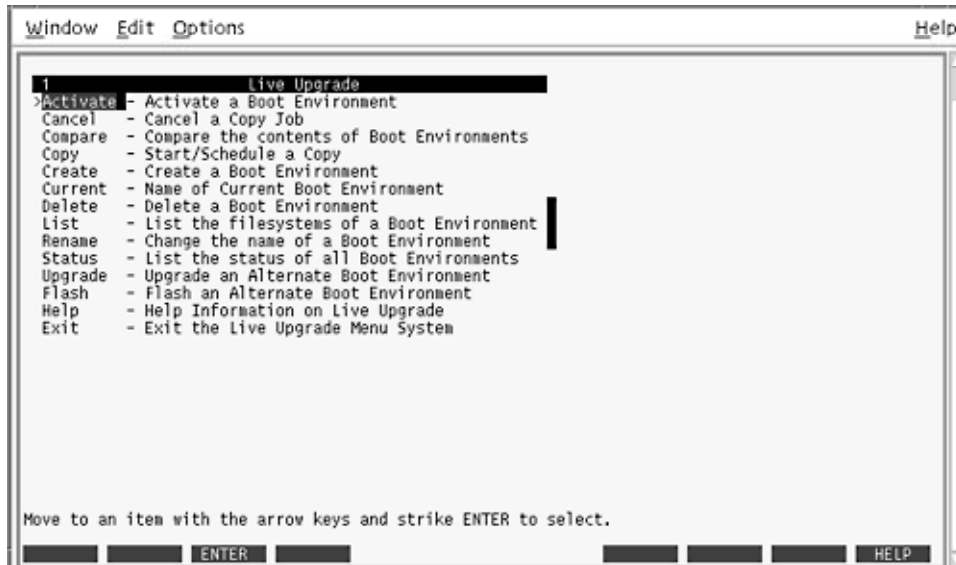


図 35-1 Solaris Live Upgrade のメインメニュー

Solaris Live Upgrade キャラクターユーザーインタフェースのメニュー間の移動には、矢印キーとファンクションキーを使用します。上下に移動する場合やフィールド内にカーソルを置く場合は、矢印キーを使用してください。処理を実行する場合は、ファンクションキーを使用してください。メニューの最下部には、キーボード上のファンクションキーを示す黒い矩形が表示されます。これらは、最初の矩形が F1、2 番目の矩形が F2 という順序で並んでいます。有効な矩形には、「Save」などの処理を示す語句が表示されています。「Configuration」メニューでは、矩形ではなくファンクションキーの番号と対応する動作が表示されます。

- F3 を使用すると、どのメニューの場合も作業の保存 (SAVE) が行われてそのメニューが終了します。
- F6 を使用すると、どのメニューにおいても作業が取り消され (CANCEL)、変更の保存が行われないままメニューが終了します。
- ほかのファンクションキーの動作は、メニューによって異なります。

以下に説明する作業において、ファンクションキーを押すように指示されています。キーボード上のキーが Solaris Live Upgrade メニュー上のファンクションキーと適切に対応していない場合は、Control-F および該当する番号を使用してください。

---

## 作業マップ: Solaris Live Upgrade のインストールとブート環境の作成

表 35-1 作業マップ: Solaris Live Upgrade の使用

作業	説明	参照先
Solaris Live Upgrade パッケージのインストール	オペレーティング環境にパッケージをインストールします。	415 ページの「Solaris Live Upgrade のインストール」
Solaris Live Upgrade の起動	Solaris Live Upgrade のメインメニューを起動します。	417 ページの「Solaris Live Upgrade の起動と停止 (キャラクタインタフェース)」
ブート環境の作成	非アクティブブート環境にファイルシステムをコピーして再構成します。	418 ページの「新しいブート環境の作成」

---

## Solaris Live Upgrade のインストール

現在のオペレーティング環境に Solaris Live Upgrade パッケージをインストールする必要があります。アップグレード後のオペレーティング環境のリリース番号と同じリリース番号の Solaris Live Upgrade パッケージを選択してください。たとえば、オペレーティング環境を Solaris 8 から Solaris 9 にアップグレードする場合、Solaris 9 リリースの Solaris Live Upgrade パッケージが必要になります。

パッケージのインストールには、`pkgadd` コマンドまたは Solaris Live Upgrade インストーラを使用します。416 ページの「`pkgadd` コマンドを使用して Solaris Live Upgrade をインストールする方法」、または416 ページの「Solaris Web Start インストーラを使用して Solaris Live Upgrade をインストールする方法」を参照してください。

---

注 – Solaris 2.6、Solaris 7、または Solaris 8 を実行している場合、Solaris Live Upgrade インストーラを実行できないことがあります。これらのリリースには、Java 2 Runtime Environment の実行に必要なパッチのセットが含まれていません。Solaris Live Upgrade インストーラを実行してパッケージをインストールするには、Java 2 Runtime Environment の推奨パッチクラスタが必要です。Solaris Live Upgrade パッケージをインストールするには、pkgadd コマンドを使用します。または、<http://sunsolve.sun.com> から入手可能な Java 2 Runtime Environment 推奨パッチクラスタをインストールします。

---

## ▼ pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法

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. 次の順序でパッケージをインストールします。

```
# pkgadd -d path_to_packages SUNWlur SUNWluu
path_to_packages ソフトウェアパッケージの絶対パスを指定します。
```

3. 指定したパッケージが正常にインストールされていることを確認します。

```
# pkgchk -v SUNWlur SUNWluu
```

## ▼ Solaris Web Start インストーラを使用して Solaris Live Upgrade をインストールする方法

1. Solaris DVD または Solaris SOFTWARE 2 of 2 CD を挿入します。

2. インストーラを実行します。

- Solaris DVD を使用している場合は、インストーラのあるディレクトリに移動し、インストーラを実行します。

```
# cd /cdrom/cdrom0/Solaris_9/Tools/installers
# ./liveupgrade20
```

Solaris Web Start インストーラが表示されます。

- Solaris SOFTWARE 2 of 2 CD を使用している場合は、インストーラを実行します。

```
% ./installer
```



Solaris Web Start インストーラが表示されます。

3. 「インストール形式の選択 (**Select Type of Install**)」パネルで「カスタム (**Custom**)」をクリックします。
4. 「ロケールの選択 (**Locale Selection**)」パネルで、インストールする言語をクリックします。
5. インストールするソフトウェアを選択します。
  - DVD の場合、「コンポーネントの選択 (**Component Selection**)」パネルの「次へ (**Next**)」をクリックしてパッケージをインストールします。
  - CD の場合、「製品の選択 (**Product Selection**)」パネルの Solaris Live Upgrade の項目で「デフォルトインストール (**Default Install**)」をクリックします。
6. **Solaris Web Start** の指示に従って、ソフトウェアをインストールします。

---

## Solaris Live Upgrade の起動と停止 (キャラクタインタフェース)

Solaris Live Upgrade のメニュープログラムの起動と停止方法について説明します。

### ▼ Solaris Live Upgrade メニューを起動する

---

注 - キャラクタインタフェースを tip 回線などを介してリモートで表示する場合は、必要に応じて TERM 環境変数を VT220 に設定してください。また、共通デスクトップ環境 (CDE) を使用する場合は、TERM 変数の値を xterm ではなく dtterm に設定してください。

---

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# /usr/sbin/lu
```

Solaris Live Upgrade のメインメニューが表示されます。

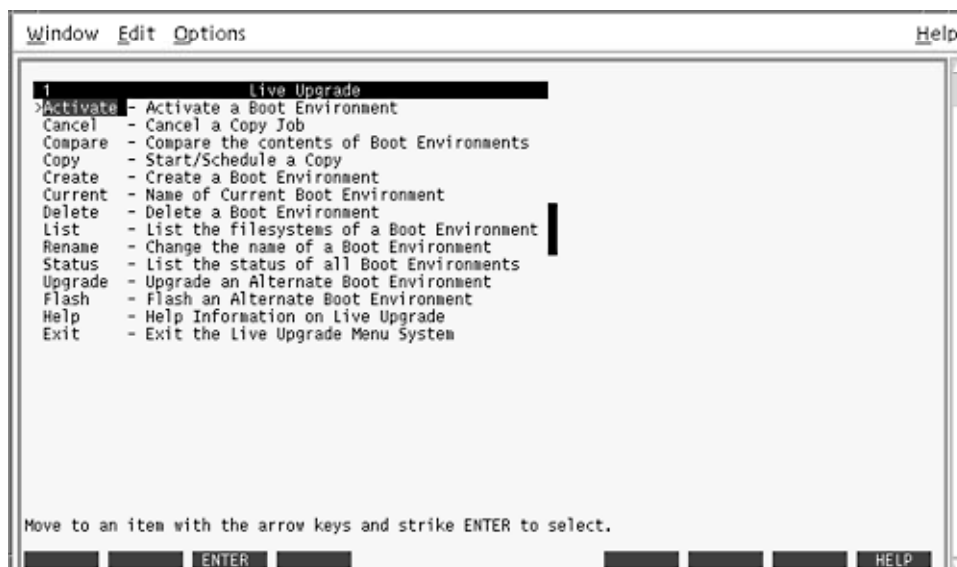


図 35-2 Solaris Live Upgrade のメインメニュー

## ▼ Solaris Live Upgrade のメニューを終了するには

- メインメニューの「Exit」を選択します。

---

## 新しいブート環境の作成

ブート環境を作成すると、アクティブブート環境から新しいブート環境にクリティカルファイルシステムをコピーできます。必要に応じてディスクを再編成し、ファイルシステムをカスタマイズして、クリティカルファイルシステムを新しいブート環境にコピーするには、CUI の「Create」メニュー、「Configuration」サブメニュー、lucreate コマンドを使用します。

ファイルシステムは、新しいブート環境にコピーする前にカスタマイズできます。このため、クリティカルファイルシステムディレクトリを親のディレクトリにマージすることも、親ディレクトリから分離することも可能になります。ユーザー定義の(共有可能)ファイルシステムは、デフォルトで複数のブート環境で共有されます。ただし、必要に応じて共有可能なファイルシステムをコピーすることもできます。ス

ワップ (共有可能なファイルシステム) の分割やマージも可能です。クリティカルファイルシステムと共有可能ファイルシステムの概要については、380 ページの「ファイルシステムのタイプ」を参照してください。

## ▼ ブート環境を作成する (キャラクタインタフェース)

1. メインメニューから「**Create**」を選択します。  
「Create a Boot Environment」サブメニューが表示されます。

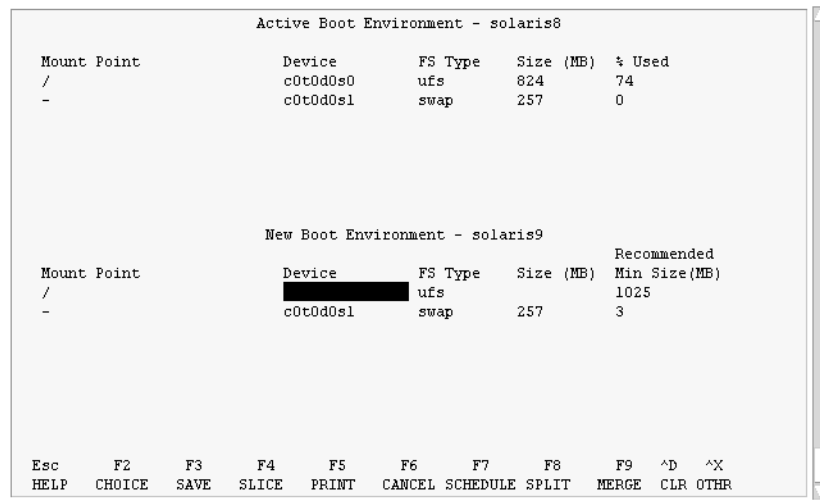
2. アクティブブート環境 (必要に応じて) と新しいブート環境の名前を入力し、確定します。アクティブブート環境の名前の入力が必要なのは、最初にブート環境を作成するときだけです。

ブート環境名は英数字で 30 文字以内とします。使用できるのは英数文字だけで、マルチバイト文字を使用することはできません。

Name of Current Boot Environment:     **solaris8**

Name of New Boot Environment:     **solaris9**

3. **F3** を押して変更を保存します。  
「Configuration」メニューが表示されます。



The screenshot shows a terminal window titled "Active Boot Environment - solaris8" and "New Boot Environment - solaris9". It displays a table of mount points, devices, file system types, and sizes for both environments. The "New Boot Environment" section has a redacted device name. At the bottom, there is a keyboard menu with function keys and their corresponding actions.

Active Boot Environment - solaris8				
Mount Point	Device	FS Type	Size (MB)	% Used
/	c0t0d0s0	ufs	824	74
-	c0t0d0s1	swap	257	0

New Boot Environment - solaris9				
Mount Point	Device	FS Type	Size (MB)	Recommended Min Size (MB)
/	[REDACTED]	ufs		1025
-	c0t0d0s1	swap	257	3

Esc	F2	F3	F4	F5	F6	F7	F8	F9	^D	^X
HELP	CHOICE	SAVE	SLICE	PRINT	CANCEL	SCHEDULE	SPLIT	MERGE	CLR	OTHR

図 35-3 Solaris Live Upgrade の「Configuration」メニュー

「Configuration」メニューには次の項目があります。

- 元のブート環境は、画面の上部に表示されます。画面の下部には作成されるブート環境が表示されます。
  - 「Device」フィールドには次の情報が含まれています。
    - ディスクデバイス名。/dev/dsk/cwtxdysz の形式で表されます。
    - Solaris ボリュームマネージャのメタデバイス名。/dev/md/dsk/dnum の形式で表されます。
    - Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume\_name の形式で表されます。
    - クリティカルファイルシステムの選択領域は、クリティカルファイルシステムを選択するまで空白のままです。/usr、/var、/opt などのクリティカルファイルシステムは、分割可能です。ルート (/) ファイルシステムとマージすることもできます。
    - /export などの共有可能なファイルシステム (スワップ) は「Device」フィールドに表示されます。これらのファイルシステムは、ソースブート環境とターゲットブート環境の両方に同じマウントポイントを持っています。スワップはデフォルトで共有されますが、スワップスライスの分割とマージ (追加と削除) も行うことができます。  
クリティカルファイルシステムと共有可能なファイルシステムの概要については、380 ページの「ファイルシステムのタイプ」を参照してください。
  - 「FS\_Type」フィールドでは、ファイルシステムの種類を変更できます。ファイルシステムの種類は、次のいずれかになります。
    - vxfs: Veritas ファイルシステムを示す
    - swap: スワップファイルシステムを示す
    - ufs: UFS ファイルシステムを示す
4. (省略可能) 以下の作業は、必要に応じて行うことができます。
- 画面上の情報を ASCII ファイルに出力するには F5 を押します。
  - ファイルシステムのリストをスクロールするには、Control-X を押します。  
この操作で、アクティブブート環境のファイルシステムと新しいブート環境のファイルシステムを切り替えてスクロールできるようになります。
  - 「Configuration」メニューを閉じるには、F6 を押します。
    - 「Configuration」メニューが表示されている場合は、変更が保存されず、ファイルシステムは修正されません。
    - 「Configuration」サブメニューが表示されている場合は、「Configuration」メニューに戻ります。
5. F2 を押して、利用できるスライスを選択します。  
「Choices」メニューでは、カーソルが置かれているフィールドに、そのシステム上で利用できるスライスが表示されます。表示されるのは「Slice」フィールドと「file system FS\_Type」フィールドです。

- a. 矢印キーを使用してフィールド内にカーソルを置き、スライスまたはファイルシステムの種類を選択します。
  - 「Slice」フィールドにカーソルを置くと、すべての空きスライスが表示されます。ルート (/) の場合、「Choices」に表示されるのはルート (/) ファイルシステムの制限事項を満たす空きスライスだけです。403 ページの「ルート (/) ファイルシステムのスライスを選択するための指針」を参照してください。
  - 「FS\_Type」フィールドにカーソルを置くと、利用できるすべてのファイルシステムタイプが表示されます。
  - 現在のファイルシステムには、ボールド書体のスライスを選択できます。スライスのサイズは、ファイルシステムのサイズにアップグレード用に 30% を加えることによって概算できます。
  - ボールド書体ではないスライスは、そのファイルシステムをサポートするにはサイズが小さすぎることを意味します。ディスクスライスを作成し直す方法については、手順 6 を参照してください。
- b. **Return** キーを押してスライスを選択します。

選択したスライスが「Slice」フィールドに表示されるか、あるいは「FS\_Type」フィールド内でファイルシステムの種類が変化します。
6. (省略可能) 空きスライスが最小要件を満たしていない場合は、**F4** を押して任意の空きスライスを分割し直してください。

「Solaris Live Upgrade Slice Configuration」メニューが表示されます。

新しいスライスを作成できるように、`format` コマンドが実行されます。画面の指示に従って新しいスライスを作成してください。`format` コマンドについての詳細は、`format` のマニュアルページを参照してください。

矢印キーを使って、「Device」フィールドと「FS\_Type」フィールド間を移動できます。デバイスを選択すると、「Size (Mbytes)」フィールドに値が自動的に表示されます。

  - a. デバイスを解放するには、**Control-D** を押します。

以上の操作でスライスが利用できるようになり、「Choices」メニューに表示されます。
  - b. **F3** を押して「**Configuration**」メニューに戻ります。
7. (省略可能) クリティカルファイルシステムを分割すると、そのファイルシステムは別々のマウントポイントに配置されます。ファイルシステムを分割するには、次の手順に従ってください。

(ファイルシステムのマージについては、手順 8 を参照してください。)

  - a. 分割するファイルシステムを選択します。

`/usr`、`/var`、`/opt` などのファイルシステムは、それらの親ディレクトリから分割できます。

---

注 - ブート環境のファイルシステムを作成する場合のルールは、Solaris オペレーティング環境のファイルシステムを作成する場合と同じです。Solaris Live Upgrade では、クリティカルなファイルシステムに無効な構成を作成してしまうことを回避できません。たとえば、lucreate コマンドを用いて、ルート (/) と /kernel を別々のファイルシステムに作成することができますが、このようにルート (/) を分割するのは誤りです。

---

- b. **F8** を押します。
  - c. 新しいブート環境のファイルシステム名を入力します。例:  

```
Enter the directory that will be a separate file system
on the new boot environment: /opt
```

新しいファイルシステムが検証されると、画面に新しい行が追加されます。
  - d. **F3** を押して「**Configuration**」メニューに戻ります。  
「**Configuration**」メニューが表示されます。
8. (省略可能) マージを行うと、ファイルシステムは同じマウントポイントに配置されます。ファイルシステムをその親ディレクトリにマージするには、次の操作を行います。  
(ファイルシステムの分割については、手順 7 を参照してください。)
- a. マージするファイルシステムを選択します。  
/usr、/var、/opt などのファイルシステムをそれらの親ディレクトリにマージできます。
  - b. **F9** を押します。  
次の例のように、結合されるファイルシステムが表示されます。  

```
/opt will be merged into /.
```
  - c. **Return** キーを押します。
  - d. **F3** を押して「**Configuration**」メニューに戻ります。  
「**Configuration**」メニューが表示されます。
9. (省略可能) スワップスライスを追加するか削除するかを決定します。
- スワップスライスを分割して新しいスライスに配置したい場合は、手順 10 に進みます。
  - スワップスライスを削除したい場合は、手順 11 に進みます。
10. (省略可能) スワップスライスを分割するには、次の手順に従います。
- a. 「**Device**」フィールドで、分割したいスワップスライスを選択します。

b. **F8** を押します。

c. プロンプトに対して、次のコマンドを入力します。

```
Enter the directory that will be a separate filesystem on  
the new BE: swap
```

d. **F2 (Choice)** を押します。

「Choice」メニューに、スワップに利用できるスライスが表示されます。

e. スワップを配置するスライスを選択します。

そのスライスが「Device」フィールドに表示され、スワップの新しいスライスとなります。

11. (省略可能) スワップスライスを削除するには、次の手順に従います。

a. 「Device」フィールドで、削除するスワップスライスを選択します。

b. **F9** を押します。

c. プロンプトが表示されたら「y」と入力します。

```
Slice /dev/dsk/c0t4d0s0 will not be swap partition.  
Please confirm? [y, n]: y
```

このスワップスライスが削除されました。

12. 今すぐにブート環境を作成するか、後で作成するようにスケジュールするかを決定します。

■ すぐに新しいブート環境を作成する場合は、**F3** を押します。

構成が保存され、構成画面が閉じます。このファイルシステムがコピーされ、ブート環境がブート可能になり、非アクティブブート環境が作成されます。

ブート環境の作成には、システム構成に応じて、1時間以上かかる場合があります。続いて、Solaris Live Upgrade メインメニューが表示されます。

■ 後で作成されるようにスケジュールする場合は、次の例に示すように「y」と入力し、続いて開始時刻と電子メールアドレスを入力します。

```
Do you want to schedule the copy? y  
Enter the time in 'at' format to schedule create: 8:15 PM  
Enter the address to which the copy log should be mailed: someone@anywhere.com
```

処理の完了は、電子メールで通知されます。

時間の書式については、at (1) のマニュアルページを参照してください。

スケジュールできるのは一度に1つのジョブだけです。

作成が完了すると、非アクティブブート環境をアップグレードできるようになります。第36章を参照してください。

## ▼ ブート環境を初めて作成する (コマンド行インタフェース)

lucreate コマンドに `-m` オプションを指定することにより、新しいブート環境に作成するファイルシステムとその数を指定できます。作成するファイルシステムの数だけ、このオプションを繰り返し指定する必要があります。たとえば、`-m` オプションを 1 回だけ使用した場合、すべてのファイルシステムが指定の場所に格納されます。元のブート環境のすべてのファイルシステムが、`-m` で指定されたファイルシステムにマージされます。`-m` オプションを 2 回使用すると、ファイルシステムが 2 つ作成されます。`-m` オプションを使ってファイルシステムを作成する場合、次の項目に留意してください。

- `-m` オプションを 1 個指定して、新しいブート環境のルート (`/`) ファイルシステムを作成する必要があります。`-m` オプションを指定しないで `lucreate` を実行すると、「Configuration」メニューが表示されます。「Configuration」メニューでは、新しいマウントポイントにファイルを変更して新しいブート環境をカスタマイズできます。
- 現在のブート環境にクリティカルファイルシステムがある場合、このファイルシステムは、`-m` オプションで指定しなくても新しく作成されたファイルシステムの上位 2 番目のファイルシステムにマージされます。
- 新しいブート環境には、`-m` オプションで指定されたファイルシステムだけが作成されます。現在のブート環境に複数のファイルシステムがあり、新しいブート環境にも同じ数のファイルシステムを作成したい場合は、ファイルシステムごとに 1 個ずつ `-m` オプションを指定します。たとえば、ルート (`/`) ファイルシステム、`/opt` ファイルシステム、`/var` ファイルシステムがある場合、これらを新しいブート環境に作成するには、それぞれに `-m` オプションを 1 個ずつ指定します。
- マウントポイントが重複しないようにしてください。たとえば、ルート (`/`) ファイルシステムを 2 つ作成することはできません。

1. スーパーユーザーとしてログインします。
2. 次のように入力して新しいブート環境を作成します。

```
# lucreate [-A 'BE_description'] -c BE_name \  
-m mountpoint:device[,metadevice]:fs_options [-m ...] -n BE_name
```

`-A 'BE_description'`

(省略可能) ブート環境名 (`BE_name`) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

`-c BE_name`

アクティブブート環境に名前 `BE_name` を割り当てます。このオプションは省略可能で、最初のブート環境を作成する場合だけ使用されます。`lucreate` を初めて実行する場合に `-c` オプションを省略すると、デフォルトの名前が作成されます。

デフォルトの名前は、次の基準に従って選択されます。



- 物理ブートデバイスが判別可能な場合は、その物理ブートデバイスのベース名が現在のブート環境の名前になります。

たとえば、物理ブートデバイスが `/dev/dsk/c0t0d0s0` であれば、現在のブート環境には `c0t0d0s0` という名前が与えられます。

- 物理ブートデバイスが判別不可能な場合は、`uname` コマンドの `-s` オプションと `-r` オプションで取得される名前が組み合わせられます。

たとえば、`uname -s` で取得されるオペレーティング環境の名前が `SunOS`、`uname -r` で取得されるリリース名が `5.9` であれば、現在のブート環境には `SunOS5.9` という名前が与えられます。

- 上記のどちらの方法でも名前を決定できない場合、現在のブート環境には `current` という名前が与えられます。

---

注 - 最初のブート環境を作成した後は、`-c` オプションを指定しても無視されるか、エラーメッセージが表示されます。

- 現在のブート環境と同じ名前を指定すると、このオプションは無視されません。
- 現在のブート環境と異なる名前を指定すると、作成は失敗し、エラーメッセージが表示されます。次の例は、ブート環境の名前によってエラーメッセージが発生するようすを示しています。

```
# lucurr
c0t0d0s0
# lucreate -c c1t1d1s1 -n newbe -m /:c1t1d1s1:ufs
ERROR: current boot environment name is c0t0d0s0: cannot change name
using <-c c1t1d1s1>
```

---

`-m mountpoint:device[,metadevice]:fs_options [-m ...]`

新しいブート環境のファイルシステム構成を `vfstab` で指定します。 `-m` に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す `-` (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。
  - ディスクデバイスの名前。 `/dev/dsk/cwt.xdysz` の形式で表されます。
  - Solaris ボリュームマネージャのボリューム名。 `/dev/md/dsk/dnum` の形式で表されます。
  - Veritas Volume Manager のボリューム名。 `/dev/md/vxfs/dsk/dnum` の形式で表されます。
  - キーワード `merged`。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。
- `fs_options` フィールドには、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す
- vxfs: Veritas ファイルシステムを示す
- swap: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
- 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、438 ページの「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。

-n *BE\_name*

作成するブート環境の名前。BE\_name 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。

例 35-1 ブート環境を作成する (コマンド行)

この例では、アクティブブート環境の名前は first\_disk です。-m オプションはファイルシステムのマウントポイントを表します。ルート (/) ファイルシステムと /usr ファイルシステムが作成されます。新しいブート環境の名前は second\_disk です。「mydescription」という記述は、second\_disk に対応しています。新しいブート環境 second\_disk のスワップは、自動的にソースである first\_disk から共有されます。

```
# lucreate -A 'mydescription' -c first_disk
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s3:ufs \
-n second_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

## ▼ ブート環境を作成しファイルシステムをマージする (コマンド行インタフェース)

---

注 -lucreate コマンドに -m オプションを指定することにより、新しいブート環境に作成するファイルシステムとその数を指定できます。作成するファイルシステムの数だけ、このオプションを繰り返し指定する必要があります。たとえば、-m オプションを 1 回だけ使用した場合、すべてのファイルシステムが指定の場所に格納されます。元のブート環境のファイルシステムがすべて 1 つのファイルシステムにマージされます。-m オプションを 2 回使用すると、ファイルシステムが 2 つ作成されます。

---

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# lucreate [-A 'BE_description'] \
-m mountpoint:device[,metadevice]:fs_options \
[-m [...]] -m mountpoint:merged:fs_options -n BE_name
```

**-A BE\_description**  
(省略可能) ブート環境名 (BE\_name) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

**-m mountpoint:device[,metadevice]:fs\_options [-m...]**  
新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- *mountpoint* には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- *device* フィールドには、次のいずれかを指定できます。
  - ディスクデバイスの名前。/dev/dsk/cwt.xdysz の形式で表されます。
  - Solaris ボリュームマネージャのメタデバイス名。/dev/md/dsk/dnum の形式で表されます。
  - Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume\_name の形式で表されます。
  - キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。
- *fs\_options* フィールドには、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す
  - vxfs: Veritas ファイルシステムを示す
  - swap: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
  - 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、438 ページの「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。

**-n BE\_name**  
作成するブート環境の名前。BE\_name 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。

**例 35-2** ブート環境を作成しファイルシステムをマージする (コマンド行インタフェース)

この例の場合、現在のブート環境には、ルート (/) ファイルシステム、/usr ファイルシステム、/opt ファイルシステムがあります。/opt ファイルシステムは、親ファイルシステム /usr にマージされます。新しいブート環境の名前は second\_disk です。「mydescription」という記述は、second\_disk に対応しています。

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s1:ufs \
```

例 35-2 ブート環境を作成しファイルシステムをマージする (コマンド行インタフェース) (続き)

```
-m /usr/opt:merged:ufs -n second_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

## ▼ ブート環境を作成しファイルシステムを分割する (コマンド行インタフェース)

---

注 - ブート環境のファイルシステムを作成する場合のルールは、Solaris オペレーティング環境のファイルシステムを作成する場合と同じです。Solaris Live Upgrade では、クリティカルなファイルシステムに無効な構成を作成してしまうことを回避できません。たとえば、lucreate コマンドを用いて、ルート (/) と /kernel を別々のファイルシステムに作成することができますが、このようにルート (/) を分割するのは誤りです。

---

1 つのディレクトリを複数のマウントポイントに分割すると、ファイルシステム間でハードリンクが維持されなくなります。たとえば、/usr/stuff1/file が /usr/stuff2/file にハードリンクされている場合に /usr/stuff1 と /usr/stuff2 を別々のファイルシステムに分割すると、ファイル間のリンクは解除されます。lucreate から警告メッセージが表示され、解除されたハードリンクの代わりとなるシンボリックリンクが作成されます。

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# lucreate [-A 'BE_description' ] \  
-m mountpoint:device[,metadevice]:fs_options \  
[-m mountpoint:device[,metadevice]:fs_options] -n new_BE
```

```
-A 'BE_description'
```

(省略可能) ブート環境名 (BE\_name) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

```
-m mountpoint:device[,metadevice]:fs_options [-m...]
```

新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- *mountpoint* には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- *device* フィールドには、次のいずれかを指定できます。

- ディスクデバイスの名前。/dev/dsk/cwt.xdyysz の形式で表されます。
  - Solaris ボリュームマネージャのメタデバイス名。/dev/md/dsk/dnum の形式で表されます。
  - Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume\_name の形式で表されます。
  - キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。
- fs\_options フィールドには、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す
    - vxfs: Veritas ファイルシステムを示す
    - swap: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
    - 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、438 ページの「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。

-n BE\_name

作成するブート環境の名前。BE\_name 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。

例 35-3 ブート環境を作成しファイルシステムを分割する (コマンド行インタフェース)

この例では、前述のコマンドによって ルート (/) ファイルシステムを新しいブート環境内の複数のディスクスライスに分割しています。ここでは、/usr、/var、および /opt をすべてルート (/) に置いている次のソースブート環境を想定してください。/dev/dsk/c0t0d0s0 / です。

新しいブート環境で、次に示すように別々のスライスにマウントすることによって、ファイルシステム /usr、/var、/opt を分割します。

```
/dev/dsk/c0t1d0s0 /
/dev/dsk/c0t1d0s1 /var
/dev/dsk/c0t1d0s7 /usr
/dev/dsk/c0t1d0s5 /opt
```

「mydescription」という記述は、ブート環境名 second\_disk に対応していません。

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t1d0s0:ufs -m /usr:/dev/dsk/c0t1d0s7:ufs \
-m /var:/dev/dsk/c0t1d0s1:ufs -m /opt:/dev/dsk/c0t1d0s5:ufs \
-n second_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

## ▼ ブート環境を作成しスワップを再構成する (コマンド行インタフェース)

スワップスライスは、デフォルトでは複数のブート環境で共有されます。-m オプションでスワップを「指定しない」場合、現在のブート環境と新しいブート環境が同じスワップスライスを共有します。新しいブート環境のスワップを構成し直したい場合は、-m オプションを使用してそのブート環境に対してスワップスライスの追加または削除を行なってください。

---

注 - スワップスライスを分割したりマージしたりするには、現在のブート環境 (-s オプションを使用した場合はソースブート環境) 以外のブート環境では、スワップスライ스가使用中であってはならないという制限があります。ファイルシステムの種類 (スワップ、ufs など) にかかわらず、スワップスライスはほかのブート環境によって使用されている場合、ブート環境の作成は失敗します。

既存のスワップスライスをを使用してブート環境を作成した後、`vfstab` ファイルを編集することができます。

---

1. スーパーユーザーとしてログインします。

2. 次のコマンドを入力します。

```
# lucreate [-A 'BE_description'] \  
-m mountpoint:device[,metadevice]:fs_options \  
-m -:device:swap -n BE_name
```

```
-A 'BE_description'
```

(省略可能) ブート環境名 (BE\_name) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

```
-m mountpoint:device[,metadevice]:fs_options [-m..]
```

新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。
  - ディスクデバイスの名前。/dev/dsk/cwtxydsz の形式で表されます。
  - Solaris ボリュームマネージャのメタデバイス名。/dev/md/dsk/dnum の形式で表されます。
  - Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume\_name の形式で表されます。
  - キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。

- `fs_options` フィールドには、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す
  - `vxfs`: Veritas ファイルシステムを示す
  - `swap`: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
  - 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、438 ページの「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。

`-n BE_name`

作成するブート環境の名前。BE\_name は、一意となるように指定する必要があります。

スワップが別のスライスまたはデバイスに移動し、新しいブート環境が作成されません。

例 35-4 ブート環境を作成しスワップを再構成する (コマンド行インタフェース)

この例の場合、現在のブート環境には、`/dev/dsk/c0t0d0s0` にルート (/)、`/dev/dsk/c0t0d0s1` にスワップがあります。新しいブート環境はルート (/) を `/dev/dsk/c0t4d0s0` にコピーし、`/dev/dsk/c0t0d0s1` と `/dev/dsk/c0t4d0s1` の両方をスワップスライスとして使用します。「mydescription」という記述は、ブート環境名 `second_disk` に対応していません。

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -m -:/dev/dsk/c0t0d0s1:swap \
-m -:/dev/dsk/c0t4d0s1:swap -n second_disk
```

これらのスワップ割り当てでは、`second_disk` からブートが行われて初めて有効になります。スワップスライスが多数存在する場合は、`-M` オプションを使用してください。431 ページの「リストを使用してブート環境を作成しスワップを再構成する (コマンド行インタフェース)」を参照してください。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

## ▼ リストを使用してブート環境を作成しスワップを再構成する (コマンド行インタフェース)

スワップスライスが多数存在する場合は、スワップリストを作成してください。lucreate は、新しいブート環境のスワップスライスにこのリストを使用します。

---

注 – スワップスライスを分割したりマージしたりするには、現在のブート環境 (-s オプションを使用した場合はソースブート環境) 以外のブート環境では、スワップスライ스가使用中であってはならないという制限があります。スワップスライスのファイルシステムの種類 (swap、ufs) などにかかわらず、スワップスライ스가他のブート環境によって使用されている場合、ブート環境の作成は失敗します。

---

1. 新しいブート環境で使用されるスワップスライスのリストを作成します。このファイルの場所と名前はユーザーが決定できます。この例では、**/etc/lu/swapslices** ファイルにはデバイスとスライスが挙げられています。

```
 -: /dev/dsk/c0t3d0s2:swap
 -: /dev/dsk/c0t3d0s2:swap
 -: /dev/dsk/c0t4d0s2:swap
 -: /dev/dsk/c0t5d0s2:swap
 -: /dev/dsk/c1t3d0s2:swap
 -: /dev/dsk/c1t4d0s2:swap
 -: /dev/dsk/c1t5d0s2:swap
```

2. 次のコマンドを入力します。

```
# lucreate [-A 'BE_description' \
-m mountpoint:device[,metadevice]:fs_options \
-M slice_list -n BE_name
```

```
-A 'BE_description'
```

(省略可能) ブート環境名 (BE\_name) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

```
-m mountpoint:device[,metadevice]:fs_options [-m...]
```

新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- **mountpoint** には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- **device** フィールドには、次のいずれかを指定できます。
  - ディスクデバイスの名前。/dev/dsk/cwt.xdysz の形式で表されます。
  - Solaris ボリュームマネージャのメタデバイス名。/dev/md/dsk/dnum の形式で表されます。
  - Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume\_name の形式で表されます。
  - キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。
- **fs\_options** フィールドには、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す
  - vxfs: Veritas ファイルシステムを示す



- `swap`: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
- 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、438 ページの「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。

#### `-M slice_list`

ファイル `slice_list` 中には、`-m` オプションのリストが記述されています。これらの引数は、`-m` に指定されている書式で指定してください。ハッシュ記号 (#) で始まるコメント行は無視されます。`-m` オプションは、ブート環境用のファイルシステムが多数存在する場合に便利です。`-m` オプションと `-M` オプションは一緒に使えます。たとえば、`slice_list` にスワップスライスを記録しておき、`-m` を使用して、ルート (/) スライスと `/usr` スライスを指定できます。

`-m` オプションと `-M` オプションでは、特定のマウントポイントについて複数のスライスを指定できます。これらのスライスを処理する場合、`lucreate` は利用不可能なスライスをスキップして利用できる最初のスライスを選択します。

#### `-n BE_name`

作成するブート環境の名前。`BE_name` は、一意となるように指定する必要があります。

**例 35-5** リストを使用してブート環境を作成しスワップを再構成する (コマンド行インタフェース)

この例では、新しいブート環境のスワップは、`/etc/lu/swapslices` ファイルに挙げられている一連のスライスです。「`mydescription`」という記述は、`second_disk` に対応しています。

```
# lucreate -A 'mydescription' -c first_disk \
-m /:/dev/dsk/c02t4d0s0:ufs -m /usr:/dev/dsk/c02t4d0s1:ufs \
-M /etc/lu/swapslices -n second_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

## ▼ ブート環境を作成し共有可能ファイルシステムをコピーする (コマンド行インタフェース)

新しいブート環境に共有可能ファイルシステムをコピーしたい場合は、`-m` オプションを使用してマウントポイントがコピーされるように指定してください。それ以外の場合、共有可能なファイルシステムはデフォルトで共有され、`vfstab` ファイルに指定された同じマウントポイントを使用します。共有可能ファイルシステムに対する更新は、両方の環境に適用されます。

1. スーパーユーザーとしてログインします。

## 2. ブート環境を作成します。

```
# lucreate [-A 'BE_description' ] \  
-m mountpoint:device[,metadevice]:fs_options \  
[-m ]mountpoint:device[,metadevice]:fs_options -n BE_name
```

-A 'BE\_description'

(省略可能) ブート環境名 (BE\_name) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

-m mountpoint:device[,metadevice]:fs\_options [-m...]

新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- *mountpoint* には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- *device* フィールドには、次のいずれかを指定できます。
  - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます。
  - Solaris ポリユーモネージャのメタデバイス名。/dev/md/dsk/dnum の形式で表されます。
  - Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume\_name の形式で表されます。
  - キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。
- *fs\_options* フィールドには、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す
  - vxfs: Veritas ファイルシステムを示す
  - swap: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
  - 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、438 ページの「RAID-1 ポリユーモ (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。

-n BE\_name

作成するブート環境の名前。BE\_name は、一意となるように指定する必要があります。

例 35-6 ブート環境を作成し共有可能ファイルシステムをコピーする (コマンド行インタフェース)

この例の場合、現在のブート環境には、ルート (/) ファイルシステムと /home ファイルシステムがあります。新しいブート環境では、ルート (/) ファイルシステムがルート (/) と /usr の 2 つのファイルシステムに分割されます。/home ファイルシステムは新しいブート環境にコピーされます。「mydescription」という記述は、ブート環境名 second\_disk に対応しています。

例 35-6 ブート環境を作成し共有可能ファイルシステムをコピーする (コマンド行インタフェース) (続き)

```
# lucreate -A 'mydescription' -c first_disk \  
-m /:/dev/dsk/c0t4d0s0:ufs -m /usr:/dev/dsk/c0t4d0s3:ufs \  
-m /home:/dev/dsk/c0t4d0s4:ufs -n second_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

## ▼ 別々のソースから単一のブート環境を作成 (コマンド行インタフェース)

lucreate コマンドは、アクティブブート環境内のファイルシステムに基づいてブート環境を作成します。アアクティブブート環境以外のブート環境に基づいてブート環境を作成する場合は、`-s` オプションを指定して lucreate を実行します。

---

注 - 新しいブート環境をアクティブにした後、フォールバックを行う必要がある場合は、ソースブート環境ではなく最後にアクティブだったブート環境に戻ります。

---

1. スーパーユーザーとしてログインします。
2. ブート環境を作成します。

```
# lucreate [-A 'BE_description'] -s source_BE_name  
-m mountpoint:device[,metadevice]:fs_options -n BE_name
```

`-A 'BE_description'`  
(省略可能) ブート環境名 (BE\_name) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

`-s source_BE_name`  
新しいブート環境に対するソースブート環境を指定します。このソースはアクティブブート環境ではありません。

`-m mountpoint:device[,metadevice]:fs_options [-m...]`  
新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。
  - ディスクデバイスの名前。/dev/dsk/cwt.xdysz の形式で表されます。

- Solaris ボリュームマネージャのメタデバイス名。/dev/md/dsk/dnum の形式で表されます。
- Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume\_name の形式で表されます。
- キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。
- fs\_options フィールドには、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す
  - vxfs: Veritas ファイルシステムを示す
  - swap: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
  - 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、438 ページの「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。

-n BE\_name

作成するブート環境の名前。BE\_name 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。

例 35-7 別々のソースから単一のブート環境を作成 (コマンド行インタフェース)

この例では、ソースブート環境 third\_disk 内のルートファイルシステム (/) に基づいてブート環境を作成します。Third\_disk はアクティブブート環境ではありません。「mydescription」という記述は、ブート環境名 second\_disk に対応しています。

```
# lucreate -A 'mydescription' -s third_disk \
-m /:/dev/dsk/c0t4d0s0:ufs -n second_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

## ▼ フラッシュアーカイブ用の空のブート環境の作成 (コマンド行インタフェース)

lucreate コマンドは、アクティブブート環境内のファイルシステムに基づいてブート環境を作成します。lucreate コマンドに -s - オプションを指定して実行すると、空のブート環境を短時間で作成できます。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、フラッシュアーカイブがインストールされる時にはじめて作成されることとなります。空のブート環境にアーカイブがインストールされると、ファイルシステムは予約されたスライスにインストールされます。

1. スーパーユーザーとしてログインします。

## 2. 空のブート環境を作成します。

```
# lucreate -A 'BE_name' -s - \  
-m mountpoint:device[,metadevice]:fs_options -n BE_name
```

-A 'BE\_description'

(省略可能) ブート環境名 (BE\_name) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

-s -

空のブート環境を作成します。

-m mountpoint:device[,metadevice]:fs\_options [-m...]

新しいブート環境のファイルシステム構成を指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- *mountpoint* には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- *device* フィールドには、次のいずれかを指定できます。
  - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます。
  - Solaris ボリュームマネージャのメタデバイス名。/dev/md/dsk/dnum の形式で表されます。
  - Veritas Volume Manager のボリューム名。/dev/vx/dsk/volume\_name の形式で表されます。
  - キーワード merged。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。
- *fs\_options* フィールドには、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す
  - vxfs: Veritas ファイルシステムを示す
  - swap: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
  - 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、438 ページの「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。

-n BE\_name

作成するブート環境の名前。BE\_name 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。

### 例 35-8 フラッシュアーカイブ用の空のブート環境の作成 (コマンド行インタフェース)

この例では、ファイルシステムを一切含まないブート環境を作成します。「mydescription」という記述は、ブート環境名 second\_disk に対応しています。

```
# lucreate -A 'mydescription' -s - \  
-m /:/dev/dsk/c0t1d0s0:ufs -n second_disk
```

例 35-8 フラッシュアーカイブ用の空のブート環境の作成 (コマンド行インタフェース)  
(続き)

空のブート環境の作成が完了したら、フラッシュアーカイブをインストールし、アクティブ (ブート可能な状態) にすることができます。第 36 章を参照してください。

空のブート環境の作成とアーカイブのインストールの例については、498 ページの「空のブート環境を作成してフラッシュアーカイブをインストールする例 (コマンド行インタフェース)」を参照してください。

図 35-4 は、空のブート環境の作成の様子を示しています。

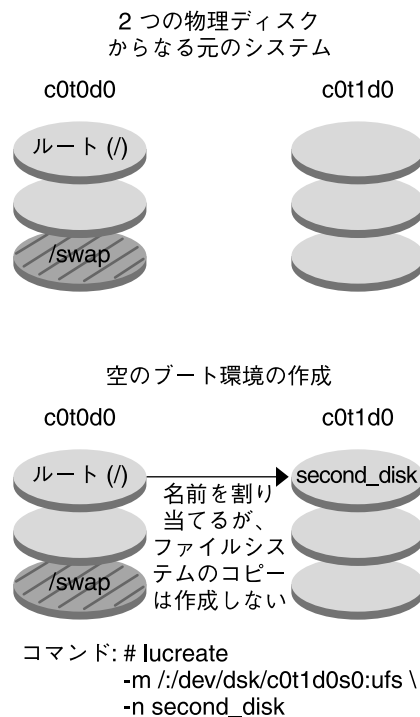


図 35-4 空のブート環境の作成

## ▼ 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)

ブート環境を作成するとき、Solaris Live Upgrade は Solaris ボリュームマネージャテクノロジーを使って RAID-1 ボリュームを作成します。ブート環境を作成するとき、Solaris Live Upgrade を使って次の作業を行うことができます。

- 単一スライスの連結 (サブミラー) を RAID-1 ボリューム (ミラー) から切り離す。必要な場合は、内容を保存して新しいブート環境の内容にすることができます。内容はコピーされないため、新しいブート環境を短時間で作成できます。ミラーから切り離されたサブミラーは、元のミラーの一部ではなくなります。サブミラーに対する読み取りや書き込みがミラーを介して実行されることはなくなります。
- ミラーを含んだブート環境を作成する
- 新しく作成したミラーに単一スライスの連結を接続する。

Solaris Live Upgrade のミラー化機能を使用するには、状態データベースと状態データベースの複製を作成する必要があります。状態データベースでは、Solaris ボリュームマネージャ構成の状態に関する情報がディスクに保存されます。

- 状態データベースの作成については、『Solaris ボリュームマネージャの管理』の「状態データベース (概要)」を参照してください。
- Solaris ボリュームマネージャの概要と、Solaris Live Upgrade で実行できる作業については、384 ページの「ミラー化されたファイルシステムを持つブート環境の作成」を参照してください。
- Solaris Live Upgrade を使用するときには使用できない Solaris ボリュームマネージャの複雑な構成については、『Solaris ボリュームマネージャの管理』の「記憶装置管理の概念」を参照してください。

1. スーパーユーザーとしてログインします。
2. 次のように入力して新しいブート環境を作成します。

```
# lucreate [-A 'BE_description' \
-m mountpoint:device[,metadevice]:fs_options [-m... ] \
-n BE_name
```

```
-A 'BE_description'
```

(省略可能) ブート環境名 ( *BE\_name* ) の説明を記述できます。記述の長さ、使用できる文字に制限はありません。

```
-m mountpoint:device[,metadevice]:fs_options [-m...]
```

新しいブート環境のファイルシステム構成を `vfstab` で指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- *mountpoint* には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- *device* フィールドには、次のいずれかを指定できます。
  - ディスクデバイスの名前。/dev/dsk/cwtxdysz の形式で表されます。
  - Solaris ボリュームマネージャのボリューム名。/dev/md/dsk/dnum の形式で表されます。
  - Veritas Volume Manager のボリューム名。/dev/md/vxfs/dsk/dnum の形式で表されます。
  - キーワード `merged`。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。

- *fs\_options* フィールドには、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す
    - *vxfs*: Veritas ファイルシステムを示す
    - *swap*: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
    - 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。
      - *mirror* を指定すると、指定したデバイスに RAID-1 ボリューム (ミラー) を作成できます。その後の *-m* オプションで *attach* を指定して、少なくとも 1 つの連結を新しいミラーに接続する必要があります。指定するデバイスには、正しく名前が付けられている必要があります。たとえば、論理デバイスの名前 */dev/md/dsk/d10* や短縮名 *d10* をミラー名として使用できます。デバイスの命名の詳細は、『Solaris ボリュームマネージャの管理』の「Solaris ボリュームマネージャコンポーネントの概要」を参照してください。
      - *detach* を指定すると、指定したマウントポイントに関連付けられているボリュームから連結を切り離すことができます。ボリュームを指定する必要はありません。
      - *attach* を指定すると、指定したマウントポイントに関連付けられているミラーに連結を接続できます。指定した物理ディスクスライスは、単一デバイスの連結になり、ミラーに接続されます。ディスクに接続する連結を指定するには、デバイス名の後ろにコンマと連結の名前を付加します。コンマと連結の名前を省略して *lucreate* を実行すると、空いているボリュームが連結用に選択されます。
- lucreate* で作成できるのは、単一の物理スライスから成る連結だけです。このコマンドでは、1 つのミラーに 3 つまで連結を接続できません。
- *preserve* を指定すると、既存のファイルシステムとその内容を保存できます。このキーワードを使うと、ソースブート環境の内容をコピーする処理を省略できます。内容を保存することで、新しいブート環境を短時間で作成できます。特定のマウントポイントについて、*preserve* で指定できるのは 1 つの物理デバイスだけです。*preserve* を指定して *lucreate* コマンドを実行すると、指定したファイルシステムに対してデバイスの内容が適切かどうかを検査されます。この検査は限定的なものなので、適合性を保証することはできません。
- preserve* キーワードは、物理スライスと Solaris ボリュームマネージャのボリュームの両方に使用できます。
- UFS ファイルシステムが物理スライスに置かれている場合に *preserve* キーワードを使用すると、UFS ファイルシステムの内容がそのスライスに保存されます。次の *-m* オプションの例では、



preserve キーワードを使って、物理デバイス c0t0d0s0 の内容をルート (/) ファイルシステムとして保存します。

```
-m /:c0t0d0s0:preserve,ufs
```

- UFS ファイルシステムがボリュームに置かれている場合に preserve キーワードを使用すると、UFS ファイルシステムの内容がそのボリュームに保存されます。

次の -m オプションの例では、preserve キーワードを使って、RAID-1 ボリューム (ミラー) d10 の内容をルート (/) ファイルシステムとして保存します。

```
-m /:d10:preserve,ufs
```

次の -m オプションの例では、RAID-1 ボリューム (ミラー) d10 がルート (/) ファイルシステムとして構成されます。単一スライスの連結 d20 が現在のミラーから切り離されます。d20 がミラー d10 に接続されます。ルート(/) ファイルシステムは、サブミラー d20 に保持されます。

```
-m /:d10:mirror,ufs -m /:d20:detach,attach,preserve
```

**-n BE\_name**

作成するブート環境の名前。BE\_name 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。

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

**例 35-9** ミラーを持つブート環境の作成とデバイスの指定 (コマンド行)

この例では、ファイルシステムのマウントポイントを -m オプションで指定します。

- 「mydescription」という記述は、another\_disk に対応しています。
- lucreate コマンドにより、ルート (/) マウントポイントの UFS ファイルシステムが構成されます。d10 というミラーが作成されます。このミラー d10 に、現在のブート環境のルート (/) ファイルシステムがコピーされます。ミラー d10 にあるデータはすべて上書きされます。
- 2 つのスライス c0t0d0s0 および c0t1d0s0 は、サブミラー d1 および d2 として指定されています。これら 2 つのサブミラーは、ミラー d10 に追加されず。
- 新しいブート環境には another\_disk という名前が付けられます。

```
# lucreate -A 'mydescription' \  
-m /:/dev/md/dsk/d10:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0,d1:attach \  
-m /:/dev/dsk/c0t1c0s0,d2:attach -n another_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

例 35-10 ミラーを持つブート環境の作成とサブミラー名の省略(コマンド行)

この例では、ファイルシステムのマウントポイントを `-m` オプションで指定します。

- 「mydescription」という記述は、another\_disk に対応しています。
- `lucreate` コマンドにより、ルート (/) マウントポイントの UFS ファイルシステムが構成されます。d10 というミラーが作成されます。このミラー d10 に、現在のブート環境のルート (/) ファイルシステムがコピーされます。ミラー d10 にあるデータはすべて上書きされます。
- 2つのスライス `c0t1d0s0` および `c0t2d0s0` は、サブミラーとして指定されています。サブミラーを指定せずに `lucreate` コマンドを実行すると、利用可能なボリューム名の一覧から名前が選択されます。これら2つのサブミラーは、ミラー d10 に接続されます。
- 新しいブート環境には another\_disk という名前が付けられます。

```
# lucreate -A 'mydescription' \  
-m /:/dev/md/dsk/d10:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0:attach \  
-m /:/dev/dsk/c0t1d0s0:attach -n another_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第 36 章を参照してください。

例 35-11 ブート環境の作成とサブミラーの切り離し(コマンド行)

この例では、ファイルシステムのマウントポイントを `-m` オプションで指定します。

- 「mydescription」という記述は、another\_disk に対応しています。
- `lucreate` コマンドにより、ルート (/) マウントポイントの UFS ファイルシステムが構成されます。d10 というミラーが作成されます。
- スライス `c0t0d0s0` がその現在のミラーから切り離されます。このスライスはサブミラー d1 として指定され、ミラー d10 に追加されます。このサブミラーの内容であるルート (/) ファイルシステムは保存され、コピー処理は発生しません。スライス `c0t1d0s0` はサブミラー d2 として指定され、ミラー d10 に追加されます。
- 新しいブート環境には another\_disk という名前が付けられます。

```
# lucreate -A 'mydescription' \  
-m /:/dev/md/dsk/d10:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0,d1:detach,attach,preserve \  
-m /:/dev/dsk/c0t1d0s0,d2:attach -n another_disk
```

この例は、次の例のように簡略化できます。物理デバイスや論理デバイスの短縮名が使用されています。サブミラー d1 および d2 の指示子は省略されています。

```
# lucreate -A 'mydescription' \  
-m /:d10:ufs,mirror \  
-m /:c0t0d0s0:detach,attach,preserve \  

```

例 35-11 ブート環境の作成とサブミラーの切り離し (コマンド行) (続き)

```
-m /:c0t1d0s0:attach -n another_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

例 35-12 ブート環境の作成、サブミラーの切り離しと内容の保存 (コマンド行)

この例では、ファイルシステムのマウントポイントを `-m` オプションで指定します。

- 「mydescription」という記述は、another\_disk に対応しています。
- `lucreate` コマンドにより、ルート (/) マウントポイントの UFS ファイルシステムが構成されます。d20 というミラーが作成されます。
- スライス `c0t0d0s0` が現在のミラーから切り離され、ミラー `d20` に追加されます。サブミラーの名前は指定されていません。このサブミラーの内容であるルート (/) ファイルシステムは保存され、コピー処理は発生しません。
- 新しいブート環境には another\_disk という名前が付けられます。

```
# lucreate -A 'mydescription' \  
-m /:/dev/md/dsk/d20:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0:detach,attach,preserve \  
-n another_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

例 35-13 2つのミラーを持つブート環境の作成 (コマンド行)

この例では、ファイルシステムのマウントポイントを `-m` オプションで指定します。

- 「mydescription」という記述は、another\_disk に対応しています。
- `lucreate` コマンドにより、ルート (/) マウントポイントの UFS ファイルシステムが構成されます。d10 というミラーが作成されます。このミラー `d10` に、現在のブート環境のルート (/) ファイルシステムがコピーされます。ミラー `d10` にあるデータはすべて上書きされます。
- 2つのスライス `c0t0d0s0` および `c0t1d0s0` は、サブミラー `d1` および `d2` として指定されています。これら2つのサブミラーは、ミラー `d10` に追加されます。
- `lucreate` コマンドにより、`/opt` マウントポイントの UFS ファイルシステムが構成されます。d11 というミラーが作成されます。このミラー `d11` に、現在のブート環境の `/opt` ファイルシステムがコピーされます。ミラー `d11` にあるデータはすべて上書きされます。
- 2つのスライス `c2t0d0s1` および `c3t1d0s1` は、サブミラー `d3` および `d4` として指定されています。これら2つのサブミラーは、ミラー `d11` に追加されません。

例 35-13 2つのミラーを持つブート環境の作成 (コマンド行) (続き)

- 新しいブート環境には `another_disk` という名前が付けられます。

```
# lucreate -A 'mydescription' \  
-m /:/dev/md/dsk/d10:ufs,mirror \  
-m /:/dev/dsk/c0t0d0s0,d1:attach \  
-m /:/dev/dsk/c0t1d0s0,d2:attach \  
-m /opt:/dev/md/dsk/d11:ufs,mirror \  
-m /opt:/dev/dsk/c2t0d0s1,d3:attach \  
-m /opt:/dev/dsk/c3t1d0s1,d4:attach -n another_disk
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする (ブート可能な状態にする) ことができます。第 36 章を参照してください。

## ▼ ブート環境の作成と内容のカスタマイズ (コマンド行インタフェース)

次のオプションを使って、新しいブート環境のファイルシステムの内容を変更できます。ディレクトリやファイルは新しいブート環境にコピーされません。

1. スーパーユーザーとしてログインします。
2. 次のように入力して新しいブート環境を作成します。

```
# lucreate -m mountpoint:device[,metadevice]:fs_options [-m ...] \  
[-x exclude_dir] [-y include] [-Y include_list_file] [-f exclude_list_file] \  
[-z filter_list] [-I] -n BE_name
```

```
-m mountpoint:device[,metadevice]:fs_options [-m ...]
```

新しいブート環境のファイルシステム構成を `vfstab` で指定します。-m に引数として指定されるファイルシステムは、同じディスク上のファイルシステムでも、複数のディスク上のファイルシステムでも構いません。このオプションは、作成したいファイルシステムの数だけ使用します。

- `mountpoint` には、任意の有効なマウントポイント、またはスワップパーティションを示す - (ハイフン) を指定できます。
- `device` フィールドには、次のいずれかを指定できます。
  - ディスクデバイスの名前。 `/dev/dsk/cwt.xdysz` の形式で表されます。
  - Solaris ボリュームマネージャのボリューム名。 `/dev/md/dsk/dnum` の形式で表されます。
  - Veritas Volume Manager のボリューム名。 `/dev/md/vxfs/dsk/dnum` の形式で表されます。
  - キーワード `merged`。指定されたマウントポイントのファイルシステムがその親とマージされることを示します。
- `fs_options` フィールドには、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す

- `vxfst`: Veritas ファイルシステムを示す
- `swap`: スワップファイルシステムを示す。スワップマウントポイントはハイフン (-) で表す
- 論理デバイス (ミラー) であるファイルシステムについては、いくつかのキーワードを使って、そのファイルシステムに対して実行するアクションを指定できます。論理デバイスの作成、論理デバイスの構成変更、論理デバイスの削除などを行うキーワードがあります。これらのキーワードの詳細は、438 ページの「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。

**-x `exclude_dir`**

ファイルやディレクトリを除外して、新しいブート環境にコピーされないようにします。このオプションのインスタンスを複数使用して、複数のファイルまたはディレクトリを除外できます。

`exclude_dir` には、除外するディレクトリやファイルの名前を指定します。

**-y `include_dir`**

指定されたディレクトリやファイルを新しいブート環境にコピーします。ディレクトリは除外するが、個々のサブディレクトリやファイルを含める場合、このオプションを使用します。

`include_dir` には、含めるサブディレクトリやファイルの名前を指定します。

**-Y `list_filename`**

リスト内のディレクトリやファイルを新しいブート環境にコピーします。ディレクトリは除外するが、個々のサブディレクトリやファイルを含める場合、このオプションを使用します。

- `list_filename` は、リストを含むファイルのフルパスです。
- `list_filename` ファイルでは、1 ファイルを 1 行で表す必要があります。
- 行でディレクトリを指定すると、そのディレクトリの下にあるすべてのサブディレクトリとファイルが含まれます。行でファイルを指定すると、そのファイルだけが含まれます。

**-f `list_filename`**

リスト内のファイルやディレクトリを除外して、新しいブート環境にコピーされないようにします。

- `list_filename` は、リストを含むファイルのフルパスです。
- `list_filename` ファイルでは、1 ファイルを 1 行で表す必要があります。

**-z `list_filename`**

リスト内のディレクトリやファイルを新しいブート環境にコピーします。リスト内の各ファイルまたはディレクトリには、プラス「+」またはマイナス「-」記号を付けます。プラスはファイルやディレクトリを含めることを、マイナスはファイルやディレクトリを除外することを示します。

- `list_filename` は、リストを含むファイルのフルパスです。
- `list_filename` ファイルでは、1 ファイルを 1 行で表す必要があります。プラスまたはマイナスとファイル名との間には 1 つの半角スペースが必要です。

- 行でディレクトリとプラス (+) を指定すると、そのディレクトリの下にあるすべてのサブディレクトリとファイルがコピーされます。行でファイルとプラス (+) を指定すると、そのファイルだけがコピーされます。

-I

システムファイルの整合性チェックを無効にします。このオプションは慎重に使用してください。

重要なシステムファイルをブート環境から除外してしまうことを防ぐために、`lucreate` は整合性チェックを実行します。このチェックにより、システムパッケージデータベースに登録されたすべてのファイルが検査され、そのいずれかが除外されると、ブート環境の作成が停止します。このオプションを指定すると、この整合性チェックが無効になります。このオプションを指定すると、より短時間でブート環境を作成できますが、問題を検出できなくなる可能性があります。

-n *BE\_name*

作成するブート環境の名前。*BE\_name* 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。

例 35-14 ブート環境の作成とファイルの除外(コマンド行)

この例では、新しいブート環境に `second_disk` という名前が付けられます。ソースブート環境には、ファイルシステムとしてルート (/) があります。新しいブート環境では、`/var` ファイルシステムがルート (/) から分割され、別のスライスに置かれます。`lucreate` コマンドにより、ルート (/) マウントポイントと `/var` マウントポイントの UFS ファイルシステムが構成されます。また、2つの `/var` メールファイル `root` および `staff` は、新しいブート環境にコピーされません。ソースブート環境と新しいブート環境の間で自動的にスワップが共有されません。

```
# lucreate -n second_disk \  
-m /:/dev/dsk/c0t1d0s0:ufs -m /var/mail:c0t2d0s0:ufs \  
-x /var/mail/root -x /var/mail/staff
```

例 35-15 ブート環境を作成し、ファイルを除外または含める(コマンド行)

この例では、新しいブート環境に `second_disk` という名前が付けられます。ソースブート環境には、オペレーティング環境のファイルシステムとしてルート (/) があります。ソースブート環境には `/mystuff` というファイルシステムもあります。`lucreate` コマンドにより、ルート (/) マウントポイントと `/mystuff` マウントポイントの UFS ファイルシステムが構成されます。`/mystuff` の2つのディレクトリ `/latest` および `/backup` だけが、新しいブート環境にコピーされます。ソースブート環境と新しいブート環境の間で自動的にスワップが共有されません。

```
# lucreate -n second_disk \  
-m /:/dev/dsk/c01t0d0s0:ufs -m /mystuff:c1t1d0s0:ufs \  
-x /mystuff -y /mystuff/latest -y /mystuff/backup
```

新しいブート環境の作成が終わると、この環境をアップグレードしてアクティブにする(ブート可能な状態にする)ことができます。第 36 章を参照してください。

# Solaris Live Upgrade によるアップグレード (作業)

---

この章では、Solaris Live Upgrade を使用して非アクティブブート環境のアップグレードとアクティブ化を行う方法について説明します。また、アクティブ化によって発生した問題を解決する方法についても説明します。この章の内容は次のとおりです。

- 448 ページの「作業マップ: ブート環境のアップグレード」
- 448 ページの「ブート環境のアップグレード」
- 459 ページの「ブート環境へのフラッシュアーカイブのインストール」
- 464 ページの「ブート環境のアクティブ化」
- 468 ページの「問題の解決: 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)」

Solaris Live Upgrade は、メニューを介して使用することもコマンド行インタフェースで使用することもできます。以下の説明では、両方のインタフェースについて手順を説明しています。これらの手順では、Solaris Live Upgrade の使用に関する詳しい説明は省略しています。コマンドの詳細については、第 39 章 を参照してください。また、コマンド行インタフェースのオプションの詳細については、各マニュアルページを参照してください。

---

# 作業マップ: ブート環境のアップグレード

表 36-1 作業マップ: Solaris Live Upgrade によるアップグレード

作業	説明	参照先
ブート環境のアップグレードまたはフラッシュアーカイブのインストール	<ul style="list-style-type: none"><li>■ オペレーティング環境イメージを使用して非アクティブブート環境をアップグレードします。</li><li>■ 非アクティブブート環境へフラッシュアーカイブをインストールします。</li></ul>	<ul style="list-style-type: none"><li>■ 448 ページの「ブート環境のアップグレード」</li><li>■ 459 ページの「ブート環境へのフラッシュアーカイブのインストール」</li></ul>
非アクティブブート環境のアクティブ化	変更を有効にし、非アクティブブート環境をアクティブに切り替えます。	464 ページの「ブート環境のアクティブ化」
(任意) アクティブ化で問題が発生した場合のフォールバック	問題が発生する場合は元のブート環境をアクティブに戻します。	468 ページの「問題の解決: 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)」

---

## ブート環境のアップグレード

「Upgrade」メニューまたは `luupgrade` コマンドを使用してブート環境をアップグレードします。この節では、以下の場所に置かれているファイル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手順について説明します。

- ネットワークファイルシステム (NFS) サーバー
- ローカルファイル
- ローカルテープ
- ローカルデバイス (DVD または CD)

---

注 - インストールに複数の CD が必要な場合は、コマンド行インタフェースを使用してアップグレードする必要があります。451 ページの「複数の CD を使用してオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)」を参照してください。

---



## アップグレードのガイドライン

最新のオペレーティング環境へアップグレードする間、アクティブブート環境への影響はありません。新しいファイルは非アクティブブート環境のクリティカルファイルシステムとマージされますが、共有可能ファイルシステムは変更されません。

アップグレードを行う代わりに、フラッシュアーカイブを作成して、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。それらの新しいファイルは非アクティブブート環境のクリティカルファイルシステムを上書きしますが、共有可能ファイルシステムは変更されません。459 ページの「ブート環境へのフラッシュアーカイブのインストール」を参照してください。

物理ディスクスライス、Solaris ボリュームマネージャのボリューム、および Veritas Volume Manager のボリュームがどのような組み合わせで含まれているブート環境もアップグレードできます。ルート(/) ファイルシステム用のスライスは、RAID-1 ボリューム (ミラー) に含まれている単一スライスの連結でなければなりません。ミラー化されたファイルシステムを持つブート環境の作成手順については、438 ページの「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。

## Solaris Live Upgrade によるパッケージの追加と削除



---

注意 – Solaris Live Upgrade でパッケージやパッチのアップグレードや追加、削除を行うためには、パッケージやパッチが SVR4 パッケージ化ガイドラインに準拠していません。サンのパッケージはこのガイドラインに準拠していますが、サードパーティベンダーのパッケージがこれに準拠しているとは限りません。パッケージがガイドラインに準拠していないと、パッケージ追加ソフトウェアに問題が発生したり、場合によっては、アクティブなブート環境が変更されたりすることがあります。

---

Solaris Live Upgrade によるパッケージの追加や削除については、次の項目を参照してください。

- 計画情報については、400 ページの「Solaris Live Upgrade によるパッケージやパッチの管理」を参照してください。
- コマンドとオプションの詳細は、`luupgrade (1M)` のマニュアルページを参照してください。
- パッケージ化の要件については、付録 G を参照してください。

## ▼ ブート環境のオペレーティングシステムイメージをアップグレードする (キャラクタインタフェース)

この手順でアップグレードを行う場合は、1つのCD、DVD、または結合されたインストールイメージを使用する必要があります。複数のCDが必要なインストールでは、451ページの「複数のCDを使用してオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)」を参照してください。

1. **Solaris Live Upgrade** のメインメニューから「**Upgrade**」を選択します。  
「Upgrade」メニューが表示されます。
2. 新しいブート環境の名前を入力します。
3. **Solaris** インストールイメージが置かれている場所のパスを入力します。

インストールメディアの種類	説明
ネットワークファイルシステム	インストールイメージが置かれているネットワークファイルシステムのパスを指定します。
ローカルファイル	インストールイメージが置かれているローカルファイルシステムのパスを指定します。
ローカルテープ	インストールイメージが置かれているローカルテープデバイスとテープ上の位置を指定します。
ローカルデバイス、DVD、またはCD	ローカルデバイスと、インストールイメージのパスを指定します。

- DVD または 1 つの CD を使用する場合は、次の例のように、そのディスクへのパスを入力します。

```
Package Media: /cdrom/Solaris_9/s0
```

- 1 つに結合されたイメージがネットワーク上に存在する場合は、次の例のようにそのネットワークファイルシステムのパスを入力します。

```
Package Media: /net/installmachine/export/Solaris_9/os_image
```

4. **F3** を押してアップグレードします。
5. アップグレードが完了すると、メインメニューが表示されます。

## ▼ ブート環境のオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)

この手順でアップグレードを行う場合は、1つのCD、DVD、または結合されたインストールイメージを使用する必要があります。複数のCDが必要なインストールでは、451ページの「複数のCDを使用してオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)」を参照してください。

1. スーパーユーザーとしてログインします。
2. 次のように入力して、アップグレードするブート環境とインストールソフトウェアのパスを指定します。

```
# luupgrade -u -n BE_name -s os_image_path
-u                ブート環境のオペレーティングシステムイメージをアップグ
                  レードします。
-n BE_name        アップグレード対象のブート環境の名前を指定します。
-s os_image_path オペレーティングシステムイメージが置かれているディレク
                  トリのパス名を指定します。
```

例 36-1 ブート環境のオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)

この例では、second\_disk ブート環境をアップグレードします。

```
# luupgrade -u -n second_disk \
-s /net/installmachine/export/Solaris_9/OS_image
```

## ▼ 複数のCDを使用してオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)

オペレーティングシステムイメージが複数のCDに入っている場合は、このアップグレード方法を使用してください。

1. スーパーユーザーとしてログインします。
2. 次のように入力して、アップグレードするブート環境とインストールソフトウェアのパスを指定します。

```
# luupgrade -u -n BE_name -s os_image_path
-u                ブート環境のオペレーティングシステムイメージをアップグ
                  レードします。
```

- n *BE\_name*           アップグレード対象のブート環境の名前を指定します。
- s *os\_image\_path*   オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。

3. 1 枚目の CD のインストーラ処理が完了したら、2 枚目の CD を挿入します。
4. この手順は前述のものと同じですが、**-u** オプションではなく **-i** オプションを使用します。メニューまたはテキストモードで、2 枚目の CD 上のインストーラを実行するように選択してください。

- 次のコマンドは、メニューを使用して 2 枚目の CD 上のインストーラを実行します。

```
# luupgrade -i -n BE_name -s os_image_path
```

- 次のコマンドは、テキストモードで 2 枚目の CD 上のインストーラを実行します。ユーザーに入力を求めるプロンプトは表示されません。

```
# luupgrade -i -n BE_name -s os_image_path -O '-nodisplay -noconsole'
```

- i                           指定されたメディア上のインストールプログラムを探し、そのプログラムを実行します。インストーラプログラムは、**-s** で指定します。
- n *BE\_name*               アップグレード対象のブート環境の名前を指定します。
- s *os\_image\_path*       オペレーティングシステムイメージが置かれているディレクトリのパス名を指定します。
- O '-nodisplay -noconsole' (任意) テキストモードで 2 枚目の CD 上のインストーラを実行します。ユーザーに入力を求めるプロンプトは表示されません。

このようにして、ブート環境をアクティブにする準備が整います。464 ページの「ブート環境のアクティブ化」を参照してください。

例 36-2 SPARC: 複数の CD によりオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース)

この例では、`second_disk` ブート環境をアップグレードします。インストールイメージは、Solaris SOFTWARE 1 of 2 と Solaris SOFTWARE 2 of 2 CD に入っています。`-u` オプションでは、CD セットのすべてのパッケージを格納するだけの領域があるかどうかを判定します。`-O` オプションと `-nodisplay` および `-noconsole` オプションを指定すると、2 枚目の CD の読み取りの後にキャラクターユーザーインタフェースは表示されません。したがって、入力情報のプロンプトは表示されません。このインタフェースを表示する場合は、これらのオプションを省略します。

Solaris SOFTWARE 1 of 2 CD を挿入し、次のように入力します。

```
# luupgrade -u -n second_disk -s /dev/cdrom/cdrom0/s0
```

例 36-2 SPARC: 複数の CD によりオペレーティングシステムイメージをアップグレードする (コマンド行インタフェース) (続き)

Solaris SOFTWARE 2 of 2 CD を挿入し、次のように入力します。

```
# luupgrade -i -n second_disk -s /dev/cdrom/cdrom0 -O '-nodisplay \
-noconsole'
```

---

注 - ほかの CD がある場合は、luupgrade コマンドに -i オプションを指定してそれらの CD をインストールします。

---

## ▼ Solaris Live Upgrade で使用されるプロファイルを作成する

ここでは、Solaris Live Upgrade で使用されるプロファイルの作成方法について説明します。luupgrade コマンドに -j オプションを指定してこのプロファイルを使用し、非アクティブブート環境をアップグレードできます。

このプロファイルの使用方法については、次の 2 つのセクションを参照してください。

- プロファイルを使用したアップグレードについては、458 ページの「プロファイルを使用して Solaris Live Upgrade でアップグレードする (コマンド行インタフェース)」を参照してください。
  - プロファイルを使用したフラッシュのインストールについては、462 ページの「プロファイルを使用した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。
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。プロファイルの名前は、システムに Solaris ソフトウェアをインストールするためにそのプロファイルをどのように使用するかを示すものにしてください。たとえば、このプロファイルに upgrade\_Solaris\_9 という名前を付けます。
  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。  
Solaris Live Upgrade プロファイルで使用できるのは、次の表に示されたアップグレードキーワードだけです。
    - upgrade または flash\_install の Install\_type キーワード値を使ったインストールについては、表 36-2 を参照してください。
    - flash\_update の Install\_type キーワード値を使ったインストールについては、表 36-3 を参照してください。

表 36-2 Solaris Live Upgrade プロファイルのキーワードと説明

キーワード	説明
(必須) <code>Install_type</code>	<p>システムの既存の Solaris 環境をアップグレードするか、システムに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。このキーワードには、次の値を使用します。</p> <ul style="list-style-type: none"> <li>■ アップグレードの場合は、<code>upgrade</code></li> <li>■ フラッシュインストールの場合は、<code>flash_install</code></li> <li>■ フラッシュ差分インストールの場合は、<code>flash_update</code></li> </ul>
(フラッシュアーカイブの場合は必須) <code>archive_location</code>	<p>指定された位置からフラッシュアーカイブを取得します。このキーワードで使用可能な値のリストについては、342 ページの「<code>archive_location</code> プロファイルキーワード」を参照してください。</p>
(省略可能) <code>cluster</code> (クラスタの追加または削除)	<p>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対してクラスタを追加するか、または削除するかを指定します。このキーワードで使用可能な値のリストについては、352 ページの「<code>cluster</code> プロファイルキーワード (ソフトウェアグループの追加)」を参照してください。</p>
(省略可能) <code>geo</code>	<p>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。このキーワードで使用可能な値のリストについては、360 ページの「<code>geo</code> プロファイルキーワード」を参照してください。</p>
(省略可能) <code>isa_bits</code>	<p>64 ビットの Solaris 9 パッケージと 32 ビットの Solaris 9 パッケージのどちらをインストールするかを指定します。このキーワードで使用可能な値のリストについては、362 ページの「<code>isa_bits</code> プロファイルキーワード」を参照してください。</p>
(省略可能) <code>local_customization</code>	<p>クローンシステムに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。<code>local_customization</code> 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。この値は、クローンシステム上のスクリプトへのパスです。配置前および配置後スクリプトの詳細は、215 ページの「カスタムスクリプトの作成」を参照してください。</p>
(省略可能) <code>locale</code>	<p>インストールまたはアップグレード時に追加するロケールパッケージを指定します。このキーワードで使用可能な値のリストについては、364 ページの「<code>locale</code> プロファイルキーワード」を参照してください。</p>
(省略可能) <code>package</code>	<p>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対してパッケージを追加するか、または削除するかを指定します。このキーワードで使用可能な値のリストについては、366 ページの「<code>package</code> プロファイルキーワード」を参照してください。</p>

表 36-3 差分アーカイブを作成するための Solaris Live Upgrade プロファイルキーワードと説明

キーワード	説明
(必須) <code>Install_type</code>	システムに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。差分アーカイブを表す値は <code>flash_update</code> です。
(必須) <code>archive_location</code>	指定された位置からフラッシュアーカイブを取得します。このキーワードで使用可能な値のリストについては、342 ページの「 <code>archive_location</code> プロファイルキーワード」を参照してください。
(省略可能) <code>forced_deployment</code>	ソフトウェアで想定されているものとは異なるクローンシステムに、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。 <code>forced_deployment</code> を使用すると、クローンシステムをソフトウェアが想定してい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。ファイルが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。
(省略可能) <code>local_customization</code>	クローンシステムに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。 <code>local_customization</code> 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。この値は、クローンシステム上のスクリプトへのパスです。配置前および配置後スクリプトの詳細は、215 ページの「カスタムスクリプトの作成」を参照してください。
(省略可能) <code>no_content_check</code>	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、 <code>no_content_check</code> 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にある場合を除き、このキーワードの使用は避けてください。
(省略可能) <code>no_master_check</code>	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、 <code>no_master_check</code> キーワードを使用してファイルの検証を省略できます。クローンシステムのファイルの検証は行われません。この検証により、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てい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にある場合を除き、このキーワードの使用は避けてください。

3. プロファイルをローカルシステムのディレクトリに保存します。
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) プロファイル进行测试します。

プロファイルのテスト方法については、457 ページの「Solaris Live Upgrade で使用されるプロファイル进行测试する」を参照してください。

例 36-3 Solaris Live Upgrade プロファイルの作成

この例では、プロファイルはアップグレードパラメータを提供します。このプロファイルでは、Solaris Live Upgrade の `luupgrade` コマンドに `-u` および `-j` オプションを指定して、非アクティブなブート環境をアップグレードします。このプロファイルでは、パッケージおよびクラスタを追加します。地域ロケールおよび追加ロケールも、プロファイルに追加されます。プロファイルにロケールを追加する場合、ブート環境がディスク容量に余裕を持って作成されていることを確認してください。

```
# profile keywords      profile values
# -----
install_type           upgrade
package                SUNWxwman add
cluster                SUNWCacc add
geo                    C_Europe
locale                 zh_TW
locale                 zh_TW.BIG5
locale                 zh_TW.UTF-8
locale                 zh_HK.UTF-8
locale                 zh_HK.BIG5HK
locale                 zh
locale                 zh_CN.GB18030
locale                 zh_CN.GBK
locale                 zh_CN.UTF-8
```

例 36-4 差分アーカイブをインストールするための Solaris Live Upgrade プロファイルの作成

次のプロファイルの例は、Solaris Live Upgrade でクローンシステムに差分アーカイブをインストールするためのものです。差分アーカイブで指定されているファイルだけが、追加、削除、または変更されます。フラッシュアーカイブは、NFS サーバーから取得されます。イメージは元のマスターシステムから構築されたものなので、クローンシステムのイメージの妥当性検査は行われません。このプロファイルは、Solaris Live Upgrade の `luupgrade` コマンドの `-u` オプションと `-j` オプションで使用します。

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

`luupgrade` コマンドによる差分アーカイブのインストールについては、462 ページの「プロファイルを使用した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。



## ▼ Solaris Live Upgrade で使用されるプロファイル进行测试する

プロファイルを作成したら、luupgrade コマンドを使用してプロファイル进行测试します。luupgrade が生成するインストール出力を調べることによって、意図したとおりにプロファイルが動作するかを簡単に調べることができます。

### ● プロファイル进行测试する

```
# luupgrade -u -n BE_name -D -s os_image_path -j profile_path
```

-u	ブート環境のオペレーティングシステムイメージをアップグレードします。
-n BE_name	アップグレード対象のブート環境の名前を指定します。
-D	luupgrade コマンドは、選択されたブート環境のディスク構成を使用して、-j オプションと共に渡されたプロファイルオプション进行测试します。
-s os_image_path	オペレーティングシステムイメージを含むディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。
-j profile_path	アップグレード用に構成されたプロファイルのパス。プロファイルは、ローカルマシンのディレクトリに存在する必要があります。

### 例 36-5 Solaris Live Upgrade を使用したプロファイルのテスト

次の例では、プロファイル名は upgrade\_Solaris\_9 です。このプロファイルは ulb08 という名前の非アクティブブート環境でのテストに成功します。

```
# luupgrade -u -n ulb08 -D -s /net/installsvr/export/u1/combined.ulwos \  
-j /var/tmp/flash_profile  
Validating the contents of the media /net/installsvr/export/u1/combined.ulwos.  
The media is a standard Solaris media.  
The media contains an operating system upgrade image.  
The media contains Solaris version 9.  
Locating upgrade profile template to use.  
Locating the operating system upgrade program.  
Checking for existence of previously scheduled Live Upgrade requests.  
Creating upgrade profile for BE ulb08.  
Determining packages to install or upgrade for BE ulb08.  
Simulating the operating system upgrade of the BE ulb08.  
The operating system upgrade simulation is complete.  
INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of the  
upgrade operation.  
INFORMATION: var/sadm/system/data/upgrade_cleanup contains a log of  
cleanup operations required.  
The Solaris upgrade of the boot environment ulb08 is complete.
```

これで、プロファイルを使用して非アクティブブート環境をアップグレードできません。

## ▼ プロファイルを使用して Solaris Live Upgrade でアップグレードする (コマンド行インタフェース)

ここでは、プロファイルを使用したオペレーティング環境のアップグレード方法を、手順を追って説明します。

プロファイルを使用してフラッシュアーカイブをインストールする場合は、462 ページの「プロファイルを使用した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。

プロファイルにロケールを追加した場合、ブート環境がディスク容量に余裕を持って作成されていることを確認してください。

1. スーパーユーザーとしてログインします。

2. プロファイルを作成します。

Solaris Live Upgrade プロファイルで使用可能なアップグレードキーワードのリストについては、表 36-2 を参照してください。

3. 次のコマンドを入力します。

```
# luupgrade -u -n BE_name -s os_image_path -j profile_path
```

-u                   ブート環境のオペレーティングシステムイメージをアップグレードします。

-n *BE\_name*           アップグレード対象のブート環境の名前を指定します。

-s *os\_image\_path*    オペレーティングシステムイメージを含むディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。

-j *profile\_path*      プロファイルへのパス。プロファイルは、ローカルマシンのディレクトリに存在する必要があります。プロファイル作成についての詳細は、453 ページの「Solaris Live Upgrade で使用されるプロファイルを作成する」を参照してください。

このようにして、ブート環境をアクティブにする準備が整います。

例 36-6 カスタム JumpStart プロファイルを使用したブート環境のアップグレード (コマンド行インタフェース)

この例では、アーカイブは `second_disk` ブート環境にインストールされます。 `-j` オプションを使用して、プロファイルにアクセスします。このようにして、ブート環境をアクティブにする準備が整います。プロファイルを作成する場合は、453 ページの「Solaris Live Upgrade で使用されるプロファイルを作成する」を参照してください。

```
# luupgrade -u -n second_disk \  
-s /net/installmachine/export/solarisX/OS_image \  
-j /var/tmp/profile
```

---

## ブート環境へのフラッシュアーカイブのインストール

この節では、以下の場所に格納されているフラッシュアーカイブを、Solaris Live Upgrade を使用してインストールする手順を説明します。

- HTTP サーバー
- FTP サーバー - このパスは、コマンド行だけから使用します。
- ネットワークファイルシステム (NFS) サーバー
- ローカルファイル
- ローカルテープ
- ローカルデバイス (DVD または CD)

---

注 - アーカイブ記憶域に関連するパスの構文例については、342 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。

---

フラッシュアーカイブをインストールすると、新しいブート環境上に存在する共有ファイル以外のすべてのファイルが上書きされます。

フラッシュのインストール機能を使用するには、マスターシステムをインストールし、フラッシュアーカイブを作成しておく必要があります。フラッシュの詳細は、第 19 章を参照してください。フラッシュアーカイブを作成するには、次の手順を参照してください。

- 220 ページの「初期インストール用フラッシュアーカイブの作成方法」
- 225 ページの「更新されたマスターイメージを使用してフラッシュ差分アーカイブを作成する方法」
- 228 ページの「Solaris Live Upgrade を使用してフラッシュ差分アーカイブを作成する方法」

## ▼ ブート環境へのフラッシュアーカイブのインストール (キャラクタインタフェース)

1. **Solaris Live Upgrade** のメインメニューから「**Flash**」を選択します。  
「Flash an Inactive Boot Environment」メニューが表示されます。
2. フラッシュアーカイブをインストールするブート環境の名前と、インストールメディアの場所を入力します。

ブート環境の名前: **Solaris\_9**

Package media: **/net/install-svr/export/Solaris\_9/latest**

3. アーカイブを追加するために **F1** を押します。  
「Archive Selection」サブメニューが表示されます。

```
Location          - Retrieval Method
<No Archives added> - Select ADD to add archives
```

このメニューでは、アーカイブのリストを作成できます。アーカイブの追加または削除を行うには、次の手順を実行します。

- a. アーカイブをリストに追加する場合は **F1** を押します。  
「Select Retrieval Method」サブメニューが表示されます。

```
HTTP
NFS
Local File
Local Tape
Local Device
```

- b. 「**Select Retrieval Method**」メニューで、フラッシュアーカイブの場所を選択します。

選択された媒体	プロンプト
HTTP	フラッシュアーカイブのアクセスに必要となる URL 情報とプロキシ情報を指定します。
NFS	フラッシュアーカイブが置かれているネットワークファイルシステムのパスを指定します。アーカイブのファイル名も指定できます。
ローカルファイル	フラッシュアーカイブが置かれているローカルファイルシステムのパスを指定します。
ローカルテープ	フラッシュアーカイブが置かれているローカルテープドライブとテープ上の位置を指定します。
ローカルデバイス	フラッシュアーカイブが置かれているローカルデバイス、パス、ファイルシステムの種類を指定します。

次のような「Retrieval」サブメニューが表示されます。表示は選択されたメディアによって異なります。

NFS Location:

- c. 次の例のように、アーカイブのパスを入力します。

NFS Location: **host:/path/to archive.flar**

- d. **F3** を押してリストにアーカイブを追加します。

- e. (省略可能) アーカイブをリストから削除する場合は **F2** を押します。

- f. インストールしたいアーカイブがリストに含まれた時点で **F6** を押して終了します。

4. **F3** を押して **1** つまたは複数のアーカイブをインストールします。

フラッシュアーカイブがブート環境にインストールされます。ブート環境上のファイルは、共有可能ファイルを除きすべて上書きされます。

このようにして、ブート環境をアクティブにする準備が整います。465 ページの「ブート環境のアクティブ化 (キャラクタインタフェース)」を参照してください。

## ▼ ブート環境への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)

1. スーパーユーザーとしてログインします。

2. 次のコマンドを入力します。

```
# luupgrade -f -n BE_name -s os_image_path -a archive
```

-f オペレーティングシステムをフラッシュアーカイブからインストールすることを示します。

-n *BE\_name* アーカイブを使用してインストールするブート環境の名前を指定します。

-s *os\_image\_path* オペレーティングシステムイメージを含むディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。

-a *archive* フラッシュアーカイブへのパス (ローカルファイルシステムでそのアーカイブが利用できる場合)。-s オプションと -a オプションを使用して指定するオペレーティングシステムイメージのバージョンは、同じでなければなりません。

例 36-7 ブート環境へフラッシュアーカイブをインストールする (コマンド行インタフェース)

この例では、アーカイブは `second_disk` ブート環境にインストールされます。アーカイブはローカルシステムに存在します。-s および -a オプションで指定するオペレーティングシステムバージョンは、どちらも Solaris 9 リリースです。`second_disk` 上のファイルは、共有可能ファイルを除いてすべて上書きされます。

```
# luupgrade -f -n second_disk \  
-s /net/installmachine/export/Solaris_9/OS_image \  
-a /net/server/archive/Solaris_9
```

このようにして、ブート環境をアクティブにする準備が整います。

## ▼ プロファイルを使用した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)

ここでは、プロファイルを使用してフラッシュアーカイブまたは差分アーカイブをインストールする手順を説明します。

プロファイルにロケールを追加した場合、ブート環境がディスク容量に余裕を持って作成されていることを確認してください。

1. スーパーユーザーとしてログインします。

2. プロファイルを作成します。

Solaris Live Upgrade プロファイルで使用可能なキーワードのリストについては、表 36-2を参照してください。

3. 次のコマンドを入力します。

```
# luupgrade -f -n BE_name -s os_image_path -j profile_path
```

-f	オペレーティングシステムをフラッシュアーカイブからインストールすることを示します。
-n BE_name	アップグレード対象のブート環境の名前を指定します。
-s os_image_path	オペレーティングシステムイメージを含むディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。
-j profile_path	フラッシュインストール用に構成されたプロファイルへのパス。プロファイルは、ローカルマシンのディレクトリに存在する必要があります。

このようにして、ブート環境をアクティブにする準備が整います。

例 36-8 プロファイルを使用してブート環境にフラッシュアーカイブをインストールする (コマンド行インタフェース)

この例では、インストールするアーカイブの場所をプロファイルで指定していません。

```
# profile keywords          profile values
# -----
install_type                flash_install
archive_location            nfs installserver:/export/solaris/flasharchive/solarisarchive
```

プロファイルを作成した後、luupgrade コマンドを実行してアーカイブをインストールできます。-j オプションを使用して、プロファイルにアクセスします。

```
# luupgrade -f -n second_disk \
-s /net/installmachine/export/solarisX/OS_image \
-j /var/tmp/profile
```

このようにして、ブート環境をアクティブにする準備が整います。プロファイルを作成する場合は、453 ページの「Solaris Live Upgrade で使用されるプロファイルを作成する」を参照してください。

## ▼ プロファイルキーワードを使用した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)

ここでは、プロファイルファイルではなく、archive\_location キーワードをコマンド行で使用して、フラッシュアーカイブをインストールする手順を説明します。プロファイルファイルを使用せずに、アーカイブを取得できます。

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# luupgrade -f -n BE_name -s os_image_path -J 'profile'
```

-f                    オペレーティングシステムをフラッシュアーカイブからアップグレードすることを示します。

-n BE\_name            アップグレード対象のブート環境の名前を指定します。

-s os\_image\_path      オペレーティングシステムイメージを含むディレクトリのパス名を指定します。このディレクトリはインストールメディア (DVD-ROM や CD-ROM など) 上でも NFS または UFS ディレクトリであってもかまいません。

-J 'profile'           archive\_location プロファイルキーワードを指定します。キーワードの値については、342 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。

このようにして、ブート環境をアクティブにする準備が整います。

例 36-9 プロファイルキーワードを使用した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)

この例では、アーカイブは `second_disk` ブート環境にインストールされます。-J オプションと `archive_location` キーワードを使用して、アーカイブを取得します。`second_disk` 上のファイルは、共有可能ファイルを除いてすべて上書きされます。

```
# luupgrade -f -n second_disk \  
-s /net/installmachine/export/solarisX/OS_image \  
-J 'archive_location http://example.com/myflash.flar'
```

---

## ブート環境のアクティブ化

ブート環境をアクティブにすると、次のシステムリブートでブート可能になります。新しいアクティブブート環境で何か問題が発生する場合は、元のブート環境に戻すことができます。詳細は、468 ページの「問題の解決: 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)」を参照してください。

ブート環境を正常にアクティブにするためには、そのブート環境が以下の条件を満たしている必要があります。

- ブート環境のステータスは「complete」でなければなりません。ステータスを確認するには、476 ページの「すべてのブート環境のステータスの表示」を参照してください。
- 現在のブート環境とは別のブート環境をアクティブにする場合は、`lumount (1M)` または `mount (1M)` を使用してそのブート環境のパーティションをマウントすることはできません。
- 比較処理で使用中のブート環境はアクティブにできません。481 ページの「ブート環境の比較」を参照してください。

---

注 - スワップを再構成する場合は、非アクティブブート環境をブートする前に実行してください。デフォルトでは、すべてのブート環境が同じスワップデバイスを共有します。スワップを再構成する場合は、「新しいブート環境を作成する」、手順 9、または 430 ページの「ブート環境を作成しスワップを再構成する (コマンド行インタフェース)」を参照してください。

---



## ▼ x86: (省略可能) アクティブ化の前にブート用フロッピーディスクを更新する

システムのブートに Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) フロッピーディスクを使用しない場合は、この手順は省略できます。Device Configuration Assistant を使用してブートする場合は、ブート用フロッピーディスクを更新する必要があります。この手順では、既存のフロッピーディスクを上書きするか、あるいは新しいフロッピーディスクに書き込むことにより、使用中のリリースと一致するようにブート用フロッピーディスクを更新します。

1. **Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版)** フロッピーディスクを挿入します。これは、上書きされる既存のフロッピーディスクまたは新しいフロッピーディスクのいずれかです。
2. このリリース用の最新のイメージに、ブートフロッピーディスクを更新します。
3. ブートフロッピーディスクを取り出します。
4. 次のコマンドを入力します。

```
volcheck
```

5. このフロッピーディスクに新しいブート環境の **boot/solaris/bootenv.rc** ファイルをコピーします。

```
cp /a/boot/solaris/bootenv.rc /floppy/floppy0/solaris/bootenv.rc
```

6. フロッピーディスク上の入力デバイスと出力デバイスをチェックして、それらが正しいことを確認します。正しくない場合はそれらを更新してください。  
このようにして、新しいブート環境をアクティブにする準備が整います。

## ▼ ブート環境のアクティブ化 (キャラクタインタフェース)

新しく作成したブート環境で初めてブートする時に、Solaris Live Upgrade は新しいブート環境と以前のアクティブブート環境の同期をとります。ここでいう「同期」とは、前にアクティブであったブート環境にあるクリティカルなシステムファイルやディレクトリを、ブートされているブート環境にコピーすることです。Solaris Live Upgrade では、強制的に同期を行うようにプロンプトで指定しない限り、2 回目以降のブート時には同期は行われません。

同期の詳細については、409 ページの「ブート環境間でのファイルの同期」を参照してください。

1. **Solaris Live Upgrade** のメインメニューで「**Activate**」を選択します。

2. アクティブにするブート環境の名前を入力します。

```
Name of Boot Environment: Solaris_9  
Do you want to force a Live Upgrade sync operations: no
```

3. ファイルの同期を実行することも、実行せずに処理を続けることもできます。

- **Return** キーを押して続けます。  
ブート環境の最初のブートでは、ファイルの同期が自動的に行われます。
- ファイルの同期を実行できますが、この機能は慎重に使用してください。同期するファイルに、それぞれのブート環境のオペレーティングシステムが対応していなければなりません。ファイルの同期を実行する場合は、次のように入力します。

```
Do you want to force a Live Upgrade sync operations: yes
```



---

注意 - 以前のアクティブブート環境で発生した変更によりユーザーが気付いていない場合や、それらの変更を制御できない場合もあるため、同期を使用する際には十分注意してください。たとえば、現在のブート環境で Solaris 9 ソフトウェアを実行しているとします。この場合、Solaris 7 リリースをブートし、強制的な同期を実行すると、Solaris 7 リリースのファイルは変更されることがあります。ファイルはオペレーティング環境のリリースに依存しているため、Solaris 7 リリースのブートは失敗することがあります。Solaris 9 のファイルと Solaris 7 のファイルは互換性があるとは限らないからです。

---

4. **F3** を押して、アクティブ化の処理を開始します。
5. **Return** キーを押して続けます。  
新しいブート環境は、次のリブート時にアクティブになります。
6. 非アクティブブート環境をリブートしてアクティブにします。

```
# init 6
```

## ▼ ブート環境をアクティブにする (コマンド行インタフェース)

1. スーパーユーザーとしてログインします。
2. 次のように入力して、ブート環境をアクティブにします。

```
# /usr/sbin/luactivate BE_name  
BE_name   アクティブにするブート環境の名前を指定します。
```

3. リブートします。

```
# init 6
```



---

注意 - リブートには `init` か `shutdown` コマンドを使用してください。 `reboot` や `halt`、`uadmin` コマンドを使用すると、ブート環境の切り替えは行われません。最後にアクティブであったブート環境が再びブートされます。

---

例 36-10 ブート環境をアクティブにする (コマンド行インタフェース)

この例では、次のリブート時に `second_disk` ブート環境がアクティブになります。

```
# /usr/sbin/luactivate second_disk
# init 6
```

## ▼ ブート環境をアクティブにしてファイルを同期させる (コマンド行インタフェース)

新しく作成したブート環境で初めてブートする時に、Solaris Live Upgrade は新しいブート環境と以前のアクティブブート環境の同期をとります。ここでいう「同期」とは、前にアクティブであったブート環境にあるクリティカルなシステムファイルやディレクトリを、ブートされているブート環境にコピーすることです。Solaris Live Upgrade では、強制的に同期を行うように `luactivate` コマンドの `-s` オプションで指定しない限り、2 回目以降のブート時には同期は行われません。

同期の詳細については、409 ページの「ブート環境間でのファイルの同期」を参照してください。

1. スーパーユーザーとしてログインします。
2. 次のように入力して、ブート環境をアクティブにします。

```
# /usr/sbin/luactivate -s BE_name
```

`-s` 最後アクティブであったブート環境と新しいブート環境のファイルを同期します。ブート環境の最初のアクティブ化ではファイル間の同期が行われますが、それ以降のアクティブ化では `-s` オプションを指定しない限りファイルの同期は行われません。



---

注意 - 以前のアクティブブート環境で発生した変更によりユーザーが気付いていない場合や、それらの変更を制御できない場合もあるため、このオプションを使用する際には十分注意してください。たとえば、現在のブート環境で Solaris 9 ソフトウェアを実行しているとします。この場合、Solaris 7 リリースをブートし、強制的な同期を実行すると、Solaris 7 リリースのファイルは変更されることがあります。ファイルはオペレーティング環境のリリースに依存しているため、Solaris 7 リリースのブートは失敗することがあります。Solaris 9 のファイルと Solaris 7 のファイルは互換性があるとは限らないからです。

---

`BE_name` アクティブにするブート環境の名前を指定します。

### 3. リブートします。

```
# init 6
```

例 36-11 ブート環境をアクティブにする (コマンド行インタフェース)

この例では、次のリブート時に `second_disk` ブート環境がアクティブになり、ファイルの同期がとられます。

```
# /usr/sbin/luactivate -s second_disk
# init 6
```

---

## 問題の解決: 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)

アップグレードの後に障害が検出されたり、アップグレードされたコンポーネントにアプリケーションが対応できない場合は、次の手順のどれか (プラットフォームによって異なる) を使って元のブート環境にフォールバックする必要があります。SPARC ベースシステムの場合は、最初の手順として `luactivate` を実行して元のブート環境に戻します。この処理が失敗した場合は、2 つ目の手順によって DVD-ROM、CD-ROM、またはネットイメージからブートしてください。x86 ベースシステムの場合は、`root (/)` ファイルシステムが同じ物理ディスクにマウントされているか、別の物理ディスクにマウントされているかに基づいて手順を選択します。

## ▼ 新しいブート環境の作成に成功した場合のフォールバック

新しいブート環境のアクティブ化に成功したが、その結果に満足できない場合は、この手順を使用します。

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# /usr/sbin/luactivate BE_name
```

*BE\_name* アクティブにするブート環境の名前を指定します。

3. リブートします。

```
# init 6
```

前の稼働ブート環境がアクティブブート環境になります。

## ▼ SPARC: ブート環境のアクティブ化に失敗した場合のフォールバック

- 新しいブート環境のブートに失敗した場合で、元のブート環境をシングルユーザーモードでブートできる場合は、この手順を使って元のブート環境にフォールバックします。
- メディアやネットインストールイメージからブートする場合は、470 ページの「SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする」を参照してください。

1. **OK** プロンプトで、**Solaris DVD**、**Solaris INSTALLATION CD**、ネットワーク、またはローカルディスクからマシンをシングルユーザーモードで起動します。

```
OK boot cdrom -s
```

または、

```
OK boot net -s
```

または、

```
OK boot device_name -s
```

*device\_name* システムをブートするデバイスの名前を指定します (たとえば、`/dev/dsk/c0t0d0s0`)。

2. 次のコマンドを入力します。

```
# /sbin/luactivate BE_name
```

*BE\_name* アクティブにするブート環境の名前を指定します。

- このコマンドを実行した結果プロンプトが表示されない場合は、470 ページの「SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする」へ進みます。
- プロンプトが表示される場合は、次の手順に進んでください。

3. プロンプトに対して、次のコマンドを入力します。

```
Do you want to fallback to activate boot environment <disk name>
(yes or no)? yes
```

フォールバックによるアクティブ化が正常に行われたことを示すメッセージが表示されます。

4. リブートします。

```
# init 6
```

前の稼働ブート環境がアクティブブート環境になります。

## ▼ SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする

DVD、CD、またはネットワークインストールイメージからブートする場合は、この手順を使用します。この場合、以前のアクティブブート環境からルート (/) スライスをマウントする必要があります。その後で、ブート環境の切り替えを行う `luactivate` コマンドを実行してください。リブートすると、1つ前のアクティブブート環境が再び起動されます。

1. **OK** プロンプトで、**Solaris DVD**、**Solaris INSTALLATION CD**、ネットワーク、またはローカルディスクからマシンをシングルユーザーモードで起動します。

```
OK boot cdrom -s
```

または、

```
OK boot net -s
```

または、

```
OK boot device_name -s
```

`device_name` オペレーティングシステムのコピーが格納されているディスクおよびスライスの名前を指定します (たとえば、`/dev/dsk/c0t0d0s0`)。

2. 必要に応じて、フォールバックブート環境のルート (/) が置かれているファイルシステムの完全性を確認します。

```
# fsck device_name
```

- device\_name*    フォールバックしたい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名前を指定します。デバイス名の形式は、`/dev/dsk/cwtxdysz` です。
3. アクティブブート環境のルート (/) スライスディレクトリ (`/mnt` など) にマウントします。
 

```
# mount device_name /mnt
```

*device\_name*    フォールバックしたい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名前を指定します。デバイス名の形式は、`/dev/dsk/cwtxdysz` です。
  4. アクティブブート環境のルート (/) スライスから、次のように入力します。
 

```
# /mnt/sbin/luactivate
```

前の稼働ブート環境がアクティブになり、結果が示されます。
  5. `/mnt` をマウント解除します。
 

```
# umount /mnt
```
  6. リブートします。
 

```
# init 6
```

前の稼働ブート環境がアクティブブート環境になります。

## ▼ x86: 別のディスクに存在するブート環境をフォールバックする

新しいブート環境のブート中に障害に遭遇する場合があります。ブート環境のルート (/) ファイルシステムが別の物理ディスク上に存在する場合、次の手順を実行して元のブート環境にフォールバックします。

1. システムをリブートし、該当する **BIOS** メニューに入ります。
  - ブートデバイスが **SCSI** の場合は、**SCSI** コントローラのマニュアルを参照して **SCSI BIOS** へ入る方法を確認してください。
  - ブートデバイスがシステム **BIOS** で管理されている場合は、システム **BIOS** のマニュアルを参照してシステム **BIOS** へ入る方法を確認してください。
2. 該当する **BIOS** のマニュアルに従って、元のブート環境のブートデバイスに戻るようにブートデバイスを変更します。
3. **BIOS** の変更を保存します。
4. ブート処理を開始するために、**BIOS** メニューを閉じます。

5. **b -s** と入力して、シングルユーザー状態でマシンをブートします。

6. 次のコマンドを入力します。

```
# /sbin/luactivate
```

7. リブートします。

```
# init 6
```

## ▼ x86: 同じディスクに存在するブート環境をフォールバックする

ブート中に障害が発生する場合があります。ルート (/) ファイルシステムが同じ物理ディスク上に存在する場合、次の手順を実行して、元のブート環境にフォールバックします。この場合、以前のアクティブブート環境からルート (/) スライスをマウントする必要があります。その後で、ブート環境の切り替えを行う `luactivate` コマンドを実行してください。リブートすると、1つ前のアクティブブート環境が再び起動されます。

1. システムのブート方法を決定します。

- Solaris DVD または Solaris INSTALLATION 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、システムのコモニティディスクドライブに Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) を挿入します。

---

**x86** のみ – 付録 I の手順に従って Solaris DVD (x86 版) または Solaris SOFTWARE 2 of 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。

---

画面の指示に従って進み、「Current Boot Parameters」メニューを表示します。

2. **b -s** と入力して、シングルユーザー状態でマシンをブートします。



- 必要に応じて、フォールバックブート環境のルート (/) が置かれているファイルシステムの完全性を確認します。

```
# fsck mount_point
```

*mount\_point* 信頼性のあるルートファイルシステム

- アクティブブート環境のルートスライスをディレクトリ (/mnt/sbin など) にマウントします。

```
# mount device_name /mnt/sbin
```

*device\_name* フォールバックしたい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名前を指定します。デバイス名の形式は、/dev/dsk/cwtxdysz です。

- アクティブブート環境のルートスライスから、次のように入力します。

```
# /mnt/sbin/luactivate
```

前の稼働ブート環境がアクティブになり、結果が示されます。

- /mnt/sbin のマウントを解除します。

```
# umount device_name /mnt/sbin
```

*device\_name* フォールバックしたい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名前を指定します。デバイス名の形式は、/dev/dsk/cwtxdysz です。

- リブートします。

```
# init 6
```

前の稼働ブート環境がアクティブブート環境になります。



## 第 37 章

# Solaris Live Upgrade ブート環境の管理 (作業)

この章では、ブート環境のファイルシステムを最新の状態に維持したり、ブート環境を削除するなど、さまざまな管理作業について説明します。この章の内容は次のとおりです。

- 475 ページの「Solaris Live Upgrade 管理作業の概要」
- 476 ページの「すべてのブート環境のステータスの表示」
- 478 ページの「以前に構成されたブート環境の更新」
- 480 ページの「スケジュールされた処理 (作成/アップグレード/コピー) の取り消し」
- 481 ページの「ブート環境の比較」
- 482 ページの「非アクティブブート環境の削除」
- 483 ページの「アクティブブート環境の名前の表示」
- 484 ページの「ブート環境の名前の変更」
- 486 ページの「ブート環境名に関連付ける説明の作成または変更」
- 488 ページの「ブート環境の構成の表示」

## Solaris Live Upgrade 管理作業の概要

表 37-1 Solaris Live Upgrade 管理作業の概要

作業	説明	参照先
(省略可能) ステータスの表示	■ ブート環境の状態 (アクティブ、アクティブ化の処理中、アクティブになるようにスケジュールされている、比較処理中) を表示します。	■ 476 ページの「すべてのブート環境のステータスの表示」

表 37-1 Solaris Live Upgrade 管理作業の概要 (続き)

作業	説明	参照先
	<ul style="list-style-type: none"> <li>■ アクティブブート環境と非アクティブブート環境を比較します。</li> <li>■ アクティブブート環境の名前を表示します。</li> <li>■ ブート環境の構成を表示します。</li> </ul>	<ul style="list-style-type: none"> <li>■ 481 ページの「ブート環境の比較」</li> <li>■ 483 ページの「アクティブブート環境の名前の表示」</li> <li>■ 488 ページの「ブート環境の構成の表示」</li> </ul>
(省略可能) 非ブアクティブブート環境の更新	ファイルシステムの構成を変更することなく、アクティブブート環境からファイルシステムを再度コピーします。	478 ページの「以前に構成されたブート環境の更新」
(省略可能) その他の作業	<ul style="list-style-type: none"> <li>■ ブート環境を削除します。</li> <li>■ ブート環境の名前を変更します。</li> <li>■ ブート環境の名前に関連付ける説明を作成または変更します。</li> <li>■ スケジュールされているジョブを取り消します。</li> </ul>	<ul style="list-style-type: none"> <li>■ 482 ページの「非アクティブブート環境の削除」</li> <li>■ 484 ページの「ブート環境の名前の変更」</li> <li>■ 486 ページの「ブート環境名に関連付ける説明の作成または変更」</li> <li>■ 480 ページの「スケジュールされた処理 (作成/アップグレード/コピー) の取り消し」</li> </ul>

## すべてのブート環境のステータスの表示

ブート環境についての情報を表示するには、「Status」メニューまたは `lustatus` コマンドを使用してください。ブート環境を指定しない場合は、システム上のすべてのブート環境のステータス情報が表示されます。

各ブート環境について、次の詳細情報が表示されます。

- Name – 各ブート環境の名前
- Complete – コピー処理、作成処理とも進行中ではなく、ブート環境をブートできる状態であることを示します。作成処理またはアップグレード処理が進行中であつたり失敗した場合などは、ステータスは未完了として示されます。たとえば、あるブート環境のコピー処理が進行中であるか、コピー処理がスケジュールされている場合は、そのブート環境は未完了とみなされます。
- Active – アクティブブート環境であるかどうかを示します。

- **ActiveOnReboot** – システムの次のリブート時にそのブート環境がアクティブになるかどうかを示します。
- **CopyStatus** – ブート環境の作成またはコピーの状態 (作成またはコピーがスケジューリングされている、アクティブ、またはアップグレード中) を示します。ステータスが SCHEDULED の場合、Solaris Live Upgrade のコピー、名前変更、アップグレードの各処理を行うことはできません。

## ▼ すべてのブート環境のステータスを表示する (キャラクタインタフェース)

- メインメニューから「**Status**」を選択します。  
次のような表が表示されます。

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
disk_a_S7	yes	yes	yes	no	-
disk_b_S7database	yes	no	no	yes	COPYING
disk_b_S8	no	no	no	yes	-

注 – この例では、disk\_b\_S8 は未完了の状態、disk\_b\_S7database は CopyStatus が SCHEDULED になっています。このためこれらの環境では、コピー、名前変更、アップグレードの各処理を行うことはできません。

## ▼ すべてのブート環境のステータスを表示する (コマンド行インタフェース)

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# lustatus BE_name
```

*BE\_name* ステータスを表示する非アクティブブート環境の名前を指定します。  
*BE\_name* を省略すると、lustatus によりシステム内のすべてのブート環境のステータスが表示されます。

この例では、すべてのブート環境のステータスが表示されます。

```
# lustatus second_disk
```

boot environment Name	Is Complete	Active Now	Active OnReboot	Can Delete	Copy Status
disk_a_S7	yes	yes	yes	no	-
disk_b_S7database	yes	no	no	yes	COPYING

```
disk_b_S8          no          no          no          yes          -
```

---

注 - この例では、disk\_b\_S8 は未完了の状態、disk\_b\_S7database は CopyStatus が SCHEDULED になっています。このためこれらの環境では、コピー、名前変更、アップグレードの各処理を行うことはできません。

---

## 以前に構成されたブート環境の更新

「Copy」メニューまたは lumake コマンドを使用して、以前に構成されたブート環境の内容を更新できます。アクティブ (ソース) ブート環境のファイルシステムがターゲットブート環境にコピーされると、ターゲット上にあったデータは破棄されます。コピー元のブート環境のステータスは、「complete」である必要があります。ブート環境のステータスを確認する方法については、476 ページの「すべてのブート環境のステータスの表示」を参照してください。

コピー作業は後で行われるようにスケジュールできます。スケジュールできるのは一度に1つのジョブだけです。スケジュールされたコピー処理を取り消す方法については、480 ページの「スケジュールされた処理 (作成/アップグレード/コピー) の取り消し」を参照してください。

### ▼ 以前に構成されたブート環境を更新する (キャラクタインタフェース)

1. メインメニューから「Copy」を選択します。
2. 更新する非アクティブブート環境の名前を入力します。  
Name of Target Boot Environment: **solaris8**
3. コピー処理を継続するか、または後でコピーが実行されるようにスケジュールします。
  - コピーを継続するには、Return キーを押します。  
以上の手順で、非アクティブブート環境が更新されます。
  - 後でコピーが実行されるようにスケジュールするには、「y」と入力し、時刻 (at コマンドの書式を使用) と、結果の送信先電子メールアドレスを指定します。

```
Do you want to schedule the copy? y  
Enter the time in 'at' format to schedule copy: 8:15 PM
```

```
Enter the address to which the copy log should be mailed:
someone@anywhere.com
```

時間の書式については、at (1) のマニュアルページを参照してください。

以上の手順で、非アクティブブート環境が更新されます。

スケジュールされたコピー処理を取り消す方法については、480 ページの「スケジュールされた処理 (作成/アップグレード/コピー) の取り消し」を参照してください。

## ▼ 以前に構成されたブート環境を更新する (コマンド行インタフェース)

この手順では、以前に作成されたブート環境上の古いファイルを上書きしてソースファイルをコピーします。

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# lumake -n BE_name [-s source_BE] [-t time] [-m email_address]
```

-n <i>BE_name</i>	ファイルシステムを書き換えるブート環境の名前を指定します。
-s <i>source_BE</i>	(省略可能) ターゲットブート環境にコピーするファイルシステムがあるソースブート環境の名前を指定します。このオプションを省略すると、lumake は現在のブート環境をソースとして使用します。
-t <i>time</i>	(省略可能) 指定されたブート環境上のファイルを指定された時刻に上書きするバッチジョブを設定します。時刻は、at(1) のマニュアルページに指定されている書式で入力します。
-m <i>email_address</i>	(省略可能) コマンドが完了した時点で、ここで指定する電子メールアドレスに lumake の出力を送ります。email_address はチェックされません。このオプションは、-t と併用する必要があります。

例 37-1 以前に構成されたブート環境を更新する (コマンド行インタフェース)

この例では、`first_disk` のファイルシステムが `second_disk` にコピーされます。処理が完了した時点で、電子メールが `anywhere.com` の Joe に送信されます。

```
# lumake -n second_disk -s first_disk -m joe@anywhere.com
```

例 37-1 以前に構成されたブート環境を更新する (コマンド行インタフェース) (続き)

first\_disk 上のファイルが second\_disk にコピーされ、通知の電子メールが送信されます。スケジュールされたコピー処理を取り消す方法については、480 ページの「スケジュールされた処理 (作成/アップグレード/コピー) の取り消し」を参照してください。

---

## スケジュールされた処理 (作成/アップグレード/コピー) の取り消し

ブート環境のスケジュールされた処理 (作成、アップグレード、コピー) は、その処理の開始前に取り消すことができます。GUI では、「Create a Boot Environment」、 「Upgrade a Boot Environment」、または「Copy a Boot Environment」メニューを使用して、特定の時間に処理が実行されるようにスケジュールすることができます。CLI では、lucancel コマンドを使用して、処理をスケジュールできます。システムでスケジュールできるジョブは一度に 1 つだけです。

### ▼ スケジュールされた処理 (作成/アップグレード/コピー) を取り消す (キャラクタインタフェース)

1. メインメニューから「**Cancel**」を選択します。
2. 取り消しが可能なブート環境の一覧を表示するには、**F2** を押します。
3. 取り消すブート環境を選択します。  
これで、指定されている時刻に処理は実行されなくなります。

### ▼ スケジュールされた処理 (作成/アップグレード/コピー) を取り消す (コマンド行インタフェース)

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。  

```
# lucancel
```

これで、指定されている時刻に処理は実行されなくなります。



---

## ブート環境の比較

アクティブブート環境とその他のブート環境の相違を確認するには、「Compare」メニューまたは `lucompare` コマンドを使用します。比較をするためには、非アクティブブート環境は完了状態で、コピー処理がスケジュールされていない必要があります。476 ページの「すべてのブート環境のステータスの表示」を参照してください。

指定するブート環境には、`lumount` または `mount` を使用してマウントされたパーティションを含むことはできません。

### ▼ ブート環境を比較する (キャラクタインタフェース)

1. メインメニューから「Compare」を選択します。
2. 「Compare to Original」または「Compare to an Active Boot Environment」を選択します。
3. **F3** を押します。
4. 次に示すように、元の (アクティブ) ブート環境の名前、非アクティブブート環境の名前、およびファイルのパスを入力します。

```
Name of Parent: solaris8
Name of Child: solaris8-1
Full Pathname of the file to Store Output: /tmp/compare
```

5. **F3** を押してファイルに保存します。  
「Compare」メニューに以下の属性が表示されます。

- 「モード」
- 「リンクの数」
- 「所有者」
- 「グループ」
- チェックサム - 指定されたブート環境内のファイルとこれに対応するアクティブブート環境内のファイルが、前述したすべてのフィールドにおいて一致する場合だけ、チェックサムを計算します。すべて一致するがチェックサムは異なるという場合には、異なるチェックサムが比較対象ファイルのエントリに付加されます。
- 「サイズ」
- いずれか一方のブート環境だけに存在するファイル

6. **F3** を押して「**Compare**」メニューに戻ります。

## ▼ ブート環境を比較する (コマンド行インタフェース)

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# /usr/sbin/lucompare -i infile
```

または

```
# /usr/sbin/lucompare -t -o outfile BE_name
```

**-i infile** *infile* 中に指定されたファイルを比較します。比較するファイルは、絶対パスで指定する必要があります。ファイルのエントリがディレクトリである場合、比較はディレクトリに対して再帰的に行われます。このオプションまたは **-t** のいずれか一方を使用できます (両方は使用できません)。

**-t** バイナリ以外のファイルだけを比較します。この比較では、ファイルごとに `file(1)` コマンドを使用してそのファイルがテキストファイルであるかを確認します。ユーザーは、このオプションまたは **-i** のいずれか一方を使用できます (両方は使用できません)。

**-o outfile** 相違についての出力を *outfile* にリダイレクトします。

**BE\_name** アクティブブート環境と比較するブート環境の名前を指定します。

### 例 37-2 ブート環境の比較 (コマンド行インタフェース)

この例では、`first_disk` ブート環境 (ソース) と `second_disk` ブート環境が比較され、結果がファイルに出力されます。

```
# /usr/sbin/lucompare -i /etc/lu/compare/ \  
-o /var/tmp/compare.out second_disk
```

---

## 非アクティブブート環境の削除

「Delete」メニューまたは `ludelete` を使用してください。アクティブブート環境および次のリブートでアクティブになるブート環境は、削除できません。削除するブート環境は完了状態でなければなりません。完了状態のブート環境とは、ステータスを

変更する処理が終了している環境を指します。ブート環境のステータスを確認する方法については、476 ページの「すべてのブート環境のステータスの表示」を参照してください。また、`lumount` を使用してファイルシステムをマウントしているブート環境も削除できません。

## ▼ 非アクティブブート環境を削除する (キャラクタインタフェース)

1. メインメニューから「Delete」を選択します。
2. 削除する非アクティブブート環境の名前を入力します。

```
Name of boot environment: solaris8
指定された非アクティブブート環境が削除されます。
```

## ▼ 非アクティブブート環境を削除する (コマンド行インタフェース)

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# ludelete BE_name
BE_name    削除する非アクティブブート環境の名前を指定します。
```

例 37-3 非アクティブブート環境の削除 (コマンド行インタフェース)

この例では、ブート環境 `second_disk` が削除されます。

```
# ludelete second_disk
```

---

## アクティブブート環境の名前の表示

現在動作しているブート環境の名前を表示するには、「Current」メニューまたは `lucurr` コマンドを使用してください。システム上に構成されたブート環境がない場合は、「No Boot Environments are defined」というメッセージが表示されます。

lucurr で表示されるのは現在のブート環境の名前だけです。次のブート時にアクティブになるブート環境の名前は表示されません。ブート環境のステータスを確認する方法については、476 ページの「すべてのブート環境のステータスの表示」を参照してください。

## ▼ アクティブブート環境の名前を表示する (キャラクタインタフェース)

- メインメニューから「**Current**」を選択します。  
アクティブブート環境の名前または「No Boot Environments are defined」というメッセージが表示されます。

## ▼ アクティブブート環境の名前を表示する (コマンド行インタフェース)

- 次のコマンドを入力します。

```
# /usr/sbin/lucurr
```

例 37-4 アクティブブート環境の名前の表示 (コマンド行インタフェース)

この例では、現在のブート環境の名前が表示されます。

```
# /usr/sbin/lucurr  
solaris8
```

---

## ブート環境の名前の変更

ブート環境の名前の変更は、ブート環境の Solaris リリースを別のリリースにアップグレードする場合などに便利です。たとえば、オペレーティングシステムのアップグレード処理においてブート環境 solaris 7 を solaris 8 に変更できます。非アクティブブート環境の名前を変更する場合は、「Rename」メニューまたは `lurename` コマンドを使用してください。

新しい名前の設定は、以下の規則に従う必要があります。

- 長さが 30 文字を超えない
- 英数字または ASCII 文字 (UNIX シェルで特別な意味を持つ文字を除く) で構成されている。sh(1) の「クォート」の節を参照してください。

- シングルバイトの 8 ビット文字だけ含む
- システム上に 1 つしか存在しない

ブート環境の名前を変更するためには、そのステータスが「complete」である必要があります。ブート環境のステータスを確認する方法については、476 ページの「すべてのブート環境のステータスの表示」を参照してください。lumount または mount を使用してファイルをマウントしているブート環境の名前は、変更できません。

## ▼ 非アクティブブート環境の名前を変更する (キャラクタインタフェース)

1. メインメニューから「Rename」を選択します。
2. 名前を変更するブート環境を指定し、続いて新しい名前を入力します。
3. **F3** を押して変更を保存します。

## ▼ 非アクティブブート環境の名前を変更する (コマンド行インタフェース)

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# lurename -e BE_name -n new_name
```

-e *BE\_name* 変更する非アクティブブート環境の名前を指定します。

-n *new\_name* 非アクティブブート環境の新しい名前を指定します。

この例では、second\_disk が third\_disk に変更されます。

```
# lurename -e second_disk -n third_disk
```

---

## ブート環境名に関連付ける説明の作成または変更

ブート環境名に説明を付けることができます。この説明によって名前が置き換わることはありません。ブート環境名は長さや文字に制限がありますが、この説明は長さ、内容に制限がありません。シンプルなテキストでも、gif ファイルのような複雑なものでもかまいません。この説明は、次の時点で作成できます。

- ブート環境を作成する時点 (-A オプション指定で `lucreate` コマンドを使用する)
- ブート環境の作成後 (`ludesc` コマンドを使用する)

`lucreate` コマンドで -A オプションを使用する方法の詳細は、424 ページの「ブート環境を初めて作成する (コマンド行インタフェース)」を参照してください。ブート環境の作成後に説明を作成する方法については、以下の説明または `ludesc (1M)` のマニュアルページを参照してください。

### ▼ テキストを使用してブート環境名の説明を作成または変更する方法

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# /usr/sbin/ludesc -n BE_name 'BE_description'
-n BE_name 'BE_description'    ブート環境名を指定します。
```

例 37-5 テキストを使用してブート環境名に説明を加える

この例では、`second_disk` というブート環境に説明が加えられています。この説明は、単一引用符で囲まれたテキストで記述されます。

```
# /usr/sbin/ludesc -n second_disk 'Solaris 9 test build'
```

### ▼ ファイルを使用してブート環境名の説明を作成または変更する方法

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# /usr/sbin/ludesc -n BE_name -f file_name
```

`-n BE_name` ブート環境名を指定します。  
`file_name` ブート環境名に関連付ける説明が書かれているファイルを指定します。

例 37-6 ファイルを使用してブート環境名に説明を加える

この例では、`second_disk` というブート環境に説明が加えられています。説明は、`gif` ファイル内に入っています。

```
# /usr/sbin/ludesc -n second_disk -f rose.gif
```

## ▼ テキストで記述された説明からブート環境名を確認する方法

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# /usr/sbin/ludesc -A 'BE_description'  
-A 'BE_description' ブート環境名に関連付けられている説明を指定します。
```

例 37-7 説明からブート環境名を確認する

この例では、説明を指定して `-A` オプションを使用することでブート環境名 `second_disk` を確認しています。

```
# /usr/sbin/ludesc -A 'Solaris 9 test build'  
second_disk
```

## ▼ ファイル内の説明からブート環境名を確認する方法

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# /usr/sbin/ludesc -f file_name  
-f file_name ブート環境名に関連付けられている説明を含むファイルを指定します。
```

例 37-8 ファイル内の説明からブート環境名を確認する

この例では、`-f` オプションと、説明を含むファイルの名前を使用することでブート環境の名前 `second_disk` を確認しています。

例 37-8 ファイル内の説明からブート環境名を確認する (続き)

```
# /usr/sbin/ludesc -f rose.gif
second_disk
```

## ▼ 名前からブート環境説明を確認する方法

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# /usr/sbin/ludesc -n BE_name
-n BE_name    ブート環境名を指定します。
```

例 37-9 ブート環境名から説明を確認する

この例では、ブート環境名を指定して `-n` オプションを使用することで説明を確認しています。

```
# /usr/sbin/ludesc -n second_disk
Solaris 9 test build
```

---

## ブート環境の構成の表示

ブート環境の構成を表示するには、「List」メニューまたは `lufslist` コマンドを使用してください。出力される情報は、各ブート環境マウントポイントのディスクスライス(ファイルシステム)、ファイルシステムの種類、およびファイルシステムサイズです。

## ▼ 非アクティブブート環境の構成を表示する (キャラクターインタフェース)

1. メインメニューから「List」を選択します。
2. 構成を表示したいブート環境の名前を入力します。

```
Name of Boot Environment: solaris8
```

3. **F3** を押します。  
次の例ではリストを表示しています。



Filesystem	fstype	size (Mb)	Mounted on
/dev/dsk/c0t0d0s1	swap	512.11	-
/dev/dsk/c0t4d0s3	ufs	3738.29	/
/dev/dsk/c0t4d0s4	ufs	510.24	/opt

4. **F6** を押して「**List**」メニューに戻ります。

## ▼ ブート環境の構成を表示する (コマンド行インタフェース)

1. スーパーユーザーとしてログインします。
2. 次のコマンドを入力します。

```
# lufslist
```

*BE\_name* ファイルシステムの詳細を表示するブート環境の名前を指定します。  
次の例ではリストを表示しています。

Filesystem	fstype	size (Mb)	Mounted on
/dev/dsk/c0t0d0s1	swap	512.11	-
/dev/dsk/c0t4d0s3	ufs	3738.29	/
/dev/dsk/c0t4d0s4	ufs	510.24	/opt



# Solaris Live Upgrade (例)

---

この章では、ブート環境を作成したあと、新しいブート環境を更新およびアクティブ化して、これを新たな稼動環境にする例を示します。この章の内容は次のとおりです。

- 491 ページの「Solaris Live Upgrade によるアップグレードの使用例 (コマンド行インタフェース)」
- 494 ページの「RAID 1 ボリューム (ミラー) の一方を切り離してアップグレードする例 (コマンド行インタフェース)」
- 498 ページの「既存のボリュームから Solaris ボリュームマネージャ RAID-1 ボリュームへの移行例 (コマンド行インタフェース)」
- 498 ページの「空のブート環境を作成してフラッシュアーカイブをインストールする例 (コマンド行インタフェース)」
- 501 ページの「Solaris Live Upgrade によるアップグレードの例 (キャラクタインタフェース)」

---

## Solaris Live Upgrade によるアップグレードの使用例 (コマンド行インタフェース)

この例では、Solaris 7 リリースを使用しているシステムで、`lucreate` コマンドを使用して新しいブート環境を作成します。この新しいブート環境は、`luupgrade` コマンドを使用して Solaris 9 にアップグレードされ、次に、`luactivate` コマンドによってアクティブ化されます。この節では、以前のブート環境にフォールバックする例も示します。

## アクティブブート環境で Live Upgrade をインストールする

1. Solaris DVD または Solaris SOFTWARE 2 of 2 CD を挿入します。
2. 使用しているメディアに合わせて操作を行います。
  - Solaris DVD を使用している場合は、インストーラのあるディレクトリに移動し、インストーラを実行します。

```
# cd /cdrom/cdrom0/Solaris_9/Tools/Installers
# ./liveupgrade20
```

Solaris Web Start インストーラが表示されます。
  - Solaris SOFTWARE 2 of 2 CD を使用している場合は、インストーラを実行します。

```
% ./installer
```

Solaris Web Start インストーラが表示されます。
3. 「インストール形式の選択 (Select Type of Install)」パネルで「カスタム (Custom)」をクリックします。
4. 「ロケールの選択 (Locale Selection)」パネルで、インストールする言語をクリックします。
5. インストールするソフトウェアを選択します。
  - DVD の場合、「コンポーネントの選択 (Component Selection)」パネルの「次へ (Next)」をクリックしてパッケージをインストールします。
  - CD の場合、「製品の選択 (Product Selection)」パネルの Solaris Live Upgrade の項目で「デフォルトインストール (Default Install)」をクリックします。
6. Solaris Web Start の指示に従って、ソフトウェアをインストールします。

## ブート環境を作成する

-c オプションを使用して、ソースブート環境に c0t4d0s0 という名前を付けます。ソースブート環境の名前設定は最初のブート環境を作成するときだけ必要です。-c オプションによる名前付けの詳細は、手順 2 を参照してください。

新しいブート環境の名前は c0t15d0s0 です。-A オプションを用いて、このブート環境名に関連付けられる説明を作成します。

ルート (/) ファイルシステムを新しいブート環境にコピーし、ソースブート環境のスワップスライスは共有せずに、新しいスワップスライスを作成します。

```
# lucreate -A 'BE_description' -c c0t4d0s0 -m /:/dev/dsk/c0t15d0s0:ufs \
-m -:/dev/dsk/c0t15d0s1:swap -n c0t15d0s0
```

## 非アクティブブート環境をアップグレードする

非アクティブブート環境の名前は c0t15d0s0 です。アップグレードに使用されるオペレーティングシステムイメージはネットワークから取得します。

```
# luupgrade -n c0t15d0s0 -u -s /net/ins-svr/export/Solaris_9 \
combined.solaris_wos
```

## ブート環境がブート可能か確認する

lustatus コマンドは、ブート環境の作成が完了したかどうかを報告します。  
lustatus コマンドは、ブート環境がブート可能であるかどうかを報告します。

```
# lustatus
boot environment   Is           Active      Active      Can          Copy
Name              Complete    Now         OnReboot    Delete       Status
-----
c0t4d0s0           yes         yes         yes         no          -
c0t15d0s0         yes         no          no          yes         -
```

## 非アクティブブート環境をアクティブにする

luactivate コマンドを使用して c0t15d0s0 ブート環境 をブート可能にします。続いてシステムをリブートします。これで c0t15d0s0 がアクティブブート環境になり、c0t4d0s0 ブート環境が非アクティブになります。

```
# luactivate c0t15d0s0
# init 6
```

## ソースブート環境へフォールバックする

新しいブート環境のアクティブ化の状況に応じて、次の3つの作業からいずれかを選択します。

- アクティブ化が正常に行われたが元のブート環境に戻りたいという場合は、例 38-1を参照してください。
- アクティブ化に失敗したが、元のブート環境からブート可能な場合は、例 38-2を参照してください。
- アクティブ化に失敗し、メディアまたはネットインストールイメージを使用して元のブート環境に戻る必要がある場合は、例 38-3を参照してください。

例 38-1 ブート環境作成は正常に完了したが元のブート環境にフォールバックさせる場合

この例では、新しいブート環境のアクティブ化が正常に完了したにもかかわらず、元の c0t4d0s0 ブート環境をアクティブブート環境として復元しています。デバイス名は first\_disk です。

例 38-1 ブート環境作成は正常に完了したが元のブート環境にフォールバックさせる場合 (続き)

```
# /usr/sbin/luactivate first_disk
# init 6
```

例 38-2 SPARC: ブート環境のアクティブ化に失敗した場合のフォールバック

この例では、新しいブート環境のブートに失敗しています。シングルユーザーモードで元のブート環境 c0t4d0s0 からブートさせるために、OK プロンプトを表示させる必要があります。

```
OK boot net -s
# /sbin/luactivate first_disk
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# init 6
```

元のブート環境 c0t4d0s0 がアクティブブート環境になります。

例 38-3 SPARC: DVD、CD、またはネットワークインストールイメージを使って元のブート環境にフォールバックする

この例では、新しいブート環境のブートに失敗しています。元のブート環境からはブートできないためメディアまたは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。デバイスは /dev/dsk/c0t4d0s0 です。元のブート環境 c0t4d0s0 がアクティブブート環境になります。

```
OK boot net -s
# fsck /dev/dsk/c0t4d0s0
# mount /dev/dsk/c0t4d0s0 /mnt
# /mnt/sbin/luactivate
Do you want to fallback to activate boot environment c0t4d0s0
(yes or no)? yes
# umount /mnt
# init 6
```

---

## RAID 1 ボリューム (ミラー) の一方を切り離してアップグレードする例 (コマンド行インタフェース)

この例では、次の作業の手順を示します。

- 新しいブート環境に RAID-1 ボリューム (ミラー) を作成する
- ミラーの一方を切り離し、アップグレードする

- ミラー (連結) の他方を新しいミラーに接続する

図 38-1 は、3 つの物理ディスクから成る現在のブート環境を示します。

RAID-1 ボリューム (ミラー) の一方を切り離し、アップグレードする

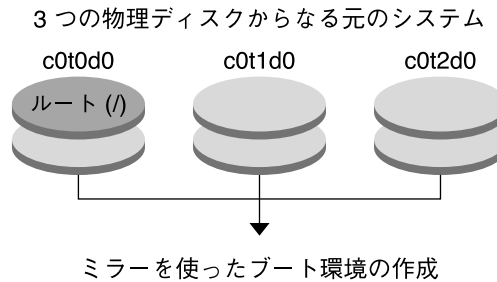


図 38-1 RAID-1 ボリューム (ミラー) の一方を切り離し、アップグレードする

1. ミラーを持つ新しいブート環境 `second_disk` を作成します。

次のコマンドは、次のような処理を実行します。

- `lucreate` コマンドにより、ルート (`/`) マウントポイントの UFS ファイルシステムが構成されます。d10 というミラーが作成されます。このミラー d10 に、現在のブート環境のルート (`/`) ファイルシステムがコピーされます。ミラー d10 にあるデータはすべて上書きされます。
- 2 つのスライス `c0t1d0s0` および `c0t2d0s0` は、サブミラーとして指定されています。これら 2 つのサブミラーは、ミラー d10 に接続されます。

```
# lucreate -c first_disk -n second_disk \
-m /:/dev/md/dsk/d10:ufs,mirror \
-m /:/dev/dsk/c0t1d0s0:attach \
-m /:/dev/dsk/c0t2d0s0:attach
```

2. ブート環境 `second_disk` をアクティブ化します。

```
# /usr/sbin/luactivate second_disk
# init 6
```

3. 別のブート環境 `third_disk` を作成します。

次のコマンドは、次のような処理を実行します。

- `lucreate` コマンドにより、ルート (`/`) マウントポイントの UFS ファイルシステムが構成されます。d20 というミラーが作成されます。
- スライス `c0t1d0s0` がその現在のミラーから切り離され、ミラー d20 に追加されます。このサブミラーの内容であるルート (`/`) ファイルシステムは保持され、コピー処理は発生しません。

```
# lucreate -n third_disk \
-m /:/dev/md/dsk/d20:ufs,mirror \
-m /:/dev/dsk/c0t1d0s0:detach,attach,preserve
```

4. フラッシュアーカイブをインストールして、新しいブート環境 `third_disk` をアップグレードします。アーカイブはローカルシステムに存在します。 `-s` および `-a` オプションで指定するオペレーティングシステムバージョンは、どちらも Solaris 9 リリースです。 `third_disk` 上のファイルは、共有可能ファイルを除いてすべて上書きされます。

```
# luupgrade -f -n third_disk \  
-s /net/installmachine/export/Solaris_9/OS_image \  
-a /net/server/archive/Solaris_9
```

5. ブート環境 `third_disk` をアクティブ化して、このブート環境からシステムを実行します。

```
# /usr/sbin/luactivate third_disk  
# init 6
```

6. ブート環境 `second_disk` を削除します。

```
# ludelete second_disk
```

7. 次のコマンドは、次のような処理を実行します。

- ミラー `d10` を消去します。
- `c0t2d0s0` の連結の数を調べます。
- `metastat` コマンドで見つけた連結を、ミラー `d20` に接続します。  
`metattach` コマンドは、新しく接続した連結と、ミラー `d20` の連結とを同期します。連結にあるデータはすべて上書きされます。

```
# metaclear d10  
metastat -p | grep c0t2d0s0  
dnum 1 1 c0t2d0s0  
metattach d20 dnum
```

`num` `metastat` コマンドで見つかった連結の数。

これで、新しいブート環境 `third_disk` がアップグレードされ、この環境からシステムが実行されます。 `third_disk` には、ミラー化されたルート (`/`) ファイルシステムが含まれています。

図 38-2 は、上記の例のコマンドでミラーを切り離してアップグレードする手順の全体を示しています。



RAID-1 ボリューム (ミラー) の一方を切り離し、アップグレードする (続き)

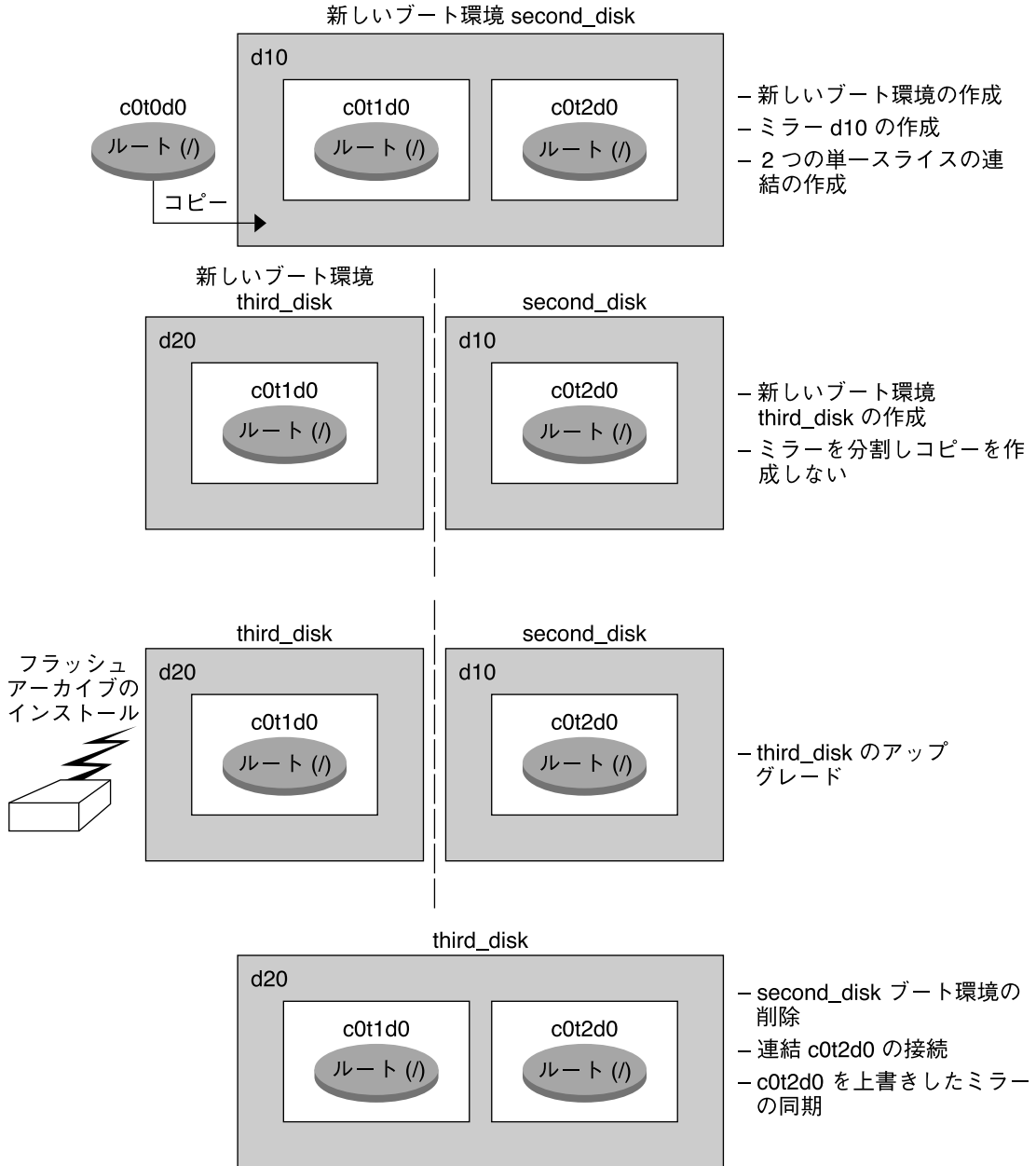


図 38-2 RAID-1 ボリューム (ミラー) の一方を切り離し、アップグレードする (続き)

---

## 既存のボリュームから Solaris ボリュームマネージャ RAID-1 ボリュームへの移行例 (コマンド行インタフェース)

Solaris Live Upgrade では、RAID-1 ボリューム (ミラー) 上に新しいブート環境を作成できます。現在のブート環境のファイルシステムは、次のいずれかです。

- 物理ストレージデバイス
- Solaris ボリュームマネージャの制御下の RAID-1 ボリューム
- Veritas VXFS 制御下のボリューム

ただし、新しいブート環境のターゲットは、Solaris ボリュームマネージャ RAID-1 ボリュームでなければなりません。たとえば、ルート (/) ファイルシステムのコピー用に指定するスライスは、/dev/md/dsk/rootvol となります。rootvol はルート (/) ファイルシステムを含むボリュームです。

この例では、現在のブート環境のルート (/) ファイルシステムは Solaris ボリュームマネージャボリューム以外のボリューム上にあります。新しいブート環境では、Solaris ボリュームマネージャ RAID-1 ボリュームである c0t2d0s0 上にルート (/) ファイルシステムが作成されます。lucreate コマンドは、現在のボリュームを Solaris ボリュームマネージャボリュームに移行させるコマンドです。新しいブート環境の名前は svm\_be です。lustatus コマンドを使用すると、新しいブート環境のアクティブ化とリブートの準備ができているかどうかわかります。ブート環境がアクティブ化され、現在のブート環境になります。

```
# lucreate -n svm_be -m /:/dev/md/dsk/dl:mirror,ufs \  
-m /:/dev/dsk/c0t2d0s0:attach  
# lustatus  
# luactivate svm_be  
# lustatus  
# init 6
```

---

## 空のブート環境を作成してフラッシュアーカイブをインストールする例 (コマンド行インタフェース)

以下の手順を 3 段階に分けて説明します。

- 空のブート環境の作成

- アーカイブのインストール
- ブート環境をアクティブにし、現在実行中のブート環境にする

lucreate コマンドは、アクティブなブート環境内のファイルシステムに基づいたブート環境を作成します。lucreate コマンドに `-s` オプションを指定して実行すると、空のブート環境を短時間で作成できます。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、フラッシュアーカイブがインストールされる時にはじめて作成されることとなります。空のブート環境にアーカイブがインストールされると、ファイルシステムは予約されたスライスにインストールされます。その後、ブート環境をアクティブ化します。

## 空のブート環境の作成

最初の手順で、空のブート環境を作成します。指定されたファイルシステム用にスライスが予約されますが、現在のブート環境からファイルシステムがコピーされることはありません。新しいブート環境の名前は `second_disk` です。

```
# lucreate -s - -m /:/dev/dsk/c0t1d0s0:ufs \  
-n second_disk
```

これで、ブート環境にフラッシュアーカイブを格納する準備ができました。

図 38-3 は、空のブート環境の作成の様子を示しています。

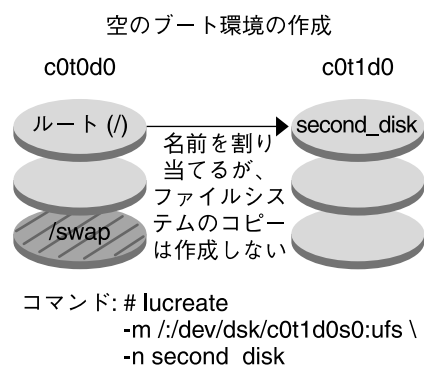
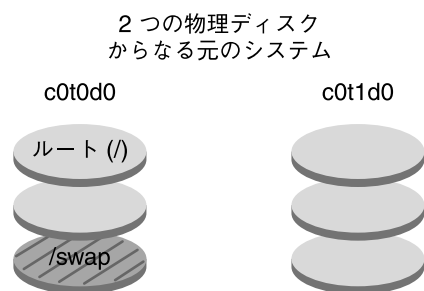


図 38-3 空のブート環境の作成

## 新しいブート環境へのフラッシュアーカイブのインストール

2番目の手順では、前の例で作成した second\_disk ブート環境に、アーカイブをインストールします。アーカイブはローカルシステムに存在します。-s および -a オプションで指定するオペレーティングシステムのバージョンは、どちらも Solaris 9 リリースです。アーカイブの名前は Solaris\_9.flar です。

```

# luupgrade -f -n second_disk \
-s /net/installmachine/export/Solaris_9/OS_image \
-a /net/server/archive/Solaris_9.flar

```

これで、ブート環境をアクティブにする準備ができました。

## 新しいブート環境のアクティブ化

最後の手順では、`luactivate` コマンドを使用して、`second_disk` ブート環境をブート可能な状態にします。続いてシステムをリブートします。これで `second_disk` がアクティブブート環境になります。

```
# luactivate second_disk
# init 6
```

- 空のブート環境の詳しい作成手順については、436 ページの「フラッシュアーカイブ用の空のブート環境の作成 (コマンド行インタフェース)」を参照してください。
- フラッシュアーカイブの詳しい作成手順については、第 21 章を参照してください。
- ブート環境をアクティブ化したり、元のブート環境にフォールバックしたりする手順については、468 ページの「問題の解決: 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)」を参照してください。

---

## Solaris Live Upgrade によるアップグレードの例 (キャラクタインタフェース)

この例では、Solaris 2.6 リリースを使用しているシステム上に新しいブート環境を作成しています。Solaris 9 リリースにアップグレードし、アップグレードされたこのブート環境をアクティブにしています。

### アクティブブート環境で Live Upgrade をインストールする

1. Solaris DVD または Solaris SOFTWARE 2 of 2 CD を挿入します。
2. インストーラを実行します。
  - Solaris DVD を使用している場合は、インストーラのあるディレクトリに移動し、インストーラを実行します。

```
# cd /cdrom/cdrom0/Solaris_9/Tools/Installers
# ./liveupgrade20
```

Solaris Web Start インストーラが表示されます。
  - Solaris SOFTWARE 2 of 2 CD を使用している場合は、インストーラを実行します。

```
% ./installer
```

Solaris Web Start インストーラが表示されます。

3. 「インストール形式の選択 (Select Type of Install)」パネルで「カスタム (Custom)」をクリックします。
4. 「ロケールの選択 (Locale Selection)」パネルで、インストールする言語をクリックします。
5. インストールするソフトウェアを選択します。
  - DVD の場合、「コンポーネントの選択 (Component Selection)」パネルの「次へ (Next)」をクリックしてパッケージをインストールします。
  - CD の場合、「製品の選択 (Product Selection)」パネルの Solaris Live Upgrade の項目で「デフォルトインストール (Default Install)」をクリックします。
6. Solaris Web Start の指示に従って、ソフトウェアをインストールします。

## ブート環境を作成する

この例では、ソースブート環境の名前は c0t4d0s0 です。ルート (/) ファイルシステムを新しいブート環境にコピーし、ソースブート環境のスワップスライスは共有せずに、新しいスワップスライスを作成します。

1. キャラクタインタフェースを表示します。

```
# /usr/sbin/lu
```

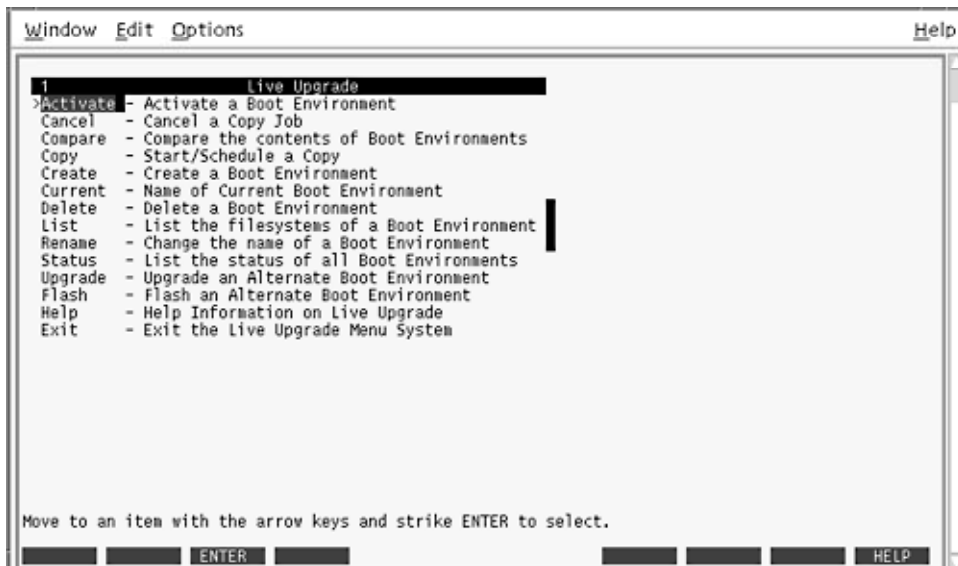


図 38-4 Solaris Live Upgrade のメインメニュー

2. メインメニューから「Create」を選択します。

Name of Current Boot Environment: **c0t4d0s0**

Name of New Boot Environment: **c0t15d0s0**

3. F3 を押します。

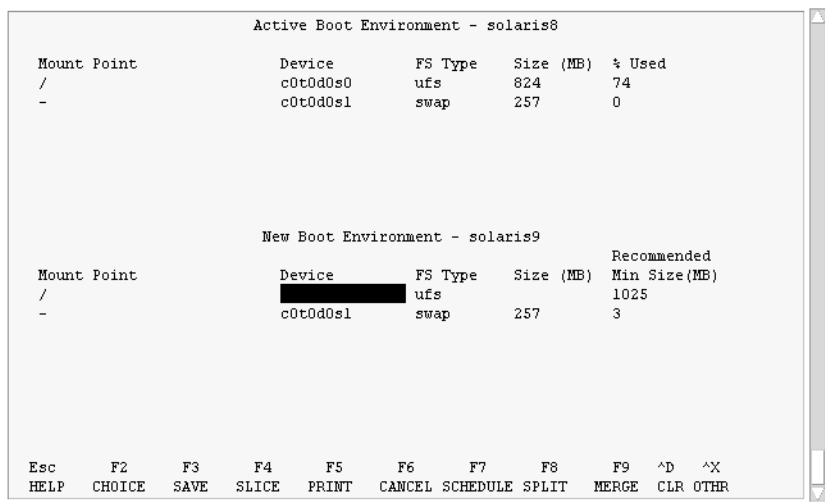


図 38-5 Solaris Live Upgrade の「Configuration」メニュー

4. 「Configuration」メニューで F2 を押して「Choices」メニューを表示します。
5. ディスク c0t15d0 からルート (/) 用としてスライス 0 を選択します。
6. 構成メニューで、分割するスワップスライスを選択して c0t15d0 上にスワップ用の新しいスライスを作成します。
7. F2 を押して「Choices」メニューを表示します。
8. 新しいスワップスライスとして、ディスク c0t15d0 からスライス 1 を選択します。
9. F3 を押して新しいブート環境を作成します。

## 非アクティブブート環境をアップグレードする

アップグレードのためにネットワークからオペレーティングシステムイメージを取得します。

1. メインメニューから「Upgrade」を選択します。

```
Name of New Boot Environment:  c0t15d0s0
Package Media: /net/ins3-svr/export/Solaris_9/combined.solaris_wos
```

2. F3 を押します。



## 非アクティブブート環境をアクティブにする

c0t15d0s0 ブート環境をブート可能にします。続いてシステムをリブートします。これで c0t15d0s0 がアクティブブート環境になり、c0t4d0s0 ブート環境が非アクティブになります。

1. メインメニューから「Activate」を選択します。

```
Name of Boot Environment: c0t15d0s0  
Do you want to force a Live Upgrade sync operations: no
```

2. F3 を押します。
3. Return キーを押します。
4. 次のコマンドを入力します。

```
# init 6
```

フォールバックが必要な場合は、前述の例のコマンド行による作業を行います。493 ページの「ソースブート環境へフォールバックする」を参照してください。



## 第 39 章

# Solaris Live Upgrade (コマンドリファレンス)

次の表に、メニューを使用せずにコマンド行で実行できるコマンドを示します。Solaris Live Upgrade には、次の表に示すすべてのコマンド行ユーティリティのマニュアルページが含まれています。

表 39-1 Solaris Live Upgrade のコマンド

作業	コマンド名
非アクティブブート環境をアクティブにする	luactivate (1M)
スケジュールされた処理 (コピーまたは作成) を取り消す	lucancel (1M)
アクティブブート環境を非アクティブブート環境と比較する	lucompare (1M)
非アクティブブート環境を更新するためにファイルシステムをコピーし直す	lucopy
ブート環境を作成する	lucreate (1M)
アクティブブート環境に名前を付ける	lucurr (1M)
ブート環境を削除する	ludelete (1M)
ブート環境の名前に記述を追加する	ludesc (1M)
各ブート環境の重要なファイルシステムを表示する	lufslist (1M)
ブート環境内のすべてのファイルシステムをマウントできる。このコマンドを使用すると、ブート環境がアクティブでない時にそのブート環境内のファイルを変更できる	lumount (1M)
ブート環境の名前を変更する	lurename (1M)
すべてのブート環境のステータスを表示する	lustatus (1M)

表 39-1 Solaris Live Upgrade のコマンド (続き)

作業	コマンド名
ブート環境に存在するすべてのファイルシステムのマウントを解除する。このコマンドを使用すると、ブート環境がアクティブでない時にそのブート環境内のファイルを変更できる	luumount (1M)
非アクティブブート環境上のオペレーティング環境をアップグレードするか、あるいは非アクティブブート環境上にフラッシュアーカイブをインストールする	luupgrade (1M)

## 第 40 章

# WAN ブートによる広域ネットワーク 経路のインストール (トピック)

---

以下の章では、広域ネットワーク (WAN) 経路でシステムをインストールする方法について説明します。

第 41 章	WAN ブートインストールの概要について説明します。
第 42 章	WAN ブートインストールを実行する前に理解しておく必要のある情報について説明します。
第 43 章	WAN ブートインストールの準備方法について説明します。
第 44 章	クライアントに対して WAN 経路でインストールを行う手順について説明します。
第 45 章	WAN ブートインストールの例を示します。
第 46 章	WAN ブートのコマンドおよび構成ファイルのキーワードと構文の一覧を示します。



## 第 41 章

---

### WAN ブート (概要)

---

この章では、WAN ブートインストールの概要について説明します。この章の内容は次のとおりです。

- 511 ページの「WAN ブートとは」
- 512 ページの「どのような場合に WAN ブートを使用するか」
- 513 ページの「WAN ブートのしくみ (概要)」
- 517 ページの「WAN ブートでサポートされているセキュリティ構成 (概要)」

---

### WAN ブートとは

WAN ブートインストールでは、HTTP を使って広域ネットワーク (WAN) 経由でソフトウェアのブートとインストールを行うことができます。WAN ブートを使用すると、大規模なパブリックネットワークを介して Solaris オペレーティング環境を SPARC ベースのシステムにインストールできますが、このようなネットワークはインフラストラクチャの信頼性が低い場合があります。WAN ブートをセキュリティ機能とともに使用することによって、データの機密性とインストールイメージの完全性を保護できます。

WAN ブートインストールでは、パブリックネットワークを介して SPARC ベースのリモートクライアントに、暗号化したフラッシュアーカイブを転送できます。次に、WAN ブートプログラムは、カスタム JumpStart インストールを実行して、クライアントシステムをインストールします。インストールの完全性を保護するために、非公開鍵を使ってデータの認証および暗号化を行うことができます。また、デジタル証明書を使用するようにシステムを構成すると、HTTPS 接続を介してインストールデータやファイルを転送できます。

WAN ブートインストールを実行するには、HTTP または HTTPS 接続を介して Web サーバーから次の情報をダウンロードして、SPARC ベースのシステムをインストールします。

- wanboot プログラム – wanboot プログラムは、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です。wanboot プログラムは、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 ブートファイルシステム – WAN ブートは、クライアントシステムをインストールするために、いくつかのファイルを使ってクライアントの構成やデータの取得を行います。これらのファイルは、Web サーバーの /etc/netboot ディレクトリに置かれています。wanboot-cgi プログラムは、これらのファイルを1つのファイルシステムとしてクライアントに転送します。このファイルシステムは WAN ブートファイルシステムと呼ばれます。
- WAN ブートミニルート – WAN ブートミニルートは、WAN ブートインストールを実行するために Solaris ミニルートに変更を加えたものです。Solaris ミニルートと同様に、WAN ブートミニルートには、カーネルのほか、Solaris 環境のインストールに最低限必要なソフトウェアが格納されています。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入っています。
- カスタム JumpStart 構成ファイル – WAN ブートは、システムをインストールするために、sysidcfg、rules.ok、およびプロファイルファイルをクライアントに転送します。次に、WAN ブートはこれらのファイルを使って、クライアントシステムに対してカスタム JumpStart インストールを実行します。
- フラッシュアーカイブ – フラッシュアーカイブは、マスターシステムからコピーされたファイルの集合体です。このアーカイブは、クライアントシステムをインストールするために使用できます。WAN ブートは、カスタム JumpStart インストールを使って、フラッシュアーカイブをクライアントシステムにインストールします。アーカイブをクライアントシステムにインストールすると、クライアント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。

次に、カスタム JumpStart インストールを使って、アーカイブをクライアントにインストールします。

上記の情報を転送するとき、鍵とデジタル証明書を使って保護することもできます。

WAN ブートインストールで発生するイベントの順序の詳細については、513 ページの「WAN ブートのしくみ (概要)」を参照してください。

---

## どのような場合に WAN ブートを使用するか

WAN ブートインストールを使用すると、地理的に離れた場所にある SPARC ベースのシステムに対してインストールを実行できます。WAN ブートを使用すると、パブリックネットワーク経由でのみアクセス可能なりモートのサーバーやクライアントに対しても、インストールを実行できます。



ローカルエリアネットワーク (LAN) 内にあるシステムに対してインストールを行いたい場合、WAN ブートインストールを使用すると、必要以上の構成や管理が必要になることがあります。LAN 上のシステムに対するインストール方法については、第 12 章を参照してください。

---

## WAN ブートのしくみ (概要)

WAN ブートは、サーバー、構成ファイル、CGI (Common Gateway Interface) プログラム、およびインストールファイルを組み合わせることで、SPARC ベースのリモートクライアントに対してインストールを行います。ここでは、WAN ブートインストールで発生するイベントの通常の順序について説明します。

### WAN ブートインストールでのイベントの順序

図 41-1 は、WAN ブートインストールで発生するイベントの基本的な順序を示しています。この図で、SPARC ベースのクライアントは、構成データとインストールファイルを、Web サーバーとインストールサーバーから WAN 経由で取得します。

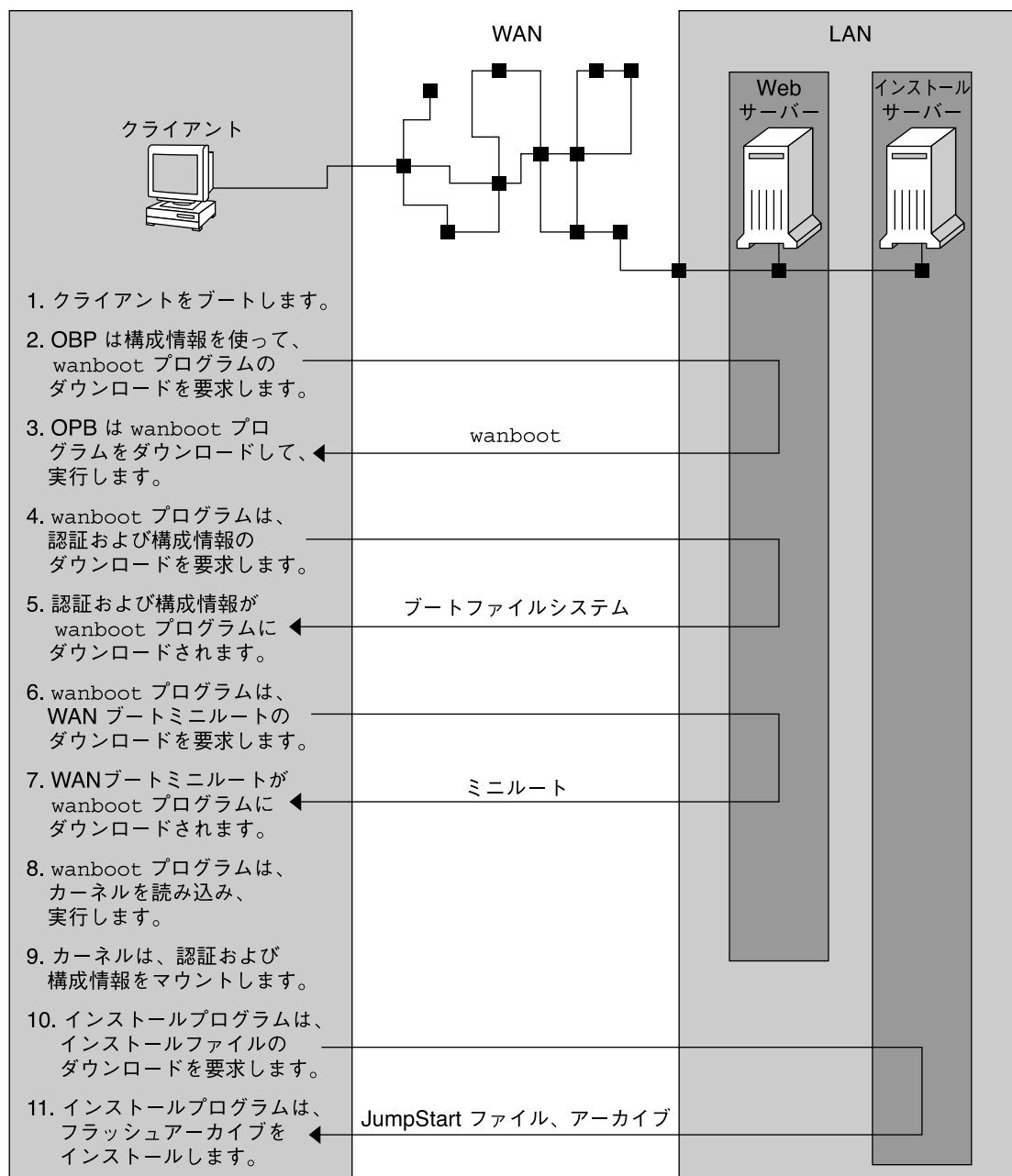


図 41-1 WAN ブートインストールでのイベントの順序

1. 次のどれかの方法で、クライアントをブートします。

- OpenBoot PROM (OBP) のネットワークインタフェース変数を設定することによって、ネットワークからブートする
  - DHCP オプションを使ってネットワークからブートする
  - ローカル CD-ROM からブートする
2. クライアントの OBP は、次のどちらかから構成情報を取得します。
    - ユーザーがコマンド行に入力したブート引数の値から
    - ネットワークで DHCP が使用されている場合は、DHCP サーバーから
  3. クライアントの OBP は、WAN ブートの二次レベルのブートプログラム (wanboot) を要求します。  
 クライアントの OBP は、wanboot プログラムを次のどちらかからダウンロードします。
    - WAN ブートサーバーと呼ばれる特別な Web サーバーから、ハイパーテキストトランスファープロトコル (HTTP) を使って
    - ローカル CD-ROM から (上記の図には示されていない)
  4. wanboot プログラムは、WAN ブートサーバーに対し、クライアント構成情報を要求します。
  5. wanboot プログラムは、wanboot-cgi プログラムによって WAN ブートサーバーから転送される構成ファイルをダウンロードします。構成ファイルは、WAN ブートファイルシステムとしてクライアントに転送されます。
  6. wanboot プログラムは、WAN ブートサーバーに対し、WAN ブートミニルートのダウンロードを要求します。
  7. wanboot プログラムは、HTTP または HTTPS を使って、WAN ブートサーバーから WAN ブートミニルートダウンロードします。
  8. wanboot プログラムは、WAN ブートミニルートから UNIX カーネルを読み込み、実行します。
  9. UNIX カーネルは、Solaris インストールプログラムで使用できるように、WAN ブートファイルシステムを見つけてマウントします。
  10. インストールプログラムは、インストールサーバーに対し、フラッシュアーカイブとカスタム JumpStart ファイルのダウンロードを要求します。  
 インストールプログラムは、HTTP または HTTPS 接続を介して、アーカイブとカスタム JumpStart ファイルをダウンロードします。
  11. インストールプログラムは、カスタム JumpStart インストールを実行して、フラッシュアーカイブをクライアントにインストールします。

## WAN ブートインストール時のデータの保護

WAN ブートインストールでは、ハッシュキー、暗号化鍵、およびデジタル証明書を使って、インストール中にシステムデータを保護できます。ここでは、WAN ブートインストールでサポートされている各種のデータ保護方法について簡単に説明します。

## ハッシュキーによるデータ完全性のチェック

WAN ブートサーバーからクライアントに転送するデータを保護するために、HMAC (Hashed Message Authentication Code) キーを生成します。このハッシュキーを、WAN ブートサーバーとクライアントの両方にインストールします。WAN ブートサーバーはこのキーを使って、クライアントに転送するデータに署名します。クライアントはこのキーを使って、WAN ブートサーバーから転送されるデータの完全性を確認します。クライアントにハッシュキーをインストールすると、クライアントは以降の WAN ブートインストールにこのキーを使用します。

ハッシュキーの使用方法については、548 ページの「ハッシュキーと暗号化鍵の作成」を参照してください。

## 暗号化鍵によるデータの暗号化

WAN ブートインストールでは、WAN ブートサーバーからクライアントに転送するデータを暗号化できます。WAN ブートのユーティリティを使って、3DES (Triple Data Encryption Standard) または AES (Advanced Encryption Standard) の暗号化鍵を作成できます。この鍵を、WAN ブートサーバーとクライアントの両方に渡します。WAN ブートサーバーはこの暗号化鍵を使って、クライアントに転送するデータを暗号化します。クライアントはこの鍵を使って、インストール時に暗号化されて転送された構成ファイルとセキュリティファイルを、復号化できます。

クライアントに暗号化鍵をインストールすると、クライアントは以降の WAN ブートインストールにこの鍵を使用します。

サイトで暗号化鍵の使用が許可されていない場合もあります。サイトで暗号化を使用できるかどうかについては、サイトのセキュリティ管理者に問い合わせてください。サイトで暗号化を使用できる場合は、3DES 暗号化鍵または AES 暗号化鍵のどちらを使用すべきかを、セキュリティ管理者に尋ねてください。

暗号化鍵の使用方法については、548 ページの「ハッシュキーと暗号化鍵の作成」を参照してください。

## HTTPS によるデータの保護

WAN ブートでは、WAN ブートサーバーとクライアントの間のデータ転送に HTTPS (Secure Sockets Layer を介した HTTP) を使用できます。HTTPS を使用すると、サーバーに対して、あるいはサーバーとクライアントの両方に対して、インストール時に身分証明を行うよう要求できます。また、HTTPS では、インストール時にサーバーからクライアントに転送されるデータが暗号化されます。

HTTPS では、ネットワーク上でデータを交換するシステムに対して、デジタル証明書による認証が行われます。デジタル証明書は、オンライン通信を行うときにシステム (サーバーまたはクライアント) が信頼できるシステムであることを示すためのファイルです。外部の認証局に依頼してデジタル証明書を取得するか、独自の証明書と認証局を作成します。

クライアントがサーバーを信頼してサーバーからのデータを受け入れるようにするには、サーバーにデジタル証明書をインストールする必要があります。次に、この証明書を信頼するようにクライアントに指示します。サーバーに対して身分証明を行うよう、クライアントに要求することもできます。そのためには、クライアントにデジタル証明書を用意します。次に、インストール時にクライアントが証明書を提出したらその証明書の署名者を受け入れるように、サーバーに指示します。

インストール時にデジタル証明書を使用するには、HTTPS を使用するように Web サーバーを構成する必要があります。HTTPS の使用方法については、Web サーバーのマニュアルを参照してください。

WAN ブートインストールでデジタル証明書を使用するための要件については、527 ページの「デジタル証明書の要件」を参照してください。WAN ブートインストールでデジタル証明書を使用する方法については、546 ページの「サーバー認証とクライアント認証のためのデジタル証明書の使用」を参照してください。

---

## WAN ブートでサポートされているセキュリティ構成 (概要)

WAN ブートでは、さまざまなレベルのセキュリティがサポートされています。WAN ブートでサポートされているセキュリティ機能を組み合わせて使用することで、ネットワークのニーズに対応できます。より安全な構成にするほど、多くの管理が必要になりますが、システムデータをより広範に保護できます。パブリックネットワーク経由でインストールを行う、高いセキュリティを必要とするシステムには、517 ページの「セキュリティ保護された WAN ブートインストール構成」で説明する構成を選択できます。半私設のネットワーク上にあり、それほどセキュリティを必要としないシステムには、518 ページの「セキュリティ保護されていない WAN ブートインストール構成」で説明する構成を検討してください。

ここでは、WAN ブートインストールのセキュリティレベルを設定するための各種構成について簡単に説明します。また、これらの構成に必要なセキュリティメカニズムについても説明します。

### セキュリティ保護された WAN ブートインストール構成

この構成は、サーバーとクライアントの間で交換されるデータの完全性を保護し、内容の機密性を保つために役立ちます。この構成は、HTTPS 接続を使用するとともに、クライアント構成ファイルを暗号化するために 3DES または AES アルゴリズムを使用します。また、この構成では、サーバーはインストール時にクライアントに対して身分証明を行うよう要求されます。セキュリティ保護された WAN ブートインストールを行うには、次のセキュリティ機能が必要です。

- WAN ブートサーバーとインストールサーバーで、HTTPS が有効になっていること
- WAN ブートサーバーとクライアントに、HMAC SHA1 ハッシュキーが、インストールされていること
- WAN ブートサーバーとクライアントに、3DES または AES 暗号化鍵がインストールされていること
- WAN ブートサーバーに関する認証局のデジタル証明書

インストール時にクライアントの認証も行う場合は、次のセキュリティ機能を使用する必要があります。

- WAN ブートサーバーの非公開鍵
- クライアントのデジタル証明書

この構成を使ってインストールを行うために必要な作業については、表 43-1 を参照してください。

## セキュリティ保護されていない WAN ブートインストール構成

この構成では、管理に必要な労力は最小限に抑えられますが、Web サーバーからクライアントへのデータ転送のセキュリティは最も低くなります。ハッシュキー、暗号化鍵、およびデジタル証明書を作成する必要はありません。HTTPS を使用するように Web サーバーを構成する必要もありません。ただし、この構成によるインストールでは、インストールデータとファイルは HTTP 接続を介して転送されるので、ネットワーク上での妨害に対して無防備になります。

転送されたデータの完全性をクライアントでチェックできるようにするには、この構成とともに HMAC SHA1 ハッシュキーを使用します。ただし、フラッシュアーカイブはハッシュキーで保護されません。インストール時にサーバーとクライアントの間で転送されるアーカイブは、セキュリティ保護されません。

この構成を使ってインストールを行うために必要な作業については、表 43-2 を参照してください。

## 第 42 章

# WAN ブートによるインストールの準備 (計画)

この章では、WAN ブートインストールを行うためにネットワークを準備する方法について説明します。この章の内容は次のとおりです。

- 519 ページの「WAN ブートの要件とガイドライン」
- 528 ページの「WAN ブートのセキュリティ限界」
- 528 ページの「WAN ブートインストールに必要な情報の収集」

## WAN ブートの要件とガイドライン

ここでは、WAN ブートインストールを実行するためのシステム要件について説明します。

表 42-1 WAN ブートインストールを行うためのシステム要件

システムと説明	要件
WAN ブートサーバー – wanboot プログラム、構成ファイルとセキュリティファイル、および WAN ブートミニルートを提供する Web サーバー	<ul style="list-style-type: none"><li>■ オペレーティングシステム – Solaris 9 12/03 オペレーティング環境、またはその互換バージョン</li><li>■ Web サーバーとして構成されていること</li><li>■ Web サーバーソフトウェアで HTTP 1.1 がサポートされていること</li><li>■ デジタル証明書を使用する場合は、Web サーバーソフトウェアで HTTPS がサポートされていること</li></ul>

表 42-1 WAN ブートインストールを行うためのシステム要件 (続き)

システムと説明	要件
インストールサーバー-クライアントのインストールに必要なフラッシュアーカイブとカスタム JumpStart ファイルを提供する	<ul style="list-style-type: none"> <li>■ ディスク容量-各フラッシュアーカイブに必要な容量</li> <li>■ メディアドライブ-CD-ROM ドライブまたは DVD-ROM ドライブ</li> <li>■ オペレーティングシステム-Solaris 9 12/03 オペレーティング環境、またはその互換バージョン</li> </ul> <p>WAN ブートサーバーとは別のシステムで稼働している場合、インストールサーバーは次の追加要件を満たす必要がある</p> <ul style="list-style-type: none"> <li>■ Web サーバーとして構成されていること</li> <li>■ Web サーバーソフトウェアで HTTP 1.1 がサポートされていること</li> <li>■ デジタル証明書を使用する場合は、Web サーバーソフトウェアで HTTPS がサポートされていること</li> </ul>
クライアントシステム-WAN 経由でインストールを行う対象のリモートシステム	<ul style="list-style-type: none"> <li>■ メモリー-256M バイト以上の RAM</li> <li>■ CPU-UltraSPARC II プロセッサ以上</li> <li>■ ハードディスク-2G バイト以上のハードディスク容量</li> <li>■ OBP-WAN ブート対応の PROM</li> </ul> <p>適切な PROM を持っていないクライアントには、CD-ROM ドライブが必要</p> <p>クライアントの PROM が WAN ブートに対応しているかどうかを調べる方法については、567 ページの「クライアント OBP での WAN ブート対応を確認する方法」を参照</p>
(省略可能) DHCP サーバー-DHCP サーバーを使ってクライアント構成情報を提供できる	<p>SunOS DHCP サーバーを使用している場合は、次のどれかの作業を実行する必要がある</p> <ul style="list-style-type: none"> <li>■ サーバーを EDHCP サーバーにアップグレードする</li> <li>■ Sun ベンダーオプションの名前を変更して、オプションに対する 8 文字の制限を満たすようにする。WAN インストール固有の Sun ベンダーオプションの詳細については、563 ページの「(省略可能) DHCP による構成情報の提供」を参照</li> </ul> <p>DHCP サーバーがクライアントとは異なるサブネットにある場合は、BOOTP リレーエージェントを構成する必要がある。BOOTP リレーエージェントの構成方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP サービスの構成 (手順)」を参照</p>



表 42-1 WAN ブートインストールを行うためのシステム要件 (続き)

システムと説明	要件
(省略可能) ログサーバー - デフォルトでは、WAN インストール時のブートログメッセージおよびインストールログメッセージは、すべてクライアントのコンソールに表示される。これらのメッセージを別のシステムに表示するには、ログサーバーとして使用するシステムを指定する	Web サーバーとして構成されている必要がある 注 - インストール時に HTTPS を使用する場合は、WAN ブートサーバーと同じシステムにログサーバーを置く必要がある
(省略可能) プロキシサーバー - インストールデータとファイルのダウンロード時に HTTP プロキシを使用するように WAN ブート機能を構成できる	インストールで HTTPS を使用する場合は、HTTPS トンネリングを行うようにプロキシサーバーを構成する必要がある

## Web サーバーソフトウェアの要件とガイドライン

WAN ブートサーバーとインストールサーバーで使用する Web サーバーソフトウェアは、次の要件を満たす必要があります。

- オペレーティングシステム - WAN ブートでは、wanboot-cgi という CGI (Common Gateway Interface) プログラムが、クライアントマシンが受け付ける特定のフォーマットにデータやファイルを変換します。これらのスクリプトを使って WAN ブートインストールを実行するには、Solaris 9 12/03 オペレーティング環境またはその互換バージョンで Web サーバーソフトウェアを実行する必要があります。
- ファイルサイズの制限 - Web サーバーソフトウェアによっては、HTTP を介して転送できるファイルサイズが制限される場合もあります。Web サーバーのマニュアルを参照して、フラッシュアーカイブを転送できることを確認してください。
- SSL サポート - WAN ブートインストールで HTTPS を使用するには、Web サーバーソフトウェアで SSL バージョン 3 がサポートされている必要があります。

## サーバー構成オプション

WAN ブートに必要なサーバーの構成をカスタマイズすることで、ネットワークのニーズに対応できます。すべてのサーバーを単一のシステムに置くことも、複数のシステムに置くこともできます。

- 単一のサーバー - WAN ブートのデータとファイルを 1 台のシステムに集中化させたい場合は、すべてのサーバーを同じマシンで稼働させることができます。各種のサーバーを 1 台のシステムで管理できるほか、1 台のシステムを Web サーバーとして構成するだけで済みます。ただし、単一のサーバーでは、多数の WAN ブート

インストールが同時に発生した場合に、必要なトラフィック量をサポートできないことがあります。

- 複数のサーバー–インストールデータとファイルをネットワーク上に分散させたい場合は、これらのサーバーを複数のマシンで稼働させることができます。たとえば、中心となる WAN ブートサーバーを 1 台セットアップし、複数のインストールサーバーを構成してフラッシュアーカイブをネットワーク上に分散できます。インストールサーバーとログサーバーを別々のマシンで稼働させる場合は、どちらのサーバーも Web サーバーとして構成する必要があります。

## ドキュメントルートディレクトリへのインストールファイルと構成ファイルの保存

WAN ブートインストール時に、`wanboot-cgi` プログラムによって次のファイルが転送されます。

- `wanboot` プログラム
- WAN ブートミニルート
- カスタム JumpStart ファイル
- フラッシュアーカイブ

`wanboot-cgi` プログラムでこれらのファイルを転送できるようにするには、Web サーバーソフトウェアがアクセスできるディレクトリに、これらのファイルを保存する必要があります。たとえば、Web サーバーのドキュメントルートにこれらのファイルを置くと、これらのファイルへのアクセスが可能になります。

ドキュメントルートは、Web サーバー上の主要なドキュメントディレクトリであり、クライアントに公開するファイルはここに保存されます。Web サーバーソフトウェアを使って、このディレクトリの名前や構成を変更できます。Web サーバー上のドキュメントルートディレクトリを設定する方法については、Web サーバーのマニュアルを参照してください。

ドキュメントルートディレクトリにいくつかのサブディレクトリを作成して、それぞれ異なるインストールファイルと構成ファイルを保存することもできます。たとえば、インストール対象であるクライアントのグループごとに、固有のサブディレクトリを作成します。また、ネットワーク上に何種類かのリリースの Solaris オペレーティング環境をインストールする場合は、リリースごとにサブディレクトリを作成します。

図 42-1 は、ドキュメントルートディレクトリの基本的な構造の例を示しています。この例で、WAN ブートサーバーとインストールサーバーは同じマシンに置かれています。このサーバーでは、Apache Web サーバーソフトウェアが実行されています。

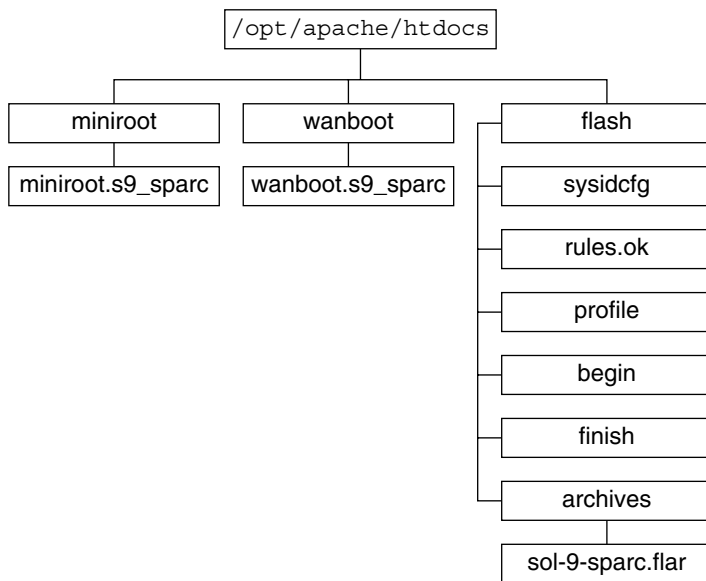


図 42-1 ドキュメントルートディレクトリの構造の例

この例のドキュメントディレクトリは、次のような構造を使用しています。

- /opt/apache/htdocs ディレクトリは、ドキュメントルートディレクトリです。
- WAN ブートミニルート (miniroot) ディレクトリには、WAN ブートミニルートが置かれています。
- wanboot ディレクトリには、wanboot プログラムが置かれています。
- Solaris フラッシュ (flash) ディレクトリには、クライアントのインストールに必要なカスタム JumpStart ファイルと、サブディレクトリ archives が置かれています。archives ディレクトリには、Solaris 9 のフラッシュアーカイブが置かれています。

---

注 - WAN ブートサーバーとインストールサーバーがそれぞれ別のシステムで稼働している場合は、flash ディレクトリをインストールサーバーに置くこともできます。WAN ブートサーバーがこれらのファイルやディレクトリにアクセスできることを確認してください。

---

ドキュメントルートディレクトリの作成方法については、Web サーバーのマニュアルを参照してください。インストールファイルの作成および保存の方法については、550 ページの「カスタム JumpStart インストールファイルの作成」を参照してください。

## /etc/netboot ディレクトリへの構成情報とセキュリティ情報の保存

/etc/netboot ディレクトリには、WAN ブートインストールに必要な、構成情報、非公開鍵、デジタル証明書、および認証局が保存されます。ここでは、WAN ブートインストールをカスタマイズするために /etc/netboot ディレクトリ内に作成できるファイルとディレクトリについて説明します。

### WAN ブートインストールの適用範囲のカスタマイズ

インストール時に wanboot-cgi プログラムは、WAN ブートサーバーの /etc/netboot ディレクトリ内でクライアント情報を検索します。wanboot-cgi プログラムは、この情報を WAN ブートファイルシステムに変換してから、WAN ブートファイルシステムをクライアントに転送します。/etc/netboot ディレクトリ内にサブディレクトリを作成することで、WAN ブートインストールの適用範囲をカスタマイズできます。次のディレクトリ構造を使って、インストール対象のクライアント間で構成情報をどのように共有するかを定義します。

- グローバルな構成 – ネットワーク上のすべてのクライアントで構成情報を共有するには、共有する構成ファイルを /etc/netboot ディレクトリに保存します。
- ネットワーク固有の構成 – 特定のサブネット上のマシンだけで構成情報を共有するには、共有する構成ファイルを /etc/netboot ディレクトリのサブディレクトリに保存します。サブディレクトリは、次の規則に従って名前を付けてください。

`/etc/netboot/net-ip`

この例で、*net-ip* はクライアントのサブネットの IP アドレスです。たとえば、192.168.255.0 という IP アドレスを持つサブネット上のすべてのシステムで構成ファイルを共有するには、`/etc/netboot/192.168.255.0` というディレクトリを作成します。その後、このディレクトリに構成ファイルを保存します。

- クライアント固有の構成 – 特定のクライアントだけでブートファイルシステムを使用するには、ブートファイルシステムのファイルを /etc/netboot のサブディレクトリに保存します。サブディレクトリは、次の規則に従って名前を付けてください。

`/etc/netboot/net-ip/client-ID`

この例で、*net-ip* はサブネットの IP アドレスです。*client-ID* は、DHCP サーバーによって割り当てられるクライアント ID か、ユーザー指定のクライアント ID です。たとえば、サブネット 192.168.255.0 にあって 010003BA152A42 というクライアント ID を持つシステムで、特定の構成ファイルを使用するには、`/etc/netboot/192.168.255.0/010003BA152A42` というディレクトリを作成します。その後、該当するファイルをこのディレクトリに保存します。

## /etc/netboot ディレクトリにおけるセキュリティ情報 と構成情報の指定

次のファイルを作成して /etc/netboot ディレクトリに保存することで、セキュリティ情報と構成情報を指定します。

- `wanboot.conf` – このファイルは、WAN ブートインストール用のクライアント構成情報を指定します。
- システム構成ファイル (`system.conf`) – このシステム構成ファイルは、クライアントの `sysidcfg` ファイルおよびカスタム `JumpStart` ファイルの場所を指定します。
- `keystore` – このファイルには、クライアントの HMAC SHA1 ハッシュキー、3DES または AES 暗号化鍵、および SSL 非公開鍵が保存されます。
- `truststore` – このファイルには、クライアントが信頼すべき、認証局のデジタル証明書が保存されます。これら信頼できる証明書に従って、クライアントはインストール時にサーバーを信頼します。
- `certstore` – このファイルには、クライアントのデジタル証明書が保存されます。

---

注 – `certstore` ファイルは、クライアント ID のディレクトリに置く必要があります。/etc/netboot ディレクトリのサブディレクトリに関する詳細は、524 ページの「WAN ブートインストールの適用範囲のカスタマイズ」を参照してください。

---

これらのファイルの作成方法と保存方法については、次の手順を参照してください。

- 557 ページの「システム構成ファイルの作成」
- 559 ページの「`wanboot.conf` ファイルの作成」
- 548 ページの「ハッシュキーと暗号化鍵の作成」
- 546 ページの「サーバー認証とクライアント認証のためのデジタル証明書の使用」

## /etc/netboot ディレクトリにおけるセキュリティ情報 と構成情報の共有

ネットワーク上のクライアントに対してインストールを行うとき、いくつかのクライアントで、あるいはすべてのサブネットで、セキュリティファイルと構成ファイルを共有することもできます。これらのファイルを共有するには、`/etc/netboot/net-ip/client-ID`、`/etc/netboot/net-ip`、および `/etc/netboot` の各ディレクトリに構成情報を置きます。インストール時に、`wanboot-cgi` プログラムはこれらのディレクトリから構成情報を検索し、クライアントに最もよく適合する構成情報を使用します。

`wanboot-cgi` プログラムは、次の順序でクライアント情報を検索します。

1. /etc/netboot/net-ip/client-ID – wanboot-cgi プログラムはまず、クライアントマシンに固有の構成情報を検索します。/etc/netboot/net-ip/client-ID ディレクトリにすべてのクライアント構成情報が揃っている場合、wanboot-cgi プログラムが /etc/netboot ディレクトリの他の場所の構成情報を検索することはありません。
2. /etc/netboot/net-ip – 必要な情報が /etc/netboot/net-ip/client-ID ディレクトリに揃っていない場合、wanboot-cgi プログラムは /etc/netboot/net-ip ディレクトリでサブネット構成情報を検索します。
3. /etc/netboot – 必要な情報が /etc/netboot/net-ip ディレクトリにも見つからない場合、wanboot-cgi プログラムは /etc/netboot ディレクトリでグローバルな構成情報を検索します。

図 42-2 は、/etc/netboot ディレクトリを設定して WAN ブートインストールをカスタマイズする方法を示しています。

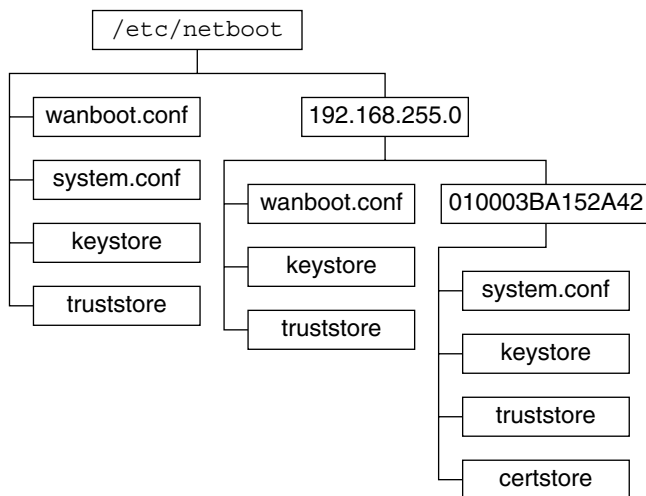


図 42-2 /etc/netboot ディレクトリの例

図 42-2 の /etc/netboot ディレクトリレイアウトでは、次のような WAN ブートインストールを実行できます。

- クライアント 010003BA152A42 に対してインストールを行うときは、/etc/netboot/192.168.255.0/010003BA152A42 ディレクトリにある次のファイルが wanboot-cgi プログラムによって使用されます。
  - system.conf
  - keystore
  - truststore
  - certstore

次に、/etc/netboot/192.168.255.0 ディレクトリにある wanboot.conf ファイルが、wanboot-cgi プログラムによって使用されます。

- 192.168.255.0 サブネット上のクライアントに対してインストールを行うときは、`/etc/netboot/192.168.255.0` ディレクトリにある `wanboot.conf`、`keystore`、および `truststore` の各ファイルが、`wanboot-cgi` プログラムによって使用されます。次に、`/etc/netboot` ディレクトリにある `system.conf` ファイルが、`wanboot-cgi` プログラムによって使用されます。
- 192.168.255.0 サブネット上にないクライアントマシンに対してインストールを行うときは、`/etc/netboot` ディレクトリにある次のファイルが、`wanboot-cgi` プログラムによって使用されます。
  - `wanboot.conf`
  - `system.conf`
  - `keystore`
  - `truststore`

## wanboot-cgi プログラムの保存

`wanboot-cgi` プログラムは、WAN ブートサーバーからクライアントにデータとファイルを転送します。このプログラムは、WAN ブートサーバー上でクライアントがアクセスできるディレクトリに置く必要があります。たとえば、WAN ブートサーバーの `cgi-bin` ディレクトリにこのプログラムを置くと、クライアントがこのプログラムにアクセスできるようになります。`wanboot-cgi` プログラムを CGI プログラムとして使用するよう Web サーバーソフトウェアを構成する必要がある場合もあります。CGI プログラムの要件については、Web サーバーのマニュアルを参照してください。

## デジタル証明書の要件

WAN ブートインストールのセキュリティを高めるには、デジタル証明書を使ってサーバーとクライアントの認証を有効にします。WAN ブートでは、オンライントランザクションの間に、デジタル証明書を使ってサーバーまたはクライアントの識別情報が確立されます。デジタル証明書は認証局 (CA) によって発行されます。これらの証明書には、シリアル番号、有効期限、証明書所有者の公開鍵のコピー、および認証局のデジタル署名が含まれています。

サーバーに対して、あるいはサーバーとクライアントの両方に対して、インストール時に認証を行うには、サーバーにデジタル証明書をインストールする必要があります。デジタル証明書を使用するときは、次のガイドラインに従ってください。

- デジタル証明書を使用する場合、デジタル証明書は PKCS#12 (Public-Key Cryptography Standards #12) ファイルの一部としてフォーマットされている必要がある
- 独自の証明書を作成する場合は、PKCS#12 ファイルとして作成する必要がある
- 第三者機関である認証局から証明書を取得する場合は、PKCS#12 フォーマットの証明書を依頼する

WAN ブートインストールで PKCS#12 証明書を使用する方法については、546 ページの「サーバー認証とクライアント認証のためのデジタル証明書の使用」を参照してください。

---

## WAN ブートのセキュリティ限界

WAN ブートには各種のセキュリティ機能が用意されていますが、次のような潜在的問題には対応していません。

- サービス妨害攻撃 – サービス妨害 (DoS) 攻撃にはさまざまな形式がありますが、その目的はユーザーが特定のサービスにアクセスできないようにすることです。たとえば、大量のデータでネットワークに負担をかけたり、限られたリソースを強引に消費したりする DoS 攻撃があります。また、システム間で転送中のデータに対して操作を加える DoS 攻撃もあります。WAN ブートでは、DoS 攻撃に対するサーバーやクライアントの保護は行われません。
- サーバー上のバイナリの破壊 – WAN ブートインストールでは、インストールの実行前に WAN ブートミニルートやフラッシュアーカイブの完全性がチェックされることはありません。インストールを実行する前に、<http://sunsolve.sun.com> の Solaris フィンガープリントデータベースと比較して、Solaris バイナリの完全性を確認してください。
- 暗号化鍵とハッシュキーの機密性 – WAN ブートで暗号化鍵やハッシュキーを使用する場合は、インストール時にキーの値をコマンド行に入力する必要があります。ネットワークに必要な注意事項を守り、キーの値を機密に保つようにしてください。
- ネットワークのネームサービスへの危害 – ネットワークでネームサービスを使用する場合は、WAN ブートインストールを実行する前に、ネームサーバーの完全性を確認してください。

---

## WAN ブートインストールに必要な情報の収集

WAN ブートインストールを行うためにネットワークを構成するには、さまざまな情報を収集する必要があります。WAN 経由でのインストールを準備するときに、この情報を書きとめておくとよいでしょう。

ネットワークについて WAN ブートインストール情報を記録するには、次のワークシートを使用してください。

- 表 42-2



■ 表 42-3

表 42-2 サーバー情報を収集するためのワークシート

必要な情報	記入欄
インストールサーバーの情報 ■ インストールサーバー上の WAN ブートミニルートへのパス ■ インストールサーバー上のカスタム JumpStart ファイルへのパス	
WAN ブートサーバーの情報 ■ WAN ブートサーバー上の wanboot プログラムへのパス ■ WAN ブートサーバー上の wanboot-cgi プログラムの URL ■ WAN ブートサーバー上の /etc/netboot 階層にあるクライアントのサブディレクトリへのパス ■ (省略可能) PKCS#12 証明書ファイルのファイル名 ■ (省略可能) WAN ブートサーバー以外で、WAN インストールに必要なすべてのマシンのホスト名 ■ (省略可能) ネットワークのプロキシサーバーの IP アドレスと TCP ポート番号	
オプションサーバーの情報 ■ ログサーバー上の bootlog-cgi スクリプトの URL ■ ネットワークのプロキシサーバーの IP アドレスと TCP ポート番号	

表 42-3 クライアント情報を収集するためのワークシート

インフォメーション	記入欄
クライアントのサブネットの IP アドレス	
クライアントのルーターの IP アドレス	
クライアントの IP アドレス	
クライアントのサブネットマスク	
クライアントのホスト名	
クライアントの MAC アドレス	



## 第 43 章

# WAN ブートによるインストールの準備 (作業)

---

この章では、WAN ブートインストールを行うためにネットワークを準備する方法について説明します。必要な作業は次のとおりです。

- 531 ページの「広域ネットワーク経由でインストールを行うための準備 (作業マップ)」
- 535 ページの「WAN ブートサーバーの構成」
- 550 ページの「カスタム JumpStart インストールファイルの作成」
- 556 ページの「構成ファイルの作成」
- 563 ページの「(省略可能) DHCP による構成情報の提供」
- 544 ページの「(省略可能) WAN ブートログサーバーの構成」

---

## 広域ネットワーク経由でインストールを行うための準備 (作業マップ)

次の表は、WAN ブートインストールの準備に必要な作業の一覧です。

- セキュリティ保護された WAN ブートインストールの準備に必要な作業の一覧については、表 43-1 を参照してください。

HTTPS によるセキュリティ保護された WAN ブートインストールについては、517 ページの「セキュリティ保護された WAN ブートインストール構成」を参照してください。

- セキュリティ保護されていない WAN ブートインストールの準備に必要な作業の一覧については、表 43-2 を参照してください。

セキュリティ保護されていない WAN ブートインストールについては、518 ページの「セキュリティ保護されていない WAN ブートインストール構成」を参照してください。

DHCP サーバーやログサーバーを使用するには、表の末尾にある追加作業を実行する必要があります。

表 43-1 作業マップ: セキュリティ保護された WAN ブートインストールを実行するための準備

作業	説明	参照先
インストールで使用するセキュリティ機能を決定する	セキュリティ機能と構成について検討し、WAN ブートインストールで使用するセキュリティのレベルを決定する	515 ページの「WAN ブートインストール時のデータの保護」 517 ページの「WAN ブートでサポートされているセキュリティ構成 (概要)」
WAN ブートインストール情報を収集する	ワークシートを使って、WAN ブートインストールの実行に必要なすべての情報を記録する	528 ページの「WAN ブートインストールに必要な情報の収集」
WAN ブートサーバーにドキュメントルートディレクトリを作成する	構成ファイルとインストールファイルを提供するために、ドキュメントルートディレクトリと必要に応じてサブディレクトリを作成する	536 ページの「ドキュメントルートディレクトリの作成」
WAN ブートミニルートを作成する	setup_install_server コマンドを使って、WAN ブートミニルートを作成する	537 ページの「SPARC: WAN ブートミニルートを作成する方法」
WAN ブートサーバーに wanboot プログラムをインストールする	WAN ブートサーバーのドキュメントルートディレクトリに wanboot プログラムをコピーする	539 ページの「WAN ブートサーバーへの wanboot プログラムのインストール」
WAN ブートサーバーに wanboot-cgi プログラムをインストールする	WAN ブートサーバーの CGI ディレクトリに wanboot-cgi プログラムをコピーする	543 ページの「WAN ブートサーバーに wanboot-cgi プログラムをコピーする方法」
(省略可能) ログサーバーを構成する	ブートおよびインストールのログメッセージを表示するための専用システムを構成する	544 ページの「(省略可能) WAN ブートログサーバーの構成」
/etc/netboot 階層を設定する	WAN ブートインストールに必要な構成ファイルとセキュリティファイルを /etc/netboot 階層に格納する	540 ページの「WAN ブートサーバーに /etc/netboot ディレクトリを作成する」
WAN ブートインストールのセキュリティを高めるために、HTTPS を使用するように Web サーバーを構成する	HTTPS を使って WAN インストールを実行するための Web サーバー要件に合わせる	545 ページの「(省略可能) HTTPS によるデータの保護」

表 43-1 作業マップ: セキュリティ保護された WAN ブートインストールを実行するための準備 (続き)

作業	説明	参照先
WAN ブートインストールのセキュリティを高めるために、デジタル証明書の形式を変更する	PKCS#12 ファイルを、WAN インストールで使用できるように非公開鍵と証明書に分割する	546 ページの「信頼できる証明書およびクライアントの非公開鍵の作成」
WAN ブートインストールのセキュリティを高めるために、ハッシュキーと暗号化鍵を作成する	wanbootutil keygen コマンドを使って、HMAC SHA1、3DES、または AES キーを作成する	548 ページの「ハッシュキーと暗号化鍵を作成する方法」
フラッシュアーカイブを作成する	flar create コマンドを使って、クライアントにインストールするソフトウェアのアーカイブを作成する	551 ページの「フラッシュアーカイブを作成する方法」
カスタム JumpStart インストール用のインストールファイルを作成する	テキストエディタを使って、次のファイルを作成する <ul style="list-style-type: none"> <li>■ sysidcfg</li> <li>■ プロファイル</li> <li>■ rules.ok</li> <li>■ begin スクリプト</li> <li>■ finish スクリプト</li> </ul>	552 ページの「sysidcfg ファイルの作成」 553 ページの「プロファイルの作成」 554 ページの「rules ファイルの作成」 556 ページの「(省略可能) begin スクリプトと finish スクリプトの作成」
システム構成ファイルを作成する	system.conf ファイルに構成情報を設定する	557 ページの「システム構成ファイルを作成する方法」
WAN ブート構成ファイルを作成する	wanboot.conf ファイルに構成情報を設定する	560 ページの「wanboot.conf ファイルを作成する方法」
(省略可能) WAN ブートインストールをサポートするように DHCP サーバーを構成する	DHCP サーバーに Sun ベンダーオプションとマクロを設定する	75 ページの「DHCP サービスによるシステム構成情報の事前設定 (作業)」

表 43-2 作業マップ: セキュリティ保護されていない WAN ブートインストールを実行するための準備

作業	説明	参照先
インストールで使用するセキュリティ機能を決定する	セキュリティ機能と構成について検討し、WAN ブートインストールで使用するセキュリティのレベルを決定する	515 ページの「WAN ブートインストール時のデータの保護」 517 ページの「WAN ブートでサポートされているセキュリティ構成 (概要)」
WAN ブートインストール情報を収集する	ワークシートを使って、WAN ブートインストールの実行に必要なすべての情報を記録する	528 ページの「WAN ブートインストールに必要な情報の収集」
WAN ブートサーバーにドキュメントルートディレクトリを作成する	構成ファイルとインストールファイルを提供するために、ドキュメントルートディレクトリと必要に応じてサブディレクトリを作成する	536 ページの「ドキュメントルートディレクトリの作成」
WAN ブートミニルートを作成する	<code>setup_install_server</code> コマンドを使って、WAN ブートミニルートを作成する	537 ページの「SPARC: WAN ブートミニルートを作成する方法」
WAN ブートサーバーに <code>wanboot</code> プログラムをインストールする	WAN ブートサーバーのドキュメントルートディレクトリに <code>wanboot</code> プログラムをコピーする	539 ページの「WAN ブートサーバーへの <code>wanboot</code> プログラムのインストール」
WAN ブートサーバーに <code>wanboot-cgi</code> プログラムをインストールする	WAN ブートサーバーの CGI ディレクトリに <code>wanboot-cgi</code> プログラムをコピーする	543 ページの「WAN ブートサーバーに <code>wanboot-cgi</code> プログラムをコピーする方法」
(省略可能) ログサーバーを構成する	ブートおよびインストールのログメッセージを表示するための専用システムを構成する	544 ページの「(省略可能) WAN ブートログサーバーの構成」
<code>/etc/netboot</code> 階層を設定する	WAN ブートインストールに必要な構成ファイルとセキュリティファイルを <code>/etc/netboot</code> 階層に格納する	540 ページの「WAN ブートサーバーに <code>/etc/netboot</code> ディレクトリを作成する」

表 43-2 作業マップ: セキュリティ保護されていない WAN ブートインストールを実行するための準備 (続き)

作業	説明	参照先
(省略可能) ハッシュキーを作成する	<p>wanbootutil keygen コマンドを使って、HMAC SHA1 キーを作成する</p> <p>セキュリティ保護されていないインストールで、データの完全性をチェックする場合は、この作業を実行して HMAC SHA1 ハッシュキーを作成する</p>	548 ページの「ハッシュキーと暗号化鍵を作成する方法」
フラッシュアーカイブを作成する	flar create コマンドを使って、クライアントにインストールするソフトウェアのアーカイブを作成する	551 ページの「フラッシュアーカイブを作成する方法」
カスタム JumpStart インストール用のインストールファイルを作成する	<p>テキストエディタを使って、次のファイルを作成する</p> <ul style="list-style-type: none"> <li>■ sysidcfg</li> <li>■ プロファイル</li> <li>■ rules.ok</li> <li>■ begin スクリプト</li> <li>■ finish スクリプト</li> </ul>	<p>552 ページの「sysidcfg ファイルの作成」</p> <p>553 ページの「プロファイルの作成」</p> <p>554 ページの「rules ファイルの作成」</p> <p>556 ページの「(省略可能) begin スクリプトと finish スクリプトの作成」</p>
システム構成ファイルを作成する	system.conf ファイルに構成情報を設定する	557 ページの「システム構成ファイルを作成する方法」
WAN ブート構成ファイルを作成する	wanboot.conf ファイルに構成情報を設定する	560 ページの「wanboot.conf ファイルを作成する方法」
(省略可能) WAN ブートインストールをサポートするように DHCP サーバーを構成する	DHCP サーバーに Sun ベンダーオプションとマクロを設定する	75 ページの「DHCP サービスによるシステム構成情報の事前設定 (作業)」

## WAN ブートサーバーの構成

WAN ブートサーバーは、WAN ブートインストール時にブートデータと構成データを提供する Web サーバーです。WAN ブートサーバーのシステム要件のリストについては、表 42-1 を参照してください。

ここでは、WAN ブートインストールを行うために WAN ブートサーバーを構成する方法について説明します。必要な作業は次のとおりです。

- 536 ページの「ドキュメントルートディレクトリの作成」
- 536 ページの「WAN ブートミニルートの作成」
- 539 ページの「WAN ブートサーバーへの wanboot プログラムのインストール」
- 540 ページの「WAN ブートサーバーに /etc/netboot ディレクトリを作成する」
- 543 ページの「WAN ブートサーバーへの WAN ブート CGI プログラムのコピー」
- 545 ページの「(省略可能) HTTPS によるデータの保護」

## ドキュメントルートディレクトリの作成

構成ファイルとインストールファイルを提供するには、WAN ブートサーバーの Web サーバーソフトウェアがこれらのファイルにアクセスできるようにする必要があります。たとえば、WAN ブートサーバーのドキュメントルートディレクトリにこれらのファイルを置くと、これらのファイルへのアクセスが可能になります。

構成ファイルとインストールファイルの提供にドキュメントルートディレクトリを使用するには、このディレクトリを作成する必要があります。ドキュメントルートディレクトリの作成方法については、Web サーバーのマニュアルを参照してください。ドキュメントルートディレクトリの設計方法については、522 ページの「ドキュメントルートディレクトリへのインストールファイルと構成ファイルの保存」を参照してください。

## WAN ブートミニルートの作成

WAN ブートでは、WAN ブートインストール用に変更された特別な Solaris ミニルートが使用されます。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入っています。WAN ブートインストールを実行するには、Solaris DVD または Solaris SOFTWARE 1 of 2 CD から、このミニルートを WAN ブートサーバーにコピーする必要があります。-w オプションを指定して `setup_install_server` コマンドを実行し、Solaris ソフトウェアのメディアからシステムのハードディスクに WAN ブートミニルートをコピーします。

次の手順では、SPARC メディアを使って SPARC WAN ブートミニルートを作成します。x86 ベースのサーバーから SPARC WAN ブートミニルートを提供するには、まず SPARC マシンにミニルートを作成する必要があります。次に、作成したミニルートを、x86 ベースのサーバーのドキュメントルートディレクトリにコピーします。

`setup_install_server` コマンドの詳細は、第 15 章を参照してください。



## ▼ SPARC: WAN ブートミニルートを作成する方法

この手順では、WAN ブートサーバーでボリュームマネージャを実行していると仮定します。ボリュームマネージャを使用していない場合は、ボリュームマネージャを使用せずにリムーバブルメディアを管理する方法について、『Solaris のシステム管理 (基本編)』を参照してください。

### 1. WAN ブートサーバー上でスーパーユーザーになります。

システムの必要条件は以下のとおりです。

- CD-ROM ドライブまたは DVD-ROM ドライブを備えていること
  - サイトのネットワークおよびネームサービスに組み込まれていること
- ネームサービスを使用する場合は、システムがすでに NIS、NIS+、DNS、LDAP のいずれかのネームサービスに登録されていなければなりません。ネームサービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。

### 2. Solaris SOFTWARE 1 of 2 CD または Solaris DVD をインストールサーバーのドライブに挿入します。

### 3. WAN ブートミニルートと Solaris インストールイメージを置くためのディレクトリを作成します。

```
# mkdir -p wan-dir-path install-dir-path
```

`-p` 目的のディレクトリを作成するときに、必要な親ディレクトリもすべて作成するよう `mkdir` コマンドに指示します。

`wan-dir-path` WAN ブートミニルートの作成先となる、インストールサーバー上のディレクトリを指定します。このディレクトリには、ミニルートを格納できる容量が必要です。ミニルートの標準サイズは 250 M バイトです。

`install-dir-path` Solaris ソフトウェアイメージのコピー先となる、インストールサーバー上のディレクトリを指定します。この手順の後半で、このディレクトリは削除できます。

### 4. マウントされたディスクの `Tools` ディレクトリに移動します。

```
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
```

上の例では、`cdrom0` は、Solaris オペレーティング環境のメディアを含むドライブへのパスです。

### 5. WAN ブートミニルートと Solaris ソフトウェアイメージを、WAN ブートサーバーのハードディスクにコピーします。

```
# ./setup_install_server -w wan-dir-path install-dir-path
```

`wan-dir-path` WAN ブートミニルートをコピーするディレクトリを指定します。

*install-dir-path* Solaris ソフトウェアイメージをコピーするディレクトリを指定します。

---

注 - `setup_install_server` コマンドは、Solaris SOFTWARE ディスクイメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、`df -k1` コマンドを使用します。

---

`setup_install_server -w` コマンドは、WAN ブートミニルートと、Solaris ソフトウェアのネットワークインストールイメージを作成します。

6. (省略可能) ネットワークインストールイメージを削除します。

フラッシュアーカイブを使って WAN インストールを実行する場合、Solaris ソフトウェアイメージは不要です。ほかのネットワークインストールに使用する予定がない場合は、ネットワークインストールイメージを削除して、ディスクの空き領域を増やすことができます。ネットワークインストールイメージを削除するには、次のコマンドを入力します。

```
# rm -rf install-dir-path
```

7. 次のどちらかの方法で、WAN ブートサーバーが WAN ブートミニルートにアクセスできるようにします。

- WAN ブートサーバーのドキュメントルートディレクトリに、WAN ブートミニルートへのシンボリックリンクを作成します。

```
# cd document-root-directory/miniroot  
# ln -s wan-dir-path/miniroot .
```

*document-root-directory/miniroot* WAN ブートサーバーのドキュメントルートディレクトリにあるディレクトリで、WAN ブートミニルートにリンクするものを指定します。

*/wan-dir-path/miniroot* WAN ブートミニルートへのパスを指定します。

- WAN ブートサーバーのドキュメントルートディレクトリに、WAN ブートミニルートを移動します。

```
# mv wan-dir-path/miniroot document-root-directory/miniroot/miniroot-name
```

*wan-dir-path/miniroot* WAN ブートミニルートへのパスを指定します。

*/document-root-directory/miniroot/* WAN ブートサーバーのドキュメントルートディレクトリにある WAN ブートミニディレクトリへのパスを指定します。

*miniroot-name*

WAN ブートミニルートの名前を指定します。miniroot.s9\_sparc のように、わかりやすい名前をファイルに付けます。

## WAN ブートサーバーへの wanboot プログラムのインストール

WAN ブートでは、特別な二次レベルのブートプログラム wanboot が、クライアントのインストールに使用されます。wanboot プログラムは、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みます。

WAN ブートインストールを実行するには、インストール時に wanboot プログラムをクライアントに提供する必要があります。次の方法で、このプログラムをクライアントに提供できます。

- クライアントの PROM が WAN ブートに対応している場合は、WAN ブートサーバーからクライアントにプログラムを転送できます。クライアントの PROM が WAN ブートに対応しているかどうかを調べる方法については、567 ページの「クライアント OBP での WAN ブート対応の確認」を参照してください。
- クライアントの PROM が WAN ブートに対応していない場合は、ローカル CD を使ってクライアントにプログラムを提供する必要があります。クライアントの PROM が WAN ブートに対応していない場合は、540 ページの「WAN ブートサーバーに /etc/netboot ディレクトリを作成する」に進み、インストールの準備を続行します。

### ▼ SPARC: WAN ブートサーバーに wanboot プログラムをインストールする方法

この手順では、WAN ブートサーバーでボリュームマネージャを実行していると仮定します。ボリュームマネージャを使用していない場合は、ボリュームマネージャを使用せずにリムーバブルメディアを管理する方法について、『Solaris のシステム管理 (基本編)』を参照してください。

1. インストールサーバー上でスーパーユーザーになります。
2. **Solaris SOFTWARE 1 of 2 CD** または **Solaris DVD** をインストールサーバーのドライブに挿入します。
3. **Solaris SOFTWARE 1 of 2 CD** または **Solaris DVD** の **sun4u** プラットフォームディレクトリに移動します。

```
# cd /cdrom/cdrom0/s0/Solaris_9/Tools/Boot/platform/sun4u/
```

4. インストールサーバーに **wanboot** プログラムをコピーします。

```
# cp wanboot /document-root-directory/wanboot/wanboot-name
```

*document-root-directory*      WAN ブートサーバーのドキュメントルートディレクトリを指定します。

*wanboot-name*                wanboot プログラムの名前を指定します。  
wanboot.s9\_sparc のように、わかりやすい名前を  
ファイルに付けます。

5. 次のどちらかの方法で、**WAN** ブートサーバーが **wanboot** プログラムにアクセスできるようにします。

- WAN ブートサーバーのドキュメントルートディレクトリに、wanboot プログラムへのシンボリックリンクを作成します。

```
# cd /document-root-directory/wanboot  
# ln -s /wan-dir-path/wanboot .
```

*document-root-directory/wanboot*      WAN ブートサーバーのドキュメントルートディレクトリにあるディレクトリで、wanboot プログラムにリンクするものを指定します。

*/wan-dir-path/wanboot*                wanboot プログラムへのパスを指定します。

- WAN ブートサーバーのドキュメントルートディレクトリに、WAN ブートミニルートを移動します。

```
# mv /wan-dir-path/wanboot /document-root-directory/wanboot/wanboot-name
```

*wan-dir-path/wanboot*                wanboot プログラムへのパスを指定します。

*/document-root-directory/wanboot/*      WAN ブートサーバーのドキュメントルートディレクトリにある wanboot プログラムディレクトリへのパスを指定します。

*wanboot-name*                        wanboot プログラムの名前を指定します。wanboot.s9\_sparc のように、わかりやすい名前をファイルに付けます。

## WAN ブートサーバーに /etc/netboot ディレクトリを作成する

インストール時に WAN ブートは、Web サーバーの /etc/netboot ディレクトリの内容を参照して、インストールの実行方法に関する指示を取得します。このディレクトリには、WAN ブートインストールに必要な、構成情報、非公開鍵、デジタル証明

書、および認証局が保存されます。インストール時、この情報は `wanboot-cgi` プログラムによって WAN ブートファイルシステムに変換されます。その後、`wanboot-cgi` プログラムは WAN ブートファイルシステムをクライアントに転送します。

`/etc/netboot` ディレクトリ内にサブディレクトリを作成することで、WAN ブートインストールの適用範囲をカスタマイズできます。次のディレクトリ構造を使って、インストール対象のクライアント間で構成情報をどのように共有するかを定義します。

- グローバルな構成 – ネットワーク上のすべてのクライアントで構成情報を共有するには、共有する構成ファイルを `/etc/netboot` ディレクトリに保存します。
- ネットワーク固有の構成 – 特定のサブネット上のマシンだけで構成情報を共有するには、共有する構成ファイルを `/etc/netboot` ディレクトリのサブディレクトリに保存します。サブディレクトリは、次の規則に従って名前を付けてください。

```
/etc/netboot/net-ip
```

この例で、`net-ip` はクライアントのサブネットの IP アドレスです。

- クライアント固有の構成 – 特定のクライアントだけでブートファイルシステムを使用するには、ブートファイルシステムのファイルを `/etc/netboot` のサブディレクトリに保存します。サブディレクトリは、次の規則に従って名前を付けてください。

```
/etc/netboot/net-ip/client-ID
```

この例で、`net-ip` はサブネットの IP アドレスです。`client-ID` は、DHCP サーバーによって割り当てられるクライアント ID か、ユーザー指定のクライアント ID です。

`/etc/netboot` ディレクトリの詳しい設計方法については、524 ページの「`/etc/netboot` ディレクトリへの構成情報とセキュリティ情報の保存」を参照してください。

## ▼ `/etc/netboot` ディレクトリを作成する方法

1. WAN ブートサーバー上でスーパーユーザーになります。

2. `/etc/netboot` ディレクトリを作成します。

```
# mkdir /etc/netboot
```

3. `/etc/netboot` ディレクトリのアクセス権を **700** に変更します。

```
# chmod 700 /etc/netboot
```

4. `/etc/netboot` ディレクトリの所有者を、**Web** サーバーの所有者に変更します。

```
# chown web-server-user:web-server-group /etc/netboot/
```

*web-server-user* Web サーバプロセスの所有者であるユーザーを指定します。

*web-server-group* Web サーバプロセスの所有者であるグループを指定します。

5. スーパーユーザーを終了します。

```
# exit
```

6. **Web** サーバ所有者の役割になります。

7. **/etc/netboot** ディレクトリに、クライアントのサブディレクトリを作成します。

```
# mkdir -p /etc/netboot/net-ip/client-ID
```

-p 目的のディレクトリを作成するときに、必要な親ディレクトリもすべて作成するように `mkdir` コマンドに指示します。

(省略可能) *net-ip* クライアントのサブネットのネットワーク IP アドレスを指定します。

(省略可能) *client-ID* クライアント ID を指定します。クライアント ID は、ユーザーが定義した値か、DHCP クライアント ID です。*client-ID* ディレクトリは、*net-ip* ディレクトリのサブディレクトリである必要があります。

8. **/etc/netboot** ディレクトリ内の各サブディレクトリについて、アクセス権を **700** に変更します。

```
# chmod 700 /etc/netboot/dir-name
```

*dir-name* /etc/netboot ディレクトリ内のサブディレクトリの名前を指定します。

例 43-1 WAN ブートサーバに **/etc/netboot** ディレクトリを作成する

次の例は、サブネット 192.168.255.0 にあるクライアント 010003BA152A42 に対応する **/etc/netboot** ディレクトリの作成方法を示しています。この例では、*nobody* というユーザーと *admin* というグループが、Web サーバプロセスを所有しています。

この例のコマンドは、次の処理を行います。

- **/etc/netboot** ディレクトリを作成します。
- **/etc/netboot** ディレクトリのアクセス権を **700** に変更します。
- **/etc/netboot** ディレクトリの所有権を Web サーバプロセスの所有者に渡します。
- Web サーバユーザーと同じ役割になります。
- **/etc/netboot** ディレクトリに、サブネット名と同じ名前のサブディレクトリ 192.168.255.0 を作成します。

例 43-1 WAN ブートサーバーに /etc/netboot ディレクトリを作成する (続き)

- このサブネットディレクトリに、クライアント ID と同じ名前のサブディレクトリを作成します。
- /etc/netboot のサブディレクトリのアクセス権を 700 に変更します。

```
# cd /
# mkdir /etc/netboot/
# chmod 700 /etc/netboot
# chown nobody:admin /etc/netboot
# exit
server# su nobody
Password:
nobody# mkdir -p /etc/netboot/192.168.255.0/010003BA152A42
nobody# chmod 700 /etc/netboot/192.168.255.0
nobody# chmod 700 /etc/netboot/192.168.255.0/010003BA152A42
```

## WAN ブートサーバーへの WAN ブート CGI プログラムのコピー

wanboot-cgi プログラムは、WAN ブートサーバーからクライアントに次のファイルを転送するデータストリームを作成します。

- wanboot プログラム
- WAN ブートファイルシステム
- WAN ブートミニルート

Solaris 9 12/03 オペレーティング環境をインストールすると、wanboot-cgi プログラムもシステムにインストールされます。WAN ブートサーバーがこのプログラムを使用できるようにするには、WAN ブートサーバーの cgi-bin ディレクトリにこのプログラムをコピーします。

### ▼ WAN ブートサーバーに wanboot-cgi プログラムをコピーする方法

1. WAN ブートサーバー上でスーパーユーザーになります。
2. WAN ブートサーバーに wanboot-cgi プログラムをコピーします。

```
# cp /usr/lib/inet/wanboot/wanboot-cgi /WAN-server-root/cgi-bin/wanboot-cgi
/WAN-server-root
```

WAN ブートサーバー上の Web サーバーソフトウェアのルートディレクトリを指定します。

3. WAN ブートサーバーで、CGI プログラムのアクセス権を 755 に変更します。

```
# chmod 755 /WAN-server-root/cgi-bin/wanboot-cgi
```

## (省略可能) WAN ブートログサーバーの構成

クライアント以外のシステムでブートログメッセージとインストールログメッセージを記録するには、ログサーバーを設定する必要があります。インストール時に HTTPS を介してログサーバーを使用するには、WAN ブートサーバーをログサーバーとして構成する必要があります。

ログサーバーを構成するには、次の手順を実行します。

### ▼ ログサーバーを構成する方法

1. ログサーバーの CGI スクリプトディレクトリに **bootlog-cgi** スクリプトをコピーします。

```
# cp /usr/lib/inet/wanboot/bootlog-cgi \
log-server-root/cgi-bin
```

log-server-root/cgi-bin ログサーバーの Web サーバーディレクトリにある cgi-bin ディレクトリを指定します。

2. **bootlog-cgi** スクリプトのアクセス権を **755** に変更します。

```
# chmod 755 log-server-root/cgi-bin/bootlog-cgi
```

3. **wanboot.conf** ファイル内の **boot\_logger** パラメータの値を設定します。

wanboot.conf ファイルに、ログサーバー上の bootlog-cgi スクリプトの URL を指定します。

wanboot.conf ファイルのパラメータの設定方法については、559 ページの「wanboot.conf ファイルの作成」を参照してください。

インストール時、ログサーバーの /tmp ディレクトリに、ブートログメッセージとインストールログメッセージが記録されます。ログファイルの名前は bootlog.hostname となります。hostname は、クライアントのホスト名です。

例 43-2 HTTPS を介して WAN ブートインストールを行う場合のログサーバーの構成

次の例では、WAN ブートサーバーをログサーバーとして構成します。

```
# cp /usr/lib/inet/wanboot/bootlog-cgi /opt/apache/cgi-bin/
# chmod 755 /opt/apache/cgi-bin/bootlog-cgi
```



## (省略可能) HTTPS によるデータの保護

WAN ブートサーバーからクライアントへの転送データを保護するには、HTTPS (Secure Sockets Layer を介した HTTP) を使用します。517 ページの「セキュリティ保護された WAN ブートインストール構成」に説明されている、より高いセキュリティで保護されたインストール構成を使用するには、Web サーバーで HTTPS を使用できるようにする必要があります。

WAN ブートサーバーの Web サーバーソフトウェアで HTTPS を使用できるようにするには、次の作業を実行します。

- Web サーバーソフトウェアの SSL (Secure Sockets Layer) サポートを有効にします。  
SSL サポートとクライアント認証を有効にする手順は、Web サーバーによって異なります。Web サーバーでこれらのセキュリティ機能を有効にする方法については、このマニュアルでは説明していません。これらの機能については、次のマニュアルを参照してください。
  - SunONE Web サーバーおよび iPlanet Web サーバーで SSL を有効にする方法については、<http://docs.sun.com> にある Sun ONE および iPlanet のマニュアルコレクションを参照してください。
  - Apache Web サーバーで SSL を有効にする方法については、<http://httpd.apache.org/docs-project/> にある Apache Documentation Project を参照してください。
  - 上記以外の Web サーバーソフトウェアを使用している場合は、そのソフトウェアのマニュアルを参照してください。
- WAN ブートサーバーにデジタル証明書をインストールします。  
WAN ブートでデジタル証明書を使用する方法については、546 ページの「サーバー認証とクライアント認証のためのデジタル証明書の使用」を参照してください。
- 信頼できる証明書をクライアントに提供します。  
信頼できる証明書の作成方法については、546 ページの「サーバー認証とクライアント認証のためのデジタル証明書の使用」を参照してください。
- ハッシュキーと暗号化鍵を作成します。  
キーの作成方法については、548 ページの「ハッシュキーと暗号化鍵の作成」を参照してください。
- (省略可能) クライアント認証をサポートするように Web サーバーソフトウェアを構成します。  
クライアント認証をサポートするように Web サーバーを構成する方法については、Web サーバーのマニュアルを参照してください。

## サーバー認証とクライアント認証のためのデジタル証明書の使用

WAN ブートインストールでは、PKCS#12 ファイルを使って、サーバー認証またはサーバー認証とクライアント認証の両方を伴うインストールを HTTPS で実行できます。PKCS#12 ファイルを使用するための要件とガイドラインについては、527 ページの「デジタル証明書の要件」を参照してください。

WAN ブートインストールで PKCS#12 ファイルを使用するには、次の作業を実行します。

- PKCS#12 ファイルを、SSL 非公開鍵のファイルと信頼できる証明書のファイルに分割します。
- /etc/netboot ディレクトリにあるクライアントの truststore ファイルに、信頼できる証明書を挿入します。信頼できる証明書に従って、クライアントはサーバーを信頼します。
- (省略可能) /etc/netboot ディレクトリにあるクライアントの keystore ファイルに、SSL 非公開鍵ファイルの内容を挿入します。

wanbootutil コマンドには、上記の作業を実行するためのオプションが用意されています。

PKCS#12 ファイルを分割する前に、WAN ブートサーバーの /etc/netboot ディレクトリに適切なサブディレクトリを作成してください。

- /etc/netboot ディレクトリの概要については、524 ページの「/etc/netboot ディレクトリへの構成情報とセキュリティ情報の保存」を参照してください。
- /etc/netboot ディレクトリの作成方法については、540 ページの「WAN ブートサーバーに /etc/netboot ディレクトリを作成する」を参照してください。

### ▼ 信頼できる証明書およびクライアントの非公開鍵の作成

1. WAN ブートサーバーで、Web サーバーユーザーと同じ役割になります。
2. 信頼できる証明書を PKCS#12 ファイルから抽出します。/etc/netboot ディレクトリにあるクライアントの truststore ファイルに、この証明書を挿入します。

```
# wanbootutil p12split -i p12cert \  
-t /etc/netboot/net-ip/client-ID/truststore
```

p12split

wanbootutil コマンドのオプションです。PKCS#12 ファイルを非公開鍵ファイルと証明書ファイルに分割します。

```
-i p12cert
```

分割する PKCS#12 ファイルの名前を指定します。

```
-t /etc/netboot/net-ip/client-ID/truststore
```

クライアントの truststore ファイルに証明書を挿入します。net-ip は、クライアントのサブネットの IP アドレスです。client-ID は、ユーザーが定義した ID か、DHCP クライアント ID です。

3. (省略可能) クライアント認証を要求するかどうかを決定します。

- 要求する場合は、引き続き次の手順を実行します。
- 要求しない場合は、548 ページの「ハッシュキーと暗号化鍵の作成」へ進みます。

a. クライアントの certstore にクライアントの証明書を挿入します。

```
# wanbootutil p12split -i p12cert -c \  
/etc/netboot/net-ip/client-ID/certstore -k keyfile
```

p12split

wanbootutil コマンドのオプションです。PKCS#12 ファイルを非公開鍵ファイルと証明書ファイルに分割します。

-i p12cert

分割する PKCS#12 ファイルの名前を指定します。

-c /etc/netboot/net-ip/client-ID/certstore

クライアントの certstore にクライアントの証明書を挿入します。net-ip は、クライアントのサブネットの IP アドレスです。client-ID は、ユーザーが定義した ID か、DHCP クライアント ID です。

-k keyfile

PKCS#12 ファイルの分割によって作成する、クライアントの SSL 非公開鍵ファイルの名前を指定します。

b. クライアントの keystore に非公開鍵を挿入します。

```
# wanbootutil keymgmt -i -k keyfile \  
-s /etc/netboot/net-ip/client-ID/keystore -o type=rsa
```

keymgmt -i

クライアントの keystore に SSL 非公開鍵を挿入します。

-k keyfile

前の手順で作成したクライアントの非公開鍵ファイルの名前を指定します。

-s /etc/netboot/net-ip/client-ID/keystore

クライアントの keystore へのパスを指定します。

-o type=rsa

キータイプとして RSA を指定します。

例 43-3 サーバー認証用の信頼できる証明書を作成する

次の例では、サブネット 192.168.255.0 にあるクライアント 010003BA152A42 に対して、PKCS#12 ファイルを使ってインストールを行います。このコマンド例は、client.p12 という名前の PKCS#12 ファイルから証明書を抽出します。次に、この信頼できる証明書の内容を、クライアントの truststore ファイルに挿入します。

例 43-3 サーバー認証用の信頼できる証明書を作成する (続き)

これらのコマンドを実行する前に、まず Web サーバーユーザーと同じユーザー役割になる必要があります。この例の場合、Web サーバーユーザー役割は nobody です。

```
server# su nobody
Password:
nobody# wanbootutil pl2split -i client.p12 \
-t /etc/netboot/192.168.255.0/010003BA152A42/truststore
nobody# chmod 600 /etc/netboot/192.168.255.0/010003BA152A42/truststore
```

## ハッシュキーと暗号化鍵の作成

HTTPS を使ってデータを転送するには、HMAC SHA1 ハッシュキーと暗号化鍵を作成する必要があります。半私設のネットワーク上でインストールを行うときなど、インストールデータの暗号化が不要な場合もあります。HMAC SHA1 ハッシュキーを使用すると、wanboot プログラムの完全性を確認できます。ハッシュキーと暗号化鍵の概要については、515 ページの「WAN ブートインストール時のデータの保護」を参照してください。

wanbootutil keygen コマンドを使用すると、これらのキーを生成し、/etc/netboot の適切なディレクトリに保存できます。

### ▼ ハッシュキーと暗号化鍵を作成する方法

1. WAN ブートサーバーで、Web サーバーユーザーと同じ役割になります。
2. HMAC SHA1 マスターキーを作成します。

```
# wanbootutil keygen -m
keygen -m WAN ブートサーバーの HMAC SHA1 マスターキーを作成します。
```

3. このマスターキーから、クライアントの HMAC SHA1 ハッシュキーを作成します。

```
# wanbootutil keygen -c -o [net=net-ip,{cid=client-ID,}]type=sha1
-c クライアントのハッシュキーをマスターキーから作成します。
-o wanbootutil keygen コマンドに追加オプションが含まれていることを示します。
```

(省略可能) net=net-ip クライアントのサブネットの IP アドレスを指定します。net オプションを指定しない場合、キーは /etc/netboot/keystore ファイルに保存され、すべての WAN ブートクライアントで使用可能になります。

(省略可能) `cid=client-ID` クライアント ID を指定します。クライアント ID は、ユーザーが定義した ID か、DHCP クライアント ID です。cid オプションの前には、`net=` に有効な値を指定する必要があります。net オプションを指定し、cid オプションを指定しない場合、キーは `/etc/netboot/net-ip/keystore` ファイルに保存されます。このキーは、`net-ip` サブネットにあるすべての WAN ブートクライアントで使用可能になります。

`type=sha1` クライアントの HMAC SHA1 ハッシュキーを作成するよう、`wanbootutil keygen` ユーティリティに指示します。

#### 4. クライアントの暗号化鍵を作成する必要があるかどうかを決定します。

HTTPS を介して WAN ブートインストールを実行するには、暗号化鍵を作成する必要があります。クライアントが WAN ブートサーバーと HTTPS 接続を確立する前に、WAN ブートサーバーは、暗号化されたデータと情報をクライアントに転送します。クライアントは暗号化鍵を使ってこの情報を復号化し、インストール時にこの情報を使用することができます。

- サーバー認証を伴う、より高いセキュリティで保護された WAN インストールを HTTPS で実行する場合は、次の手順に進みます。
- `wanboot` プログラムの完全性チェックだけを行う場合は、暗号化鍵を作成する必要はありません。手順 6 に進みます。

#### 5. クライアントの暗号化鍵を作成します。

```
# wanbootutil keygen -c -o [net=net-ip,{cid=client-ID,}]type=key-type
```

`-c` クライアントの暗号化鍵を作成します。

`-o` `wanbootutil keygen` コマンドに追加オプションが含まれていることを示します。

(省略可能) `net=net-ip` クライアントのネットワーク IP アドレスを指定します。net オプションを指定しない場合、キーは `/etc/netboot/keystore` ファイルに保存され、すべての WAN ブートクライアントで使用可能になります。

(省略可能) `cid=client-ID` クライアント ID を指定します。クライアント ID は、ユーザーが定義した ID か、DHCP クライアント ID です。cid オプションの前には、`net=` に有効な値を指定する必要があります。net オプションを指定し、cid オプションを指定しない場合、キーは `/etc/netboot/net-ip/keystore` ファイルに保存されます。このキーは、`net-ip` サブネットにあるすべての WAN ブートクライアントで使用可能になります。

`type=key-type` クライアントの暗号化鍵を作成するよう、`wanbootutil keygen` ユーティリティに指示しま

す。key-type には、3des または aes という値を指定できます。

6. クライアントシステムにキーをインストールします。

クライアントにキーをインストールする方法については、569 ページの「クライアントに対するキーのインストール」を参照してください。

例 43-4 HTTPS を介して WAN ブートインストールを実行するために必要なキーを作成する

次の例では、WAN ブートサーバーの HMAC SHA1 マスターキーを作成します。また、サブネット 192.168.255.0 にあるクライアント 010003BA152A42 用に、HMAC SHA1 ハッシュキーと 3DES 暗号化鍵を作成します。

これらのコマンドを実行する前に、まず Web サーバーユーザーと同じユーザー役割になる必要があります。この例の場合、Web サーバーユーザー役割は nobody です。

```
server# su nobody
Password:
nobody# wanbootutil keygen -m
nobody# wanbootutil keygen -c -o net=192.168.255.0,cid=010003BA152A42,type=sha1
nobody# wanbootutil keygen -c -o net=192.168.255.0,cid=010003BA152A42,type=3des
```

---

## カスタム JumpStart インストールファイルの作成

WAN ブートは、カスタム JumpStart インストールを実行して、フラッシュアーカイブをクライアントにインストールします。カスタム JumpStart 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を自動的にかつ同時に行うことができる、コマンド行インタフェースです。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。

WAN ブートインストール用のカスタム JumpStart ファイルを準備するには、次の作業を実行します。

- 551 ページの「フラッシュアーカイブの作成」
- 552 ページの「sysidcfg ファイルの作成」
- 554 ページの「rules ファイルの作成」
- 553 ページの「プロファイルの作成」

- 556 ページの「(省略可能) begin スクリプトと finish スクリプトの作成」

カスタム JumpStart インストールの詳細は、第 25 章を参照してください。

## フラッシュアーカイブの作成

フラッシュインストール機能を利用すると、マスターシステムと呼ばれるシステム上の Solaris オペレーティング環境を、単一の参照用インストールイメージとして使用できます。また、マスターシステムのイメージを複製して、フラッシュアーカイブを作成できます。ネットワーク内のほかのシステムにこのフラッシュアーカイブをインストールすることで、クローンシステムを作成できます。

ここでは、WAN ブートインストールで使用するフラッシュアーカイブの作成方法について説明します。フラッシュアーカイブを作成する前に、マスターシステムのインストールを行う必要があります。

- マスターシステムのインストール方法については、214 ページの「マスターシステムへのインストール」を参照してください。
- フラッシュアーカイブの詳細については、第 21 章を参照してください。

### ▼ フラッシュアーカイブを作成する方法

フラッシュアーカイブの作成方法の詳細は、220 ページの「フラッシュアーカイブの作成」を参照してください。

1. マスターシステムをブートします。

できるだけ静的な状態でマスターシステムを稼働させます。可能であれば、システムをシングルユーザーモードで実行してください。これが不可能な場合、アーカイブしたいアプリケーションおよび大量のオペレーティングシステムリソースを必要とするアプリケーションを停止します。

2. **flar create** コマンドを使用して、アーカイブを作成します。

```
# flar create -n name [optional-parameters] document-root/flash/filename
```

*name* アーカイブに指定する名前です。指定する *name* は、*content\_name* キーワードの値になります。

*optional-parameters* flar create コマンドには、フラッシュアーカイブをカスタマイズするためのオプションをいくつか指定できます。これらのオプションの詳細は、第 23 章を参照してください。

*document-root/flash* インストールサーバーのドキュメントルートディレクトリにある Solaris Flash サブディレクトリへのパスです。

*filename* アーカイブファイルの名前です。

ディスク容量を節約するために、`flar create` コマンドに `-c` オプションを指定してアーカイブを圧縮することもできます。ただし、アーカイブを圧縮すると、WAN ブートインストールのパフォーマンスに影響する場合があります。圧縮されたアーカイブの作成方法については、`flar create (1M)` のマニュアルページを参照してください。

- アーカイブの作成が正常に完了すると、`flar create` コマンドは終了コード 0 を返します。
- アーカイブの作成が失敗すると、`flar create` コマンドは 0 以外の終了コードを返します。

フラッシュアーカイブの作成方法の例については、221 ページの「例 — 初期インストール用アーカイブの作成」を参照してください。

## sysidcfg ファイルの作成

`sysidcfg` ファイルに一連のキーワードを指定すると、システムを事前設定できます。`sysidcfg` のキーワードと値の詳細は、65 ページの「`sysidcfg` ファイルによる事前設定」を参照してください。

### ▼ `sysidcfg` ファイルを作成する方法

1. インストールサーバーで、テキストエディタを使って `sysidcfg` というファイルを作成します。

2. 必要な `sysidcfg` のキーワードを入力します。

`sysidcfg` のキーワードの詳細については、67 ページの「`sysidcfg` ファイルキーワード」を参照してください。

3. WAN ブートサーバーがアクセスできる場所に、この `sysidcfg` ファイルを保存します。

このファイルを次のどちらかの場所に保存します。

- WAN ブートサーバーとインストールサーバーが同じマシンに置かれている場合は、WAN ブートサーバーのドキュメントルートディレクトリの `flash` サブディレクトリにこのファイルを保存します。
- WAN ブートサーバーとインストールサーバーがそれぞれ別のマシンに置かれている場合は、インストールサーバーのドキュメントルートディレクトリの `flash` サブディレクトリにこのファイルを保存します。

#### 例 43-5 WAN ブートインストール用の `sysidcfg` ファイル

SPARC ベースのシステムで使用される `sysidcfg` ファイルの例を次に示します。このシステムのホスト名、IP アドレス、およびネットマスクは、ネームサービスを編集することにより、すでに事前設定されています。



例 43-5 WAN ブートインストール用の sysidcfg ファイル (続き)

```
network_interface=primary {hostname=seahag
                             default_route=192.168.88.1
                             ip_address=192.168.88.210
                             netmask=255.255.0.0
                             protocol_ipv6=no}

timezone=US/Central
system_locale=C
terminal=xterm
timeserver=localhost
name_service=NIS {name_server=matter(192.168.255.255)
                  domain_name=mind.over.example.com
                  }
security_policy=none
```

## プロファイルの作成

プロファイルは、システムへの Solaris ソフトウェアのインストール方法をカスタム JumpStart プログラムに指示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。

プロファイルの作成方法の詳細は、274 ページの「プロファイルの作成」を参照してください。

### ▼ プロファイルを作成する方法

1. インストールサーバーで、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。  
プロファイルの名前は、システムに Solaris ソフトウェアをインストールするためにそのプロファイルをどのように使用するかを示すものにしてください。(例: basic\_install、eng\_profile、user\_profile)。
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。  
プロファイルのキーワードと値の一覧は、340 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。  
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せん。
3. **WAN** ブートサーバーがアクセスできる場所に、このプロファイルを保存します。このプロファイルを次のどちらかの場所に保存します。
  - WAN ブートサーバーとインストールサーバーが同じマシンに置かれている場合は、WAN ブートサーバーのドキュメントルートディレクトリの flash サブディレクトリにこのファイルを保存します。

- WAN ブートサーバーとインストールサーバーがそれぞれ別のマシンに置かれている場合は、インストールサーバーのドキュメントルートディレクトリの `flash` サブディレクトリにこの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) プロファイル进行测试します。

プロファイルのテストに関する情報については、284 ページの「プロファイルのテスト」を参照してください。

#### 例 43-6 HTTPS サーバーから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 プログラムに対し、HTTPS サーバーからフラッシュアーカイブを取得するように指示します。

```
# 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。

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

## rules ファイルの作成

`rules` ファイルは、Solaris オペレーティング環境のインストール先となる各システムグループのルールを記述し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基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

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に属する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。

`rules` ファイルの作成方法の詳細については、270 ページの「`rules` ファイルの作成」を参照してください。

## ▼ rules ファイルを作成する方法

1. インストールサーバーで、**rules** という名前のテキストファイルを作成します。
2. インストール対象であるシステムのグループごとに、適用するルールを **rules** ファイルに追加します。  
`rules` ファイルの作成方法の詳細については、270 ページの「`rules` ファイルの作成」を参照してください。
3. インストールサーバーに **rules** ファイルを保存します。
4. **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 9 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 DVD、または Solaris SOFTWARE 1 of 2 CD 上のイメージです。

システムが以前のバージョンの Solaris を実行している場合、このオプションを使用して、最新バージョンの `check` スクリプトを実行します。

**-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` ファイルの有効性と各プロファイルの有効性の検査結果が表示されます。エラーが検出されないと、`The custom JumpStart configuration is ok` と表示されます。`check` スクリプトによって `rules.ok` ファイルが作成されます。

5. **WAN** ブートサーバーがアクセスできる場所に、この **rules.ok** ファイルを保存します。

このファイルを次のどちらかの場所に保存します。

- WAN ブートサーバーとインストールサーバーが同じマシンに置かれている場合は、WAN ブートサーバーのドキュメントルートディレクトリの `flash` サブディレクトリにこのファイルを保存します。
  - WAN ブートサーバーとインストールサーバーがそれぞれ別のマシンに置かれている場合は、インストールサーバーのドキュメントルートディレクトリの `flash` サブディレクトリにこのファイルを保存します。
6. `root` が `rules.ok` 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` ファイルの例については、273 ページの「`rules` ファイルの例」を参照してください。

## (省略可能) `begin` スクリプトと `finish` スクリプトの作成

「`begin` スクリプト」と「`finish` スクリプト」は、ユーザーが定義する Bourne シェルスクリプトで、`rules` ファイル内で指定します。`begin` スクリプトは、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。`finish` スクリプトは、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。これらのスクリプトは、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。

`begin` スクリプトを使用すると、動的プロファイルを作成できます。`finish` スクリプトを使用すると、ファイル、パッケージ、パッチ、ほかのソフトウェアの追加など、各種のポストインストール作業を実行できます。

`begin` スクリプトと `finish` スクリプトは、インストールサーバー上で `sysidcfg`、`rules.ok`、およびプロファイルの各ファイルと同じディレクトリに保存する必要があります。

- `begin` スクリプトの作成方法については、291 ページの「`begin` スクリプトの作成」を参照してください。
- `finish` スクリプトの作成方法については、293 ページの「`finish` スクリプトの作成」を参照してください。

---

## 構成ファイルの作成

WAN ブートでは、WAN ブートインストールに必要なデータとファイルの場所が、次のファイルによって指定されます。

- システム構成ファイル (system.conf)
- wanboot.conf ファイル

ここでは、これら2つのファイルの作成方法と保存方法について説明します。

## システム構成ファイルの作成

システム構成ファイルでは、次のファイルの場所を WAN ブートインストールプログラムに指示できます。

- sysidcfg ファイル
- rules.ok ファイル
- カスタム JumpStart プロファイル

WAN ブートは、システム構成ファイルのポインタに従って、クライアントに対してインストールと構成を行います。

システム構成ファイルは、プレーンテキストファイルで、次の書式に従う必要があります。

設定=値

システム構成ファイルを使って sysidcfg、rules.ok、およびプロファイルの各ファイルの場所を WAN インストールプログラムに指示するには、次の手順を実行します。

### ▼ システム構成ファイルを作成する方法

1. WAN ブートサーバーで、Web サーバーユーザーと同じ役割になります。
2. テキストファイルを作成します。sys-conf.s9-sparc のように、わかりやすい名前をファイルに付けます。
3. このシステム構成ファイルに、次のエントリを追加します。

SsysidCF=*sysidcfg-file-URL*

sysidcfg ファイルが置かれているインストールサーバー上の flash ディレクトリを指定します。この URL は、552 ページの「sysidcfg ファイルの作成」で作成した sysidcfg ファイルへのパスと一致するようにしてください。

HTTPS を使って WAN インストールを行う場合は、有効な HTTPS URL を設定してください。

Sjumpscf=*jumpstart-files-URL*

rules.ok ファイル、プロファイルファイル、および begin スクリプトと finish スクリプトが置かれているインストールサーバー上の Solaris Flash ディレクトリを指定します。この URL は、553 ページの「プロファイルの作

成」および 554 ページの「rules ファイルの作成」で作成したカスタム JumpStart ファイルへのパスと一致するようにしてください。

HTTPS を使って WAN インストールを行う場合は、有効な HTTPS URL を設定してください。

4. WAN ブートサーバーがアクセスできるディレクトリに、このファイルを保存します。

管理上の目的から、WAN ブートサーバーの /etc/netboot ディレクトリにある適切なクライアントのディレクトリに、このファイルを保存することもできます。

5. システム構成ファイルのアクセス権を **600** に変更します。

```
# chmod 600 /path/system-conf-file
```

*path* システム構成ファイルが置かれているディレクトリへのパスを指定します。

*system-conf-file* システム構成ファイルの名前を指定します。

#### 例 43-7 HTTPS を介して WAN ブートインストールを行う場合のシステム構成ファイル

次の例で、WAN ブートプログラムは、Web サーバー `https://www.example.com` のポート 1234 上で `sysidcfg` ファイルとカスタム JumpStart ファイルを検索します。インストール時、Web サーバーは HTTPS を使ってデータとファイルを暗号化します。

`sysidcfg` ファイルとカスタム JumpStart ファイルは、ドキュメントルートディレクトリ `htdocs` の `flash` サブディレクトリに置かれています。

```
SsysidCF=https://www.example.com:1234/htdocs/flash
SjumpsCF=https://www.example.com:1234/htdocs/flash
```

#### 例 43-8 セキュリティ保護されていない WAN ブートインストールを行う場合のシステム構成ファイル

次の例で、WAN ブートプログラムは、Web サーバー `http://www.example.com` で `sysidcfg` ファイルとカスタム JumpStart ファイルを検索します。Web サーバーは HTTP を使用するため、インストール時にデータやファイルは保護されません。

`sysidcfg` ファイルとカスタム JumpStart ファイルは、ドキュメントルートディレクトリ `htdocs` の `flash` サブディレクトリに置かれています。

```
SsysidCF=http://www.example.com/htdocs/flash
SjumpsCF=http://www.example.com/htdocs/flash
```

## wanboot.conf ファイルの作成

wanboot.conf ファイルは、プレーンテキストの構成ファイルで、WAN インストールを実行するために WAN ブートプログラムで使用されます。wanboot-cgi プログラム、ブートファイルシステム、および WAN ブートミニルートはすべて、wanboot.conf ファイルに保存されている情報を使ってクライアントマシンのインストールを行います。

WAN ブートサーバー上の /etc/netboot ディレクトリにある該当のクライアントのサブディレクトリに、この wanboot.conf ファイルを保存してください。/etc/netboot ディレクトリを使って WAN ブートインストールの適用範囲を定義する方法については、540 ページの「WAN ブートサーバーに /etc/netboot ディレクトリを作成する」を参照してください。

WAN ブートサーバーで Solaris 9 12/03 オペレーティング環境が稼働している場合は、/etc/netboot/wanboot.conf.sample に wanboot.conf ファイルの例があります。この例は、WAN ブートインストール用のテンプレートとして使用できます。

wanboot.conf ファイルには、次の情報を指定する必要があります。

表 43-3 wanboot.conf ファイルに指定する情報

情報の種類	説明
WAN ブートサーバーの情報	<ul style="list-style-type: none"><li>■ WAN ブートサーバー上の wanboot プログラムへのパス</li><li>■ WAN ブートサーバー上の wanboot-cgi プログラムの URL</li></ul>
インストールサーバーの情報	<ul style="list-style-type: none"><li>■ インストールサーバー上の WAN ブートミニルートへのパス</li><li>■ WAN ブートサーバー上のシステム構成ファイルへのパス。システム構成ファイルは、sysidcfg ファイルとカスタム JumpStart ファイルの場所を指定する</li></ul>
セキュリティ情報	<ul style="list-style-type: none"><li>■ WAN ブートファイルシステムまたは WAN ブートミニルートの署名タイプ</li><li>■ WAN ブートファイルシステムの暗号化タイプ</li><li>■ WAN ブートインストール時にサーバー認証を行うかどうか</li><li>■ WAN ブートインストール時にクライアント認証を行うかどうか</li></ul>
省略可能な情報	<ul style="list-style-type: none"><li>■ WAN ブートインストール時にクライアントに対して解決する必要のある、追加のホスト</li><li>■ ログサーバー上の bootlog-cgi スクリプトの URL</li></ul>

これらの情報を指定するには、パラメータとその値を次の書式で列挙します。

パラメータ=値

wanboot.conf ファイルのパラメータと構文については、610 ページの「wanboot.conf ファイルのパラメータと構文」を参照してください。

## ▼ wanboot.conf ファイルを作成する方法

1. WAN ブートサーバーで、Web サーバーユーザーと同じ役割になります。

2. wanboot.conf テキストファイルを作成します。

wanboot.conf という名前の新しいテキストファイルを作成するか、  
/etc/netboot/wanboot.conf.sample にあるファイル例を使用します。  
ファイル例を使用する場合は、パラメータを追加したあとで、ファイルの名前を  
wanboot.conf に変更してください。

3. インストール用のパラメータと値を wanboot.conf に入力します。

wanboot.conf のパラメータと値については、610 ページの「wanboot.conf  
ファイルのパラメータと構文」を参照してください。

4. /etc/netboot ディレクトリの適切なサブディレクトリに wanboot.conf  
ファイルを保存します。

/etc/netboot ディレクトリの作成方法については、540 ページの「WAN ブート  
サーバーに /etc/netboot ディレクトリを作成する」を参照してください。

5. wanboot.conf ファイルの妥当性検査を行います。

```
# bootconfchk /etc/netboot/path-to-wanboot.conf/wanboot.conf
```

```
path-to-wanboot.conf WAN ブートサーバーにあるクライアントの  
wanboot.conf ファイルへのパスを指定します。
```

- wanboot.conf ファイルが構造的に有効であれば、bootconfchk コマンドは  
終了コード 0 を返します。
- wanboot.conf ファイルが無効であれば、bootconfchk コマンドは 0 以外の  
終了コードを返します。

6. wanboot.conf ファイルのアクセス権を 600 に変更します。

```
# chmod 600 /etc/netboot/path-to-wanboot.conf/wanboot.conf
```

例 43-9 HTTPS を介して WAN ブートインストールを行う場合の wanboot.conf ファイル

次の wanboot.conf ファイル例には、HTTPS を使った WAN インストールを行うた  
めの構成情報が記述されています。この wanboot.conf ファイルには、インストー  
ルで 3DES 暗号化鍵を使用することも指示されています。

```
boot_file=/wanboot/wanboot.s9_sparc  
root_server=https://www.example.com:1234/cgi-bin/wanboot-cgi  
root_file=/miniroot/miniroot.s9_sparc  
signature_type=shal
```



例 43-9 HTTPS を介して WAN ブートインストールを行う場合の wanboot.conf ファイル (続き)

```
encryption_type=3des
server_authentication=yes
client_authentication=no
resolve_hosts=
boot_logger=https://www.example.com:1234/cgi-bin/bootlog-cgi
system_conf=system.conf
```

この wanboot.conf ファイルで指定されている構成は次のとおりです。

`boot_file=/wanboot/wanboot.s9_sparc`  
二次レベルのブートプログラムの名前は `wanboot.s9_sparc` です。このプログラムは、WAN ブートサーバーのドキュメントルートディレクトリ内の `/wanboot` ディレクトリに置かれています。

`root_server=https://www.example.com:1234/cgi-bin/wanboot-cgi`  
WAN ブートサーバー上の `wanboot-cgi` プログラムの場所は `https://www.example.com:1234/cgi-bin/wanboot-cgi` です。URL の `https` という部分は、この WAN ブートインストールで HTTPS を使用することを示しています。

`root_file=/miniroot/miniroot.s9_sparc`  
WAN ブートミニルートの名前は `miniroot.s9_sparc` です。このミニルートは、WAN ブートサーバーのドキュメントルートディレクトリ内の `/miniroot` ディレクトリに置かれています。

`signature_type=sha1`  
`wanboot.s9_sparc` プログラムと WAN ブートファイルシステムは、HMAC SHA1 ハッシュキーで署名されます。

`encryption_type=3des`  
`wanboot.s9_sparc` プログラムと WAN ブートファイルシステムは、3DES 暗号化鍵で暗号化されます。

`server_authentication=yes`  
インストール時にサーバー認証が行われます。

`client_authentication=no`  
インストール時にクライアント認証は行われません。

`resolve_hosts=`  
WAN インストールの実行に必要な追加のホスト名はありません。必要なファイルと情報はすべて、WAN ブートサーバーのドキュメントルートディレクトリに置かれています。

`boot_logger=https://www.example.com:1234/cgi-bin/bootlog-cgi`  
HTTPS を使って WAN ブートサーバー上でブートログメッセージおよびインストールログメッセージが記録されます。

WAN ブートインストールのログサーバーの設定方法については、544 ページの「(省略可能) WAN ブートログサーバーの構成」を参照してください。

例 43-9 HTTPS を介して WAN ブートインストールを行う場合の wanboot.conf ファイル (続き)

```
system_conf=system.conf
sysidcfg ファイルとカスタム JumpStart ファイルの場所が記述されたシステム構成ファイルは、/etc/netboot ディレクトリのサブディレクトリに置かれています。システム構成ファイルの名前は system.conf です。
```

例 43-10 セキュリティ保護されていない WAN ブートインストールを行う場合の wanboot.conf ファイル

次の wanboot.conf ファイル例には、HTTP を使ったセキュリティの低い WAN インストールを行うための構成情報が記述されています。この wanboot.conf ファイルには、インストールで暗号化鍵やハッシュキーを使用しないことも指示されています。

```
boot_file=/wanboot/wanboot.s9_sparc
root_server=http://www.example.com/cgi-bin/wanboot-cgi
root_file=/miniroot/miniroot.s9_sparc
signature_type=
encryption_type=
server_authentication=no
client_authentication=no
resolve_hosts=
boot_logger=http://www.example.com/cgi-bin/bootlog-cgi
system_conf=system.conf
```

この wanboot.conf ファイルで指定されている構成は次のとおりです。

```
boot_file=/wanboot/wanboot.s9_sparc
二次レベルのブートプログラムの名前は wanboot.s9_sparc です。このプログラムは、WAN ブートサーバーのドキュメントルートディレクトリ内の /wanboot ディレクトリに置かれています。
```

```
root_server=http://www.example.com/cgi-bin/wanboot-cgi
WAN ブートサーバー上の wanboot-cgi プログラムの場所は
http://www.example.com/cgi-bin/wanboot-cgi です。このインストールでは HTTPS を使用しません。
```

```
root_file=/miniroot/miniroot.s9_sparc
WAN ブートミニルートの名前は miniroot.s9_sparc です。このミニルートは、WAN ブートサーバーのドキュメントルートディレクトリ内の /miniroot サブディレクトリに置かれています。
```

```
signature_type=
wanboot.s9_sparc プログラムと WAN ブートファイルシステムは、ハッシュキーで署名されません。
```

```
encryption_type=
wanboot.s9_sparc プログラムとブートファイルシステムは、暗号化されません。
```

```
server_authentication=no
インストール時に、キーや証明書によるサーバー認証は行われません。
```

例 43-10 セキュリティ保護されていない WAN ブートインストールを行う場合の wanboot.conf ファイル (続き)

client\_authentication=no

インストール時に、キーや証明書によるクライアント認証は行われません。

resolve\_hosts=

インストールの実行に必要な追加のホスト名はありません。必要なファイルと情報はすべて、WAN ブートサーバーのドキュメントルートディレクトリに置かれています。

boot\_logger=http://www.example.com/cgi-bin/bootlog-cgi

WAN ブートサーバー上でブートログメッセージおよびインストールログメッセージが記録されます。

WAN ブートインストールのログサーバーの設定方法については、544 ページの「(省略可能) WAN ブートログサーバーの構成」を参照してください。

system\_conf=system.conf

sysidcfg ファイルとカスタム JumpStart ファイルの場所が記述されたシステム構成ファイルは、system.conf という名前です。このファイルは、/etc/netboot ディレクトリにある適切なクライアントのサブディレクトリに置かれています。

---

## (省略可能) DHCP による構成情報の提供

ネットワークで DHCP サーバーを使用する場合は、次の情報を提供するように DHCP サーバーを構成できます。

- プロキシサーバーの IP アドレス
- wanboot-cgi プログラムの場所

WAN ブートインストールでは、次の DHCP ベンダーオプションを使用できます。

SHTTPproxy ネットワークのプロキシサーバーの IP アドレスを指定します。

SbootURI WAN ブートサーバー上の wanboot-cgi プログラムの URL を指定します。

Solaris DHCP サーバーにこれらのベンダーオプションを設定する方法については、75 ページの「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 DHCP サーバーの設定方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP サービスの構成 (手順)」を参照してください。



## 第 44 章

# SPARC: WAN ブートによるインストール (作業)

この章では、SPARC ベースのクライアントに対して WAN ブートインストールを実行する方法について説明します。WAN ブートインストールの準備方法については、第 43 章を参照してください。

この章の内容は次のとおりです。

- 566 ページの「WAN ブートインストールを行うためのクライアントの準備」
- 574 ページの「クライアントのインストール」

## 作業マップ: WAN ブートによるクライアントのインストール

次の表では、クライアントのインストールを WAN 経由で実行するために必要な作業の一覧を示しています。

表 44-1 作業マップ: WAN ブートインストールの実行

作業	説明	参照先
WAN ブートインストールを行うためにネットワークを準備する	WAN ブートインストールの実行に必要なサーバーとファイルを設定する	第 43 章
クライアントシステムが WAN ブートに対応していることを確認する	クライアントの OBP をチェックして、WAN ブートのブート引数がサポートされていることを確認する	567 ページの「クライアント OBP での WAN ブート対応の確認」

表 44-1 作業マップ: WAN ブートインストールの実行 (続き)

作業	説明	参照先
クライアントの OBP に net デバイス別名が正しく設定されていることを確認する	devalias コマンドを使って、net デバイス別名に主ネットワークインタフェースが設定されていることを確認する	568 ページの「クライアント OBP の net デバイス別名の確認」
クライアントにキーを提供する	OBP の変数を設定するか、インストール時にキーの値を入力することで、クライアントにキーを提供する  この作業は、セキュリティ保護されたインストール構成に必要。セキュリティ保護されていないインストールで、データの完全性をチェックする場合は、この作業を実行して HMAC SHA1 ハッシュキーをクライアントに提供する	569 ページの「クライアントに対するキーのインストール」
クライアントに対して広域ネットワーク経由でインストールを実行する	適切な方法を選択してクライアントのインストールを実行する	575 ページの「自動 WAN ブートインストールの実行」  577 ページの「対話式 WAN ブートインストールの実行」  581 ページの「DHCP サーバーを使ったインストール」  583 ページの「ローカルの CD メディアを使ったインストール」

## WAN ブートインストールを行うためのクライアントの準備

クライアントシステムのインストールを行う前に、次の作業を実行してクライアントを準備してください。

- 567 ページの「クライアント OBP での WAN ブート対応の確認」
- 568 ページの「クライアント OBP の net デバイス別名の確認」
- 569 ページの「クライアントに対するキーのインストール」

## クライアント OBP での WAN ブート対応の確認

WAN ブートインストールを自動的に実行するには、クライアントの OpenBoot PROM (OBP) が WAN ブートに対応している必要があります。クライアントの OBP が WAN ブートに対応しているかどうかを調べるには、次の手順に従ってください。

### ▼ クライアント OBP での WAN ブート対応を確認する方法

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. WAN ブート対応を表す OBP 構成変数を調べます。

```
# eeprom | grep network-boot-arguments
```

- 変数 `network-boot-arguments` が表示されるか、上記のコマンドが `network-boot-arguments: data not available` という出力を返した場合、OBP は WAN ブートインストールに対応しています。WAN ブートインストールを実行する前に OBP を更新する必要はありません。
- 上記のコマンドから何の出力も得られない場合、OBP は WAN ブートインストールに対応していません。次のどちらかの作業を実行する必要があります。
  - クライアントの OBP を更新します。OBP の更新方法については、システムのマニュアルを参照してください。
  - ローカル CD-ROM ドライブ内の Solaris 9 SOFTWARE CD から WAN ブートインストールを実行します。ローカル CD-ROM ドライブからクライアントをブートする方法については、583 ページの「ローカルの CD メディアを使ったインストール」を参照してください。

例 44-1 クライアント上で OBP が WAN ブートに対応しているかどうかを確認する

次のコマンドは、クライアントの OBP が WAN ブートに対応しているかどうかを確認する方法を示しています。

```
# eeprom | grep network-boot-arguments
network-boot-arguments: data not available
```

この例では、出力に `network-boot-arguments: data not available` が表示されているので、クライアント OBP は WAN ブートに対応しています。

## クライアント OBP の net デバイス別名の確認

boot net コマンドを使って WAN からクライアントをブートするには、net デバイス別名にクライアントの主ネットワークデバイスが設定されている必要があります。ほとんどのシステムで、この別名はすでに正しく設定されています。ただし、使用するネットワークデバイスがデバイス別名に設定されていない場合は、別名を変更する必要があります。

クライアント上で net デバイス別名を確認するには、次の手順に従ってください。

### ▼ net デバイス別名を確認する方法

1. クライアント上でスーパーユーザーになります。

```
# init 0
ok プロンプトが表示されます。
```

3. ok プロンプトで、OBP に設定されているデバイス別名を調べます。

```
ok devalias
devalias コマンドは、次の例のような情報を出力します。

screen                /pci@1f,0/pci@1,1/SUNW,m64B@2
net                   /pci@1f,0/pci@1,1/network@c,1
net2                  /pci@1f,0/pci@1,1/network@5,1
disk                  /pci@1f,0/pci@1/scsi@8/disk@0,0
cdrom                 /pci@1f,0/pci@1,1/ide@d/cdrom@0,0:f
keyboard              /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse                 /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
```

- インストール時に使用するネットワークデバイスが net 別名に設定されている場合は、別名を設定し直す必要はありません。569 ページの「クライアントに対するキーのインストール」に進み、インストールを続行します。
- 使用するネットワークデバイスが net 別名に設定されていない場合は、別名を設定し直す必要があります。次の手順へ進みます。

4. net デバイス別名を設定します。

次のどちらかのコマンドを使って、net デバイス別名を設定します。

- このインストールに限り net デバイス別名を変更する場合は、devalias コマンドを使用します。

```
ok devalias net device-path

net device-path     デバイス device-path を net 別名に割り当てます。
```

- net デバイス別名を固定的に変更する場合は、nvalias コマンドを使用します。

```
ok nvalias net device-path
```



`net device-path` デバイス `device-path` を `net` 別名に割り当てます。

#### 例 44-2 net デバイス別名の確認と設定変更

次のコマンドは、`net` デバイス別名を確認して設定し直す方法を示しています。

デバイス別名を調べます。

```
ok devalias
screen                /pci@1f,0/pci@1,1/SUNW,m64B@2
net                   /pci@1f,0/pci@1,1/network@c,1
net2                  /pci@1f,0/pci@1,1/network@5,1
disk                  /pci@1f,0/pci@1/scsi@8/disk@0,0
cdrom                 /pci@1f,0/pci@1,1/ide@d/cdrom@0,0:f
keyboard              /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse                 /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
```

`/pci@1f,0/pci@1,1/network@5,1` というネットワークデバイスを使用する場合は、次のコマンドを入力します。

```
ok devalias net /pci@1f,0/pci@1,1/network@5,1
```

デバイス別名の設定方法の詳細は、『*OpenBoot 3.x* コマンド・リファレンスマニュアル』の「デバイスツリー」を参照してください。

## クライアントに対するキーのインストール

より高いセキュリティで保護された WAN ブートインストールを行う場合や、セキュリティ保護されていないインストールでデータの完全性をチェックする場合は、クライアントにキーをインストールする必要があります。ハッシュキーや暗号化鍵を使用すると、クライアントに転送されるデータを保護できます。これらのキーは、次の方法でインストールできます。

- OBP 変数を設定する – クライアントをブートする前に、OBP のネットワークブート引数にキーの値を設定します。これらのキーは、クライアントに対する以降の WAN ブートインストールで使用されます。
- ブート処理中にキーの値を入力する – `wanboot` プログラムの `boot>` プロンプトで、キーの値を設定します。この方法でインストールしたキーは、現在の WAN ブートインストールだけに使用されます。

動作中のクライアントの OBP にキーをインストールすることもできます。動作中のクライアントにキーをインストールするには、そのシステムが Solaris 9 12/03 オペレーティング環境またはその互換バージョンで稼働していることが必要です。

クライアントにキーをインストールするときには、必ずセキュリティ保護された接続を使用して、キーの値を転送してください。キーの値の機密性を確保するために、サイトのセキュリティポリシーに従ってください。

- OBP のネットワークブート引数にキーの値を割り当てる方法については、570 ページの「クライアントの OBP にキーをインストールする方法」を参照してください。
- ブート処理中にキーをインストールする方法については、578 ページの「対話式インストールを実行する方法」を参照してください。
- 動作中のクライアントの OBP にキーをインストールする方法については、572 ページの「動作中のクライアントにハッシュキーと暗号化鍵をインストールする方法」を参照してください。

## ▼ クライアントの OBP にキーをインストールする方法

OBP のネットワークブート引数にキーの値を割り当てるには、次の手順に従ってください。

1. WAN ブートサーバーで、Web サーバーユーザーと同じ役割になります。
2. クライアントの各キーの値を表示します。

```
# wanbootutil keygen -d -c -o net=net-ip,cid=client-ID,type=key-type
```

*net-ip* クライアントのサブネットの IP アドレスを指定します。

*client-ID* インストール対象であるクライアントの ID を指定します。クライアント ID は、ユーザーが定義した ID か、DHCP クライアント ID です。

*key-type* クライアントにインストールするキーのタイプを指定します。指定できるキータイプは、3des、aes、または sha1 です。

キーの値が 16 進数で表示されます。

3. クライアントにインストールする各キータイプについて、上記の手順を繰り返します。
4. クライアントシステムの実行レベルを 0 にします。

```
# init 0
```

ok プロンプトが表示されます。

5. クライアントの ok プロンプトで、ハッシュキーの値を設定します。

```
ok set-security-key wanboot-hmac-sha1 key-value
```

*set-security-key* クライアントにキーをインストールします。

*wanboot-hmac-sha1* HMAC SHA1 ハッシュキーをインストールするよう OBP に指示します。

*key-value* 手順 2 で表示された 16 進数の文字列を指定します。

HMAC SHA1 ハッシュキーがクライアントの OBP にインストールされます。

6. クライアントの **ok** プロンプトで、暗号化鍵をインストールします。

```
ok set-security-key wanboot-3des key-value
```

**set-security-key** クライアントにキーをインストールします。

**wanboot-3des** 3DES 暗号化鍵をインストールするよう OBP に指示します。AES 暗号化鍵を使用する場合は、この値を **wanboot-aes** にしてください。

**key-value** 暗号化鍵を表す 16 進数の文字列を指定します。

3DES 暗号化鍵がクライアントの OBP にインストールされます。

キーをインストールしたら、クライアントに対するインストールの準備は完了です。クライアントシステムのインストール方法については、574 ページの「クライアントのインストール」を参照してください。

7. (省略可能) クライアントの **OBP** にキーが設定されていることを確認します。

```
ok list-security-keys
```

Security Keys:

```
wanboot-hmac-sha1
```

```
wanboot-3des
```

8. (省略可能) キーを削除するには、次のコマンドを入力します。

```
ok set-security-key key-type
```

**key-type** 削除するキーのタイプを指定します。指定できる値は、**wanboot-hmac-sha1**、**wanboot-3des**、または **wanboot-aes** です。

#### 例 44-3 クライアントの OBP に対するキーのインストール

次の例は、クライアントの OBP にハッシュキーと暗号化鍵をインストールする方法を示しています。

WAN ブートサーバー上でキーの値を表示します。

```
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

上記の例では、次の情報が使用されています。

```
net=192.168.198.0
```

クライアントのサブネットの IP アドレスを指定します。

```
cid=010003BA152A42
```

クライアント ID を指定します。

```
b482aaab82cb8d5631e16d51478c90079cc1d463
```

クライアントの HMAC SHA1 ハッシュキーの値です。

例 44-3 クライアントの OBP に対するキーのインストール (続き)

```
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
クライアントの 3DES 暗号化鍵の値です。
```

インストールで AES 暗号化鍵を使用する場合、この暗号化鍵の値を表示するには、wanboot-3des を wanboot-aes に変更してください。

クライアントシステムにキーをインストールします。

```
ok set-security-key wanboot-hmac-sha1 b482aaab82cb8d5631e16d51478c90079cc1d463
ok set-security-key wanboot-3des 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

上記のコマンドは、次の処理を実行します。

- b482aaab82cb8d5631e16d51478c90079cc1d463 という値を持つ HMAC SHA1 ハッシュキーをクライアントにインストールします。
- 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04 という値を持つ 3DES 暗号化鍵をクライアントにインストールします。  
インストールで AES 暗号化鍵を使用する場合は、wanboot-3des を wanboot-aes に変更してください。

## ▼ 動作中のクライアントにハッシュキーと暗号化鍵をインストールする方法

動作中のクライアントの OBP にハッシュキーと暗号化鍵をインストールするには、次の手順に従ってください。

---

注 - この手順では、次のように仮定します。

- クライアントシステムの電源は入っている
  - Secure Shell (ssh) などのセキュリティ保護された接続を介してクライアントにアクセスできる
- 

1. WAN ブートサーバーで、Web サーバーユーザーと同じ役割になります。

2. クライアントの各キーの値を表示します。

```
# wanbootutil keygen -d -c -o net=net-ip,cid=client-ID,type=key-type
```

*net-ip* クライアントのサブネットの IP アドレスを指定します。

*client-ID* インストール対象であるクライアントの ID を指定します。クライアント ID は、ユーザーが定義した ID か、DHCP クライアント ID です。

*key-type* クライアントにインストールするキーのタイプを指定します。指定できるキータイプは、3des、aes、または sha1 です。

キーの値が 16 進数で表示されます。

3. クライアントにインストールする各キータイプについて、上記の手順を繰り返します。
4. クライアントマシン上でスーパーユーザーになります。
5. 動作中のクライアントマシンに、必要なキーをインストールします。

```
# /usr/lib/inet/wanboot/ickey -o type=key-type  
> key-value
```

*key-type* クライアントにインストールするキーのタイプを指定します。指定できるキータイプは、3des、aes、または sha1 です。

*key-value* 手順 2 で表示された 16 進数の文字列を指定します。

6. クライアントにインストールする各キータイプについて、上記の手順を繰り返します。  
キーをインストールしたら、クライアントに対するインストールの準備は完了です。クライアントシステムのインストール方法については、574 ページの「クライアントのインストール」を参照してください。

#### 例 44-4 動作中のクライアントシステムの OBP に対するキーのインストール

次の例は、動作中のクライアントの OBP にキーをインストールする方法を示しています。

WAN ブートサーバー上でキーの値を表示します。

```
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1  
b482aaab82cb8d5631e16d51478c90079cc1d463  
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des  
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

上記の例では、次の情報が使用されています。

net=192.168.198.0

クライアントのサブネットの IP アドレスを指定します。

cid=010003BA152A42

クライアント ID を指定します。

b482aaab82cb8d5631e16d51478c90079cc1d463

クライアントの HMAC SHA1 ハッシュキーの値です。

9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04

クライアントの 3DES 暗号化鍵の値です。

インストールで AES 暗号化鍵を使用する場合、この暗号化鍵の値を表示するには、type=3des を type=aes に変更してください。

動作中のクライアントの OBP にキーをインストールします。

例 44-4 動作中のクライアントシステムの OBP に対するキーのインストール (続き)

```
# /usr/lib/inet/wanboot/ickey -o type=sha1 b482aaab82cb8d5631e16d51478c90079cc1d463
# /usr/lib/inet/wanboot/ickey -o type=3des 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

上記のコマンドは、次の処理を実行します。

- b482aaab82cb8d5631e16d51478c90079cc1d463 という値を持つ HMAC SHA1 ハッシュキーをクライアントにインストールします。
- 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04 という値を持つ 3DES 暗号化鍵をクライアントにインストールします。

---

## クライアントのインストール

WAN ブートインストールを行うためのネットワークの準備が完了したら、システムのインストール方法を次の中から選択できます。

表 44-2 クライアントのインストール方法

方法	説明	参照先
自動インストール	クライアントにキーをインストールしたりクライアント構成情報を設定したりしてからクライアントをブートする場合は、このインストール方法を使用する	<ul style="list-style-type: none"><li>■ インストール前にクライアントにキーをインストールする方法については、569 ページの「クライアントに対するキーのインストール」を参照</li><li>■ 自動インストールの実行方法については、576 ページの「自動インストールを実行する方法」を参照</li></ul>
対話式インストール	ブート処理中にクライアント構成情報を設定する場合は、このインストール方法を使用する	578 ページの「対話式インストールを実行する方法」

表 44-2 クライアントのインストール方法 (続き)

方法	説明	参照先
DHCP サーバーを使ったインストール	インストール時にクライアント構成情報を提供するようにネットワークの DHCP サーバーを構成した場合は、このインストール方法を使用する	<ul style="list-style-type: none"> <li>■ WAN ブートインストールをサポートするように DHCP サーバーを構成する方法については、563 ページの「(省略可能) DHCP による構成情報の提供」を参照</li> <li>■ インストール時に DHCP サーバーを使用する方法については、582 ページの「DHCP サーバーを使ってインストールを行う方法」を参照</li> </ul>
ローカルの CD メディアを使ったインストール	クライアントの OBP が WAN ブートに対応していない場合は、Solaris 9 12/03 SOFTWARE CD のローカルコピーからクライアントをブートする	<ul style="list-style-type: none"> <li>■ クライアントの OBP が WAN ブートに対応しているかどうかを調べる方法については、567 ページの「クライアント OBP での WAN ブート対応を確認する方法」を参照</li> <li>■ Solaris 9 12/03 SOFTWARE CD のローカルコピーを使ってクライアントのインストールを行う方法については、583 ページの「ローカルの CD メディアを使ってインストールを行う方法」を参照</li> </ul>

## 自動 WAN ブートインストールの実行

インストールを実行する前にクライアントにキーをインストールしたりクライアント構成情報を設定したりする場合は、このインストール方法を使用します。この場合、WAN からクライアントをブートし、自動的にインストールを実行できます。

この手順では、すでにクライアントの OBP にキーをインストールしてあるか、またはセキュリティ保護されないインストールを実行していると仮定します。インストール前にクライアントにキーをインストールする方法については、569 ページの「クライアントに対するキーのインストール」を参照してください。

## ▼ 自動インストールを実行する方法

1. クライアントシステムが動作中の場合は、システムの実行レベルを **0** にします。

```
# init 0
ok プロンプトが表示されます。
```

2. クライアントシステムの **ok** プロンプトで、**OBP** のネットワークブート引数を設定します。

```
ok setenv network-boot-arguments
host-ip=client-IP,router-ip=router-ip,
subnet-mask=mask-value,hostname=client-name,
http-proxy=proxy-ip:port,file=wanbootCGI-URL
```

---

注 - このコマンド例には、読みやすいように改行が挿入されています。実際には、改行を挿入せずにコマンド全体を入力してください。

---

setenv network-boot-arguments	以下のブート引数を設定するよう OBP に指示します。
host-ip=client-IP	クライアントの IP アドレスを指定します。
router-ip=router-ip	ネットワークルーターの IP アドレスを指定します。
subnet-mask=mask-value	サブネットマスクの値を指定します。
hostname=client-name	クライアントのホスト名を指定します。
(省略可能) http-proxy=proxy-ip:port	ネットワークのプロキシサーバーの IP アドレスとポートを指定します。
file=wanbootCGI-URL	Web サーバー上の wanboot-cgi プログラムの URL を指定します。

3. クライアントをブートします。

```
ok boot net - install
net - install ネットワークブート引数を使って WAN からブートするよう、クライアントに指示します。
```

クライアントのインストールが WAN 経由で実行されます。WAN ブートプログラムに必要なインストール情報が見つからない場合、不足している情報の入力を求めるプロンプトが wanboot プログラムから表示されます。プロンプトに追加情報を入力します。



#### 例 44-5 自動 WAN ブートインストール

次の例では、マシンのブート前に、クライアントシステム seahag のネットワークブート引数を設定します。この例では、クライアントにハッシュキーと暗号化鍵がすでにインストールされていると仮定します。WAN からブートする前にキーをインストールする方法については、569 ページの「クライアントに対するキーのインストール」を参照してください。

```
ok setenv network-boot-arguments host-ip=192.168.198.136,  
router-ip=192.168.198.129,subnet-mask=255.255.255.192,hostname=seahag,  
file=http://192.168.198.135/cgi-bin/wanboot-cgi
```

```
ok boot net - install  
Resetting ...
```

```
Sun Blade 100 (UltraSPARC-IIe), No Keyboard  
Copyright 1998-2003 Sun Microsystems, Inc. All rights reserved.  
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.  
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.
```

```
Rebooting with command: boot net - install  
Boot device: /pci@1f,0/network@c,1 File and args: - install
```

次の変数が設定されます。

- クライアントの IP アドレスは 192.168.198.136 に設定されます。
- クライアントのルーター IP アドレスは 192.168.198.129 に設定されます。
- クライアントのサブネットマスクは 255.255.255.192 に設定されます。
- クライアントのホスト名は seahag に設定されます。
- wanboot-cgi プログラムは <http://192.168.198.135/cgi-bin/wanboot-cgi> に置かれています。

## 対話式 WAN ブートインストールの実行

インストール時にコマンド行からクライアントにキーをインストールしたりクライアント構成情報を設定したりする場合は、このインストール方法を使用します。

## ▼ 対話式インストールを実行する方法

この手順では、WAN インストールで HTTPS を使用していると仮定します。セキュリティ保護されないインストールを実行する場合で、キーを使用しないときは、クライアントのキーの表示やインストールを行わないでください。

1. WAN ブートサーバーで、Web サーバーユーザーと同じ役割になります。
2. クライアントの各キーの値を表示します。

```
# wanbootutil keygen -d -c -o net=net-ip,cid=client-ID,type=key-type
```

*net-ip* インストール対象であるクライアントのサブネットの IP アドレスを指定します。

*client-ID* インストール対象であるクライアントの ID を指定します。クライアント ID は、ユーザーが定義した ID か、DHCP クライアント ID です。

*key-type* クライアントにインストールするキーのタイプを指定します。指定できるキータイプは、3des、aes、または sha1 です。

キーの値が 16 進数で表示されます。

3. クライアントにインストールする各キータイプについて、上記の手順を繰り返します。
4. クライアントシステムが動作中の場合は、システムの実行レベルを **0** にします。
5. クライアントシステムの **ok** プロンプトで、**OBP** のネットワークブート引数を設定します。

```
ok setenv network-boot-arguments  
host-ip=client-IP,router-ip=router-ip,  
subnet-mask=mask-value,hostname=client-name,  
http-proxy=proxy-ip:port,bootserver=wanbootCGI-URL
```

---

注 - このコマンド例には、読みやすいように改行が挿入されています。実際には、改行を挿入せずにコマンド全体を入力してください。

---

setenv network-boot-arguments 以下のブート引数を設定するよう OBP に指示します。

host-ip=*client-IP* クライアントの IP アドレスを指定します。

router-ip=*router-ip* ネットワークルーターの IP アドレスを指定します。

subnet-mask=*mask-value* サブネットマスクの値を指定します。

<code>hostname=client-name</code>	クライアントのホスト名を指定します。
(省略可能) <code>http-proxy=proxy-ip:port</code>	ネットワークのプロキシサーバーの IP アドレスとポートを指定します。
<code>bootserver=wanbootCGI-URL</code>	Web サーバー上の <code>wanboot-cgi</code> プログラムの URL を指定します。

---

注 - `bootserver` 変数の URL 値に HTTPS URL を指定することはできません。URL は `http://` で開始する必要があります。

---

6. クライアントの **ok** プロンプトで、システムをブートします。

```
ok boot net -o prompt - install
```

```
net -o prompt - install
```

ネットワークからブートとインストールを行うよう、クライアントに指示します。wanboot プログラムは、クライアント構成情報の入力を求める `boot>` プロンプトを表示します。

`boot>` プロンプトが表示されます。

7. 暗号化鍵をインストールします。

```
boot> 3des=key-value
```

```
3des=key-value
```

手順 2 で表示された 3DES 暗号化鍵の 16 進数の文字列を指定します。

AES 暗号化鍵を使用する場合は、次のコマンドを使用してください。

```
boot> aes=key-value
```

8. ハッシュキーをインストールします。

```
boot> sha1=key-value
```

```
sha1=key-value
```

手順 2 で表示されたハッシュキーの値を指定します。

9. 次のコマンドを入力して、ブート処理を続行します。

```
boot> go
```

クライアントのインストールが WAN 経由で実行されます。

10. プロンプトが表示されたら、クライアント構成情報をコマンド行に入力します。

WAN ブートプログラムに必要なインストール情報が見つからない場合、不足している情報の入力を求めるプロンプトが wanboot プログラムから表示されます。プロンプトに追加情報を入力します。

#### 例 44-6 対話式 WAN ブートインストール

次の例では、インストール時に wanboot プログラムから、クライアントシステムのキーの値を入力するようプロンプトが表示されます。

WAN ブートサーバー上でキーの値を表示します。

```
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

上記の例では、次の情報が使用されています。

```
net=192.168.198.0
```

クライアントのサブネットの IP アドレスを指定します。

```
cid=010003BA152A42
```

クライアント ID を指定します。

```
b482aaab82cb8d5631e16d51478c90079cc1d463
```

クライアントの HMAC SHA1 ハッシュキーの値です。

```
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

クライアントの 3DES 暗号化鍵の値です。

インストールで AES 暗号化鍵を使用する場合、この暗号化鍵の値を表示するには、type=3des を type=aes に変更してください。

クライアントの OBP のネットワークブート引数を設定します。

```
ok setenv network-boot-arguments host-ip=192.168.198.136,
router-ip=192.168.198.129,subnet-mask=255.255.255.192,hostname=seahag,
bootserver=http://192.168.198.135/cgi-bin/wanboot-cgi
```

次の変数が設定されます。

- クライアントの IP アドレスは 192.168.198.136 に設定されます。
- クライアントのルーター IP アドレスは 192.168.198.129 に設定されます。
- クライアントのサブネットマスクは 255.255.255.192 に設定されます。
- クライアントのホスト名は seahag に設定されます。
- wanboot-cgi プログラムは http://192.168.198.135/cgi-bin/wanboot-cgi に置かれています。

クライアントのブートとインストールを実行します。

```
ok boot net -o prompt - install
Resetting ...
```

```
Sun Blade 100 (UltraSPARC-IIe), No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.
```

#### 例 44-6 対話式 WAN ブートインストール (続き)

```
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.
```

```
Rebooting with command: boot net -o prompt  
Boot device: /pci@1f,0/network@c,1 File and args: -o prompt
```

```
boot> 3des=9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

```
boot> sha1=b482aaab82cb8d5631e16d51478c90079cc1d463
```

```
boot> go
```

上記のコマンドは、次の処理を実行します。

- 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04 という値を持つ 3DES 暗号化鍵をクライアントにインストールします。
- b482aaab82cb8d5631e16d51478c90079cc1d463 という値を持つ HMAC SHA1 ハッシュキーをクライアントにインストールします。
- インストールを開始します。

## DHCP サーバーを使ったインストール

WAN ブートオプションをサポートするように DHCP サーバーを構成した場合、DHCP サーバーを使ってインストール時にクライアント構成情報を提供できます。WAN ブートインストールをサポートするように DHCP サーバーを構成する方法については、563 ページの「(省略可能) DHCP による構成情報の提供」を参照してください。

この手順では、次のように仮定します。

- クライアントシステムが動作中である
- すでにクライアントにキーをインストールしてあるか、またはセキュリティ保護されないインストールを実行している  
インストール前にクライアントにキーをインストールする方法については、569 ページの「クライアントに対するキーのインストール」を参照してください。
- SbootURI および SHTTPproxy WAN ブートオプションをサポートするように DHCP サーバーを構成してある  
これらのオプションは、WAN ブートに必要な構成情報を DHCP サーバーで提供できるようにします。

DHCP サーバーにインストールオプションを設定する方法については、75 ページの「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。

## ▼ DHCP サーバーを使ってインストールを行う方法

1. クライアントシステムが動作中の場合は、システムの実行レベルを **0** にします。

```
# init 0
ok プロンプトが表示されます。
```

2. クライアントシステムの **ok** プロンプトで、**OBP** のネットワークブート引数を設定します。

```
ok setenv network-boot-arguments dhcp,hostname=client-name

setenv network-boot-arguments 以下のブート引数を設定するよう OBP
                                に指示します。

dhcp                            DHCP サーバーを使ってクライアント
                                を構成するよう、OBP に指示します。

hostname=client-name            クライアントに割り当てるホスト名を
                                指定します。
```

3. ネットワークからクライアントをブートします。

```
ok boot net - install

net - install   ネットワークブート引数を使って WAN からブートするよ
                う、クライアントに指示します。

クライアントのインストールが WAN 経由で実行されます。WAN ブートプログラムに必要なインストール情報が見つからない場合、不足している情報の入力を求めるプロンプトが wanboot プログラムから表示されます。プロンプトに追加情報を入力します。
```

### 例 44-7 DHCP サーバーを使った WAN ブートインストール

次の例では、ネットワーク上の DHCP サーバーからクライアント構成情報が提供されます。この例では、クライアントのホスト名として `myhost` を要求しています。

```
ok setenv network-boot-arguments dhcp,
hostname=myhost
```

```
ok boot net - install
Resetting ...
```

```
Sun Blade 100 (UltraSPARC-IIe), No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.
```

```
Rebooting with command: boot net - install
```

例 44-7 DHCP サーバーを使った WAN ブートインストール (続き)

```
Boot device: /pci@1f,0/network@c,1 File and args: - install
```

## ローカルの CD メディアを使ったインストール

クライアントの OBP が WAN ブートに対応していない場合は、Solaris SOFTWARE 1 of 2 CD をクライアントの CD-ROM ドライブに挿入して、インストールを実行できます。ローカル CD を使用する場合、クライアントは、WAN ブートサーバーからではなくローカルメディアから wanboot プログラムを取得します。

ローカル CD から WAN ブートインストールを実行するには、次の手順に従ってください。

### ▼ ローカルの CD メディアを使ってインストールを行う方法

この手順では、WAN インストールで HTTPS を使用していると仮定します。セキュリティ保護されないインストールを実行する場合は、クライアントのキーの表示やインストールを行わないでください。

1. WAN ブートサーバーで、Web サーバーユーザーと同じ役割になります。
2. クライアントの各キーの値を表示します。

```
# wanbootutil keygen -d -c -o net=net-ip,cid=client-ID,type=key-type
```

*net-ip* インストール対象であるクライアントのネットワーク IP アドレスを指定します。

*client-ID* インストール対象であるクライアントの ID を指定します。クライアント ID は、ユーザーが定義した ID か、DHCP クライアント ID です。

*key-type* クライアントにインストールするキーのタイプを指定します。指定できるキータイプは、3des、aes、または sha1 です。

キーの値が 16 進数で表示されます。

3. クライアントにインストールする各キータイプについて、上記の手順を繰り返します。
4. クライアントシステムの CD-ROM ドライブに Solaris SOFTWARE 1 of 2 CD を挿入します。
5. クライアントシステムの電源を入れます。
6. CD からクライアントをブートします。

```
ok boot cdrom -o prompt -F wanboot - install
```

- cdrom            ローカル CD-ROM からブートするよう、OBP に指示します。
- o prompt       クライアント構成情報の入力をユーザーに求めるよう、wanboot プログラムに指示します。
- F wanboot      CD-ROM から wanboot プログラムを読み込むよう、OBP に指示します。
- install       WAN ブートインストールを実行するよう、クライアントに指示します。

クライアントの OBP は、Solaris SOFTWARE 1 of 2 CD から wanboot プログラムを読み込みます。wanboot プログラムによってシステムがブートされ、boot> プロンプトが表示されます。

#### 7. 暗号化鍵の値を入力します。

```
boot> 3des=key-value
```

3des=key-value    手順 2 で表示された 3DES 暗号化鍵の 16 進数の文字列を指定します。

AES 暗号化鍵を使用する場合は、次のコマンドを使用してください。

```
boot> aes=key-value
```

#### 8. ハッシュキーの値を入力します。

```
boot> sha1=key-value
```

sha1=key-value    手順 2 で表示されたハッシュキーの 16 進数の文字列を指定します。

#### 9. ネットワークインタフェース変数を設定します。

```
boot> variable=value [,variable=value*]
```

boot> プロンプトで、次の変数と値のペアを入力します。

host-ip=client-IP                      クライアントの IP アドレスを指定します。

router-ip=router-ip                    ネットワークルーターの IP アドレスを指定します。

subnet-mask=mask-value                サブネットマスクの値を指定します。

hostname=client-name                  クライアントのホスト名を指定します。

(省略可能) http-proxy=proxy-ip:port   ネットワークのプロキシサーバーの IP アドレスとポート番号を指定します。

bootserver=wanbootCGI-URL            Web サーバー上の wanboot-cgi プログラムの URL を指定します。



---

注 - `bootserver` 変数の URL 値に HTTPS URL を指定することはできません。URL は `http://` で開始する必要があります。

---

これらの変数は、次の方法で入力できます。

- `boot>` プロンプトで、変数と値のペアを 1 組入力し、Return キーを押します。

```
boot> host-ip=client-IP
boot> subnet-mask=mask-value
```

- `boot>` プロンプトで、変数と値のすべてのペアを 1 行に入力し、Return キーを押します。変数と値の各ペアを区切るには、コンマを使用します。

```
boot> host-ip=client-IP,subnet-mask=mask-value,
router-ip=router-ip,hostname=client-name,
http-proxy=proxy-ip:port,bootserver=wanbootCGI-URL
```

#### 10. 次のコマンドを入力して、ブート処理を続行します。

```
boot> go
```

クライアントのインストールが WAN 経由で実行されます。WAN ブートプログラムに必要なインストール情報が見つからない場合、不足している情報の入力を求めるプロンプトが `wanboot` プログラムから表示されます。プロンプトに追加情報を入力します。

#### 例 44-8 ローカルの CD メディアを使ったインストール

次の例では、インストール時にローカル CD 上の `wanboot` プログラムから、クライアントのネットワークインタフェース変数を設定するようプロンプトが表示されません。

WAN ブートサーバー上でキーの値を表示します。

```
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

上記の例では、次の情報が使用されています。

```
net=192.168.198.0
```

クライアントのサブネットの IP アドレスを指定します。

```
cid=010003BA152A42
```

クライアント ID を指定します。

```
b482aaab82cb8d5631e16d51478c90079cc1d463
```

クライアントの HMAC SHA1 ハッシュキーの値です。

例 44-8 ローカルの CD メディアを使ったインストール (続き)

9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04  
クライアントの 3DES 暗号化鍵の値です。

インストールで AES 暗号化鍵を使用する場合、この暗号化鍵の値を表示するには、`type=3des` を `type=aes` に変更してください。

クライアントのブートとインストールを実行します。

```
ok boot cdrom -o prompt -F wanboot - install
Resetting ...
```

```
Sun Blade 100 (UltraSPARC-IIe), No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.
```

```
Rebooting with command: boot cdrom -F wanboot - install
Boot device: /pci@1f,0/network@c,1 File and args: -o prompt
```

```
boot> 3des=9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
boot> sha1=b482aaab82cb8d5631e16d51478c90079cc1d463
boot> host-ip=192.168.198.124
boot> subnet-mask=255.255.255.128
boot> router-ip=192.168.198.1
boot> hostname=myhost
boot> client-id=010003BA152A42
boot> bootserver=http://192.168.198.135/cgi-bin/wanboot-cgi
boot> go
```

上記のコマンドは、次の処理を実行します。

- 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04 という値を持つ 3DES 暗号化鍵をクライアントに入力します。
- b482aaab82cb8d5631e16d51478c90079cc1d463 という値を持つ HMAC SHA1 ハッシュキーをクライアントに入力します。
- クライアントの IP アドレスを 192.168.198.124 に設定します。
- クライアントのサブネットマスクを 255.255.255.128 に設定します。
- クライアントのルーターの IP アドレスを 192.168.198.1 に設定します。
- クライアントのホスト名を myhost に設定します。

例 44-8 ローカルの CD メディアを使ったインストール (続き)

- クライアント ID を 010003BA152A42 に設定します。
- wanboot-cgi プログラムの場所を `http://192.168.198.135/cgi-bin/wanboot-cgi/` に設定します。



## 第 45 章

# SPARC: WAN ブートによるインストール (例)

---

この章では、広域ネットワーク (WAN) 経由でクライアントシステムの設定とインストールを行う例を示します。この例では、HTTPS 接続を介してセキュリティ保護された WAN ブートインストールを実行する方法について説明します。

- 590 ページの「サイトの設定例」
- 591 ページの「ドキュメントルートディレクトリの作成」
- 591 ページの「WAN ブートミニルートの作成」
- 592 ページの「WAN ブートサーバーへの wanboot プログラムのインストール」
- 592 ページの「/etc/netboot ディレクトリの作成」
- 593 ページの「WAN ブートサーバーへの wanboot-cgi プログラムのコピー」
- 593 ページの「WAN ブートサーバーをログサーバーとして構成」
- 593 ページの「HTTPS を使用するための WAN ブートサーバーの構成」
- 594 ページの「クライアントに対する信頼できる証明書の提供」
- 594 ページの「(省略可能) クライアント認証用の非公開鍵と証明書の使用」
- 595 ページの「サーバーとクライアントのキーの作成」
- 596 ページの「フラッシュアーカイブの作成」
- 596 ページの「sysidcfg ファイルの作成」
- 597 ページの「クライアントのプロファイルの作成」
- 597 ページの「rules ファイルの作成と妥当性検査」
- 598 ページの「システム構成ファイルの作成」
- 599 ページの「wanboot.conf ファイルの作成」
- 600 ページの「クライアント OBP での WAN ブート対応の確認」
- 601 ページの「OBP の net デバイス別名の確認」
- 601 ページの「クライアントに対するキーのインストール」
- 602 ページの「クライアントのインストール」

---

## サイトの設定例

図 45-1 は、この例で使用するサイトの設定を示しています。

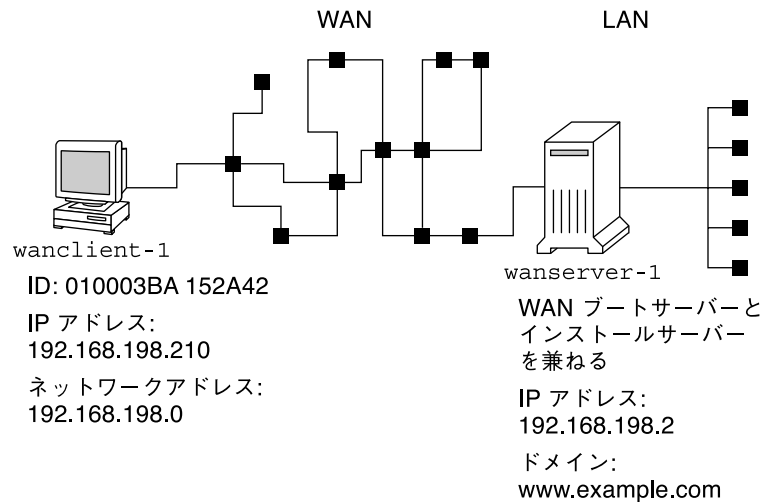


図 45-1 WAN ブートインストール用のサイト例

このサイト例には次のような特徴があります。

- サーバー wanserver-1 は、WAN ブートサーバーおよびインストールサーバーとして構成される
- wanserver-1 の IP アドレスは 192.168.198.2
- wanserver-1 のドメイン名は www.example.com
- wanserver-1 では、Solaris 9 4/04 オペレーティング環境が稼働している
- wanserver-1 では、Apache Web サーバーが稼働している。wanserver-1 の Apache ソフトウェアは、HTTPS をサポートするように構成されている
- インストール対象であるクライアントの名前は wanclient-1
- wanclient-1 は UltraSPARCII システムである
- wanclient-1 のクライアント ID は 010003BA152A42
- wanclient-1 の IP アドレスは 192.168.198.210
- クライアントのサブネットの IP アドレスは 192.168.198.0
- クライアントシステム wanclient-1 は、インターネットにアクセスできるが、wanserver-1 が置かれているネットワークには直接接続されていない
- wanclient-1 に対して、新たに Solaris 9 4/04 オペレーティング環境をインストールする

---

## ドキュメントルートディレクトリの作成

インストールファイルとデータを保存するために、wanserver-1 のドキュメントルートディレクトリ (/opt/apache/htdocs) に次のディレクトリを作成します。

- Solaris Flash のディレクトリ

```
wanserver-1# mkdir -p /opt/apache/htdocs/flash/
```

- WAN ブートミニルートのディレクトリ

```
wanserver-1# mkdir -p /opt/apache/htdocs/miniroot/
```

- wanboot プログラムのディレクトリ

```
wanserver-1# mkdir -p /opt/apache/htdocs/wanboot/
```

---

## WAN ブートミニルートの作成

setup\_install\_server (1M) に -w オプションを指定して実行することで、WAN ブートミニルートと Solaris ソフトウェアイメージを、wanserver-1 の /export/install/Solaris\_9 ディレクトリにコピーします。

wanserver-1 に接続されているメディアドライブに Solaris SOFTWAREのメディアを挿入します。次のコマンドを入力します。

```
wanserver-1# mkdir -p /export/install/sol_9_sparc
wanserver-1# cd /cdrom/cdrom0/s0/Solaris_9/Tools
wanserver-1# ./setup_install_server -w /export/install/sol_9_sparc/miniroot \
/export/install/sol_9_sparc
```

WAN ブートサーバーのドキュメントルートディレクトリ (/opt/apache/htdocs/) に、WAN ブートミニルートを移動します。

```
wanserver-1# mv /export/install/sol_9_sparc/miniroot \
/opt/apache/htdocs/miniroot/miniroot.s9_sparc
```

---

## WAN ブートサーバーへの wanboot プログラムのインストール

wanboot プログラムを WAN ブートサーバーにインストールするには、Solaris 9 4/04 ソフトウェアのメディアから WAN ブートサーバーのドキュメントルートディレクトリに、このプログラムをコピーします。

wanserver-1 に接続されているメディアドライブに Solaris DVD または Solaris SOFTWARE 1 of 2 CD を挿入し、次のコマンドを入力します。

```
wanserver-1# cd /cdrom/cdrom0/s0/Solaris_9/Tools/Boot/platform/sun4u/
wanserver-1# cp wanboot /opt/apache/htdocs/wanboot/wanboot.s9_sparc
```

---

## /etc/netboot ディレクトリの作成

WAN ブートサーバーの /etc/netboot ディレクトリに、wanclient-1 のサブディレクトリを作成します。WAN ブートインストールプログラムは、インストール時に、このディレクトリから構成情報とセキュリティ情報を取得します。

wanclient-1 は、サブネット 192.168.198.0 上にあり、010003BA152A42 というクライアント ID を持っています。/etc/netboot に wanclient-1 用のサブディレクトリを作成するには、次の作業を実行します。

- /etc/netboot ディレクトリを作成します。
- /etc/netboot ディレクトリのアクセス権を 700 に変更します。
- /etc/netboot ディレクトリの所有権を Web サーバプロセスの所有者に渡します。
- Web サーバユーザーと同じ役割になります。
- /etc/netboot ディレクトリに、サブネット名と同じ名前のサブディレクトリ 192.168.198.0 を作成します。
- このサブネットディレクトリに、クライアント ID と同じ名前のサブディレクトリを作成します。
- /etc/netboot のサブディレクトリのアクセス権を 700 に変更します。

```
wanserver-1# cd /
wanserver-1# mkdir /etc/netboot/
wanserver-1# chmod 700 /etc/netboot
wanserver-1# chown nobody:admin /etc/netboot
wanserver-1# exit
wanserver-1# su nobody
```



```
Password:
nobody# mkdir -p /etc/netboot/192.168.198.0/010003BA152A42
nobody# chmod 700 /etc/netboot/192.168.198.0
nobody# chmod 700 /etc/netboot/192.168.198.0/010003BA152A42
```

---

## WAN ブートサーバーへの wanboot-cgi プログラムのコピー

Solaris 9 4/04 オペレーティング環境が稼働しているシステムでは、wanboot-cgi プログラムは /usr/lib/inet/wanboot/ ディレクトリに置かれています。WAN ブートサーバーからインストールデータを転送できるようにするには、Web サーバーソフトウェアのディレクトリにある cgi-bin ディレクトリに wanboot-cgi プログラムをコピーします。

```
wanserver-1# cp /usr/lib/inet/wanboot/wanboot-cgi \
/opt/apache/cgi-bin/wanboot-cgi
wanserver-1# chmod 755 /opt/apache/cgi-bin/wanboot-cgi
```

---

## WAN ブートサーバーをログサーバーとして構成

WAN ブートサーバー上でブートログメッセージおよびインストールログメッセージを表示するには、wanserver-1 の cgi-bin ディレクトリに bootlog-cgi スクリプトをコピーします。

```
wanserver-1# cp /usr/lib/inet/wanboot/bootlog-cgi /opt/apache/cgi-bin/
wanserver-1# chmod 755 /opt/apache/cgi-bin/bootlog-cgi
```

---

## HTTPS を使用するための WAN ブートサーバーの構成

WAN ブートインストールで HTTPS を使用するには、Web サーバーソフトウェアの SSL サポートを有効にする必要があります。また、WAN ブートサーバーにデジタル証明書をインストールすることも必要です。この例では、wanserver-1 上の Apache

Web サーバーが SSL を使用するように構成されていると仮定しています。この例では、wanserver-1 の識別情報を確立するデジタル証明書および認証局が、すでに wanserver-1 にインストールされていると仮定しています。

SSL を使用するように Web サーバーソフトウェアを構成する方法については、Web サーバーのマニュアルを参照してください。

---

## クライアントに対する信頼できる証明書の提供

クライアントに対して身分証明を行うようサーバーに要求することで、HTTPS を介してサーバーからクライアントに転送されるデータを保護できます。サーバー認証を有効にするには、信頼できる証明書をクライアントに提供します。信頼できる証明書に従って、クライアントはインストール時にサーバーの識別情報を確認します。

信頼できる証明書をクライアントに提供するには、Web サーバーユーザーと同じユーザー役割になる必要があります。その後、証明書を分割して、信頼できる証明書を抽出します。次に、/etc/netboot ディレクトリにあるクライアントの truststore ファイルに、信頼できる証明書を挿入します。

この例では、Web サーバーユーザー役割 nobody になります。続いて、cert.p12 という名前の、サーバーの PKCS#12 証明書を分割して、/etc/netboot ディレクトリにある wanclient-1 用のディレクトリに、信頼できる証明書を挿入します。

```
wanserver-1# su nobody
Password:
wanserver-1# wanbootutil p12split -i cert.p12 -t \
/etc/netboot/192.168.198.0/010003BA152A42/truststore
```

---

## (省略可能) クライアント認証用の非公開鍵と証明書の使用

インストール時にデータをさらに保護するために、wanserver-1 に対して身分証明を行うよう wanclient-1 に要求することもできます。WAN ブートインストールでクライアント認証を有効にするには、/etc/netboot ディレクトリのクライアントのサブディレクトリに、クライアントの証明書と非公開鍵を挿入します。

クライアントに非公開鍵と証明書を提供するには、次の手順に従います。

- Web サーバーユーザーと同じ役割になります。

- PKCS#12 ファイルを、非公開鍵とクライアント証明書に分割します。
- クライアントの certstore ファイルに証明書を挿入します。
- クライアントの keystore ファイルに非公開鍵を挿入します。

この例では、Web サーバーユーザー役割 nobody になります。続いて、cert.p12 という名前の、サーバーの PKCS#12 証明書を分割します。/etc/netboot ディレクトリにある wanclient-1 用のサブディレクトリに、証明書を挿入します。wanclient.key という名前を付けた非公開鍵を、クライアントの keystore ファイルに挿入します。

```
wanserver-1# su nobody
Password:
wanserver-1# wanbootutil pl2split -i cert.p12 -c \
/etc/netboot/192.168.198.0/010003BA152A42/certstore -k wanclient.key
wanserver-1# wanbootutil keymgmt -i -k wanclient.key \
-s /etc/netboot/192.168.198.0/010003BA152A42/keystore \
-o type=rsa
```

---

## サーバーとクライアントのキーの作成

サーバーとクライアントの間で転送されるデータを保護するには、ハッシュキーと暗号化鍵を作成します。サーバーはハッシュキーを使って、wanboot プログラムの完全性を保護します。サーバーは暗号化鍵を使って、構成データとインストールデータを暗号化します。クライアントはハッシュキーを使って、ダウンロードした wanboot プログラムの完全性を確認します。クライアントは暗号化鍵を使って、インストール時にデータを復号化します。

まず、Web サーバーユーザーと同じ役割になります。この例の場合、Web サーバーユーザー役割は nobody です。

```
wanserver-1# su nobody
Password:
```

次に、wanbootutil keygen コマンドを使用して、wanserver-1 の HMAC SHA1 マスターキーを作成します。

```
wanserver-1# wanbootutil keygen -m
```

次に、wanclient-1 のハッシュキーと暗号化鍵を作成します。

```
wanserver-1# wanbootutil keygen -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
wanserver-1# wanbootutil keygen -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
```

上記のコマンドは、wanclient-1 の HMAC SHA1 ハッシュキーと 3DES 暗号化鍵を作成します。192.168.198.0 は wanclient-1 のサブネット、010003BA152A42 は wanclient-1 のクライアント ID です。

---

## フラッシュアーカイブの作成

この例では、wanserver-1 マスターシステムのクローンを作成することで、フラッシュアーカイブを作成します。このアーカイブは、sol-9-sparc という名前で、マスターシステムから正確にコピーされます。アーカイブはマスターシステムと全く同一の複製です。アーカイブは sol-9-sparc.flar に格納されます。WAN ブートサーバーのドキュメントルートディレクトリの flash/archives サブディレクトリにこのアーカイブを保存します。

```
wanserver-1# flar create -n sol-9-sparc /opt/apache/htdocs/flash/archives/sol-9-sparc.flar
```

---

## sysidcfg ファイルの作成

wanclient-1 システムを事前設定するには、キーワードと値を sysidcfg ファイルに指定します。wanserver-1 のドキュメントルートディレクトリの flash サブディレクトリにこのファイルを保存します。

例 45-1 client-1 システム用の sysidcfg ファイル

wanclient-1 用の sysidcfg ファイルの例を次に示します。このシステムのホスト名、IP アドレス、およびネットマスクは、ネームサービスを編集することにより、すでに事前設定されています。このファイルは、  
/opt/apache/htdocs/flash/sol\_9\_sparc ディレクトリに置かれます。

```
network_interface=primary {hostname=wanclient-1
                             default_route=192.168.198.1
                             ip_address=192.168.198.210
                             netmask=255.255.255.0
                             protocol_ipv6=no}

timezone=US/Central
system_locale=C
terminal=xterm
timeserver=localhost
name_service=NIS {name_server=matter(192.168.254.254)
                  domain_name=leti.example.com
                  }
security_policy=none
```

---

## クライアントのプロファイルの作成

wanclient-1 システム用に、wanclient\_1\_prof という名前のプロファイルを作成します。wanclient\_1\_prof ファイルには次のようなエントリがあり、wanclient-1 システムに Solaris 9 ソフトウェアをインストールするように定義しています。

```
# profile keywords      profile values
# -----
install_type           flash_install
archive_location       https://192.168.198.2/htdocs/flash/sol_9_sparc/archive1.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、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	圧縮されたフラッシュアーカイブが wanserver-1 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 (/) のサイズはフラッシュアーカイブのサイズに基づいて決定されます。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 上にインストールされます。

---

## rules ファイルの作成と妥当性検査

カスタム JumpStart プログラムでは、wanclient-1 システム用の正しいインストールプロファイルを選択するために、rules ファイルが使用されます。rules という名前のテキストファイルを作成します。次に、このファイルにキーワードと値を追加します。

wanclient-1 システムの IP アドレスは 192.168.198.210 です。network ルールキーワードを使って、カスタム JumpStart プログラムで client-1 のインストールに使用するプロファイルを指定します。

```
network 192.168.198.210 - wanclient_1_prof -
```

この rules ファイルは、wanclient\_1\_prof を使って wanclient-1 に Solaris 9 オペレーティング環境をインストールするよう、カスタム JumpStart プログラムに指示します。

このルールファイルに wanclient\_rule という名前を付けます。

プロファイルと rules ファイルを作成した後、check スクリプトを実行して、ファイルの妥当性を検査します。

```
wanserver-1# ./check -r wanclient_rule
```

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。

rules.ok ファイルを /opt/apache/htdocs/flash/ ディレクトリに保存します。

---

## システム構成ファイルの作成

システム構成ファイルを作成して、インストールサーバー上の sysidcfg ファイルとカスタム JumpStart ファイルの場所を記述します。WAN ブートサーバーがアクセスできるディレクトリに、このファイルを保存します。

次の例では、wanboot-cgi プログラムは、WAN ブートサーバーのドキュメントルートディレクトリ内で sysidcfg ファイルとカスタム JumpStart ファイルを検索します。WAN ブートサーバーのドメイン名は <https://www.example.com> です。WAN ブートサーバーは HTTPS を使用するように構成されているため、インストール時にデータとファイルが保護されます。

この例では、システム構成ファイルは sys.conf という名前で、WAN ブートサーバーの /etc/netboot ディレクトリ内に保存されます。sysidcfg ファイルとカスタム JumpStart ファイルは、ドキュメントルートディレクトリの Solaris Flash サブディレクトリ flash に置かれています。

```
SsysidCF=https://www.example.com/htdocs/flash/  
SjumpsCF=https://www.example.com/htdocs/flash/
```

---

## wanboot.conf ファイルの作成

WAN ブートは、wanboot.conf ファイルに記述されている情報を使ってクライアントマシンのインストールを行います。テキストエディタを使って wanboot.conf ファイルを作成します。WAN ブートサーバー上の /etc/netboot ディレクトリにある該当のクライアントのサブディレクトリに、このファイルを保存します。

次の wanclient-1 用の wanboot.conf ファイルには、HTTPS を使って WAN インストールを行うための構成情報が記述されています。このファイルは WAN ブートに対して、HMAC SHA1 ハッシュキーと 3DES 暗号化鍵を使ってデータを保護することも指示しています。

```
boot_file=/wanboot/wanboot.s9_sparc
root_server=https://www.example.com/cgi-bin/wanboot-cgi
root_file=/miniroot/miniroot.s9_sparc
signature_type=sha1
encryption_type=3des
server_authentication=yes
client_authentication=no
resolve_hosts=
boot_logger=https://www.example.com/cgi-bin/bootlog-cgi
system_conf=sys.conf
```

この wanboot.conf ファイルで指定されている構成は次のとおりです。

```
boot_file=/wanboot/wanboot.s9_sparc
```

wanboot プログラムの名前は wanboot.s9\_sparc です。このプログラムは、wanserver-1 のドキュメントルートディレクトリ内の wanboot ディレクトリに置かれています。

```
root_server=https://www.example.com/cgi-bin/wanboot-cgi
```

wanserver-1 上の wanboot-cgi プログラムの場所は https://www.example.com/cgi-bin/wanboot-cgi です。URL の https という部分は、この WAN ブートインストールで HTTPS を使用することを示しています。

```
root_file=/miniroot/miniroot.s9_sparc
```

WAN ブートミニルートの名前は miniroot.s9\_sparc です。ミニルートは、wanserver-1 のドキュメントルートディレクトリ内の miniroot ディレクトリに置かれています。

```
signature_type=sha1
```

wanboot プログラムと WAN ブートファイルシステムは、HMAC SHA1 ハッシュキーで署名されます。

```
encryption_type=3des
```

wanboot プログラムと WAN ブートファイルシステムは、3DES 暗号化鍵で暗号化されます。

`server_authentication=yes`  
インストール時にサーバー認証が行われます。

`client_authentication=no`  
インストール時にクライアント認証は行われません。

---

注 - 594 ページの「(省略可能) クライアント認証用の非公開鍵と証明書の使用」の作業を実行した場合、このパラメータの設定は `client_authentication=yes` としてください。

---

`resolve_hosts=`  
WAN インストールの実行に必要な追加のホスト名はありません。 `wanboot-cgi` プログラムに必要なホスト名はすべて、 `wanboot.conf` ファイルとクライアント証明書に指定されています。

`boot_logger=https://www.example.com/cgi-bin/bootlog-cgi`  
HTTPS を使って `wanserver-1` 上でブートログメッセージおよびインストールログメッセージが記録されます。

`system_conf=sys.conf`  
`sysid.cfg` ファイルとカスタム JumpStart ファイルの場所を指定するシステム構成ファイルは、 `sys.conf` という名前で、 `wanserver-1` の `/etc/netboot` ディレクトリに置かれています。

この例では、 `wanserver-1` の `/etc/netboot/192.168.198.0/010003BA152A42` ディレクトリに `wanboot.conf` ファイルを保存します。

---

## クライアント OBP での WAN ブート対応の確認

クライアントシステムに次のコマンドを入力して、クライアントの OBP が WAN ブートに対応しているかどうかを調べます。

```
# eeprom | grep network-boot-arguments
network-boot-arguments: data not available
```

上の例では、出力に `network-boot-arguments: data not available` が表示されているので、クライアント OBP は WAN ブートに対応しています。



---

## OBP の net デバイス別名の確認

boot net コマンドを使って WAN からクライアントをブートするには、net デバイス別名にクライアントの主ネットワークデバイスが設定されている必要があります。クライアントの ok プロンプトに devalias コマンドを入力して、net 別名に主ネットワークデバイス /pci@1f,0/pci@1,1/network@c,1 が設定されていることを確認します。

```
ok devalias
screen                /pci@1f,0/pci@1,1/SUNW,m64B@2
net                   /pci@1f,0/pci@1,1/network@c,1
net2                  /pci@1f,0/pci@1,1/network@5,1
disk                  /pci@1f,0/pci@1/scsi@8/disk@0,0
cdrom                 /pci@1f,0/pci@1,1/ide@d/cdrom@0,0:f
keyboard              /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse                 /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
```

上記の出力例では、主ネットワークデバイス /pci@1f,0/pci@1,1/network@c,1 が net 別名に割り当てられています。別名を設定し直す必要はありません。

---

## クライアントに対するキーのインストール

595 ページの「サーバーとクライアントのキーの作成」で、インストール時にデータを保護するためのハッシュキーと暗号化鍵を作成しました。インストール時に wanserver-1 から転送されるデータをクライアントが復号化できるようにするには、これらのキーを wanclient-1 にインストールします。

wanserver-1 上でキーの値を表示します。

```
wanserver-1# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
wanserver-1# wanbootutil keygen -d -c -o net=192.168.198.0,cid=010003BA152A42,type=3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

上記の例では、次の情報が使用されています。

```
net=192.168.198.0
```

クライアントのサブネットの IP アドレスを指定します。

```
cid=010003BA152A42
```

クライアント ID を指定します。

```
b482aaab82cb8d5631e16d51478c90079cc1d463
```

クライアントの HMAC SHA1 ハッシュキーの値です。

9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04  
クライアントの 3DES 暗号化鍵の値です。

インストールで AES 暗号化鍵を使用する場合、この暗号化鍵の値を表示するには、type=3des を type=aes に変更してください。

wanclient-1 の ok プロンプトで、キーをインストールします。

```
ok set-security-key wanboot-hmac-sha1
b482aaab82cb8d5631e16d51478c90079cc1d463
ok set-security-key wanboot-3des
9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04
```

上記のコマンドは、次の処理を実行します。

- b482aaab82cb8d5631e16d51478c90079cc1d463 という値を持つ HMAC SHA1 ハッシュキーを wanclient-1 にインストールします。
- 9ebc7a57f240e97c9b9401e9d3ae9b292943d3c143d07f04 という値を持つ 3DES 暗号化鍵を wanclient-1 にインストールします。

---

## クライアントのインストール

自動インストールを実行するには、wanclient-1 の ok プロンプトでネットワークブート引数を設定してから、クライアントをブートします。

```
ok setenv network-boot-arguments host-ip=192.168.198.210,
router-ip=192.168.198.1,subnet-mask=255.255.255.0,hostname=wanclient-1,
file=http://192.168.198.2/cgi-bin/wanboot-cgi
```

```
ok boot net - install
Resetting ...
```

```
Sun Blade 100 (UltraSPARC-IIe), No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.x.build_28, 256 MB memory installed, Serial #50335475.
Ethernet address 0:3:ba:e:f3:75, Host ID: 83000ef3.
```

```
Rebooting with command: boot net - install
Boot device: /pci@1f,0/network@c,1 File and args: - install
```

```
<time unavailable> wanboot progress: wanbootfs: Read 68 of 68 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
```

```
Fri Jun 20 09:16:06 wanboot progress: miniroot: Read 166067 of 166067 kB (100%)
Fri Jun 20Tue Apr 15 09:16:06 wanboot info: miniroot: Download complete
SunOS Release 5.9 Version WANboot10:04/11/03 64-bit
Copyright 1983-2003 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Configuring devices.
```

次の変数が設定されます。

- クライアントの IP アドレスは 192.168.198.210 に設定されます。
- クライアントのルーター IP アドレスは 192.168.198.1 に設定されます。
- クライアントのサブネットマスクは 255.255.255.0 に設定されます。
- クライアントのホスト名は wanclient-1 に設定されます。
- wanboot-cgi プログラムは `http://192.168.198.2/cgi-bin/wanboot-cgi` に置かれています。

クライアントのインストールが WAN 経由で実行されます。wanboot プログラムに必要なインストール情報が見つからない場合は、不足している情報の入力を求めるプロンプトが表示されることがあります。



## 第 46 章

# WAN ブート (リファレンス)

---

この章では、WAN インストールの実行に使用するコマンドとファイルについて簡単に説明します。

- 605 ページの「WAN ブートインストールのコマンド」
- 608 ページの「OBP コマンド」
- 609 ページの「システム構成ファイルの設定と構文」
- 610 ページの「wanboot.conf ファイルのパラメータと構文」

---

## WAN ブートインストールのコマンド

次の表では、WAN インストールの実行に使用するコマンドについて説明します。

- 表 46-1
- 表 46-2

表 46-1 WAN ブートのインストールファイルと構成ファイルの準備

作業と説明	コマンド
インストールサーバーのローカルディスク上の、 <i>install-dir-path</i> に Solaris インストールイメージをコピーし、 <i>wan-dir-path</i> に WAN ブートミニルートをコピーする	<code>setup_install_server -w wan-dir-path install-dir-path</code>

表 46-1 WAN ブートのインストールファイルと構成ファイルの準備 (続き)

作業と説明	コマンド
<p><i>name.flar</i> という名前のフラッシュアーカイブを作成する</p> <ul style="list-style-type: none"> <li>■ <i>name</i> は、アーカイブの名前</li> <li>■ <i>optional-parameters</i> は、アーカイブのカスタマイズに使用できる省略可能なパラメータ</li> <li>■ <i>document-root</i> は、インストールサーバーのドキュメントルートディレクトリへのパス</li> <li>■ <i>filename</i> は、アーカイブの名前</li> </ul>	<pre>flar create - n name [optional-parameters] document-root/flash/filename</pre>
<p><i>rules</i> という名前のカスタム JumpStart <i>rules</i> ファイルの妥当性を検査する</p>	<pre>./check -r rules</pre>
<p><i>wanboot.conf</i> ファイルの妥当性を検査する</p> <ul style="list-style-type: none"> <li>■ <i>net-ip</i> は、クライアントのサブネットの IP アドレス</li> <li>■ <i>client-ID</i> は、ユーザーが定義した ID か、DHCP クライアント ID</li> </ul>	<pre>bootconfchk /etc/netboot/net-ip/client-ID/wanboot.conf</pre>
<p>クライアント OBP で WAN ブートインストール対応を確認する</p>	<pre>eeeprom   grep network-boot-arguments</pre>

表 46-2 WAN ブートのセキュリティファイルの準備

作業と説明	コマンド
<p>WAN ブートサーバーの HMAC SHA1 マスターキーを作成する</p>	<pre>wanbootutil keygen -m</pre>
<p>クライアントの HMAC SHA1 ハッシュキーを作成する</p> <ul style="list-style-type: none"> <li>■ <i>net-ip</i> は、クライアントのサブネットの IP アドレス</li> <li>■ <i>client-ID</i> は、ユーザーが定義した ID か、DHCP クライアント ID</li> </ul>	<pre>wanbootutil keygen -c -o net=<i>net-ip</i>,cid=<i>client-ID</i>,type=sha1</pre>

表 46-2 WAN ブートのセキュリティファイルの準備 (続き)

作業と説明	コマンド
クライアントの暗号化鍵を作成する <ul style="list-style-type: none"> <li>■ <i>net-ip</i> は、クライアントのサブネットの IP アドレス</li> <li>■ <i>client-ID</i> は、ユーザーが定義した ID か、DHCP クライアント ID</li> <li>■ <i>key-type</i> は、3des または aes</li> </ul>	<pre>wanbootutil keygen -c -o net=<i>net-ip</i>,cid=<i>client-ID</i>,type=<i>key-type</i></pre>
PKCS#12 証明書ファイルを分割し、クライアントの truststore に証明書を挿入する <ul style="list-style-type: none"> <li>■ <i>p12cert</i> は、PKCS#12 証明書ファイルの名前</li> <li>■ <i>net-ip</i> は、クライアントのサブネットの IP アドレス</li> <li>■ <i>client-ID</i> は、ユーザーが定義した ID か、DHCP クライアント ID</li> </ul>	<pre>wanbootutil p12split -i <i>p12cert</i> -t /etc/netboot/<i>net-ip</i>/<i>client-ID</i>/truststore</pre>
PKCS#12 証明書ファイルを分割し、クライアントの certstore にクライアント証明書を挿入する <ul style="list-style-type: none"> <li>■ <i>p12cert</i> は、PKCS#12 証明書ファイルの名前</li> <li>■ <i>net-ip</i> は、クライアントのサブネットの IP アドレス</li> <li>■ <i>client-ID</i> は、ユーザーが定義した ID か、DHCP クライアント ID</li> <li>■ <i>keyfile</i> は、クライアントの非公開鍵の名前</li> </ul>	<pre>wanbootutil p12split -i <i>p12cert</i> -c /etc/netboot/<i>net-ip</i>/<i>client-ID</i>/certstore -k <i>keyfile</i></pre>
分割した PKCS#12 ファイルから、クライアントの keystore にクライアントの非公開鍵を挿入する <ul style="list-style-type: none"> <li>■ <i>keyfile</i> は、クライアントの非公開鍵の名前</li> <li>■ <i>net-ip</i> は、クライアントのサブネットの IP アドレス</li> <li>■ <i>client-ID</i> は、ユーザーが定義した ID か、DHCP クライアント ID</li> </ul>	<pre>wanbootutil keymgmt -i -k <i>keyfile</i> -s /etc/netboot/<i>net-ip</i>/<i>client-ID</i>/keystore -o type=rsa</pre>
HMAC SHA1 ハッシュキーの値を表示する <ul style="list-style-type: none"> <li>■ <i>net-ip</i> は、クライアントのサブネットの IP アドレス</li> <li>■ <i>client-ID</i> は、ユーザーが定義した ID か、DHCP クライアント ID</li> </ul>	<pre>wanbootutil keygen -d -c -o net=<i>net-ip</i>,cid=<i>client-ID</i>,type=sha1</pre>

表 46-2 WAN ブートのセキュリティファイルの準備 (続き)

作業と説明	コマンド
暗号化鍵の値を表示する <ul style="list-style-type: none"> <li>■ <i>net-ip</i> は、クライアントのサブネットの IP アドレス</li> <li>■ <i>client-ID</i> は、ユーザーが定義した ID か、DHCP クライアント ID</li> <li>■ <i>key-type</i> は、3des または aes</li> </ul>	<code>wanbootutil keygen -d -c -o net=<i>net-ip</i>, cid=<i>client-ID</i>, type=<i>key-type</i></code>
動作中のシステムにハッシュキーまたは暗号化鍵を挿入する。 <i>key-type</i> には、sha1、3des、または aes という値を指定できる	<code>/usr/lib/inet/wanboot/ickey -o type=<i>key-type</i></code>

## OBP コマンド

次の表は、WAN ブートインストールを実行するためにクライアントの ok プロンプトに入力する OBP コマンドの一覧です。

表 46-3 WAN ブートインストール用の OBP コマンド

作業と説明	OBP コマンド
自動 WAN ブートインストールを開始する	<code>boot net - install</code>
対話式 WAN ブートインストールを開始する	<code>boot net -o prompt - install</code>
ローカル CD から WAN ブートインストールを開始する	<code>boot cdrom -F wanboot - install</code>
WAN ブートインストールを開始する前にハッシュキーをインストールする。 <i>key-value</i> は、16 進数で表したハッシュキーの値	<code>set-security-key wanboot-hmac-sha1 <i>key-value</i></code>
WAN ブートインストールを開始する前に暗号化鍵をインストールする <ul style="list-style-type: none"> <li>■ <i>key-type</i> は、wanboot-3des または wanboot-aes</li> <li>■ <i>key-value</i> は、16 進数で表した暗号化鍵の値</li> </ul>	<code>set-security-key <i>key-type</i> <i>key-value</i></code>
OBP にキーの値が設定されていることを確認する	<code>list-security-keys</code>



表 46-3 WAN ブートインストール用の OBP コマンド (続き)

作業と説明	OBP コマンド
WAN ブートインストールを開始する前にクライアント構成変数を設定する <ul style="list-style-type: none"> <li>■ <i>client-IP</i> は、クライアントの IP アドレス</li> <li>■ <i>router-ip</i> は、ネットワークルーターの IP アドレス</li> <li>■ <i>mask-value</i> は、サブネットマスクの値</li> <li>■ <i>client-name</i> は、クライアントのホスト名</li> <li>■ <i>proxy-ip</i> は、ネットワークのプロキシサーバーの IP アドレス</li> <li>■ <i>wanbootCGI-path</i> は、Web サーバー上の wanbootCGI プログラムへのパス</li> </ul>	<code>setenv network-boot-arguments host-ip=client-IP, router-ip=router-ip, subnet-mask=mask-value, hostname=client-name, http-proxy=proxy-ip, file=wanbootCGI-path</code>
ネットワークデバイス別名を調べる	<code>devalias</code>
ネットワークデバイス別名を設定する。 <i>device-path</i> は、主ネットワークデバイスへのパス	<ul style="list-style-type: none"> <li>■ 現在のインストールだけに有効な別名を設定する場合は、<code>devalias net device-path</code> と入力</li> <li>■ 別名を固定的に設定する場合は、<code>nvvalias net device-path</code> と入力</li> </ul>

## システム構成ファイルの設定と構文

システム構成ファイルでは、次のファイルの場所を WAN ブートインストールプログラムに指示できます。

- `sysidcfg`
- `rules.ok`
- カスタム JumpStart プロファイル

システム構成ファイルは、プレーンテキストファイルで、次の書式に従う必要があります。

設定=値

`system.conf` ファイルには、次の設定を含める必要があります。

`SsysidCF=sysidcfg-file-URL`

この設定は、`sysidcfg` ファイルが置かれているインストールサーバー上のディレクトリを指示します。HTTPS を使って WAN インストールを行う場合は、有効な HTTPS URL を設定してください。

`SjumpsCF=jumpstart-files-URL`

この設定は、`rules.ok` ファイルとプロファイルファイルが置かれているカスタム JumpStart ディレクトリを指示します。HTTPS を使って WAN インストールを行う

場合は、有効な HTTPS URL を設定してください。

WAN ブートサーバーがアクセスできるディレクトリであればどこにでも、`system.conf` を保存できます。

---

## wanboot.conf ファイルのパラメータと構文

`wanboot.conf` ファイルは、プレーンテキストの構成ファイルで、WAN インストールを実行するために WAN ブートインストールプログラムで使用されます。次のプログラムとファイルは、`wanboot.conf` ファイルに記述されている情報を使ってクライアントマシンのインストールを行います。

- `wanboot-cgi` プログラム
- WAN ブートファイルシステム
- WAN ブートミニルート

WAN ブートサーバー上の `/etc/netboot` ディレクトリにある該当のクライアントのサブディレクトリに、この `wanboot.conf` ファイルを保存してください。`/etc/netboot` ディレクトリを使って WAN ブートインストールの適用範囲を定義する方法については、540 ページの「WAN ブートサーバーに `/etc/netboot` ディレクトリを作成する」を参照してください。

`wanboot.conf` ファイルに情報を指定するには、パラメータとその値を次の書式で列挙します。

パラメータ=値

パラメータエントリは、それぞれ 1 行に収める必要があります。先頭に `#` という文字を付加することで、ファイルにコメントを追加できます。

`wanboot.conf` ファイルの詳細については、`wanboot.conf` (4) のマニュアルページを参照してください。

`wanboot.conf` ファイルには、次のパラメータを設定する必要があります。

`boot_file=wanboot-path`

このパラメータは、`wanboot` プログラムへのパスを指定します。値は、WAN ブートサーバーのドキュメントルートディレクトリからの相対パスです。

`boot_file=/wanboot/wanboot.s9_sparc`

`root_server=wanbootCGI-URL/wanboot-cgi`

このパラメータは、WAN ブートサーバー上の `wanboot-cgi` プログラムの URL を指定します。

- クライアント認証やサーバー認証を伴わない WAN ブートインストールを実行する場合は、HTTP URL を使用します。

```
root_server=http://www.example.com/cgi-bin/wanboot-cgi
```

- サーバー認証を伴う、あるいはサーバー認証とクライアント認証の両方を伴う WAN ブートインストールを実行する場合は、HTTPS URL を使用します。

```
root_server=https://www.example.com/cgi-bin/wanboot-cgi
```

```
root_file=miniroot-path
```

このパラメータは、WAN ブートサーバー上の WAN ブートミニルートへのパスを指定します。値は、WAN ブートサーバーのドキュメントルートディレクトリからの相対パスです。

```
root_file=/miniroot/miniroot.s9_sparc
```

```
signature_type=sha1 | empty
```

このパラメータは、転送されるデータとファイルの完全性をチェックするために使用するハッシュキーのタイプを指定します。

- WAN ブートインストールでハッシュキーを使って wanboot プログラムを保護する場合は、値として sha1 を設定します。

```
signature_type=sha1
```

- ハッシュキーを使用しない、セキュリティ保護されない WAN インストールの場合、値は空白のままにします。

```
signature_type=
```

```
encryption_type=3des | aes | empty
```

このパラメータは、wanboot プログラムと WAN ブートファイルシステムの暗号化に使用する暗号化タイプを指定します。

- HTTPS を使用する WAN ブートインストールの場合は、使用する鍵の形式に応じて 3des または aes を設定します。また、signature\_type キーワードの値として sha1 を設定する必要があります。

```
encryption_type=3des
```

または、

```
encryption_type=aes
```

- 暗号化鍵を使用しない、セキュリティ保護されない WAN インストールの場合、値は空白のままにします。

```
encryption_type=
```

```
server_authentication=yes | no
```

このパラメータは、WAN ブートインストール時にサーバー認証を行うかどうかを指定します。

- サーバー認証を伴う、あるいはサーバー認証とクライアント認証の両方を伴う WAN ブートインストールの場合は、yes を設定します。また、signature\_type には sha1、encryption\_type には 3des または aes、お

および `root_server` には HTTPS の URL を設定する必要があります。

```
server_authentication=yes
```

- サーバー認証やクライアント認証を使用しない、セキュリティ保護されない WAN ブートインストールの場合は、`no` を設定します。値を空白のままにしてもかまいません。

```
server_authentication=no
```

```
client_authentication=yes | no
```

このパラメータは、WAN ブートインストール時にクライアント認証を行うかどうかを指定します。

- サーバー認証とクライアント認証を伴う WAN ブートインストールの場合は、`yes` を設定します。また、`signature_type` には `sha1`、`encryption_type` には `3des` または `aes`、および `root_server` には HTTPS の URL を設定する必要があります。

```
client_authentication=yes
```

- クライアント認証を使用しない WAN ブートインストールの場合は、`no` を設定します。値を空白のままにしてもかまいません。

```
client_authentication=no
```

```
resolve_hosts=hostname | empty
```

このパラメータは、インストール時に `wanboot-cgi` プログラムに対して解決する必要のある、追加のホストを指定します。

`wanboot.conf` ファイルやクライアントの証明書で事前に指定されていないシステムのホスト名を値として設定します。

- 必要なすべてのホストが `wanboot.conf` ファイルまたはクライアントの証明書で事前に指定されている場合、値は空白のままにします。

```
resolve_hosts=
```

- `wanboot.conf` ファイルまたはクライアントの証明書で事前に指定されていないホストがある場合は、それらのホスト名を設定します。

```
resolve_hosts=seahag,matters
```

```
boot_logger=bootlog-cgi-path | empty
```

このパラメータは、ログサーバー上の `bootlog-cgi` スクリプトの URL を指定します。

- 専用のログサーバー上でブートログメッセージやインストールログメッセージを記録する場合は、ログサーバー上の `bootlog-cgi` スクリプトの URL を値として設定します。

```
boot_logger=http://www.example.com/cgi-bin/bootlog-cgi
```

- ブートメッセージやインストールメッセージをクライアントのコンソールに表示する場合、値は空白のままにします。

```
boot_logger=
```

`system_conf=system.conf | custom-system-conf`

このパラメータは、`sysidcfg` ファイルとカスタム JumpStart ファイルの場所が記述されている、システム構成ファイルへのパスを指定します。

Web サーバー上の `sysidcfg` ファイルとカスタム JumpStart ファイルへのパスを値として設定します。

`system_conf=sys.conf`



## 付録 A

# Solaris ソフトウェアに関するリファレンス情報 (トピック)

以下では、リファレンス情報について説明します。

付録 B	Solaris 9 のメディアキットに含まれる主な DVD と CD について説明します。
付録 C	さまざまなハードウェアプラットフォームのプラットフォーム名とグループのリストを示します。
付録 D	locale キーワードをプロファイルに設定したり、ロケールを事前設定したりするときに必要な値のリストを示します。
付録 E	Solaris ソフトウェアのインストール時に発生する特定のエラーメッセージの一覧と、一般的な問題について説明します。
付録 F	DVD-ROM ドライブまたは CD-ROM ドライブが直接接続されていないシステムまたはドメインに、Solaris Web Start プログラムを使って Solaris オペレーティング環境をインストールまたはアップグレードする方法について説明します。
付録 G	カスタム JumpStart インストールを非対話的に行い、動作中の現在のシステムを変更せずに Solaris Live Upgrade を使ってアップグレードを行うための要件について説明します。
付録 H	Solaris オペレーティング環境をアップグレードする前にパッチを確認する手順について説明します。
付録 I	ブートイメージを Solaris Device Configuration Assistant (デバイス構成用補助) からフロッピーディスクにコピーする方法について説明します。
付録 J	Solaris 9 以降この本に加えられた変更点について説明します。





## 付録 B

# Solaris 9 メディアの構成 (リファレンス)

この章では、Solaris 9 ソフトウェアのメディアキットに含まれる主な DVD メディアおよび CD メディアについて説明します。

## SPARC: Solaris 9 メディア

次の表に、Solaris 9 リリースの主な DVD メディアおよび CD メディアのリストを示します。製品には、英語、その他の言語、およびロケールソフトウェアの DVD メディアまたは CD メディアが入ったマルチリンガル版メディアキットが入っています。

表 B-1 SPARC: MULTILINGUAL DVD メディア

DVD タイトル	説明
Solaris DVD (SPARC 版)	次のものが入っています。 <ul style="list-style-type: none"><li>■ Solaris 製品およびすべての部分ロケールをインストールするためのソフトウェア、ツール、および構成情報</li><li>■ サポートまたは未サポートのソフトウェアが入っている ExtraValue ソフトウェア</li><li>■ インストーラ</li><li>■ 一部のサードパーティパブリックドメインソフトウェアのソースコード</li><li>■ ローカライズされたインタフェースソフトウェアとドキュメント</li><li>■ 英語、ヨーロッパ言語、アジア言語 (日本語を含む) 用の Solaris マニュアルセット</li></ul> ディレクトリ構造については 図 B-1 を参照してください。

表 B-2 SPARC: MULTILINGUAL CD メディア

CD タイトル	説明
Solaris INSTALLATION (MULTILINGUAL) CD (SPARC 版)	Solaris ソフトウェアおよびすべての部分ロケールをインストールするスクリプトおよびローカライズされたインストーラが入っています。ディレクトリ構造については、図 B-2 を参照してください。
Solaris SOFTWARE 1 of 2 CD (SPARC 版)	Solaris 製品のインストールに必要なソフトウェア、ツール、および構成情報が入っています。ディレクトリ構造については、図 B-3 を参照してください。
Solaris SOFTWARE 2 of 2 CD (SPARC 版)	次のものが入っています。 <ul style="list-style-type: none"> <li>■ 必要に応じてインストールするソフトウェア用のパッケージ。ソフトウェアごとにインストールするかどうかの指示が表示されます。</li> <li>■ サポートまたは未サポートのソフトウェアが入っている ExtraValue ソフトウェア</li> <li>■ インストーラ</li> <li>■ 一部のサードパーティパブリックドメインソフトウェアのソースコード</li> </ul> <p>ディレクトリ構造については、図 B-4 を参照してください。</p> <p>必要があれば、インストールプログラムによってこの CD を求めるプロンプトが表示されます。</p>
Solaris LANGUAGES CD (SPARC 版)	ローカライズされたインタフェースソフトウェアとマニュアルが入っています。ディレクトリ構造については、図 B-5 を参照してください。
	特定の地域の言語をサポートするために必要な場合、インストールプログラムによってこの CD を求めるプロンプトが表示されます。
Solaris DOCUMENTATION 1 of 2 CD	英語およびヨーロッパ言語の Solaris マニュアルセットが入っています。
Solaris DOCUMENTATION 2 of 2 CD	アジア言語 (日本語を含む) の Solaris マニュアルセットが入っています。

## SPARC: Solaris 9 メディアのディレクトリ構造

この節では、各 DVD と CD のディレクトリ構造について説明します。

# SPARC: Solaris DVD (SPARC 版) のディレクトリ構造

Solaris DVD (SPARC 版) のディレクトリ構造は以下のとおりです。

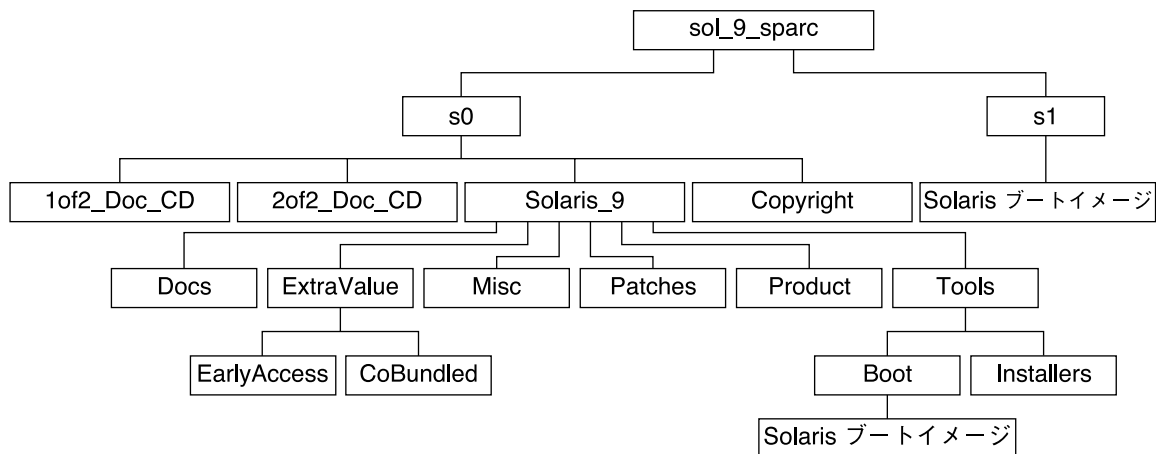


図 B-1 Solaris DVD (SPARC 版)

- スライス 0 (s0) には、Copyright と Solaris\_9 ディレクトリが入っています。Solaris\_9 ディレクトリには、Solaris 9 ソフトウェア製品のインストールに必要なツール、ソフトウェア、および構成情報が入っています。Solaris\_9 ディレクトリには、次のディレクトリが入っています。
  - ExtraValue - 次の 2 つのサブディレクトリが入っています。
    - CoBundled - Solaris オペレーティング環境には含まれないが、サポート対象の Solaris 製品 (Solaris Web Start Wizards SDK 3.0.1 および SunScreen 3.2)
    - EarlyAccess - サポート対象外の評価用ソフトウェア
  - Misc - JumpStart のソフトウェアとファイル (rules ファイル、check スクリプト、プロファイル、begin スクリプト、finish スクリプトなど) を含む jumpstart\_sample ディレクトリ
  - Patches - このリリースの時点で入手可能なすべての Solaris 9 パッチ
  - Product - Solaris 9 パッケージと制御ファイル
  - Tools - Solaris 9 インストールツール。次のものが入っています。
    - Boot サブディレクトリ - Solaris 9 ミニルートが入っています。
    - スクリプト add\_install\_client、dial、rm\_install\_client、および setup\_install\_server
    - Installers サブディレクトリ - Solaris Live Upgrade および Solaris SOFTWARE 2 of 2 CD (SPARC 版) のインストールに使用するインストーラ およびサポートされる Java™ クラスファイルが入っています。Solaris Live

Upgrade は Solaris ソフトウェアをインストールするときに自動的にインストールされます。しかし、Solaris 9 より前のリリースから Solaris Live Upgrade を使用して Solaris 9 リリースにアップグレードするには、まず、これらのパッケージを現在の Solaris リリースにインストールする必要があります。Solaris Live Upgrade をインストールする方法については、416 ページの「Solaris Web Start インストーラを使用して Solaris Live Upgrade をインストールする方法」を参照してください。

- 1of2\_Doc\_CD – 英語およびヨーロッパ言語のマニュアル
- 2of2\_Doc\_CD – アジア言語のマニュアル
- スライス 1 (s1) には、Solaris 9 ミニルートが入っています。

## SPARC: Solaris INSTALLATION CD のディレクトリ構造

次に、Solaris INSTALLATION (MULTILINGUAL) CD (SPARC 版) または Solaris INSTALLATION CD (SPARC 版) のディレクトリ構造を示します。

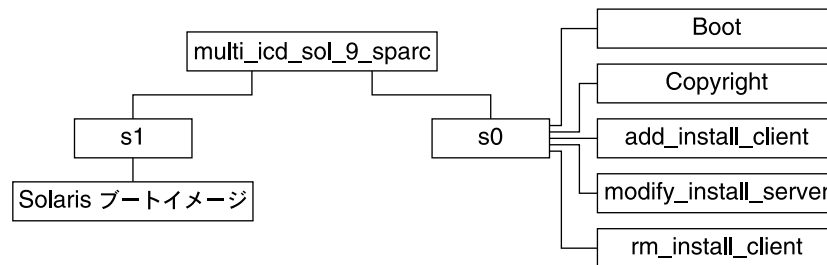


図 B-2 SPARC: Solaris INSTALLATION CD (SPARC 版)

multi\_icd\_sol\_sparc ディレクトリには、Solaris ソフトウェアのインストール用スクリプト、および Solaris 9 ミニルート用の Boot サブディレクトリが入っています。ディレクトリおよびスクリプトには、次のものが入っています。

- スライス 0 (s0) には、Solaris ソフトウェアをインストールするスクリプトと、Solaris ミニルートが入っている Boot サブディレクトリが入っています。次のようなスクリプトが入っています。
  - add\_install\_client
  - modify\_install\_server
  - rm\_install\_client
- スライス 1 (s1) には、Solaris 9 ミニルートが入っています。

## SPARC: Solaris SOFTWARE (SPARC 版) CD のディレクトリ構造

次に、Solaris SOFTWARE (SPARC 版) CD のディレクトリ構造を示します。

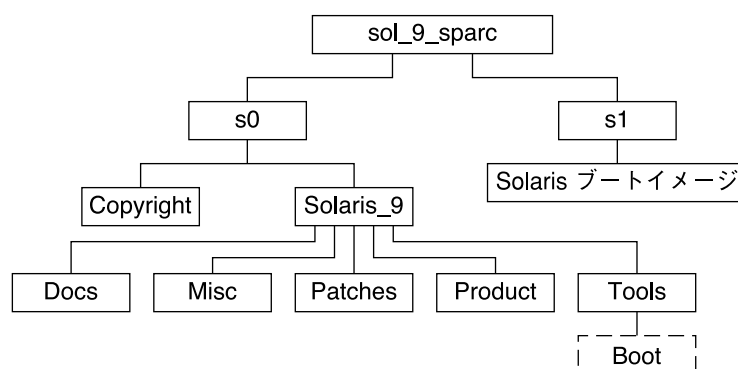


図 B-3 SPARC: Solaris SOFTWARE 1 of 2 CD (SPARC 版)

- スライス 0 (s0) には、Copyright と Solaris\_9 ディレクトリが入っています。Solaris\_9 ディレクトリには、Solaris 9 ソフトウェア製品 (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のソフトウェアグループなど) をインストールするために必要な最低限のツール、ソフトウェア、および構成情報が入っています。Solaris\_9 ディレクトリには、次のディレクトリが入っています。
  - Docs – 空のディレクトリ
  - Misc – JumpStart のソフトウェアとファイル (rules ファイル、check スクリプト、プロファイル、begin スクリプト、finish スクリプトなど) を含む jumpstart\_sample ディレクトリ
  - Patches – Solaris SOFTWARE 1 of 2 CD (SPARC 版) が作成された時点で入手可能なすべての Solaris 9 パッチ
  - Product – Solaris 9 パッケージと制御ファイル
  - Tools – Solaris 9 インストールツール (スライス 1 の boot ミニルートへのシンボリックリンク、add\_install\_client、dial、rm\_install\_client、setup\_install\_server など)
- スライス 1 (s1) には、Solaris 9 ミニルートが入っています。

次の図に、Solaris SOFTWARE 2 of 2 CD (SPARC 版) のディレクトリ構造を示します。

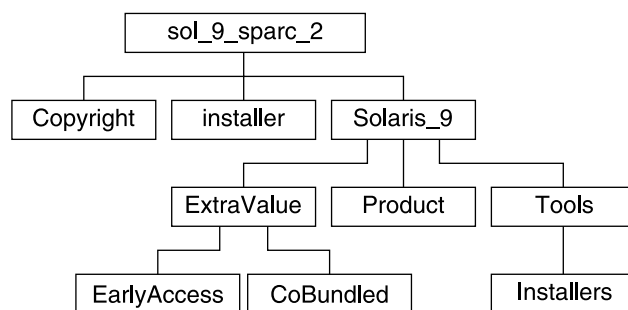


図 B-4 SPARC: Solaris SOFTWARE 2 of 2 CD (SPARC 版)

sol\_9\_sparc\_2 ディレクトリには、Copyright、Solaris Web Start installer、および Solaris\_9 ディレクトリが入っています。Solaris\_9 ディレクトリには、次のサブディレクトリが入っています。

- ExtraValue – 次の2つのサブディレクトリが入っています。
  - CoBundled – Solaris オペレーティング環境には含まれないが、サポート対象の Solaris 製品
  - EarlyAccess – サポート対象外の評価用ソフトウェア
- Product – 開発者システムサポート、全体ディストリビューション、および全体ディストリビューションと OEM サポートのソフトウェアグループ用パッケージの一部
- Tools – Solaris 9 インストールツール。次のものが入っています。
  - add\_to\_install\_server スクリプト
  - Installers サブディレクトリ - Solaris Live Upgrade および Solaris SOFTWARE 2 of 2 CD (SPARC 版) のインストールに使用するインストーラおよびサポートされる Java クラスファイルが入っています。Solaris Live Upgrade は Solaris ソフトウェアをインストールするときに自動的にインストールされます。しかし、Solaris 9 より前のリリースから Solaris Live Upgrade を使用して Solaris 9 リリースにアップグレードするには、まず、これらのパッケージを現在の Solaris リリースにインストールする必要があります。Solaris Live Upgrade をインストールする方法については、416 ページの「Solaris Web Start インストーラを使用して Solaris Live Upgrade をインストールする方法」を参照してください。

## SPARC: Solaris LANGUAGES CD (SPARC 版) のディレクトリ構造

次の図に、Solaris LANGUAGES CD (SPARC 版) のディレクトリ構造を示します。

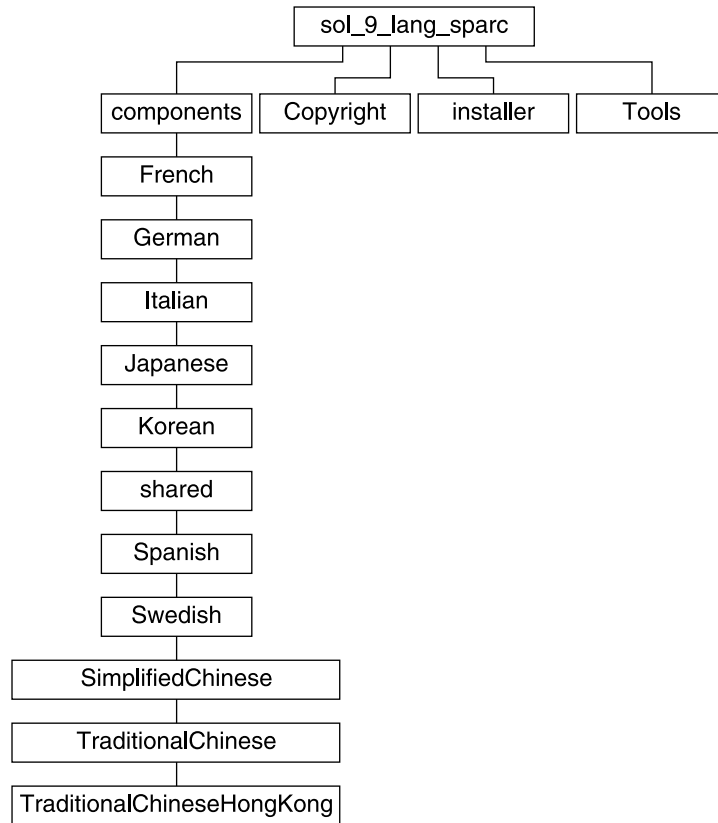


図 B-5 Solaris LANGUAGES CD (SPARC 版)

sol\_lang\_sparc ディレクトリには、Solaris の言語ソフトウェアとロケールソフトウェアをインストールする Solaris Web Start インストーラが入っています。sol\_lang\_sparc ディレクトリには次のディレクトリが入っています。

- components - ロケールパッケージ (簡体字中国語、フランス語、ドイツ語、イタリア語、日本語、韓国語、スペイン語、スウェーデン語、繁体字中国語、香港版繁体字中国語)、およびすべてのロケールで共有されるパッケージが入っています。
- Copyright - 著作権ページ
- installer - Solaris Web Start インストーラ
- Tools - 結合されたインストールイメージを作成するための add\_to\_install\_server スクリプト

ロケールソフトウェアについての詳細は、『国際化対応言語環境の利用ガイド』を参照してください。

## x86: Solaris 9 メディア

次に、Solaris 9 (x86 版) の主な DVD および CD メディアの内容を示します。製品には、英語、その他の言語、およびロケールソフトウェアの DVD メディアまたは CD メディアが入ったマルチリンガル版メディアキットが入っています。

表 B-3 x86: MULTILINGUAL DVD メディア

DVD タイトル	説明
Solaris DVD (x86 版)	<p>次のものが入っています。</p> <ul style="list-style-type: none"><li>■ Solaris 製品および部分ロケールをインストールするためのソフトウェア、ツール、および構成情報。これには、さまざまな構成作業とブート作業に使用する Solaris Device Configuration Assistant (デバイス構成用補助) が含まれます。</li><li>■ サポートまたは未サポートのソフトウェアが入っている ExtraValue ソフトウェア</li><li>■ インストーラ</li><li>■ 一部のサードパーティパブリックドメインソフトウェアのソースコード</li><li>■ ローカライズされたインタフェースソフトウェアとドキュメント</li><li>■ 英語、ヨーロッパ言語、アジア言語 (日本語を含む) 用の Solaris マニュアルセット</li></ul> <p>ディレクトリ構造については、図 B-6 を参照してください。</p>

表 B-4 x86: MULTILINGUAL CD メディア

CD タイトル	説明
Solaris INSTALLATION (MULTILINGUAL) CD (x86 版)	<p>Solaris ソフトウェアおよびすべての部分ロケールをインストールするスクリプトおよびローカライズされたインストーラが入っています。これには、さまざまな構成作業とブート作業に使用する Solaris Device Configuration Assistant (デバイス構成用補助) が含まれます。ディレクトリ構造については、図 B-7 を参照してください。</p>
Solaris SOFTWARE 1 of 2 CD (x86 版)	<p>Solaris 製品のインストールに必要なソフトウェア、ツール、および構成情報が入っています。これには、さまざまな構成作業とブート作業に使用する Solaris Device Configuration Assistant (デバイス構成用補助) が含まれます。ディレクトリ構造については、図 B-8 を参照してください。</p>



表 B-4 x86: MULTILINGUAL CD メディア (続き)

CD タイトル	説明
Solaris SOFTWARE 2 of 2 CD (x86 版)	<p>次のものが入っています。</p> <ul style="list-style-type: none"> <li>■ 必要に応じてインストールするソフトウェア用のパッケージ。ソフトウェアごとにインストールするかどうかの指示が表示されます。</li> <li>■ サポートまたは未サポートのソフトウェアが入っている ExtraValue ソフトウェア</li> <li>■ インストーラ</li> <li>■ 一部のサードパーティパブリックドメインソフトウェアのソースコード</li> </ul> <p>ディレクトリ構造については、図 B-9 を参照してください。</p> <p>必要があれば、インストールプログラムによってこの CD を求めるプロンプトが表示されます。</p>
Solaris LANGUAGES CD (x86 版)	<p>ローカライズされたインタフェースソフトウェアとマニュアルが入っています。ディレクトリ構造については、図 B-10 を参照してください。</p> <p>特定の地域の言語をサポートするために必要な場合、インストールプログラムによってこの CD を求めるプロンプトが表示されます。</p>
Solaris DOCUMENTATION 1 of 2 CD	<p>英語およびヨーロッパ言語の Solaris マニュアルセットが入っています。</p>
Solaris DOCUMENTATION 2 of 2 CD	<p>アジア言語 (日本語を含む) の Solaris マニュアルセットが入っています。</p>

## x86: Solaris 9 メディアのディレクトリ構造

この節では、各 DVD と CD のディレクトリ構造について説明します。

### Solaris DVD (x86 版) ディレクトリの構造

次に、Solaris DVD (x86 版) のディレクトリ構造を示します。

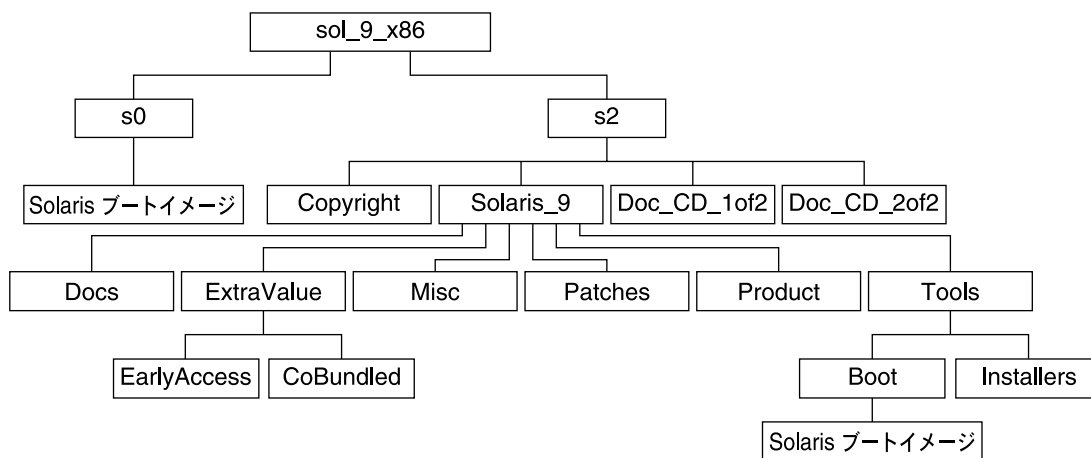


図 B-6 Solaris DVD (x86 版)

- スライス 0 (s0) には、Solaris 9 ミニルートが入っています。
- スライス 2 (s2) には、Copyright と Solaris\_9 ディレクトリが入っています。Solaris\_9 ディレクトリには、Solaris 9 ソフトウェア製品のインストールに必要なツール、ソフトウェア、および構成情報が入っています。スライス 2 には、次のディレクトリが入っています。
  - ExtraValue - 次の 2 つのサブディレクトリが入っています。
    - CoBundled - Solaris オペレーティング環境には含まれないが、サポート対象の Solaris 製品 (Solaris Web Start Wizards SDK 3.0.1 および SunScreen 3.2)
    - EarlyAccess - サポート対象外の評価用ソフトウェア
  - Misc - JumpStart のソフトウェアとファイル (rules ファイル、check スクリプト、プロファイル、begin スクリプト、finish スクリプトなど) を含む jumpstart\_sample ディレクトリ
  - Patches - このリリースの時点で入手可能なすべての Solaris 9 パッチ
  - Product - Solaris 9 パッケージと制御ファイル
  - Tools - Solaris 9 インストールツール。次のものが入っています。
    - Boot サブディレクトリ - Solaris 9 ミニルートが入っています。
    - スクリプト add\_install\_client、dial、rm\_install\_client、および setup\_install\_server
    - Installers サブディレクトリ - Solaris Live Upgrade および Solaris SOFTWARE 2 of 2 CD (x86 版) のインストールに使用するインストーラおよびサポートされる Java クラスファイルが入っています。Solaris Live Upgrade は Solaris ソフトウェアをインストールするときに自動的にインストールされます。しかし、Solaris 9 より前のリリースから Solaris Live Upgrade を使用して Solaris 9 リリースにアップグレードするには、まず、これらのパッケージを現在の Solaris リリースにインストールする必要があります。

ります。Solaris Live Upgrade をインストールする方法については、416 ページの「Solaris Web Start インストーラを使用して Solaris Live Upgrade をインストールする方法」を参照してください。

- d1\_image サブディレクトリ - フロッピーディスクにコピー可能な Solaris Device Configuration Assistant (デバイス構成用補助) ソフトウェアが入っています。
- 1of2\_Doc\_CD - 英語およびヨーロッパ言語のマニュアル
- 2of2\_Doc\_CD - アジア言語のマニュアル

## x86: Solaris INSTALLATION CD のディレクトリ構造

次に、Solaris INSTALLATION (MULTILINGUAL) CD (x86 版) または Solaris INSTALLATION CD (x86 版) のディレクトリ構造を示します。

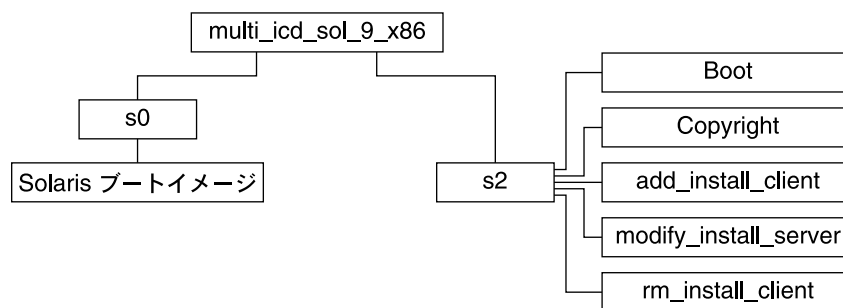


図 B-7 x86: Solaris INSTALLATION (MULTILINGUAL) CD (x86 版)

multi\_icd\_sol\_x86 ディレクトリには、Solaris ソフトウェアのインストール用スクリプト、および Solaris 9 ミニルート用の Boot サブディレクトリが入っています。ディレクトリおよびスクリプトには、次のものが入っています。

- スライス 0 (s0) には、Solaris 9 ミニルートが入っています。
- スライス 2 (s2) には、Solaris ソフトウェアをインストールするスクリプトと、Solaris ミニルートが入っている Boot サブディレクトリが入っています。次のようなスクリプトが入っています。
  - add\_install\_client
  - modify\_install\_server
  - rm\_install\_client

## x86: Solaris SOFTWARE (x86 版) CD のディレクトリ構造

次に、Solaris SOFTWARE (x86 版) CD のディレクトリ構造を示します。

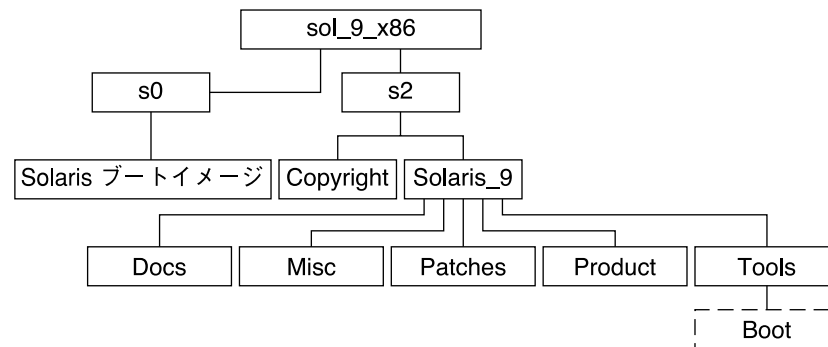


図 B-8 x86: Solaris SOFTWARE 1 of 2 CD (x86 版)

- スライス 0 (s0) には、Solaris 9 ミニルートが入っています。
- スライス 2 (s2) には、Copyright と Solaris\_9 ディレクトリが入っています。Solaris\_9 ディレクトリには、Solaris 9 ソフトウェア製品 (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のソフトウェアグループなど) をインストールするために必要な最低限のツール、ソフトウェア、および構成が入っています。Solaris\_9 ディレクトリには、次のディレクトリが入っています。
  - Docs – 空のディレクトリ
  - Misc – JumpStart のソフトウェアとファイル (rules ファイル、check スクリプト、プロファイル、begin スクリプト、finish スクリプトなど) を含む jumpstart\_sample ディレクトリ
  - Patches – Solaris SOFTWARE 1 of 2 CD (x86 版) の作成時点で入手可能なすべての Solaris 9 パッチ
  - Product – Solaris 9 パッケージと制御ファイル
  - Tools – Solaris 9 インストールツール (スライス 0 の boot ミニルートへのシンボリックリンク、add\_install\_client、dial、rm\_install\_client、setup\_install\_server など)

次に、Solaris SOFTWARE 2 of 2 CD (x86 版) のディレクトリ構造を示します。

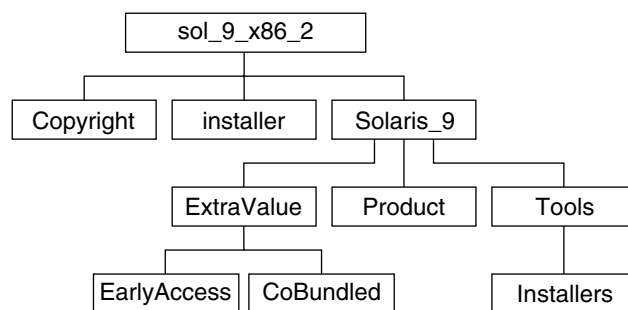


図 B-9 Solaris SOFTWARE 2 of 2 CD (x86 版)

sol\_x86\_2 ディレクトリには、Copyright、Solaris Web Start installer、および Solaris\_9 ディレクトリが入っています。Solaris\_9 ディレクトリには次のものが入っています。

- ExtraValue – 次の2つのサブディレクトリが入っています。
  - CoBundled – Solaris オペレーティング環境には含まれないが、サポート対象の Solaris 製品
  - EarlyAccess – サポート対象外の評価用ソフトウェア
- Product – 開発者システムサポート、全体ディストリビューションおよび全体ディストリビューションと OEM サポートのソフトウェアグループ用パッケージの一部
- Tools – このディレクトリには次のものが入っています。
  - add\_to\_install\_server スクリプト
  - Installers サブディレクトリ - Solaris Live Upgrade および Solaris SOFTWARE 2 of 2 CD (x86 版) のインストールに使用するインストーラおよびサポートされる Java クラスファイルが入っています。Solaris Live Upgrade は Solaris ソフトウェアをインストールするときに自動的にインストールされません。しかし、Solaris 9 より前のリリースから Solaris Live Upgrade を使用して Solaris 9 リリースにアップグレードするには、まず、これらのパッケージを現在の Solaris リリースにインストールする必要があります。Solaris Live Upgrade をインストールする方法については、416 ページの「Solaris Web Start インストーラを使用して Solaris Live Upgrade をインストールする方法」を参照してください。
  - d1\_image サブディレクトリ - フロッピーディスクにコピー可能な Solaris Device Configuration Assistant (デバイス構成用補助) ソフトウェアが入っています。

## x86: Solaris LANGUAGES CD (x86 版) のディレクトリ構造

次の図に、Solaris LANGUAGES CD (x86 版) のディレクトリ構造を示します。

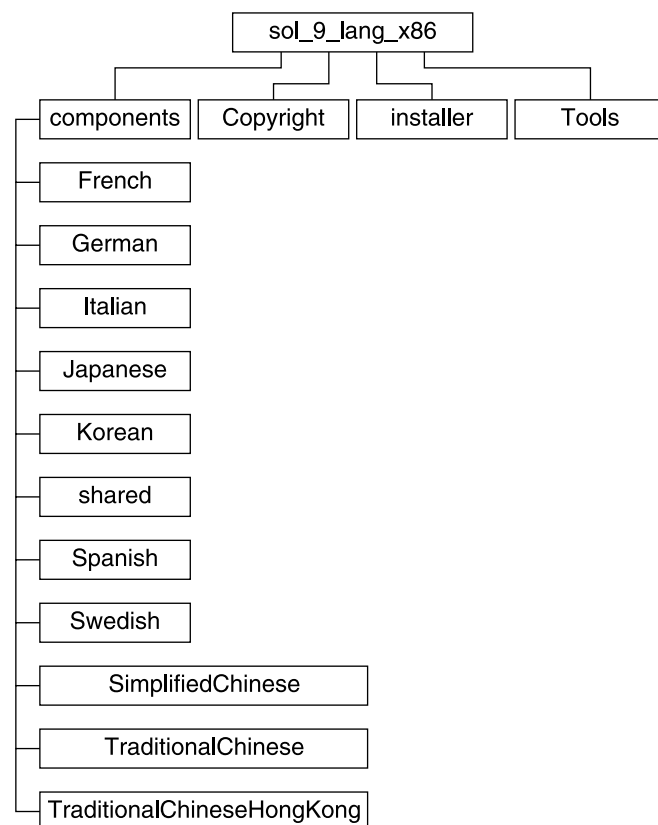


図 B-10 Solaris LANGUAGES CD (x86 版)

sol\_lang\_x86 ディレクトリには、Solaris の言語ソフトウェアとロケールソフトウェアをインストールする Solaris Web Start プログラムが入っています。デフォルトディレクトリに 9 つの言語をインストールすることも、あるいは特定の言語をインストールすることもできます。sol\_lang\_x86 ディレクトリには、次のディレクトリが入っています。

- `components` – ロケールパッケージ (簡体字中国語、フランス語、ドイツ語、イタリア語、日本語、韓国語、スペイン語、スウェーデン語、繁体字中国語、香港版繁体字中国語)、およびすべてのロケールで共有されるパッケージが入っています。
- `Copyright` – 著作権ページ
- `installer` – Web Start インストーラ

- Tools – インストールサーバーを作成するための `add_to_install_server` スクリプト

ローカルソフトウェアについての詳細は、『国際化対応言語環境の利用ガイド』を参照してください。





## 付録 C

# プラットフォーム名とプラットフォームグループ (リファレンス)

ネットワークシステムにクライアントを追加するときにはシステムアーキテクチャ (プラットフォームグループ) を、カスタム JumpStart インストール rules ファイルを作成するときにはプラットフォーム名を知っている必要があります。

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 システムの完全なリストについては、『Solaris 9 Sun ハードウェアマニュアル』を参照してください。

表 C-1 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
Ultra 5	SUNW, Ultra-5_10	sun4u
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`、システムのプラットフォームグループは `uname -m` でそれぞれ調べることもできます。



## 付録 D

# ロケールの値 (リファレンス)

以下に、プロファイル内で locale キーワードを設定する場合や、ロケールを事前構成する場合に必要な値を示します。

「ロケール」によって、特定の言語と地域についてのオンライン情報を表示する方法が決定されます。1つの言語でも、日付と時間の表記、綴り、通貨など、地域によって異なるロケールが含まれます。ロケールの詳細は、『国際化対応言語環境の利用ガイド』を参照してください。

表 D-1 アジア

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
hi_IN.UTF-8	英語	インド	UTF-8 <sup>1</sup>	ヒンディ語 (UTF-8) Unicode 3.1
ja	日本語	日本	eucJP <sup>2</sup>	日本語 (EUC) JIS X 0201-1976 JIS X 0208-1990 JIS X 0212-1990
ja_JP.eucJP	日本語	日本	eucJP	日本語 (EUC) JIS X 0201-1976 JIS X 0208-1990 JIS X 0212-1990
ja_JP.PCK	日本語	日本	PCK <sup>3</sup>	日本語 (PC kanji)

<sup>1</sup> UTF-8 は、ISO/IEC 10646-1:2000 および Unicode 3.1 で定義されている UTF-8 です。

<sup>2</sup> eucJP は、日本語 EUC コードセットを示します。ja\_JP.eucJP ロケールの仕様は UL\_OSF 日本語環境実装規約 Version 1.1 に準拠しており、ja ロケールは過去の Solaris リリースにおける従来の仕様に準拠しています。

<sup>3</sup> PCK はシフト JIS (SJIS) と呼ばれます。

表 D-1 アジア (続き)

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
				JIS X 0201-1976
				JIS X 0208-1990
ja_JP.UTF-8	日本語	日本	UTF-8	日本語 (UTF-8) Unicode 3.1
ko_KR.EUC	韓国語	韓国	1001	韓国語 (EUC) KS X 1001
ko_KR.UTF-8	韓国語	韓国	UTF-8	韓国語 (UTF-8) Unicode 3.1
th_TH.UTF-8	英語	タイ	UTF-8	タイ語 (UTF-8) Unicode 3.1
th_TH.TIS620	英語	タイ	TIS620.2533	タイ語 TIS620.2533
zh_CN.EUC	簡体字中国語	PRC	gb2312 <sup>4</sup>	簡体字中国語 (EUC) GB2312-1980
zh_CN.GBK	簡体字中国語	PRC	GBK <sup>5</sup>	簡体字中国語 (GBK)
zh_CN.GB18030	簡体字中国語	PRC	GB18030-2000	簡体字中国語 (GB18030-2000) GB18030-2000
zh_CN.UTF-8	簡体字中国語	PRC	UTF-8	簡体字中国語 (UTF-8) Unicode 3.1
zh_HK.BIG5HK	繁体字中国語	香港	Big5+HKSCS	繁体字中国語 (BIG5+HKSCS)
zh_HK.UTF-8	繁体字中国語	香港	UTF-8	繁体字中国語 (UTF-8) Unicode 3.1
zh_TW.EUC	繁体字中国語	台湾	cns11643	繁体字中国語 (EUC) CNS 11643-1992
zh_TW.BIG5	繁体字中国語	台湾	BIG5	繁体字中国語 (BIG5)
zh_TW.UTF-8	繁体字中国語	台湾	UTF-8	繁体字中国語 (UTF-8) Unicode 3.1

<sup>4</sup> gb2312 は、簡体字中国語 EUC コードセットを示します。これには GB 1988-80 および GB 2312-80 が含まれます。

<sup>5</sup> GBK は GB 拡張を示します。すべての GB 2312-80 文字、ISO/IEC 10646-1 のすべての Unified Han 文字、および日本語のひらがな文字、カタカナ文字を含みます。また、中国語、日本語、韓国語の文字セット、および ISO/IEC 10646-1 の多くの文字も含みます。

表 D-2 オーストラレーシア

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
en_AU.ISO8859-1	英語	オーストラリア	ISO8859-1	英語 (オーストラリア)
en_NZ.ISO8859-1	英語	ニュージーランド	ISO8859-1	英語 (ニュージーランド)

表 D-3 中央アメリカ

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
es_CR.ISO8859-1	スペイン語	コスタリカ	ISO8859-1	スペイン語 (コスタリカ)
es_GT.ISO8859-1	スペイン語	グアテマラ	ISO8859-1	スペイン語 (グアテマラ)
es_NI.ISO8859-1	スペイン語	ニカラグア	ISO8859-1	スペイン語 (ニカラグア)
es_PA.ISO8859-1	スペイン語	パナマ	ISO8859-1	スペイン語 (パナマ)
es_SV.ISO8859-1	スペイン語	エルサルバドル	ISO8859-1	スペイン語 (エルサルバドル)

表 D-4 中央ヨーロッパ

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
cs_CZ.ISO8859-2	英語	チェコ共和国	ISO8859-2	チェコ語 (チェコ)
de_AT.ISO8859-1	ドイツ語	オーストリア	ISO8859-1	ドイツ語 (オーストリア)
de_AT.ISO8859-15	ドイツ語	オーストリア	ISO8859-15	ドイツ語 (オーストリア、ISO8859-15 - Euro)
de_CH.ISO8859-1	ドイツ語	スイス	ISO8859-1	ドイツ語 (スイス)
de_DE.UTF-8	ドイツ語	ドイツ	UTF-8	ドイツ語 (ドイツ、Unicode 3.1)
de_DE.ISO8859-1	ドイツ語	ドイツ	ISO8859-1	ドイツ語 (ドイツ)
de_DE.ISO8859-15	ドイツ語	ドイツ	ISO8859-15	ドイツ語 (ドイツ、ISO8859-15 - Euro)
fr_CH.ISO8859-1	フランス語	スイス	ISO8859-1	フランス語 (スイス)
hu_HU.ISO8859-2	英語	ハンガリー	ISO8859-2	ハンガリー語 (ハンガリー)
pl_PL.ISO8859-2	英語	ポーランド	ISO8859-2	ポーランド語 (ポーランド)
pl_PL.UTF-8	英語	ポーランド	UTF-8	ポーランド語 (ポーランド、Unicode 3.1)
sk_SK.ISO8859-2	英語	スロバキア	ISO8859-2	スロバキア語 (スロバキア)

表 D-5 東ヨーロッパ

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
bg_BG.ISO8859-5	英語	ブルガリア	ISO8859-5	ブルガリア語 (ブルガリア)

表 D-5 東ヨーロッパ (続き)

ロケール	ユーザーインターフェース	地域	コードセット	言語サポート
et_EE.ISO8859-15	英語	エストニア	ISO8859-15	エストニア語 (エストニア)
hr_HR.ISO8859-2	英語	クロアチア	ISO8859-2	クロアチア語 (クロアチア)
lt_LT.ISO8859-13	英語	リトアニア	ISO8859-13	リトアニア語 (リトアニア)
lv_LV.ISO8859-13	英語	ラトビア	ISO8859-13	ラトビア語 (ラトビア)
mk_MK.ISO8859-5	英語	マケドニア	ISO8859-5	マケドニア語 (マケドニア)
ro_RO.ISO8859-2	英語	ルーマニア	ISO8859-2	ルーマニア語 (ルーマニア)
ru_RU.KOI8-R	英語	ロシア	KOI8-R	ロシア語 (ロシア、KOI8-R)
ru_RU.ANSI1251	英語	ロシア	ansi-1251	ロシア語 (ロシア、ANSI 1251)
ru_RU.ISO8859-5	英語	ロシア	ISO8859-5	ロシア語 (ロシア)
ru_RU.UTF-8	英語	ロシア	UTF-8	ロシア語 (ロシア、Unicode 3.1)
sh_BA.ISO8859-2@bosnia	英語	ボスニア	ISO8859-2	ボスニア語 (ボスニア)
sl_SI.ISO8859-2	英語	スロベニア	ISO8859-2	スロベニア語 (スロベニア)
sq_AL.ISO8859-2	英語	アルバニア	ISO8859-2	アルバニア語 (アルバニア)
sr_YU.ISO8859-5	英語	セルビア	ISO8859-5	セルビア語 (セルビア)
tr_TR.ISO8859-9	英語	トルコ	ISO8859-9	トルコ語 (トルコ)
tr_TR.UTF-8	英語	トルコ	UTF-8	トルコ語 (トルコ、Unicode 3.1)

表 D-6 中東

ロケール	ユーザーインターフェース	地域	コードセット	言語サポート
He	英語	イスラエル	ISO8859-8	ヘブライ語 (イスラエル)

表 D-7 北アフリカ

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
ar_EG.UTF-8	英語	エジプト	UTF-8	アラビア語 (エジプト)
Ar	英語	エジプト	ISO8859-6	アラビア語 (エジプト)

表 D-8 北アメリカ

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
en_CA.ISO8859-1	英語	カナダ	ISO8859-1	英語 (カナダ)
en_US.ISO8859-1	英語	USA	ISO8859-1	英語 (U.S.A.)
en_US.ISO8859-15	英語	USA	ISO8859-15	英語 (U.S.A., ISO8859-15 - Euro)
en_US.UTF-8	英語	USA	UTF-8	英語 (U.S.A., Unicode 3.1)
fr_CA.ISO8859-1	フランス語	カナダ	ISO8859-1	フランス語 (カナダ)
es_MX.ISO8859-1	スペイン語	メキシコ	ISO8859-1	スペイン語 (メキシコ)

表 D-9 北ヨーロッパ

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
da_DK.ISO8859-1	英語	デンマーク	ISO8859-1	デンマーク語 (デンマーク)
da_DK.ISO8859-15	英語	デンマーク	ISO8859-15	デンマーク語 (デンマーク、ISO8859-15-Euro)
fi_FI.ISO8859-1	英語	フィンランド	ISO8859-1	フィンランド語 (Unicode 3.1)
fi_FI.ISO8859-15	英語	フィンランド	ISO8859-15	フィンランド語 (フィンランド、ISO8859-15-Euro)
fi_FI.UTF-8	英語	フィンランド	UTF-8	フィンランド語 (フィンランド)
is_IS.ISO8859-1	英語	アイスランド	ISO8859-1	アイスランド語 (アイスランド)
no_NO.ISO8859-1@bokmal	英語	ノルウェー	ISO8859-1	ノルウェー語 (ノルウェー - ブークモール語)
no_NO.ISO8859-1@nyorsk	英語	ノルウェー	ISO8859-1	ノルウェー語 (ノルウェー - ニーノシク語)
sv_SE.ISO8859-1	スウェーデン語	スウェーデン	ISO8859-1	スウェーデン語 (スウェーデン)

表 D-9 北ヨーロッパ (続き)

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
sv_SE.ISO8859-15	スウェーデン語	スウェーデン	ISO8859-15	スウェーデン語 (スウェーデン、ISO8859-15-Euro)
sv_SE.UTF-8	スウェーデン語	スウェーデン	UTF-8	スウェーデン語 (スウェーデン、Unicode 3.1)

表 D-10 南アメリカ

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
es_AR.ISO8859-1	スペイン語	アルゼンチン	ISO8859-1	スペイン語 (アルゼンチン)
es_BO.ISO8859-1	スペイン語	ボリビア	ISO8859-1	スペイン語 (ボリビア)
es_CL.ISO8859-1	スペイン語	チリ	ISO8859-1	スペイン語 (チリ)
es_CO.ISO8859-1	スペイン語	コロンビア	ISO8859-1	スペイン語 (コロンビア)
es_EC.ISO8859-1	スペイン語	エクアドル	ISO8859-1	スペイン語 (エクアドル)
es_PE.ISO8859-1	スペイン語	ペルー	ISO8859-1	スペイン語 (ペルー)
es_PY.ISO8859-1	スペイン語	パラグアイ	ISO8859-1	スペイン語 (パラグアイ)
es_UY.ISO8859-1	スペイン語	ウルグアイ	ISO8859-1	スペイン語 (ウルグアイ)
es_VE.ISO8859-1	スペイン語	ベネズエラ	ISO8859-1	スペイン語 (ベネズエラ)
pt_BR.ISO8859-1	英語	ブラジル	ISO8859-1	ポルトガル語 (ブラジル)
pt_BR.UTF-8	英語	ブラジル	UTF-8	ポルトガル語 (ブラジル、Unicode 3.1)

表 D-11 南ヨーロッパ

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
ca_ES.ISO8859-1	英語	スペイン	ISO8859-1	カタロニア語 (スペイン)
ca_ES.ISO8859-15	英語	スペイン	ISO8859-15	カタロニア語 (スペイン、ISO8859-15 - Euro)
el_GR.ISO8859-7	英語	ギリシャ	ISO8859-7	ギリシャ語 (ギリシャ)
es_ES.ISO8859-1	スペイン語	スペイン	ISO8859-1	スペイン語 (スペイン)
es_ES.ISO8859-15	スペイン語	スペイン	ISO8859-15	スペイン語 (スペイン、ISO-8859-15 - Euro)



表 D-11 南ヨーロッパ (続き)

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
es_ES.UTF-8	スペイン語	スペイン	UTF-8	スペイン語 (スペイン、Unicode 3.1)
it_IT.ISO8859-1	イタリア語	イタリア	ISO8859-1	イタリア語 (イタリア)
it_IT.ISO8859-15	イタリア語	イタリア	ISO8859-15	イタリア語 (イタリア、ISO8859-15 - Euro)
it_IT.UTF-8	イタリア語	イタリア	UTF-8	イタリア語 (イタリア、Unicode 3.1)
pt_PT.ISO8859-1	英語	ポルトガル	ISO8859-1	ポルトガル語 (ポルトガル)
pt_PT.ISO8859-15	英語	ポルトガル	ISO8859-15	ポルトガル語 (ポルトガル、ISO8859-15 - Euro)

表 D-12 西ヨーロッパ

ロケール	ユーザーインタフェース	地域	コードセット	言語サポート
en_GB.ISO8859-1	英語	イギリス	ISO8859-1	英語 (イギリス)
en_IE.ISO8859-1	英語	アイルランド	ISO8859-1	英語 (アイルランド)
fr_BE.ISO8859-1	フランス語	ベルギーのワロン地域	ISO8859-1	フランス語 (ベルギーのワロン地域、Unicode 3.1)
fr_BE.UTF-8	フランス語	ベルギーのワロン地域	UTF-8	フランス語 (ベルギーのワロン地域、Unicode 3.1)
fr_FR.ISO8859-1	フランス語	フランス	ISO8859-1	フランス語 (フランス)
fr_FR.UTF-8	フランス語	フランス	UTF-8	フランス語 (フランス、Unicode 3.1)
nl_BE.ISO8859-1	英語	ベルギーのフラマン地域	ISO8859-1	オランダ語 (ベルギーのフラマン地域)
nl_NL.ISO8859-1	英語	オランダ	ISO8859-1	オランダ語 (オランダ)



# 問題発生時の解決方法 (作業)

---

この章では、Solaris 9 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、それぞれの問題の解決方法を示します。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。

- 643 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」
- 644 ページの「システムのブートに関する問題」
- 652 ページの「Solaris 9 オペレーティング環境の初期インストール時の問題」
- 654 ページの「Solaris 9 オペレーティング環境のアップグレード時の問題」

---

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 (Solaris suninstall プログラム、Solaris Web Start プログラム、またはカスタム JumpStart) の 1 つを意味します。

---

---

## ネットワークインストールの設定に関する問題

Unknown client "*host\_name*"

原因: `add_install_client` コマンドの *host\_name* 引数は、ネームサービス内のホストでなければならない。

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ない。

対処方法: ネットワークに接続されていないシステムの場合は、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: ブート用の適切な媒体が見つからない。

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 9 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。たとえば、設定時のシステムのプラットフォームグループの指定が正しいか確認します。

あるいは、Solaris DVD または Solaris SOFTWARE 1 of 2 CD、Solaris SOFTWARE 2 of 2 CD、および Solaris LANGUAGES CD のイメージをインストールサーバーにコピーしなかった場合は、インストールサーバー上の Solaris DVD または Solaris SOFTWARE 1 of 2 CD がマウントされていて、アクセスできることを確認します。

boot: cannot open /kernel/unix (SPARC システムのみ)

原因: PROM の boot- file の値を明示的に /kernel/unix に指定したときに発生するエラー。

対処方法:

- 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ない。

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 DVD または Solaris SOFTWARE 1 of 2 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 9 ソフトウェアをインストールしたとき (Solaris suninstall プログラムまたはカスタム JumpStart を使って)、ブートドライブを選択しなかった。Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) 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る。

対処方法:

- Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) 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 Device Configuration Assistant (デバイス構成用補助) (x86 版) へのアクセスの詳細は、33 ページの「x86: Solaris 9 Device Configuration Assistant (デバイス構成用補助) と PXE へのアクセス」を参照してください。
- ブート可能なメディアを使用できない場合、BIOS を編集してブート可能にしてください。詳細は、BIOS のマニュアルを参照してください。

The Solaris Installer could not find a disk that meets the criteria found in the Install documentation. Please see the documentation for more info. (x86 システムのみ)

原因: Solaris INSTALLATION CD (x86 版) からブートしようとした。システムが LBA をサポートしないため、Solaris INSTALLATIONCD を使用できない。

対処方法: CD のネットイメージ、DVD、または Solaris SOFTWARE 1 of 2 CD (x86 版) を使用してインストールします。

## 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、システムの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` プロンプトで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

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 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、Solaris 9 Supplement CD に入っています。

ブート後にインストールが失敗する。(x86 システムのみ)

原因: Solaris INSTALLATION CD からインストールする場合、Solaris 9 ルートスライスはディスクの最初の 1024 シリンダ内になければならない。

対処方法: デフォルトブートディスクの BIOS および SCSI ドライバが、LBA (Logical Block Addressing) をサポートしている必要があります。LBA により、1024 シリンダの制限を超えるマシンのブートや、複数の Solaris スライスにまたがるマシンのブートが可能になります。システムが LBA をサポートするかどうかを調べるには、表 5-2 を参照してください。システムが LBA をサポートしない場合、CD ではなく、ネットイメージからブートしてください。

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、他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ない。

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 Device Configuration Assistant (デバイス構成用補助) (x86 版) 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:

- 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台目のドライブをマスター(またはスレーブといっしょに動作するマスター)としてジャンパ設定して、2台目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、そのドライブをスレーブドライブとして指定します。ドライブがケーブルのどのドライブコネクタに接続されていてもかまいません。
- 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、次にドライブをもう一度マスターとスレーブの構成にジャンパ設定します。
- 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 Device Configuration Assistant (デバイス構成用補助) (x86 版) 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い。(x86 システムのみ)

対処方法:

- ディスクが BIOS で無効にされている場合は、Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) を使用して、ハードディスクからブートします。Solaris Device Configuration Assistant (デバイス構成用補助) へのアクセスの詳細は、33 ページの「x86: Solaris 9 Device Configuration Assistant (デバイス構成用補助) と PXE へのアクセス」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。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する。次の原因が考えられる。

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。

対処方法: 1 ネットワーク上の複数のサーバーの /etc/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調べます。複数のサーバーに登録がされている場合は、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。

対処方法: 2 インストールクライアントが複数のサーバーの /tftpboot または /rplboot ディレクトリに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のエントリがあり、別のサーバーの /etc/bootparams ファイルに、以下に示すように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している。

```
* 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する。

対処方法: 3 ネームサービスの bootparams マップまたはテーブルに \* install\_config= などのワイルドカードを使った登録がある場合は、その登録を削除し、削除したのと同じ行をブートサーバーの /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する。このシステムは、適切に設定されていない。

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。161 ページの「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を追加する」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 9 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する。

- Solaris DVD またはインストールサーバー上の Solaris DVD イメージコピー
- Solaris SOFTWARE 1 of 2 CD または インストールサーバー上の Solaris SOFTWARE 1 of 2 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 9 をインストールする場合は、Solaris DVD または Solaris SOFTWARE 1 of 2 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 DVD イメージまたは Solaris SOFTWARE 1 of 2 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。

setup\_install\_server (1M) のマニュアルページを参照してください。

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るが、認識してくれるシステムを見つけることができない。

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 suninstall プログラムが最初に NIS マップから bootparams 情報を探すことを示しています。ここで情報が見つからない場合、JumpStart プログラムまたは Solaris suninstall プログラムはブートサーバーの /etc/bootparams ファイルを調べます。

```
bootparams: nis files
```

原因: 2 クライアントの Ethernet アドレスが不正である。

対処方法: 2 インストールサーバーの /etc/ethers ファイルに登録されているクライアントの Ethernet アドレスが正しいか調べます。

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する。add\_install\_client を使用する際に不正な構成値を使用すると、この問題が発生する。たとえば、インストールしたいマシンが sun4u であるのに誤って sun4m と指定した場合などが考えられる。

対処方法: 3 正しい構成値を使用して `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: トークンリングカードを使用してシステムをブートすると、このエラーメッセージが表示される。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る。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る。

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るが、認識してくれるシステムを見つけることができない。

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るが、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ない。

対処方法: インストールサーバー上で `add_install_client` を実行します。これにより `/etc/bootparams` ファイルに適切なエントリが追加され、クライアントはネットワークからブートできます。

`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るが、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ない。

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ない。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る。

対処方法: DHCP マネージャで、オプションおよびマクロが正しく定義されていることを確認します。ルーターオプションが定義されており、その値がネットワークインストールで使用するサブネットを正しく表していることを確認します。

## ネットワークからのブート時の一般的な問題

システムはネットワークを介してブートされるが、指定したインストールサーバー以外のシステムからブートされる。

原因: このクライアントが登録された `/etc/bootparams` エントリと `/etc/ethers` エントリが別のシステム上に存在する。

対処方法: ネームサーバー上で、インストール中のシステムに対する `/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。

インストールサーバーを設定して、ネットワークを介して Solaris 9 をインストールするようにシステムを構成しても、システムをブートできない。(SPARC システムのみ)

原因: `tftpd` がインストールサーバー上で動作していない可能性がある。

対処方法: `tftpd` デーモンがインストールサーバーで実行されていることを確認します。次のコマンドを入力します。

```
# ps -ef | grep tftpd
```

このコマンドを入力しても `tftpd` デーモンが実行されていることを示す行が表示されない場合は、`/etc/inetd.conf` ファイルを編集して、次に示す行からコメント文字 (`#`) を削除します。

```
# tftp dgram udp wait root /usr/sbin/in.tftpd in.tftpd \  
-s /tftpboot
```

この変更を行なった後、システムをリブートしてください。

ネットワークからインストールするため、インストールサーバーを設定してシステムを構成しても、システムをブートできない。(x86 システムのみ)

原因: `rpld` デーモンがインストールサーバー上で動作していない可能性がある。

対処方法: `rpld` デーモンがインストールサーバーで実行されていることを確認します。次のコマンドを入力します。

```
# ps -ef | grep rpld
```

このコマンドを入力しても、`rpld` デーモンが実行されていることを示す行が表示されない場合は、次のコマンドを実行します。

```
# /usr/sbin/rpld
```

この変更を行なった後、システムをリブートしてください。

---

## Solaris 9 オペレーティング環境の初期インストール時の問題

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 DVD、Solaris INSTALLATION CD、Solaris SOFTWARE 1 of 2 CD、またはネットワークを利用してシステムをブートする必要があります。

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/Solaris_9/SUNWxxx/reloc.cpio: Broken pipe
```

説明: このエラーメッセージはインストールには影響しない。

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。システムの BIOS に設定されているデフォルトブートデバイスが、ブート時に Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) を必要とするように設定されている可能性がある。

対処方法: インストールを続けてください。必要なら、Solaris ソフトウェアのインストール後に BIOS に指定したシステムのデフォルトブートデバイスを Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) を必要としないデバイスに変更してください。

### ▼ x86: IDE ディスクの不良ブロックの検査

IDE ディスクドライブは、Solaris ソフトウェアがサポートするほかのドライブのように、不良ブロックを自動的に無効にしません。IDE ディスク上に Solaris 9 をインストールする前に、ディスクの検査を実行してください。IDE ディスクの検査を行うには、次の手順に従います。

1. シングルユーザーモードでインストールメディアをブートします。

```
# b -s
```

2. **format** 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** パーティションが必要です。

- Solaris **fdisk** パーティションがすでに存在する場合は、手順 5 に進みます。
- 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. 次のコマンドを入力します。

```
format> analyze
```

6. 次のコマンドを入力します。

```
analyze> config
```

ディスク検査の現在の設定が表示されます。

a. 設定を変更する場合は、次のコマンドを入力します。

```
analyze> setup
```

7. 次のコマンドを入力します。

```
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。

8. 次のコマンドを入力します。

```
analyze> quit
```

9. 再マップするブロックを

- 指定しない場合は、手順 10 に進みます。
- 指定する場合は、次のコマンドを入力します。

```
format> repair
```

10. 次のコマンドを入力します。

```
quit
```

**format** プログラムが終了します。

11. マルチユーザーモードでメディアを再起動する場合は、次のコマンドを入力します。

ok b

---

## Solaris 9 オペレーティング環境のアップグレード時の問題

### 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た。

対処方法: /etc/vfstab ファイルの以下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/bzczt not found

原因: パッチクラスタが必要なために Solaris Live Upgrade が失敗している。

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> でパッチを入手してください。

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 Operating Environment 1 of 2 CDROM. (x86 システムのみ)

原因: 十分な領域が存在しないため、Solaris INSTALLATION CD (x86 版) をアップグレードできない。

対処方法: アップグレードするには、512 M バイト以上のスワップスライスを作成するか、次のいずれかのアップグレード方法を使用します。

- Solaris DVD の Solaris Web Start プログラムまたはネットインストールイメージ
- Solaris SOFTWARE 1 of 2 CD (x86 版) の Solaris suninstall プログラム

- カスタム JumpStart

## 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ある。

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 ファイルが存在しない。

対処方法: 2 以下の形式で 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 が存在しない。

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:

- ミラーでなければ、vsftab ファイル内でコメントアウトします。
- ミラーの場合は、ミラーを破棄してインストールし直します。

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: アップグレード中、インストールプログラムは、システムの /etc/vfstab ファイルにリストされているすべてのファイルシステムを、アップグレードしようとしているルートファイルシステムにマウントしようとする。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、失敗して終了する。

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 suninstall 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ない。

原因: 第5章を参照してディスク容量に問題がないかを確認し、自動配置機能による領域の再配置を行わずに解決できるかどうかを調べます。

ミラー化されたルートのアップグレードに関連する問題

対処方法: Solaris ボリュームマネージャでミラー化されているルートを使用してアップグレードする際に問題が発生する場合は、『Solaris ボリュームマネージャの管理』の「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。

1. **Solaris DVD、Solaris INSTALLATION CD、Solaris SOFTWARE 1 of 2 CD**、またはネットワークを利用して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 suninstall 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。

## 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、以下の手順でアップグレードを行わないと、リブート時にシステムパニックが発生します。この問題は、パッケージが Solaris の最新のパッケージガイドラインに従っていない場合に発生します。

1. 非アクティブブート環境を作成します。418 ページの「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 /.alt.12345
```

- 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 /.alt.12345/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 /.alt.12345/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。

```
# 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、非アクティブブート環境上の **install-db** ファイルがあるディレクトリに移動します。

```
# cd /.alt.12345/etc
```

- j. 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
```

- k. 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 mount_point
```

3. 非アクティブブート環境をアップグレードします。第 36 章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。464 ページの「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 VRTSvman VRTSvmdv
```

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. リブートします。

```
# init 6
```

## x86: 既存の Service 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 Service パーティションが作成されない

Service パーティションの存在しないシステム上に Solaris 9 オペレーティング環境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 Service パーティションを作成しない場合があります。Solaris のパーティションとして同じディスクに Service パーティションを作成する場合は、Solaris 9 オペレーティング環境をインストールする前に Service パーティションを再作成する必要があります。

Solaris 8 2/02 オペレーティング環境を Sun LX50 システムにインストールする場合、インストールプログラムが Service パーティションを保持しない可能性があります。Service 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 Service パーティションを削除します。

---

注 - Solaris 8 2/02 オペレーティング環境のインストール時に Service パーティションの保持を明示的に指定しなかった場合、Service パーティションの再作成および Solaris 9 オペレーティング環境へのアップグレードを実行できない可能性があります。

---

Solaris パーティションを含むディスクに Service 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。

- Solaris Web Start インストールプログラムを使用して、Solaris 9 INSTALLATION CD からインストールする場合、次の手順を実行します。

1. ディスクの内容を削除します。
2. インストールする前に、Sun LX50 Diagnostics CD を使用して Service パーティションを作成します。

Service パーティションの作成方法の詳細は、『*Sun LX50 Server User's Manual*』および <http://cobalt-knowledge.sun.com> の Sun LX50 の Knowledge Base を参照してください。

3. Solaris 9 INSTALLATION CD を CD-ROM ドライブに挿入します。
4. Solaris 9 オペレーティング環境のインストールを開始します。

インストールプログラムが Service パーティションを検出すると、次のメッセージが表示されます。

デフォルトでは、ブートディスクの空いている領域に、x86 Boot パーティションと Solaris パーティションが配置されます。サービス fdisk パーティションが存在する場合は、デフォルトで保持されます。

継続するには次のいずれかを選択してください：

- 1) デフォルトのディスク配置を使用する
- 2) fdisk を実行し、ディスクを手動で編集する
- 3) 終了する

選択してください： [ ]

5. 「1」を入力して、デフォルトのレイアウトを使用します。

インストールプログラムにより、Service パーティションが保持され、x86 ブートパーティションと Solaris パーティションが作成されます。

---

注 – Solaris Web Start インストールプログラムにより、Solaris fdisk パーティションから 10 M バイト分の領域を確保して、x86 ブートパーティションが作成されます。このユーティリティを使用すると、既存の fdisk パーティションの変更を防ぐことができます。このパーティションは、手動で作成しないでください。

---

6. インストールが完了します。
- 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か、ネットワーク経由で Solaris 9 DVD からインストールする場合、次の手順を実行します。
    1. ディスクの内容を削除します。
    2. インストールする前に、Sun LX50 Diagnostics CD を使用して Service パーティションを作成します。

Service パーティションの作成方法の詳細は、『*Sun LX50 Server User's Manual*』および <http://cobalt-knowledge.sun.com> の Sun LX50 の Knowledge Base 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、Service パーティションが保持され、x86 ブートパーティションと Solaris パーティションが作成されます。
  - suninstall プログラムを使用して、Solaris 9 SOFTWARE 1 of 2 CD またはブートサーバー上のネットワークインストールイメージからインストールを実行するには、次の手順を実行します。
    1. ディスクの内容を削除します。
    2. インストールする前に、Sun LX50 Diagnostics CD を使用して Service パーティションを作成します。

Service パーティションの作成方法の詳細は、『*Sun LX50 Server User's Manual*』および <http://cobalt-knowledge.sun.com> の Sun LX50 の Knowledge Base を参照してください。
    3. システムをブートします。

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. 「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、Service 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。
    5. インストールが完了します。

## 付録 F

---

# リモートからのインストールまたはアップグレード (作業)

---

この付録では、DVD-ROM ドライブまたは CD-ROM ドライブが直接接続されていないシステムまたはドメインに、Solaris Web Start を使って Solaris オペレーティング環境をインストールまたはアップグレードする方法について説明します。

---

注 - マルチドメインサーバーに Solaris オペレーティング環境をインストール、またはアップグレードする場合は、インストールプロセスを開始する前に、システムコントローラまたはシステムサービスプロセッサのマニュアルを参照してください。

---

---

## SPARC: Solaris Web Start を使用してリモート DVD-ROM または CD-ROM からインストールまたはアップグレードする

DVD-ROM または CD-ROM ドライブが直接接続されていないシステムまたはドメインサーバーに Solaris オペレーティング環境をインストールしたい場合は、ほかのシステムに接続されたドライブを使用することができます。この場合、両方のシステムが同じサブネット上に接続されている必要があります。インストール手順は次のとおりです。

## ▼ SPARC: リモート DVD-ROM または CD-ROM 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

---

注 - この手順では、システムでボリュームマネージャを実行していると仮定しています。ボリュームマネージャを使用せずにメディアを管理する方法については、『Solaris のシステム管理 (基本編)』を参照してください。

---

この手順では、DVD-ROM または CD-ROM が接続されているリモートシステムをリモートシステムと呼びます。そして、インストールするクライアントをクライアントシステムと呼びます。

1. **Solaris** オペレーティング環境が動作し、かつ **DVD-ROM** ドライブまたは **CD-ROM** ドライブを備えたシステムを特定します。
2. **DVD-ROM** ドライブまたは **CD-ROM** ドライブが存在するリモートシステムで、ドライブに **Solaris DVD** または **Solaris INSTALLATION CD (SPARC 版)** を挿入します。  
ボリュームマネージャによってディスクがマウントされます。
3. リモートシステムで、**add\_install\_client** コマンドが含まれている **DVD** または **CD** のディレクトリに移ります。
  - DVD メディアの場合は、次のように入力します。

```
remote system# cd /cdrom/cdrom0/s0/Solaris_9/Tools
```
  - CD メディアの場合は、次のように入力します。

```
remote system# cd /cdrom/cdrom0/s0
```
4. リモートシステムで、クライアントとしてインストールしたいシステムを追加します。
  - DVD メディアの場合は、次のように入力します。

```
remote system# ./add_install_client \
client_system_name arch
```
  - CD メディアの場合は、次のように入力します。

```
remote system# ./add_install_client -s remote_system_name: \
/cdrom/cdrom0/s0 client_system_name arch
```

*remote\_system\_name* DVD-ROM ドライブまたは CD-ROM ドライブを備えたシステムの名前

*client\_system\_name* インストールしたいマシンの名前

*arch* インストールしたいマシンのプラットフォームグループ (例: **sun4u**)インストールしたいシステム上で、`uname -m` コマンドを実行するとプラットフォームグループが表示されます。

5. インストールしたいクライアントシステムをブートします。

```
client system: ok boot net
```

Solaris Web Start インストールが始まります。

6. 必要なら、説明に従ってシステム構成情報を入力します。
  - DVD メディアを使用している場合は、画面の説明に従ってインストールを完了します。ここで作業は終了です。
  - CD メディアを使用している場合は、マシンがリブートされ、Solaris インストールプログラムが始まります。「ようこそ (Welcome)」画面が表示された後、「ネットワーク接続性」画面で「ネットワークに接続 (Network File System)」を選択します。設定を続けると、「媒体の指定 (Specify Media)」画面が表示されます。手順 7 へ進みます。

7. 「媒体の指定 (**Specify Media**)」画面で、「次へ (**Next**)」をクリックします。「ネットワークファイルシステムのパスの指定 (**Specify Network Filesystem Path**)」画面が表示され、テキストフィールドにインストールパスが示されます。

```
client_system_ip_address:/cdrom/cdrom0/s0
```

8. DVD または CD がマウントされているリモートシステムで、ディレクトリを / (**root**) に変更します。

```
remote system# cd /
```

9. リモートシステムで、共有されているスライスへのパスがあるかチェックします。

```
remote system# share
```

10. リモートシステムで、手順 9 で見つかったパスを使って **Solaris DVD** または **Solaris INSTALLATION CD** の共有を解除します。パスが 2 つのスライスを指している場合は、両方のスライスを **unshare** します。

```
remote system# unshare absolute_path
```

*absolute\_path* share コマンドで表示される絶対パスです。

この例では、スライス 0 と 1 の共有が解除されます。

```
remote system# unshare /cdrom/cdrom0/s0
```

```
remote system# unshare /cdrom/cdrom0/s1
```

11. **Solaris INSTALLATION CD** を取り出します。

```
remote system# eject cdrom
```

12. リモートシステムの CD-ROM ドライブに **Solaris SOFTWARE 1 of 2 CD** を挿入します。

13. リモートシステムで、**Solaris SOFTWARE 1 of 2 CD** をエクスポートします。

```
remote system# share -F nfs -o ro,anon=0 /cdrom/cdrom0/s0
```

14. インストールするクライアントシステムで、「次へ (Next)」をクリックして **Solaris** のインストールを継続します。

15. **Solaris Web Start** が **Solaris SOFTWARE 2 of 2 CD** を挿入するようプロンプトを表示したら、手順 9 から手順 14 までを行い、**Solaris SOFTWARE 1 of 2 CD** の共有解除と **Solaris SOFTWARE 2 of 2 CD** のエクスポートとインストールを行います。

16. **Solaris Web Start** が **Solaris LANGUAGES CD** を挿入するようプロンプトを表示したら、手順 9 から手順 14 までを行い、**Solaris SOFTWARE 2 of 2 CD** の共有解除と **Solaris LANGUAGES CD** のエクスポートとインストールを行います。

**Solaris LANGUAGES CD** をエクスポートすると、CD-ROM がマウントされているシステム上にインストール画面が表示されます。**Solaris LANGUAGES CD** をインストールする間は、このインストール画面は無視してください。**Solaris LANGUAGES CD** のインストールが終了した後で、このインストール画面を閉じてください。



# その他の SVR4 パッケージ要件 (リファレンス)

この付録は、カスタム JumpStart プログラムや Solaris Live Upgrade を使ってパッケージ (特にサードパーティのパッケージ) のインストールや削除を行うシステム管理者のためのものです。パッケージが SVR4 パッケージ要件に準拠していれば、カスタム JumpStart インストールを非対話的に行うことができます。さらに、動作中の現在のシステムは変更されないため、Solaris Live Upgrade を使ってアップグレードを行うことができます。

## パッケージ化要件の概要

以下のドキュメントは、パッケージ化要件の背景を理解する助けになります。

- カスタム JumpStart プログラムや Solaris Live Upgrade が正しく動作するためには、パッケージが SVR4 パッケージ要件に準拠していなければなりません。『*Application Packaging Developer's Guide*』には、パッケージ化要件や用語の定義が記載されています。その中でも特に、『*Application Packaging Developer's Guide*』の「Advanced Package Creation Techniques」を参照してください。
- パッケージやユーザー管理ファイルの追加や削除に関する基本的な情報については、『*Solaris* のシステム管理 (基本編)』の「ソフトウェアの管理 (概要)」を参照してください。さらに、該当するマニュアルページも参照してください。
- この付録で参照するコマンドの詳細については、マニュアルページ `dircmp(1)`、`fssnap(1M)`、`ps(1)`、または `truss(1)` を参照してください。

表 G-1 に、Solaris Live Upgrade とカスタム JumpStart プログラムに適用される情報の参照先を示します。

表 G-1 要件情報

インストール方法	要件に関する参照先
Solaris Live Upgrade	<ul style="list-style-type: none"> <li>■ 『Application Packaging Developer's Guide』の SVR4 パッケージ要件</li> <li>■ 666 ページの「カスタム JumpStart プログラムと Solaris Live Upgrade の非アクティブブート環境要件」</li> <li>■ 668 ページの「Solaris Live Upgrade 非アクティブブート環境の要件」</li> </ul>
カスタム JumpStart プログラム	<ul style="list-style-type: none"> <li>■ 『Application Packaging Developer's Guide』の SVR4 パッケージ要件</li> <li>■ 666 ページの「カスタム JumpStart プログラムと Solaris Live Upgrade の非アクティブブート環境要件」</li> <li>■ 670 ページの「カスタム JumpStart プログラムのアップグレード要件」</li> </ul>

## カスタム JumpStart プログラムと Solaris Live Upgrade の非アクティブブート環境要件

非アクティブブート環境は、現在稼働中のシステムではなく、オペレーティング環境のコピーです。Live Upgrade やカスタム JumpStart プログラムで使用するパッケージは、次の要件に準拠していなければなりません。

- ユーザー入力なしでカスタム JumpStart インストールまたはアップグレードが可能である。
- 動作中の現在のシステムが変更されない。これは、Solaris Live Upgrade を使用する場合に必要です。

以下のリストに、非アクティブブート環境に適合するための要件を示します。

- オペレーティングシステムのインストールが成功するには、パッケージが非アクティブブート環境の指示子を認識して、それに従う必要があります。

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せん。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) パスが両方とも指定されているパッケージは、代替ルート (/) 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` 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なりません。

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なりません。インストールスクリプトを作成する場合には、そのスクリプトで、`$PKG_INSTALL_ROOT` 変数が前に付加されたディレクトリまたはファイルを参照する必要があります。パッケージでは、書き込むすべてのディレクトリやファイルの前に `$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 G-2 に、スクリプトの正しい構文を示します。

表 G-2 インストールスクリプト構文の例

スクリプトタイプ	正しい構文	正しくない構文
Bourne シェル「if」ステートメントの一部	<pre>if [ -f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf ] ; then</pre>	<pre>if [ -f /etc/myproduct.conf ] ; \ then</pre>
ファイルの削除	<pre>/bin/rm -f \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</pre>	<pre>/bin/rm -f /etc/myproduct.conf</pre>
ファイルの変更	<pre>echo "test=no"&gt; \${PKG_INSTALL_ROOT}\ /etc/myproduct.conf</pre>	<pre>echo "test=no"&gt; \ /etc/myproduct.conf</pre>

## \$PKG\_INSTALL\_ROOT と \$BASEDIR の相違点の概要

`$PKG_INSTALL_ROOT` は、パッケージを追加しようとするマシンのルート (/) ファイルシステムの場所です。この値は、`pkgadd` コマンドの `-R` 引数の後にセットされます。たとえば、次のコマンドを実行すると、パッケージのインストール時に `$PKG_INSTALL_ROOT` の値は `/a` になります。

```
# pkgadd -R /a SUNWvxxm
```

`$BASEDIR` は、再配置可能なパッケージオブジェクトがインストールされる「再配置可能」なベースディレクトリを指しています。ここにインストールされるのは、再配置可能オブジェクトだけです。再配置可能でないオブジェクト (`pkgmap` ファイルに「絶対」パスが指定されているオブジェクト) は、非アクティブブート環境の相対パスにインストールされます。`$BASEDIR` との相対パスにはインストールされません。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 (非再配置可能) 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
```

---

## Solaris Live Upgrade 非アクティブブート環境の要件

Solaris Live Upgrade を使用したり、新しいブート環境を作成する場合には、次の項目に従って問題の発生を防止してください。

- パッケージに添付されているスクリプトは、アクティブなオペレーティング環境から独立していなければなりません。パッケージに添付されているスクリプトには、パッケージのインストールや削除を行う際に特定の時点で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名で作成できるスクリプトが4つあります (preinstall, postinstall, preremove, postremove)。これらのスクリプトは、現在のアクティブなオペレーティング環境から独立していなければなりません。これは、非アクティブブート環境が Solaris Live Upgrade によって切り替えられることがあります。
- これらの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てはなりません。ps や truss はオペレーティングシステムに依存し、動作中のシステムに関する情報を報告します。
- これらのスクリプトでは、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。ただし、非アクティブブート環境を変更する場合には、666 ページの「カスタム JumpStart プログラムと Solaris Live Upgrade の非アクティブブート環境要件」に述べる規則に従ってください。
- すべてのスクリプトは Bourne シェル (/bin/sh) で書き込む必要があります。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。
- パッケージに添付されているスクリプトでは、2.6 以降のリリースで追加されたコマンドを呼び出すことはできません。たとえば、スクリプトで pgrep コマンドを呼び出すことはできません。2.6 リリース以降、多くのコマンドに新しい機

能が追加されました。スクリプトでは、2.6 リリースにないコマンドオプションを使用することはできません。たとえば、Solaris 7 リリースで、umount コマンドに `-f` オプションが追加されました。特定のコマンドまたはオプションが Solaris 2.6 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> の『Solaris 2.6 Reference Manual AnswerBook』を参照してください。

- 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。

たとえば、パッケージが `/export/SUNWvxvm` にあるなら、コマンドを次のように指定します。

```
# pkgchk -d /export SUNWvxvm
```

エラーが表示されないことを確認します。

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非アクティブブート環境にインストールしてテストします。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 SUNWvxvm
```

```
# pkgchk -R /a SUNWvxvm
```

エラーが表示されないことを確認します。

- さらに、パッケージでは、パッケージ自身が提供しているコマンドを実行しないでください。これは、ディスクレス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。

ファイルの作成や変更、削除に対する要件はさまざまなコマンドで検証できます。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` 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 ${TMPDIR} SUNWvxvm \
2>&1> /dev/null | grep -v O_RDONLY | grep -v \
'open("${TMPDIR}
```

このセクションで示したコマンドの詳細は、`dircmp(1)`、`fssnap(1M)`、`ps(1)`、`truss(1)`、`pkgadd(1M)`、`pkgchk(1M)`、または `pkgrm(1M)` のマニュアルページを参照してください。

---

## カスタム JumpStart プログラムのアップグレード要件

パッケージがカスタム JumpStart プログラムの要件に準拠していれば、次に示す従来の Solaris インストールユーティリティを使ってパッケージの追加や削除を行うことができます。

- カスタム JumpStart プログラム
- Solaris `suninstall` プログラム
- Solaris Web Start インストール

さらに、カスタム JumpStart プログラムの要件に準拠したパッケージは Solaris アップグレードに入れることができます。カスタム JumpStart プログラムに準拠するには、パッケージは 666 ページの「カスタム JumpStart プログラムと Solaris Live Upgrade の非アクティブブート環境要件」に説明されている非アクティブブート環境要件にも準拠する必要があります。

処理の途中でユーザー入力を求められないほうが、カスタム JumpStart プログラムを効果的に使用することができます。ユーザーの入力作業をなくすためには、`pkgadd` コマンドの `-a` オプションで使用する新しい管理ファイルを設定する必要があります。`-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。詳細は、マニュアルページ `admin(4)` または `pkgadd(1M)` を参照してください。

次の例で、`pkgadd` 管理ファイルの使い方を説明します。

- 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
```

- 絶対パスでファイルを指定すると、pkgadd はこれを使用します。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
```

次に示すのは、pkgadd ユーティリティ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
```





# Solaris Update リリースへのアップグレード (作業)

---

この付録では、Solaris オペレーティング環境の Update リリースにアップグレードする前にパッチをチェックする手順について説明します。

---

## Solaris Update リリースへのアップグレード

すでに Solaris 9 オペレーティング環境を実行していて、システムに個別のパッチがインストールされている場合、Solaris 9 Update リリースへのアップグレードを行うと、パッチは次のようになります。

- Solaris 9 Update リリースの一部として組み込まれているパッチは自動的に再適用されます。これらのパッチをバックアウトすることはできません。
- システムにすでにインストールされているパッチのうち、Solaris 9 リリースに組み込まれていないパッチは削除されます。

パッチアナライザを使用すれば、Solaris 9 Update リリースにアップグレードしたときに、これらのパッチのうちどれが削除されるかを確認することができます。パッチアナライザは次のような方法で使用できます。

- Solaris Web Start を使用してアップグレードする場合、「パッチの解析 (Patch Analyzer)」画面が表示されます。「はい (Yes)」を選択して解析を実行します。
- Solaris `suninstall` プログラムを使用してアップグレードする場合、「パッチの解析 (Patch Analysis)」画面で「解析 (Analyze)」を選択して解析を実行します。
- カスタム JumpStart インストールまたは Solaris Live Upgrade を使用してアップグレードする場合、`analyze_patches` スクリプトを実行して解析を実行します。詳細は、674 ページの「`analyze_patches` スクリプトを実行する方法」を参照してください。

パッチ解析結果の見方については、675 ページの「パッチアナライザの出力の確認方法」を参照してください。

## ▼ analyze\_patches スクリプトを実行する方法

---

注 - analyze\_patches スクリプトを実行するには、NFS またはローカルにマウントされたメディア上にあるスクリプトを使用して、インストールされた既存のシステム、および Solaris DVD、Solaris SOFTWARE CD、またはネットワーク上のイメージにアクセスする必要があります。

---

### 1. Misc ディレクトリに移動します。

- SPARC: イメージが、ローカルにマウントされた媒体上にある場合は、次のように入力します。

```
# cd /cdrom/sol_9_Update_sparc/s0/Solaris_9/Misc
```

このコマンドにおいて、*Update* は実際の Update 識別子を表します (例: u1)。

- x86: イメージが、ローカルにマウントされた媒体上にある場合は、次のように入力します。

```
# cd /cdrom/sol_9_Update_x86/s2/Solaris_9/Misc
```

このコマンドにおいて、*Update* は実際の Update 識別子を表します (例: u1)。

- イメージが NFS ファイルシステムにある場合は、次のように入力します。

```
# cd /NFS_mount_directory/Solaris_9/Misc
```

### 2. analyze\_patches スクリプトを実行します。

```
# ./analyze_patches [-R rootdir] [-N netdir] [-D databasedir]
```

-R *rootdir*      *rootdir* には、インストールされているシステムのルートを指定します。デフォルトで / に設定されます。

-N *netdir*      *netdir* は、インストールする OS イメージのルートへのパスです。デフォルトは /cdrom/cdrom0 です。*netdir* は、Solaris\_9 ディレクトリを含むディレクトリへのパスです。*patch\_analyzer* を NFS マウントポイントから実行する場合は、このオプションが必要です。

-D *databasedir*      スクリプトを OS イメージの Misc ディレクトリ以外のディレクトリから起動すると、パッチ解析に使用するデータベースを検索できません。したがって、-D オプションでこのデータベースへのパスを指定します。このデータベースがないと、スクリプトは正しく動作しません。このデータベースは、OS イメージの Solaris\_9/Misc/database にあります。

## ▼ パッチアナライザの出力の確認方法

解析を実行したら、次の手順で出力を検討します。

### 1. パッチアナライザの出力を確認します。

パッチアナライザからは、ほかのパッチによって削除、ダウングレード、置き換えられ削除されるパッチのリストが出力されます。パッチの置き換えは、パッチアップグレードと同等のものです。置き換えられるパッチは削除され、新しいパッチによってその修正が適用されます。次のようなメッセージが表示されます。

```
Patch 105644-03 will be removed.  
Patch 105925 will be downgraded from -02 to -01.  
Patch 105776-01 will be accumulated/obsoleted by patch 105181-05.
```

パッチアナライザからリストが出力されない場合は、システムにすでにインストールされているパッチに対する処理は何も行われません。

### 2. パッチの置き換えや削除が適切かどうかを判断します。

- 適切な場合は、システムをアップグレードします。
- 適切でない場合は、システムのアップグレードを中止します。

Update リリースでは、アップグレードのかわりに Solaris 9 Maintenance Update リリースを用いて、パッチだけをシステムに適用できます。

---

注 – Solaris 9 Maintenance Update は Web から入手できます。パッチの適用手順については、『Solaris 9 Maintenance Update ご使用にあたって』を参照してください。

---



## 付録 I

---

# x86: Solaris Device Configuration Assistant (デバイス構成用補助) または ネットワークからのブート準備 (作業)

---

この付録の内容は次のとおりです。

- 677 ページの「x86: ブートソフトウェアのフロッピーディスクへのコピー」
- 679 ページの「x86: 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」

---

## x86: ブートソフトウェアのフロッピーディスクへのコピー

Solaris Device Configuration Assistant (デバイス構成用補助) は、さまざまなハードウェア構成作業とブート作業を行うためのプログラムです。Solaris Device Configuration Assistant (デバイス構成用補助) イメージは、Solaris DVD (x86 版) または Solaris SOFTWARE 2 of 2 CD (x86 版) の Tools ディレクトリ内に存在します。次の手順を使用して、ブートイメージを 3.5 インチのフロッピーディスクにコピーしてください。

---

注 - DVD または CD メディアから直接ブートすることも、ネットイメージと PXE を使用してブートすることも可能です。これらのブート方法の詳細は、33 ページの「x86: Solaris 9 Device Configuration Assistant (デバイス構成用補助) と PXE へのアクセス」を参照してください。

---

## ▼ x86: ブートソフトウェアをフロッピーディスクにコピーする方法

---

注 - この手順では、システムでボリュームマネージャを実行していると仮定しています。ボリュームマネージャを使用せずにフロッピーディスクおよび他のディスクを管理する方法については、『Solaris のシステム管理 (基本編)』を参照してください。

---

1. フロッピーディスクドライブを持つ **x86** システムにスーパーユーザーとしてログインします。
2. **DVD-ROM** または **CD-ROM** ドライブを備えたシステムの場合、**Solaris DVD (x86 版)** または **Solaris SOFTWARE 2 of 2 CD (x86 版)** をドライブに挿入します。ボリュームマネージャによってディスクがマウントされます。
3. ブートイメージが入っているディレクトリに移動します。
  - DVD メディアの場合は、次のように入力します。

```
# cd /cdrom/sol_9_x86/s2/Solaris_9/Tools
```
  - CD メディアの場合は、次のように入力します。

```
# cd /cdrom/sol_9_x86/Solaris_9/Tools
```
4. 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。
5. ボリュームマネージャに対し、新しいメディアをチェックするように指示します。

```
# volcheck
```
6. フロッピーディスクをフォーマットします。



---

注意 - この手順によって、フロッピーディスク上のデータはすべて消去されます。

---

```
# fdformat -d -U
```

7. ファイルをフロッピーディスクにコピーします。

```
# dd if=d1_image of=/vol/dev/aliases/floppy0 bs=36k
```
8. コマンド行で **eject floppy** と入力してから、手動でドライブからフロッピーディスクを取り出します。

---

## x86: 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

この節では、ローカルのブートメディアを使用せずネットワークからインストールするように x86 ベースのシステムを設定する方法について説明します。この節の内容は次のとおりです。

- 679 ページの「PXE の概要」
- 679 ページの「x86: PXE によるブートに関するガイドライン」
- 680 ページの「PXE によるブート (作業マップ)」
- 681 ページの「x86: x86 インストールサーバーの作成」
- 684 ページの「x86: PXE を使用してネットワーク経由でインストールするシステムの追加」
- 689 ページの「x86: PXE を使用したネットワーク経由でのクライアントのブート」

### PXE の概要

PXE ネットワークブートは、「ダイレクト」なネットワークブートです。クライアントシステム上に、ブートメディアがなくても構いません。PXE では、DHCP 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できます。

PXE ネットワークブートは、Intel の Preboot Execution Environment (PXE) 仕様を実装しているデバイスでのみ動作します。システムが PXE ネットワークブートをサポートしているかどうかを判別するには、ハードウェアのマニュアルを参照してください。

Solaris ブートフロッピーディスクは、PXE をサポートしていないシステムでも従来どおり利用できます。ブート用フロッピーディスクイメージは Solaris SOFTWARE 2 of 2 CD (x86 版) に格納されています。

### x86: PXE によるブートに関するガイドライン

PXE を使用してネットワーク経由でブートするには、次のシステムが必要です。

- インストールサーバー
- DHCP サーバー
- PXE をサポートする x86 クライアント

PXE を使用してネットワーク経由でクライアントをインストールする準備を行う際、次の点に注意してください。

- インストールするクライアントシステムがあるサブネット上には、DHCP サーバーを 1 台だけ設定します。PXE ネットワークブートは、複数の DHCP サーバーがあるサブネット上では、正常に実行できません。

- 一部の初期バージョンの PXE ファームウェアでは、Solaris システムをブートすることができません。これらの古いバージョンを搭載したシステムでは、PXE ネットワークブートストラッププログラムをブートサーバーから読み込むことはできても、ブートストラップがパケットを転送しません。この問題を回避するには、アダプタの PXE ファームウェアをアップグレードしてください。ファームウェアのアップグレードに関する情報は、アダプタの製造業者の Web サイトから入手してください。詳細については、e1x1 (7D) および iprb (7D) のマニュアルページを参照してください。

## PXE によるブート (作業マップ)

PXE を使用してネットワーク経由でシステムをブートしてインストールするには、次の作業を行います

表 I-1 x86: 作業マップ: PXE を使用したネットワーク経由のブート

作業	説明	参照先
システムが PXE をサポートしていることを確認する	PXE を使用してローカルのブートメディアなしでブートできるマシンであることを確認します。	ハードウェアのマニュアルを参照してください。
インストール方法を選択する	Solaris オペレーティング環境は、複数のインストールまたはアップグレード方法を提供します。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。	第 3 章
システム情報を収集する	チェックリストやワークシートを使って、インストールやアップグレードに必要なすべての情報を収集します。	第 6 章
(省略可能) システム構成情報を事前設定する	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。	第 7 章
インストールサーバーを作成する	Solaris オペレーティング環境をネットワークからインストールするように Solaris サーバーを設定します。	681 ページの「x86: x86 インストールサーバーの作成」



表 I-1 x86: 作業マップ: PXE を使用したネットワーク経由のブート (続き)

作業	説明	参照先
ネットワークインストールするシステムを追加する	あるクライアントクラス (たとえば特定のマシンタイプ) に属するシステムまたは特定のクライアント ID に該当するシステムをインストールするために、 <code>add_install_client -d</code> を使用して DHCP サポートを追加します。	684 ページの「x86: PXE を使用してネットワーク経由でインストールするシステムの追加」
DHCP サーバーを設定する	DHCP サービスの計画と構成を行います。	『Solaris のシステム管理 (IP サービス)』の「DHCP サービスの使用計画 (手順)」
インストールパラメータ用の DHCP オプションとそのオプションを含むマクロを作成する	DHCP マネージャまたは <code>dhtadm</code> を使用して、 <code>add_install_client -d</code> コマンドから出力されるペンダーオプションとマクロを作成します。	76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」
クライアントをブートする	ネットワークからブートするようにクライアント BIOS に指示します。	689 ページの「x86: PXE を使用したネットワーク経由でのクライアントのブート」

## x86: x86 インストールサーバーの作成

インストールサーバーには、ネットワーク上でシステムをインストールするために必要なインストールイメージが含まれます。Solaris ソフトウェアをネットワークからインストールするためには、インストールサーバーを作成する必要があります。

注 – Solaris DVD メディアを使用して Solaris 2.6 または 7 オペレーティング環境の稼働するシステム上でインストールサーバーを設定する場合、最初に次のパッチのいずれかを適用する必要があります。

- Solaris 2.6 オペレーティング環境 (Intel 版) — Patch ID 107619-03
- Solaris 7 オペレーティング環境 (Intel 版) — Patch ID 107260-03

## ▼ x86 インストールサーバーを作成する方法

---

注 - この手順では、次のように仮定します。

- x86 ベースのシステムにインストールサーバーを作成する。SPARC システムを x86 版 Solaris インストールイメージとして使用する方法については、119 ページの「SPARC または x86 DVD メディアを使用して SPARC インストールサーバーを作成する方法」を参照
  - システムに CD-ROM ドライブがある
  - システムでボリュームマネージャを実行している。ボリュームマネージャを使用せずにメディアを管理する方法については、『Solaris のシステム管理 (基本編)』を参照
- 

1. インストールサーバーとして使用するシステムでスーパーユーザーになります。このシステムには CD-ROM ドライブが必要です。さらに、システムは、このサイトのネットワークに接続されネームサービスに登録されている必要があります。ネームサービスを使用する場合は、システムがすでに NIS、NIS+、DNS、LDAP のいずれかのネームサービスに登録されていなければなりません。ネームサービスを使用しない場合は、サイトのポリシーに従ってシステムの情報を供給する必要があります。

2. **Solaris SOFTWARE 1 of 2 CD** をシステムのドライブに挿入します。

3. **CD** イメージ用のディレクトリを作成します。

```
# mkdir -p install_dir_path
```

`install_dir_path` CD イメージをコピーするディレクトリを指定します。

4. マウントされたディスクの **Tools** ディレクトリに移動します。

```
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
```

上の例では、`cdrom0` は、Solaris オペレーティング環境の CD メディアを含むドライブへのパスです。

5. ドライブ内のイメージをインストールサーバーのハードディスクにコピーします。

```
# ./setup_install_server install_dir_path
```

`install_dir_path` CD イメージをコピーするディレクトリを指定します。

---

注 - `setup_install_server` コマンドは、Solaris SOFTWARE ディスクイメージをコピーする十分なディスク容量があるかどうかを調べます。利用できるディスク容量を調べるには、`df -k1` コマンドを使用します。

---

6. ルート (/) ディレクトリへ移動します。

```
# cd /
```

7. Solaris SOFTWARE 1 of 2 CD を取り出します。

8. Solaris SOFTWARE 2 of 2 CD をシステムの CD-ROM ドライブに挿入します。

9. マウントされた CD の Tools ディレクトリに移動します。

```
# cd /cdrom/cdrom0/Solaris_9/Tools
```

10. CD-ROM ドライブ内の CD をインストールサーバーのハードディスクにコピーします。

```
# ./add_to_install_server install_dir_path
```

*install\_dir\_path* CD イメージをコピーするディレクトリを指定します。

11. ルート (/) ディレクトリへ移動します。

```
# cd /
```

12. Solaris SOFTWARE 2 of 2 CD を取り出します。

13. Solaris LANGUAGES CD をシステムの CD-ROM ドライブに挿入します。

14. マウントされた CD の Tools ディレクトリに移動します。

```
# cd /cdrom/cdrom0/Tools
```

15. CD-ROM ドライブ内の CD をインストールサーバーのハードディスクにコピーします。

```
# ./add_to_install_server install_dir_path
```

*install\_dir\_path* CD イメージをコピーするディレクトリを指定します。

16. ルート (/) ディレクトリへ移動します。

```
# cd /
```

17. ネットインストールイメージのミニルート

(*/install\_dir\_path/Solaris\_9/Tools/Boot*) 内のファイルにパッチを適用する場合は、**patchadd -C** コマンドを使用します。ブートイメージに問題がある場合、ファイルにパッチを適用する必要があります。



---

注意 - patchadd -C を使用する前には必ず、パッチの README を読むか、ご購入先におたずねください。

---

例 I-1 x86: x86 CD メディアによる x86 インストールサーバーの作成

次の例は、次の CD をインストールサーバーの `/export/home/cdx86` ディレクトリにコピーして、インストールサーバーを作成する方法を示しています。

- Solaris SOFTWARE 1 of 2 CD (x86 版)
- Solaris SOFTWARE 2 of 2 CD (x86 版)
- Solaris LANGUAGES CD (x86 版)

```
# mkdir -p /export/home/cdx86
# cd /cdrom/cdrom0/s2/Solaris_9/Tools
# ./setup_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/Solaris_9/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/Tools
# ./add_to_install_server /export/home/cdx86
# cd /
# cd /cdrom/cdrom0/s2
# ./modify_install_server -p /export/home/cdx86 /cdrom/cdrom0/s2
```

この例では、各コマンドを実行する前に各 CD がドライブに挿入され、自動的にマウントされるものとします。各コマンドの実行が終了すると、CD が取り出されます。

## x86: PXE を使用してネットワーク経由でインストールするシステムの追加

インストールサーバーを作成したら、ネットワークからインストールする各システムを設定する必要があります。インストールする各システムは、次の情報にアクセスできる必要があります。

- インストールサーバー
- DHCP サーバー
- `sysidcfg` ファイル (`sysidcfg` ファイルを使用してシステム構成情報を事前設定する場合)
- ネームサーバー (ネームサービスを使用してシステム構成情報を事前設定する場合)
- プロファイルサーバー上の `JumpStart` ディレクトリにあるプロファイル (カスタム `JumpStart` インストールを使用する場合)

PXE を使用してネットワークからインストールするように x86 クライアントを設定するには、次の手順のように `add_install_client` を使用します。

## ▼ DHCP を使用してネットワーク経由でインストールするクライアントを追加する方法

1. インストールサーバーで、スーパーユーザーになります。
2. NIS、NIS+、DNS、または LDAP のネームサービスを使用する場合、インストールするシステムについての次のような情報がネームサービスに追加されていることを確認します。

- ホスト名
- IP アドレス
- Ethernet アドレス

ネームサービスの詳細は、『Solaris のシステム管理 (ネーミングとディレクトリサービス : DNS、NIS、LDAP 編)』を参照してください。

3. インストールサーバー上の Solaris 9 CD イメージの Tools ディレクトリに移動します。

```
# cd /install_dir_path/Solaris_9/Tools
```

*install\_dir\_path* Tools ディレクトリのパスを指定します。

4. ネットワークでインストールするためのシステムを設定します。

```
# ./add_install_client -d [-s install_server:install_dir_path] \  
[-c jumpstart_server:jumpstart_dir_path] [-p sysid_server:path] \  
[-t boot_image_path] [-b "boot-property=value"] \  
[-e ethernet_address] client_name platform_group
```

-d

ネットワークインストールパラメータの取得にクライアントが DHCP を使用することを指定します。-d だけを使用した場合、`add_install_client` コマンドは、同じクラスに属するすべてのクライアントシステム (たとえばすべての x86 クライアントマシン) のインストール情報を設定します。特定のクライアントのインストール情報を設定する場合は、-d と -e を併用します。

DHCP を使用した特定のクラスのインストールの詳細は、76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」を参照してください。

-s *install\_server:install\_dir\_path*

インストールサーバーの名前とパスを指定します。

- *install\_server* は、インストールサーバーのホスト名です。

- *install\_dir\_path* は、Solaris 9 CD イメージの絶対パスです。
- c *jumpstart\_server*:*jumpstart\_dir\_path*      カスタム JumpStart インストールの JumpStart ディレクトリを指定します。*jumpstart\_server* には JumpStart ディレクトリがあるサーバーのホスト名を指定します。*jumpstart\_dir\_path* は、JumpStart ディレクトリへの絶対パスです。
- p *sysid\_server*:*path*      システムの構成情報を事前設定するための *sysidcfg* ファイルのパスを指定します。*sysid\_server* は、このファイルを持っているサーバーの有効なホスト名または IP アドレスです。*path* は、*sysidcfg* ファイルを含むディレクトリの絶対パスです。
- t *boot\_image\_path*      Solaris 9 ネットインストールイメージ、CD、または DVD 上の Tools ディレクトリにあるブートイメージ以外のブートイメージを使用したい場合、別のブートイメージへのパスを指定します。
- b *boot-property=value*      **x86** システムのみ:ネットワークからクライアントをブートするとき使用するブートプロパティ変数の値を設定できます。*-b* は、*-e* オプションと組み合わせて使用してください。

ブートプロパティについては、*eeprom* (1M) のマニュアルページを参照してください。
- e *ethernet\_address*      インストールするクライアントの Ethernet アドレスを指定します。このオプションを指定すると、特定のクライアントのインストール情報を設定できません。

DHCP を使用した特定のクライアントのインストールの詳細は、76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」を参照してください。

<i>client_name</i>	ネットワークからインストールするシステムのホスト名を指定します。この名前はインストールサーバーのホスト名ではありません。
<i>platform_group</i>	インストール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。プラットフォームグループの詳細は、付録 C を参照してください。

上記のコマンドは、DHCP サーバーに追加する必要があるベンダーオプションとマクロを出力します。DHCP サーバーでこれらのベンダーオプションとマクロを定義する方法については、76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」を参照してください。

例 I-2 x86: DHCP および PXE を使用している場合の x86 インストールサーバー上の x86 インストールクライアントの追加 (CD)

次の例は、ネットワークのインストールパラメータの設定に DHCP を使用する場合に、インストールサーバーに x86 インストールクライアントを追加する方法を示しています。-d オプションを指定すると、クライアントのネットワークインストールパラメータを構成する際に、DHCP プロトコルが使用されます。DHCP クラス名 SUNW.i86pc は、このコマンドが単一のクライアントだけでなく、ネットワーク上のすべての Solaris x86 ブートクライアントに適用されることを意味します。-s オプションには、rosemary という名前のインストールサーバーからクライアントをインストールすることを指定します。Solaris SOFTWARE 1 of 2 CD (x86 版) イメージは、このサーバーの /export/home/cdx86 にあります。

DHCP を使用してネットワークインストール用のインストールパラメータを設定する方法については、『Solaris のシステム管理 (IP サービス)』の「DHCP サービスを使用した Solaris ネットワークインストールのサポート (作業マップ)」を参照してください。

```
x86_install_server# cd /export/boot/Solaris_9/Tools
x86_install_server# ./add_install_client -d -s rosemary:/export/home/cdx86 \
SUNW.i86pc i86pc
```

例 I-3 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD)

次の例は、インストールサーバーに x86 インストールクライアントを追加し、インストール時に使用するシリアルコンソールを指定する方法を示しています。この例では、インストールクライアントは次のように設定されます。

- -d オプションが指定されているので、クライアントは DHCP を使用してインストールパラメータを設定します。
- -e オプションが指定されているので、Ethernet アドレスが 00:07:e9:04:4a:bf と一致するクライアントに対してのみ、インストールが行われます。
- -b オプションが指定されているので、インストールプログラムは入出力デバイスとしてシリアルポート ttya を使用します。

例 I-3 x86: ネットワークインストール時に使用するシリアルコンソールの指定 (CD) (続き)

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" i86pc
```

-b オプションで使用可能なブートプロパティ変数の詳細情報は、eeprom(1M)のマニュアルページを参照してください。

例 I-4 x86: ネットワークインストール時に使用するブートデバイスの指定 (CD)

次の例は、インストールサーバーに x86 インストールクライアントを追加し、インストール時に使用するブートデバイスを指定する方法を示しています。インストールクライアントの設定時にブートデバイスを指定すると、インストール時、Device Configuration Assistant (デバイス構成用補助) はこの情報の入力プロンプトを表示しません。

この例では、インストールクライアントは次のように設定されます。

- -d オプションが指定されているので、クライアントは DHCP を使用してインストールパラメータを設定します。
- -e オプションが指定されているので、Ethernet アドレスが 00:07:e9:04:4a:bf と一致するクライアントに対してのみ、インストールが行われます。
- 1 番目と 2 番目の -b オプションにより、インストールプログラムの入出力デバイスがシリアルポート ttya に設定されます。
- 3 番目の -b オプションにより、インストールプログラムがインストール時に特定のブートデバイスを使用するように設定されます。

---

注-ブートデバイスパスの値は、使用するハードウェアによって異なります。

---

- i86pc プラットフォーム名から、クライアントが x86 システムであることがわかります。

```
install server# cd /export/boot/Solaris_9/Tools
install server# ./add_install_client -d -e "00:07:e9:04:4a:bf" \
-b "input-device=ttya" -b "output-device=ttya" \
-b "bootpath=/pci@0,0/pci108e,16a8@8" i86pc
```

-b オプションで使用可能なブートプロパティ変数の詳細情報は、eeprom(1M)のマニュアルページを参照してください。



## x86: PXE を使用したネットワーク経由でのクライアントのブート

ネットワーク経由でシステムをインストールするには、ネットワーク経由でブートするようにクライアントシステムに指示する必要があります。システム BIOS またはネットワークアダプタ BIOS のどちらか一方、またはその両方の BIOS 設定プログラムを使うことによって、クライアントシステム上で PXE ネットワークブートを使用できるようにします。いくつかのシステムでは、他のデバイスからのブートよりも先にネットワークブートが実行されるように、ブートデバイスの優先順位を調整する必要があります。各設定プログラムに関しては、製造業者のマニュアルを参照するか、またはブート中に表示される設定プログラムの指示を参照してください。

インストール時にプロンプトが表示されたら、システム構成情報を入力します。

システムがネットワーク経由でブートし、インストールされたら、次回以降はディスクドライブからブートするようにシステムに指示します。

PXE 対応ネットワークアダプタの中には、ブート時にしばらく表示されるプロンプトに対して特定のキーを押すと、PXE ブートを実行する機能を持つものがあります。この機能により、PXE 設定を変更することなく、PXE を使用してネットワーク経由でブートを行うことができます。アダプタにこの機能がない場合は、システムのインストール後に行われるリブート時に、BIOS の設定で PXE を使用しないように設定してください。システムがディスクドライブからブートするようになります。



# Solaris 9 での更新情報

---

この付録には、Solaris 9 以降この本に加えられた変更点を記述します。

---

## Solaris 9 9/02 リリース

バグの修正がいくつか追加されました。

---

## Solaris 9 12/02 リリース

次に、Solaris 9 12/02 リリースで追加された新機能および本書に加えられた変更点を示します。

- フラッシュのインストール機能を実行する際、利用可能ないくつかの処理およびサンプルが新たに追加されました。
- フラッシュアーカイブの作成に、`flarcreate` コマンドを使用できます。このコマンドに追加された新しいオプションを使用すると、アーカイブ作成時にアーカイブの内容をより柔軟に定義できます。1 つ以上のファイルやディレクトリを除外できるようになりました。除外されたディレクトリの中から、特定のサブディレクトリやファイルを残すこともできます。この機能は、複製する必要のない大規模なデータファイルを除外する場合に役立ちます。これらのオプションの使用方法については、208 ページの「アーカイブのファイルとディレクトリのカスタマイズ」を参照してください。
- アーカイブに関する情報の分割、マージ、および抽出を行うための新たな処理が追加されました。これらの処理の詳細については、234 ページの「フラッシュアーカイブの管理」を参照してください。

- Solaris インストールプログラムが LDAP Version 2 プロファイルをサポートするようになったため、プロキシ資格レベルを使用してシステムを構成できます。Solaris Web Start または Solaris suninstall プログラムの実行時に、LDAP プロキシのバインド識別名およびプロキシバインドパスワードを指定できます。どのインストール方法を採用する場合でも、sysidcfg ファイル内で proxy\_dn および proxy\_password キーワードを使用することにより、インストール前に LDAP を構成できます。これらのプロファイルを事前に構成する方法については、65 ページの「sysidcfg ファイルによる事前設定」を参照してください。
- Solaris Live Upgrade で使用可能な、次の処理およびサンプルが新たに追加されました。
  - プロファイルの作成
  - プロファイルのテスト
  - プロファイルを使用した、フラッシュアーカイブのアップグレードまたはインストール

詳細は、第 36 章を参照してください。

- 本書に、x86 システムに関連した手順および説明が含まれるようになりました。
- バグの修正がいくつか追加されました。

---

## Solaris 9 4/03 リリース

次に、Solaris 9 4/03 リリースで追加された新機能および本書に加えられた変更点を示します。

- フラッシュインストール機能により今回の Solaris リリースへの新しい機能強化が図られました。
  - フラッシュインストールは、小規模の変更でクローン システムをアップデートできるようになりました。クローンシステムを小規模な変更でアップデートしたい場合、元のマスターイメージとアップデート後のマスターイメージとの変更点のみを含む差分アーカイブを作成できます。差分アーカイブを使用してクローンシステムをアップデートする場合、差分アーカイブで指定されたファイルだけが変更されます。インストールは、元のマスターイメージとの一貫性を保持するソフトウェアを含むクローンシステムに限定されます。カスタム JumpStart インストール方法を使用して、クローンシステムに差分アーカイブをインストールします。差分アーカイブの作成手順については、225 ページの「更新されたマスターイメージを使用してフラッシュ差分アーカイブを作成する方法」を参照してください。
  - 特別なスクリプトを利用してマスターやクローンの構成を行ったり、アーカイブの妥当性を検査できるようになりました。これらのスクリプトを使用して、次のタスクを実行できます。

- クローンシステム上でアプリケーションを構成します。一部の複雑でない構成に対して、カスタム JumpStart スクリプトを使用できます。より複雑な構成の場合、マスターシステム上で、またはインストールの前か後にクローンシステム上で、特別な構成ファイル処理が必要な場合があります。また、ローカルのプリインストールスクリプトおよびポストインストールスクリプトをクローンに配置して、フラッシュインストールによりローカルのカスタマイズが上書きされるのを防ぐこともできます。
- 複製不可のホスト依存データを識別して、フラッシュアーカイブをホスト非依存にできます。ホスト非依存にするには、この種のデータを変更するか、アーカイブから除外します。ホスト依存データの例は、ログファイルです。
- 作成時に、アーカイブ内でソフトウェアの整合性を確認します。
- クローンシステム上でインストールを検証します。

スクリプトの作成については、215 ページの「カスタムスクリプトの作成」を参照してください。

- Solaris 9 4/03 オペレーティング環境では、Solaris Web Start および suninstall インストールプログラムは、新しい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、x86 システム上の Service パーティションに対応します。システムに Service パーティションが含まれる場合、新しい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。  
詳細については、48 ページの「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」を参照してください。
- バグの修正がいくつか追加されました。

---

## Solaris 9 8/03 リリース

次に、Solaris 9 8/03 リリースで追加された新機能および本書に加えられた変更点を示します。

- Solaris Live Upgrade に新しい拡張機能が追加されました。
  - Solaris Live Upgrade は Solaris ボリュームマネージャのテクノロジーを使用して、RAID-1 ボリューム (ミラー) を備えたファイルシステムを持つブート環境のコピーを作成します。ミラーを使用すると、ルート (/) ファイルシステムを含めどのファイルシステムでも、データの冗長性が得られます。lucreate コマンドでは、最大 3 つのサブミラーを持つ、ミラー化されたファイルシステムを作成できます。  
この概要については、384 ページの「ミラー化されたファイルシステムを持つブート環境の作成」を参照してください。手順については、438 ページの「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。

- `lucreate` コマンドを使用してブート環境を作成するときに、コピーするファイルやディレクトリを指定できます。通常であれば元のブート環境からコピーされるファイルやディレクトリの一部を、除外するように指定できます。ディレクトリを除外した場合は、そのディレクトリ内にある特定のファイルやサブディレクトリを除外しないように指定することも可能です。計画情報については、408 ページの「新しいブート環境の内容のカスタマイズ」を参照してください。手順については、444 ページの「ブート環境の作成と内容のカスタマイズ (コマンド行インタフェース)」を参照してください。
- Solaris Live Upgrade を使用して 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できるようになりました。Solaris Live Upgrade を使用して差分アーカイブをインストールする方法については、453 ページの「Solaris Live Upgrade で使用されるプロファイルを作成する」を参照してください。
- Solaris オペレーティング環境をインストールするときに、JumpStart インストールを使用して空のブート環境を作成できるようになりました。後でこの空のブート環境に Solaris フラッシュアーカイブをインストールできます。
- バグの修正がいくつか追加されました。

---

## Solaris 9 12/03 リリース

次に、Solaris 9 12/03 リリースで追加された新機能および本書に加えられた変更点を示します。

- DHCP サーバーを構成する手順が追加されました。DHCP サーバーを構成して、ネットワークインストール用にクライアント構成情報を提供できます。DHCP サーバーを構成してネットワークインストールをサポートする方法については、76 ページの「Solaris インストールパラメータ用の DHCP オプションとマクロの作成」を参照してください。
- Solaris オペレーティング環境では、リモートクライアントを広域ネットワーク (WAN) を介してインストールできます。WAN ブートインストールでは、暗号化した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リモートの SPARC ベースのクライアントに転送できます。次に、WAN ブートプログラムは、カスタム JumpStart インストールを実行して、クライアントシステムをインストールします。インストールの完全性を保護するために、非公開鍵を使ってデータの認証および暗号化を行うことができます。また、デジタル証明書を使用するようにシステムを構成すると、HTTPS 接続を介してインストールデータやファイルを転送できます。
  - WAN ブートインストールの概要については、第 41 章を参照してください。
  - WAN ブートインストールの計画については、第 42 章を参照してください。
  - WAN ブートインストールに必要なネットワークを準備する手順については、第 43 章を参照してください。

- WAN を介してリモートクライアントをインストールする方法については、第 44 章を参照してください。
- WAN ブートインストールに必要な作業の例については、第 45 章を参照してください。
- WAN ブートインストールに必要なコマンドとファイルに関するリファレンス情報については、第 46 章を参照してください。
- バグの修正がいくつか追加されました。

---

## Solaris 9 4/04 リリース

次に、Solaris 9 4/04 リリースで追加された新機能および本書に加えられた変更点を示します。

- カスタム JumpStart インストールでは、Solaris オペレーティング環境のインストール時に RAID-1 ボリューム (ミラー) を作成できます。ファイルシステムをミラー化して 2 つの物理ディスクにデータを複製することにより、システムを保護できます。ミラー化されたディスクのどちらかで障害が発生しても、もう一方のディスクからシステムデータにアクセスできます。
  - ミラー化の概要情報は、第 10 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に関する要件とガイドラインについては、第 11 章を参照してください。
  - RAID-1 ボリュームを作成する JumpStart プロファイルの例については、例 26-11 を参照してください。
  - `filesys` プロファイルキーワードを使用してミラーを作成する方法については、359 ページの「`filesys` プロファイルキーワード (ミラー化されたファイルシステムの作成)」を参照してください。
  - `metadb` プロファイルキーワードを使用して状態データベースの複製を作成する方法については、365 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。
- `add_install_client` コマンドでは、PXE ネットワークブート時に Solaris x86 クライアントのブートプロパティを設定できます。`add_install_client` コマンドの `-b` オプションでは、次の作業を実行できます。
  - ネットワークインストール時に使用する代替コンソールを指定できます。
  - インストール時にネットワークブートデバイスとして使用するデバイスを指定できます。
  - 完全に自動化されたカスタム JumpStart インストールを実行するようにクライアントに指示できます。

`add_install_client` コマンドの `-b` オプションについては、次の節を参照してください。

- DVD メディアを使用するネットワークインストールの場合、128 ページの「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を追加する」を参照してください。
- CD メディアを使用するネットワークインストールの場合、161 ページの「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を追加する」を参照してください。
- Solaris Live Upgrade に関して、いくつかの新しい手順と例が追加されました。
  - Solaris Live Upgrade パッケージのインストール手順が追加されています。416 ページの「pkgadd コマンドを使用して Solaris Live Upgrade をインストールする方法」を参照してください。
  - Solaris Live Upgrade を使用してフラッシュ差分アーカイブを作成する手順が追加されています。228 ページの「Solaris Live Upgrade を使用してフラッシュ差分アーカイブを作成する方法」を参照してください。
  - 空のブート環境の作成手順と作成例が追加されています。詳しい手順については、436 ページの「フラッシュアーカイブ用の空のブート環境の作成 (コマンド行インタフェース)」を参照してください。例については、498 ページの「空のブート環境を作成してフラッシュアーカイブをインストールする例 (コマンド行インタフェース)」を参照してください。
  - RAID-1 ボリュームでブート環境を作成するコマンドを示した例が追加されています。498 ページの「既存のボリュームから Solaris ボリュームマネージャ RAID-1 ボリュームへの移行例 (コマンド行インタフェース)」を参照してください。
- DHCP と 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ブートする方法を説明した節が追加されました。PXE を使用したネットワークインストールの手順と実行例については、679 ページの「x86: 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」を参照してください。
- バグの修正がいくつか追加されました。



## 用語集

---

<b>3DES</b>	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。
<b>AES</b>	Advanced Encryption Standard の略。対称 128 ビットブロックのデータ暗号技術。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 方式を採用した。DES に代わる米国政府の標準として、AES が採用されている。
<b>begin</b> 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する。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る。
<b>bootlog-cgi</b>	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。
<b>certstore</b>	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る。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する。
<b>CGI</b>	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る。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する。
<b>DES</b>	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 として標準化された。DES では 56 ビットの鍵を使用する。
<b>DHCP</b>	動的ホスト構成プロトコルは、アプリケーション層プロトコルであり、TCP/IP ネットワーク上の個々のコンピュータつまりクライアントが、中央管理を行なっている指定の DHCP サーバーから IP アドレ

スなどのネットワーク構成情報を抽出できるようにする。この機能は、大規模な IP ネットワークの保持、管理によるオーバーヘッドを削減する。

<b>EISA</b>	拡張 ISA (Extended Industry Standard Architecture)。x86 ベースシステムで使用されるバスの 1 つ。EISA バス標準は、ISA バスシステムよりも「洗練された」規格と言える。取り付けられたデバイスが、システムに付属の「EISA コンフィギュレータ」プログラムを使用して構成されている場合、デバイスは自動的に検出される。「ISA」も参照。
<b>/etc</b>	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。
<b>/etc/netboot</b> 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。
<b>/export</b>	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る。たとえば、 <code>/export</code> 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、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る。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 <code>/export</code> ファイルシステムに依存する。
<b>fdisk</b> 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。Solaris ソフトウェアをインストールするには、x86 システム上に 1 つ以上の Solaris 9 <code>fdisk</code> パーティションを設定する必要がある。x86 ベースのシステムでは、1 台のディスクに最大 4 つの <code>fdisk</code> パーティションを作成できる。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る。各オペレーティングシステムは、独自の <code>fdisk</code> パーティション上に存在しなければならない。システムが所有できる Solaris <code>fdisk</code> パーティションの数は、1 台のディスクにつき 1 つに限られる。
<b>finish</b> 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する。このスクリプトは、カスタム <code>JumpStart</code> インストールで使用する。
<b>HMAC</b>	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用する。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る。
<b>HTTP</b>	(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 にもとづいている。
<b>HTTPS</b>	HTTP のセキュリティ保護されたバージョン。SSL (Secure Sockets Layer) を使って実装される。

**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) である。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ない。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る。

IPv6 の詳細は、『Solaris のシステム管理 (IP サービス)』の「IPv6 (概要)」を参照。

**IP アドレス** インターネットプロトコル (Internet Protocol, IP) アドレス。TCP/IP では、ネットワーク上の個々のホストを識別する 32 ビットの一意の数値。IP アドレスは、4 つの数をピリオドで区切った形式になる (例: 192.9.9.1)。通常、IP アドレスの各部は 0~225 の番号であるが、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る。

IP アドレスは論理的には、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) の 2 つの部分に分割される。たとえば、クラス A の IP アドレスは「network.local.local.local」となり、クラス C の IP アドレスは「network.network.network.local」となる。

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

**ISA** Industry Standard Architecture の略。x86 システムのバスタイプの 1 つ。ISA バスシステムは「ダム (dumb)」タイプで、システムがデバイスを自動的に検出して構成できるようなメカニズムは提供していない。「EISA」も参照。

**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 9 ソフトウェアをシステムに自動インストールできる。

**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 ディレクトリとなる。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 ディレクトリとなる。

**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。

<b>keystore</b>	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う。
<b>LAN</b>	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。
<b>LDAP</b>	Lightweight Directory Access Protocol。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。
<b>MD5</b>	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た。
<b>NIS</b>	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る。
<b>NIS+</b>	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。
<b>/opt</b>	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。
<b>OS サーバー</b>	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、各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) 用のディスク空間が必要である。
<b>RAID-0 ボリューム</b>	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る。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ある。
<b>RAID-1 ボリューム</b>	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る。RAID-1 ボリュームはミラーと呼ばれることもある。
<b>rules.ok ファイル</b>	rules ファイルから生成されたファイル。カスタム JumpStart インストールソフトウェアは、rules.ok ファイルを使ってシステムとプロファイルを照合する。check スクリプトを使用して rules.ok ファイルを作成しなくてはならない。
<b>rules ファイル</b>	自動的にインストールしたい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する。rules ファイルは、各グループをプロファイルにリンクする。Solaris 9 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。ルールファイルは、カスタム JumpStart インストールで使用される。「プロファイル」も参照。
<b>Secure Sockets Layer</b>	(SSL) クライアントとサーバーの間にセキュリティ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る。
<b>SHA1</b>	Secure Hashing Algorithm の略。このアルゴリズムは、長さが 2 <sup>64</sup> 未満の入力に対して演算を行い、メッセージダイジェストを生成する。
<b>Solaris 9 DVD または CD イメージ</b>	システムにインストールされる Solaris 9 ソフトウェア。Solaris 9 DVD や CD から、または Solaris 9 DVD 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る。
<b>Solaris Live Upgrade</b>	アクティブブート環境が稼動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、稼動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。
<b>Solaris suninstall プログラム</b>	システムを設定し、そのシステムに Solaris 9 ソフトウェアをインストールするために使用できるコマンド行インタフェース (CLI) で、メニューから選択可能な対話式のスクリプト。
<b>Solaris Web Start</b>	グラフィカルユーザーインタフェース (GUI) とコマンド行インタフェース (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 9 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る。
<b>sysidcfg</b> 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。
<b>truststore</b>	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する。
<b>URL</b>	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る。URL の形式は <i>protocol://machine:port/document</i> 。  たとえば、 <a href="http://www.example.com/index.html">http://www.example.com/index.html</a> は URL の一例。
<b>/usr</b>	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る。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 9 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る。
<b>/var</b>	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 ファイルなどがある。

<b>WAN</b>	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。
<b>wanboot-cgi</b> 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。
<b>wanboot.conf</b> ファイル	WAN ブートインストールに必要な構成情報とセキュリティ設定値を指定するテキストファイル。
<b>wanboot</b> プログラム	WAN ブートインストールの実行に必要な、WAN ブートミニルート、クライアント構成ファイル、およびインストールファイルを読み込む、二次レベルのブートプログラム。WAN ブートインストールでは、wanboot バイナリが、ufsboot または inetboot 二次ブートプログラムと同様の処理を実行する。
<b>WAN</b> 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、暗号化したフラッシュアーカイブをパブリックネットワーク上で転送し、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る。
<b>WAN</b> ブートサーバー	WAN ブートインストールで使用される構成ファイルとセキュリティファイルを提供する Web サーバー。
<b>WAN</b> 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る。  「ミニルート」も参照
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る。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る。  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。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る。差分更新により、指定されたファイルだけが更新される。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る。
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 オペレーティング環境のアップグレードでは、Solaris オペレーティング環境の新しいバージョンがシステムのディスク上の既存のファイルにマージされる。アップグレードでは、既存の Solaris オペレーティング環境に対して行なった変更は可能な限り保存される。

アップグレードオプション	Solaris Web Start プログラムと Solaris suninstall プログラムによって提示されるオプション。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る。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、できる限り残される。
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る。  「復号化」も参照
インストールサーバー	インストール用に、Solaris 9 DVD または CD イメージをネットワーク上の他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 9 DVD 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る。
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る。
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
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 9 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る。カスタム JumpStart インストールは、ユーザーが作成する JumpStart インストールである。
カスタムプロープファイル	rules ファイルと同じ JumpStart ディレクトリに存在しなければならないファイルで、プロープと比較という 2 つのタイプの関数を含む Bourne シェルスクリプト。プロープ関数は作業を実行して、必要な情報を収集したり、定義に対応した SI_ 環境変数を設定する。プロープ関数は、プロープキーワードになる。比較関数は、対応するプロープ関数を呼び出してプロープ関数の出力を比較し、キーワードが一致する場合は 0、キーワードが一致しない場合は 1 を返す。比較関数はルールキーワードになる。「rules ファイル」も参照。
共有可能ファイルシステム	/export/home や /swap のようなユーザー定義のファイルシステム。Solaris Live Upgrade の使用時に、アクティブブート環境と非アクティブブート環境によって共有される。共有可能ファイルシステムは、アクティブブート環境の vfstab 内と非アクティブブート環境の

	<p>vfstab 内に同じマウントポイントを持つ。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る。共有可能ファイルシステムはデフォルトで共有されるが、ユーザーが宛先スライスを指定することもできる。この場合、そのファイルシステムがコピーされる。</p>
クライアント	<p>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する。</p>
クラスタ	<p>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 9 ソフトウェアは 4 つの「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」から構成されている。</p>
クリティカルファイルシステム	<p>Solaris オペレーティング環境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 vfstab では独立したマウントポイントになる。root (/)、/usr、/var、/opt などがクリティカルファイルシステムの例である。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る。</p>
クローンシステム	<p>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る。</p>
コアソフトウェアグループ	<p>システムで Solaris オペレーティング環境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。これ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るが、CDE ソフトウェアは含まれない。</p>
公開鍵	<p>公開鍵暗号方式で使用される暗号化鍵。</p>
公開鍵暗号化	<p>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。</p>
更新	<p>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る。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らない。</p>
コマンド行	<p>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) 文字で終わる。</p>
サーバー	<p>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。</p>
サブネット	<p>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。</p>



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。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する。
サブミラー	「RAID-0 ボリューム」を参照。
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る。差分更新により、指定されたファイルだけが更新される。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る。
時間帯	グリニッジ標準時間を基準に地球の表面を 24 の地域に経度分割したものの。
システム構成ファイル	(system.conf) WAN ブートインストールで使用する sysidcfg ファイルおよびカスタム JumpStart ファイルの場所を指定するテキストファイル。
状態データベース	状態データベースでは、Solaris ボリュームマネージャ構成の状態に関する情報がディスクに保存される。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。各コピーは「状態データベースの複製」と呼ばれる。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る。
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る。
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 オペレーティング環境の初期インストールでは、システムのディスクが Solaris オペレーティング環境の新しいバージョンで上書きされる。システム上で Solaris オペレーティング環境が稼動していない場合は、初期インストールを行う必要がある。
初期インストールオプション	Solaris Web Start プログラムおよび Solaris suninstall プログラムの実行時に提示される、新バージョンの Solaris でディスクを上書きするオプション。初期インストールオプションはアップグレード可能なシステムで表示される。初期インストールオプションを選択した場合、旧バージョンの Solaris ソフトウェア (ローカルの変更内容も含めて) が含まれるディスクでも上書きされる。
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。

スワップ領域	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いう。
全体ディストリビューション	Solaris 9 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 9 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する。
ソフトウェアグループ	Solaris ソフトウェアの論理グループ (クラスとパッケージ)。Solaris のインストール時には、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をインストールできる。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る。
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る。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る。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る。
ディスク (disc)	磁気ディスク (disk) に対する光学式ディスク。CD (コンパクトディスク) 業界では共通の綴りを使用する。たとえば、CD-ROM や DVD-ROM は光学式ディスク。
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る。「ディスク (disc)」も参照。
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进行测试できる。
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	システムの状態を自動的に保存し、30分間アイドル状態が続くと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る。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る。

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る。

動的プロファイル	カスタム JumpStart インストール時に、begin スクリプトによって動的に作成されるプロファイル。
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する。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。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る (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 (リモートな) 管理権限領域を表す。
認証局	(CA) 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する。
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。
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う。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、または取得できる。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らない。Sun では LDAP、NIS、および NIS+ のネームサービスをサポートしている。
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、または他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。ローカルエリアネットワーク (LAN) とも呼ばれる。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。

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る。
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 9 ソフトウェアは 4 つの「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る。
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る。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る。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する。1 方向のハッシュ関数の一例としては、MD5 および SHA-1 がある。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、小さい値に変換する。
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	手作業でも、Solaris suninstall プログラムや Solaris Web Start プログラム内でも実行できるスクリプト。パッチアナライザは、システムを解析し、Solaris 9 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する。
パネル	ウィンドウ、ダイアログボックス、アプレットの内容を編成するコンテナ。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る。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る。
非公開鍵	公開鍵暗号方式で使用される復号化鍵。
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では、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る。
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。
ブート環境	Solaris オペレーティング環境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る。  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す。単一のアクティブなブート環境からだけブートできる。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、次回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。
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う。
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
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る。「アーカイブ」も参照。
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。
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。
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ない。「ルール」も参照。
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する。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する。通常は、ルールごとに異なるプロファイルを作成するが、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。「rules ファイル」も参照。
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らない。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るが、先頭と末尾にマイナス符号は使用できない。

ボリューム	<p>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する。</p> <p>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る。標準の UNIX 用語では、擬似デバイスまたは仮想デバイスとも呼ばれる。</p>
ボリュームマネージャ	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。
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	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る。
マニフェストセクション	クローンシステムの検証に使用されるフラッシュアーカイブのセクション。マニフェストセクションには、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が一覧表示される。このセクションは、情報提供専用。このセクションは、ファイル一覧を内部形式で保存するため、スクリプトの記述には使用できない。
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る。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。
ミラー	「RAID-1 ボリューム」を参照。
メタデバイス	「ボリューム」を参照。
メディアサーバー	「インストールサーバー」を参照。
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において、ほかのすべての項目を子孫として持つ唯一の項目を指す。階層内において、ルート項目より上位の項目は存在しない。/ は、直接的または間接的にほかのすべてのディレクトリの元となるベースディレクトリ。ルートディレクトリには、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る。

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る。
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る。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る。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ない。「RAID-0 ボリューム」も参照。
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。
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ではボリュームと呼ばれる。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する。





# 索引

---

## 数字・記号

#

rules ファイル内の, 271  
プロファイル内の, 275

## 3DES 暗号化鍵

wanbootutil コマンドによる作成, 548  
WAN ブートインストールのデータの暗号化, 516, 548

## インストール

ickey コマンドによる, 572  
set-security-key コマンドによる, 570  
wanboot プログラムによる, 579  
クライアントの OBP に, 570  
動作中のクライアントに, 572

## A

add\_install\_client, 説明, 167

## add\_install\_client コマンド

JumpStart ディレクトリのアクセス, 264  
インストールサーバーの設定  
CD メディアを使用する, 162  
DVD メディアを使用する, 129  
クロスプラットフォーム CD メディアの例, 165  
シリアルコンソールの指定例, 132, 165, 687  
ブートデバイスの指定例, 133, 166, 688

## 例

DHCP を使用する (CD メディア), 164, 165, 687

add\_install\_client コマンド, 例 (続き)

DHCP を使用する (DVD メディア), 131, 132  
同じサブネット (CD メディア), 164  
シリアルコンソールの指定, 132, 165, 687  
ブートサーバー (DVD メディア), 132  
ブートサーバーの (CD メディア), 165  
ブートデバイスの指定, 133, 166, 688

add\_install\_client コマンドの -s オプション, 333

add\_to\_install\_server, 説明, 167

## AES 暗号化鍵

wanbootutil コマンドによる作成, 548  
WAN ブートインストールのデータの暗号化, 516, 548

## インストール

ickey コマンドによる, 572  
set-security-key コマンドによる, 570  
wanboot プログラムによる, 579  
クライアントの OBP に, 570  
動作中のクライアントに, 572

AND ルールフィールド, 272

## any

プローブキーワード  
説明と値, 374  
ルールキーワード, 335, 374

## archive

インストール  
空のブート環境に, 499

archive\_location キーワード, 342

arch プローブキーワード, 374

arch ルールキーワード, 335, 374

auto\_install\_sample ディレクトリ  
  check スクリプト, 289, 311  
  JumpStart ディレクトリへのファイルのコ  
    ピー, 267  
  JumpStart ディレクトリへファイルをコピー  
    , 269  
  set\_root\_pw finish スクリプト, 297  
  ファイルを JumpStart ディレクトリにコピー  
    , 263

## B

backup\_media キーワード, 91, 347  
banner コマンド, 168  
begin.log ファイル, 292  
begin スクリプト  
  アクセス権, 292  
  概要, 291  
  サイト固有のインストールプログラム, 306  
  動的プロファイルの作成, 293  
  動的プロファイルを作成する, 292  
  ルールフィールド, 272  
begin ルールフィールドの説明, 272  
boot: cannot open /kernel/unix メッセー  
  ジ, 645  
boot\_device キーワード, 349  
boot\_file パラメータ, 610  
boot\_logger パラメータ, 612  
bootconfchk コマンド, 構文, 606  
bootenv createbe キーワード, 350  
bootlog-cgi プログラム  
  wanboot.conf ファイルに指定, 612  
  WAN ブートインストールでの使用, 544  
bootlog ファイル, ログサーバーの場所の指  
  示, 544  
bootparams ファイル  
  JumpStart ディレクトリへのアクセスを有効  
    にする, 264  
  更新, 651  
bootserver 変数, 579  
Bourne シェルスクリプト、ルールフィールド  
  内の, 272  
-b オプション, add\_install\_client コマンド, 686

## C

cannot open /kernel/unix メッセージ, 645  
Can't boot from file/device メッセージ, 645  
certstore ファイル  
  クライアントの証明書の挿入, 546, 594  
  説明, 525  
CHANGE DEFAULT BOOT DEVICE メッセー  
  ジ, 652  
check スクリプト  
  custom\_probes.ok ファイルの作成, 311  
  custom\_probes ファイルの妥当性検  
    査, 311, 312  
  rules.ok ファイルの作成, 289  
  rules の妥当性検査, 289  
  rules のテスト, 555  
  rules ファイルの妥当性検査, 312  
  rules ファイルの妥当性検査, 288, 289  
  WAN ブートインストール用の構文, 606  
  動的プロファイルと, 293  
  ルールの検査, 312  
check スクリプトの -p オプション, 289, 555  
check スクリプトの -r オプション, 312  
client\_arch キーワード, 351  
client\_authentication パラメータ, 612  
client\_name、説明, 164, 687  
client\_root プロファイルキーワード, 351  
CLIENT MAC ADDR エラーメッセージ, 650  
clock gained xxx days メッセージ, 645  
cluster プロファイルキーワード  
  説明と値, 352  
  例, 276  
CPU (プロセッサ)  
  WAN ブートインストールの要件, 520  
  プローブキーワード, 374  
  ルールキーワード, 335, 374  
.cshrc ファイル, 297  
custom\_probes.ok ファイル  
  作成, 311  
  説明, 311  
custom\_probes ファイル  
  check による妥当性検査, 311  
  check を使用する妥当性検査, 312  
  custom\_probes のテスト, 312  
custom\_probes ファイル  
  名前の指定, 308  
  要件, 308

-c オプション  
  add\_install\_client コマンド, 163, 332, 333, 686  
  pfinstall コマンド, 287

## D

dd コマンド, 268  
devalias コマンド  
  WAN ブート用のデバイス別名の確認, 568  
  WAN ブート用のデバイス別名の設定, 568  
  構文, 609  
Developer Solaris Software Group, プロファイルの例, 276  
dfstab ファイル, 262, 329  
DHCP (Dynamic Host Configuration Protocol) の事前設定, 64  
DHCP サービス  
  Solaris インストール用オプションの作成, 76  
  Solaris インストール用のマクロの作成, 80  
  Solaris ネットワークの起動とインストール, 75  
  WAN ブートインストールの要件, 520  
  WAN ブートインストール用に構成, 563  
  WAN ブートインストール用の Sun ベンダーオプション, 563  
  オプションとマクロを追加するスクリプト例, 81  
  説明, 75  
DHCP マネージャ  
  インストールオプションの作成, 83  
  インストールマクロの作成, 84  
dhtadm コマンド, スクリプトに使用, 81  
disksize ルールキーワード  
  説明と値, 336, 374  
disks プロープキーワード, 説明と値, 374  
domainname プロープキーワード, 374  
domainname ルールキーワード, 336, 374  
dontuse プロファイルキーワード, 354, 370  
-d オプション  
  add\_install\_client コマンド, 162, 685

## E

eeeprom コマンド, OBP の WAN ブートインストール対応の確認, 606

encryption\_type パラメータ, 611  
eng\_profile の例, 329  
/etc/bootparams ファイル  
  JumpStart ディレクトリへのアクセスを有効にする, 264, 651  
/etc/dfs/dfstab ファイル, 262, 329  
/etc/locale ファイル, 73  
/etc/mnttab ファイル, 266  
/etc/netboot ディレクトリ  
  アクセス権, 540  
  クライアント間での構成ファイルとセキュリティファイルの共有, 524, 525  
  クライアント認証とサーバー認証の構成, 546, 548, 594  
  構成ファイルとセキュリティファイル、説明, 525  
  構成ファイルとセキュリティファイルの保存サブネット全体に対するインストール, 524, 541  
  単一のクライアントに対するインストール, 524, 541  
  ネットワーク全体に対するインストール, 524, 541  
  作成, 540, 592  
  説明, 524  
  挿入  
    暗号化鍵, 548  
    クライアントの非公開鍵, 546, 594  
    信頼できる証明書, 546, 594  
    デジタル証明書, 546, 594  
    ハッシュキー, 548  
  例, 526  
existing partitioning の値, 368  
explicit partitioning の値, 368  
/export ファイルシステム, 44  
-e オプション, add\_install\_client コマンド, 686

## F

fdformat コマンド, 266, 269, 678  
fdisk コマンド, 303  
fdisk パーティション, 要件, 47  
fdisk プロファイルキーワード  
  説明と値, 354  
  例, 276  
fileys キーワード, 357, 359

filesys プロファイルキーワード  
説明と値, 356  
例, 276  
file 変数, 576  
finish.log ファイル, 293  
finish スクリプト  
システムの root パスワードを設定する, 297  
パッケージとパッチの追加, 294  
ファイルの追加, 294  
ルート環境のカスタマイズ, 297  
ルールフィールド, 272  
finish ルールフィールドの説明, 272  
flar create コマンド, 246  
WAN ブートインストール用の構文, 606

## G

geo キーワード, 360  
getfile: RPC failed: error 5: RPC Timed out  
メッセージ, 265

## H

HMAC SHA1 ハッシュキー, 「ハッシュキー」  
を参照  
host-ip 変数, 576  
hostaddress プロープキーワード, 374  
hostaddress ルールキーワード, 336, 374  
hostname プロープキーワード, 説明と値, 374  
hostname 変数, 576  
hostname ルールキーワード  
説明と値, 336, 374  
例, 335  
HTTP over Secure Sockets Layer, 「HTTPS」を  
参照  
http-proxy 変数, 576  
HTTPS  
WAN ブートインストール時のデータの保  
護, 516  
WAN ブートで使用するための要件, 545  
説明, 516  
HTTPS によるデータの暗号化, WAN ブートイ  
ンストール, 516

## I

ickey コマンド  
WAN ブートクライアントに対するキーのイ  
ンストール, 572  
構文, 606  
IDE インタフェース  
表面検査, 652  
不良ブロックのマッピング, 652  
install\_config コマンド, 264, 265  
install\_type キーワード, 361  
install\_type プロファイルキーワード  
プロファイルのテスト, 285, 288  
要件, 274, 276  
例, 276  
installed プロープキーワード, 説明と値, 374  
installed ルールキーワード  
説明と値, 337, 374  
IPv6 の事前設定, 64  
IP アドレス  
事前設定, 64  
デフォルトルートの事前設定, 64  
デフォルトルートの指定, 54, 61  
プロープキーワード, 374  
ルールキーワード, 336, 374  
IRQ レベルの事前設定, 65  
isa\_bits キーワード, 362

## J

JumpStart ディレクトリ  
finish スクリプトでファイルを追加, 294  
finish スクリプトによるファイルの追加, 294  
rules ファイルの例, 271  
アクセス権, 262, 265  
共有, 262, 329  
作成, 262, 265, 268, 329  
x86 ベースシステム用のフロッピーディス  
ク, 678  
フロッピーディスク SPARC ベースシステ  
ム用, 266  
ファイルのコピー  
finish スクリプトを使って, 294  
インストールファイル, 263, 267  
ファイルをコピー  
インストールファイル, 269  
JumpStart ディレクトリの共有, 329

## K

karch プロンプキーワード, 374  
karch ルールキーワード, 337, 374  
Kerberos  
構成情報, 53  
事前設定, 65  
keystore ファイル  
クライアントの非公開鍵の挿入, 594  
説明, 525  
挿入  
暗号化鍵, 548  
クライアントの非公開鍵, 546  
ハッシュキー, 548

## L

layout\_constraint キーワード, 91, 362  
LBA, 48  
le0: No carrier - transceiver cable problem  
メッセージ, 645  
list-security-keys コマンド  
クライアントにインストールされているキー  
の一覧表示, 570  
構文, 608  
Live Upgrade, 「Solaris Live Upgrade」を参照  
Live Upgrade ブート環境を比較する, 481  
locale.org\_dir テーブルエントリの追加, 74  
locale キーワード, 365

## M

Makefile ファイル, 72  
marketing\_profile の例, 330  
memsize プロンプキーワード, 説明と値, 374  
memsize ルールキーワード  
説明と値, 337, 374  
metadb プロファイルキーワード, 365  
mnttab ファイル, 266  
model プロンプキーワード, 説明と値, 374  
model ルールキーワード  
説明と値, 338, 374  
modify\_install\_server, 説明, 167  
mount コマンド, 168

## N

names/naming, ホスト名, 687  
netmask の事前設定, 64  
network-boot-arguments OBP 変数  
WAN ブートインストールでの設定, 576, 578  
構文, 609  
network プロンプキーワード, 説明と値, 374  
network ルールキーワード  
説明と値, 338, 374  
net デバイス別名  
確認と再設定, 568, 569, 601  
nistbladm コマンド, 74  
no\_master\_check キーワード, 366  
No carrier - transceiver cable problem メッセージ, 645  
Not a UFS filesystem メッセージ, 645  
num\_clients プロファイルキーワード, 366  
nvalias コマンド  
WAN ブート用のデバイス別名の設定, 568  
構文, 609

## O

OBP  
net デバイス別名の確認, 569, 601  
net デバイス別名の設定, 569  
WAN ブートインストールでの変数の設定, 576, 578  
WAN ブートインストールの要件, 520  
WAN ブート対応の確認, 567, 600  
デバイス別名設定の確認, 568  
OpenBoot PROM, 「OBP」を参照  
osname プロンプキーワード, 374  
osname ルールキーワード, 338, 374

## P

package プロファイルキーワード, 説明と値, 367  
partitioning  
fdisk パーティション, 354  
fdisk パーティション, 276  
使用しないディスク, 354  
プロファイルキーワード, 367, 370  
ルートスライスの位置, 646  
例, 276

partitioning キーワード, 367  
pfinstall コマンド, 90, 284  
PKCS#12 ファイル  
    WAN ブートインストールの準備, 546, 594  
    WAN ブートインストールの要件, 527  
Preboot Execution Environment (PXE), 33  
    BIOS 設定要件, 689  
    ガイドライン, 679  
    作業, 680  
    説明, 679  
    例  
        インストールクライアントの追加, 687, 688  
        インストールサーバーの作成, 684  
printenv コマンド  
    WAN ブート対応の確認, 567, 600  
probe ルールキーワード, 説明と値, 339  
prtvtoc コマンド  
    x86: ディスク構成ファイルの作成, 303  
    SPARC: ディスク構成ファイルの作成, 301  
Public-Key Cryptography Standards #12 ファイル, 「PKCS#12 ファイル」を参照  
PXE (Preboot Execution Environment), 33  
    BIOS 設定要件, 689  
    ガイドライン, 679  
    作業, 680  
    説明, 679  
    例  
        インストールクライアントの追加, 687, 688  
        インストールサーバーの作成, 684  
-p オプション, check スクリプトの, 311

## R

RAID-0 ボリューム  
    「サブミラー」も参照  
    概念的な概要, 100  
    定義, 100  
RAID-0 ボリューム, 説明, 386  
RAID-1 ボリューム  
    概念的な概要, 101  
    定義, 101  
RAID-1 ボリューム (ミラー), Solaris Live Upgrade  
    description, 384

RAID-1 ボリューム (ミラー), Solaris Live Upgrade (続き)  
    Solaris ボリュームマネージャボリュームへの移行例, 498  
    作成, 438  
    作成およびアップグレードの例, 494  
    作成の例, 441, 442, 443  
    説明, 386  
    要件, 399, 404  
reset コマンド, 168  
resolve\_hosts パラメータ, 612  
root\_device キーワード, 368  
root\_file パラメータ, 611  
root\_server パラメータ, 610  
rootdisk  
    filesystem のスライス値, 357  
    JumpStart で設定される値, 369  
    定義, 369  
root パスワード, 事前設定, 65  
root パスワードを finish スクリプトで設定する, 297  
router-ip 変数, 576  
RPC failed: error 5: RPC Timed out メッセージ, 265  
RPC Timed out メッセージ, 265, 650  
rule\_keyword ルールフィールド, 272  
rule\_value ルールフィールド, 272  
rules  
    WAN ブートインストール用の妥当性検査, 555  
    フィールドの説明, 272  
rules.ok ファイル  
    作成, 288, 289  
    説明, 288  
    ルールの照合順序, 273, 313, 319  
rules ファイル  
    check による妥当性検査, 289  
    check を使用する妥当性検査  
        動的プロファイルと, 293  
    check を使った妥当性検査  
        カスタム JumpStart の例, 331  
    rules の妥当性検査, 289  
    WAN ブートインストール用の妥当性検査, 555  
    WAN ブートでの, 554  
    カスタム JumpStart の例, 330, 331  
    構文, 271

- rules ファイル
  - コメント, 271
- rules ファイル
  - 作成, 270
  - 説明, 270
- rules ファイル
  - 名前の指定, 271
- rules ファイル
  - 名前の指定, 271
- rules ファイル
  - 複数行のルール, 271
- rules ファイル
  - ルールの追加, 271
  - 例, 271
- rules ファイル内で新しい行に続ける, 271
- r オプション、check スクリプトの, 289

## S

- SbootURI DHCP オプション
  - WAN ブートインストールでの使用, 563
  - 説明, 79
- secure HTTP, 「HTTPS」を参照
- Secure Sockets Layer, 「SSL」を参照
- server\_authentication パラメータ, 611
- Service パーティション, インストール時に保持 (x86 システム), 48
- set\_root\_pw finish スクリプト, 297
- set-security-key コマンド
  - WAN ブートクライアントに対するキーのインストール, 570, 601
  - 構文, 608
- setenv コマンド
  - 構文, 609
  - ネットワークブート情報の設定, 576, 578
- setup\_install\_server
  - WAN ブートインストール用, 536
  - WAN ブートインストール用の構文, 605
  - 説明, 167
- setup\_install\_server コマンドの -b オプション, 328
- shareall コマンド, 262, 329
- share コマンド, JumpStart ディレクトリの共有, 329
- showmount コマンド, 168
- SHTTProxy DHCP オプション
  - WAN ブートインストールでの使用, 563

- SHTTProxy DHCP オプション (続き)
  - 説明, 79
- SI\_CONFIG\_DIR 変数, 294
- SI\_PROFILE 環境変数, 293
- signature\_type パラメータ, 611
- size
  - fdisk パーティション, 355
  - ローカルファイルシステム, 357
- SjumpsCF パラメータ, 557, 609
- slices
  - fileys の値, 357
  - プローブキーワード, 374
  - ルールキーワード, 337
- Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版), 268
  - アクセス, 33
- Solaris Live Upgrade
  - RAID-1 ボリューム (ミラー) 用スライスの選択, 404
  - アップグレード
    - 作業マップ, 448
    - ブート環境, 448
  - アップグレード障害の回復, 468
  - インストール
    - パッケージ, 415
    - フラッシュアーカイブ, 459
    - フラッシュアーカイブでプロファイルを使用, 462, 463
  - キーワード
    - プロファイル, 454, 455
  - 起動, 417
  - コピー
    - 共有可能なファイルシステム, 433
  - コマンド, 507
  - 作成
    - RAID-1 ボリューム (ミラー)、作業, 438
    - RAID-1 ボリューム (ミラー)、説明, 384
    - RAID-1 ボリューム (ミラー)、要件, 399
    - 作業マップ, 415
    - 内容のカスタマイズ、作業, 444
    - ブート環境、作業, 418
    - ブート環境、説明, 380
    - フラッシュ差分アーカイブ, 228
    - プロファイル, 453
  - 処理を取り消す, 480
  - 説明, 380
  - 停止, 417
  - ディスク容量の要件, 399

- Solaris Live Upgrade (続き)
  - 内容のカスタマイズ, 408
  - 必要なパッケージ, 400
  - 表示
    - 画面をリモートシステムから, 411
    - ブート環境の構成, 488
  - ファイルシステムの構成, 420
  - ファイルシステムのスライス, 420
  - ファイルへ出力, 420
  - ブート環境のアクティブ化, 464
  - ブート環境の削除, 482
  - ブート環境の名前の表示, 483
  - ブート環境の名前の変更, 484
  - ブート環境を比較する, 481
  - プロファイル、差分アーカイブ用の例, 456
  - プロファイル、テスト, 457
  - プロファイル、例, 456
  - ボリュームのキーワード, 440
  - 例, 491
    - RAID-1 ボリュームのアップグレード, 494, 498
    - RAID-1 ボリュームの作成, 441, 442
    - 空のブート環境の作成, 499
    - 完全なプロセス, 491
    - 内容のカスタマイズ, 446
    - フラッシュ差分アーカイブ, 229
    - ミラーの作成, 442, 443
- Solaris suninstall プログラム
  - ほかのインストール方法との比較, 36
- Solaris Web Start
  - インストールプログラム, 169
  - グラフィカルユーザーインタフェース (GUI), 169
  - コマンド行インタフェース (CLI), 170
  - 実行方法, 169
  - ほかのインストール方法との比較, 35
  - 要件
    - x86 システム上の LBA, 48
    - スワップサイズ, 47
- Solaris 9 ソフトウェア
  - グループ, 352, 353
  - アップグレード, 353
  - パッケージの指定, 367
  - プロファイルの例, 276
  - リリースまたはバージョン, 337, 374
    - installed プロンプトキーワード, 374
- Solaris 9 ソフトウェアのバージョン, installed ルールキーワード, 337
- Solaris 9 ソフトウェアのリリース
  - installed プロンプトキーワード, 374
  - installed ルールキーワード, 337, 374
  - osname プロンプトキーワード, 374
  - osname ルールキーワード, 338, 374
- Solaris Live Upgrade 処理を取り消す, 480
- Solaris Live Upgrade のコマンド, 507
- Solaris Live Upgrade ブート環境の名前を変更する, 484
- Solaris suninstall プログラム
  - 実行方法, 185
- Solaris ソフトウェア
  - リリースまたはバージョン, 338, 374
- Solaris ソフトウェアのバージョン
  - installed プロンプトキーワード, 374
  - installed ルールキーワード, 374
  - osname プロンプトキーワード, 374
  - osname ルールキーワード, 338, 374
- Solaris フラッシュアーカイブ、「アーカイブ」を参照
- Solaris ボリュームマネージャ
  - Solaris Live Upgrade で使用されるコマンド, 405
  - Solaris Live Upgrade の例
    - RAID-1 ボリュームの切り離しとアップグレード, 494
    - RAID-1 ボリュームへの移行, 498
    - カスタム JumpStart の実行時にボリュームを作成
      - example, 276
- SSL, WAN ブートインストールでの使用, 545
- SSL 非公開鍵, 「非公開鍵」を参照
- SsysidCF パラメータ, 557, 609
- stty コマンド, 51, 57, 316, 320
- subnet-mask 変数, 576
- SUNWCall グループ, 352
- SUNWCprog グループ, 352
- SUNWCreq グループ, 352
- SUNWCuser グループ, 352
- SUNWCXall グループ, 352
- swap ファイルシステム
  - サイズの決定, 352
  - メモリーのサイズと, 352
- sysidcfg ファイル
  - WAN ブート
    - 作成, 552
    - 例, 552
- sysidcfg ファイル, ガイドラインと要件, 65



sysidcfg ファイル  
    キーワード, 67  
    構文, 66  
    作成方法, 70  
system\_conf パラメータ, 613  
system.conf ファイル, 「システム構成ファイル」を参照  
system\_type プロファイルキーワード  
    説明と値, 370  
    例, 276

## T

tftpd デーモン, 651  
timed out RPC エラー, 650  
tip ライン接続とカスタム JumpStart インストール, 316, 320  
totaldisk プロブキーワード, 375  
totaldisk ルールキーワード, 340, 375  
transceiver cable problem メッセージ, 645  
Triple DES 暗号化鍵, 「3DES 暗号化鍵」を参照  
truststore ファイル  
    信頼できる証明書の挿入, 546, 594  
    説明, 525

## U

UFS, 266  
uname コマンド, 168  
Unknown client error メッセージ, 643  
upgrade\_log file, 175  
upgrade\_log ファイル, 188, 189  
usedisk プロファイルキーワード, 説明と値, 370  
/usr/sbin/rpld コマンド, 652

## V

/var/sadm/install\_data/upgrade\_log ファイル, 188, 189, 195  
/var/sadm/system/logs/begin.log ファイル, 292  
/var/sadm/system/logs/finish.log ファイル, 293  
/var/yp/Makefile, 72

/var/yp/make コマンド, 74  
volcheck コマンド, 269  
volcheck コマンド, 266, 268

## W

wanboot-cgi プログラム  
    /etc/netboot ディレクトリの検索順序, 525  
    wanboot.conf ファイルに指定, 610  
    WAN ブートサーバーへのコピー, 543, 593  
    クライアント構成情報の選択, 525  
    説明, 524  
    保存, 527  
wanboot.conf ファイル  
    WAN ブートインストールに必要な情報, 559  
    WAN ブートインストール用に作成, 559, 610  
    WAN ブートインストール用の妥当性検査, 559, 560, 599  
    構文, 610  
    説明, 525, 610  
    例  
        セキュリティ保護された WAN ブートインストール, 560, 599  
        セキュリティ保護されていない WAN ブートインストール, 562  
wanbootutil コマンド  
    PKCS#12 ファイルの分割, 546, 594  
    暗号化鍵の値の表示, 570, 578, 601  
    暗号化鍵の作成, 548, 595  
    クライアント認証とサーバー認証の構成, 546, 548, 594, 595  
    クライアントのデジタル証明書の挿入, 546, 594  
    クライアントの非公開鍵の挿入, 546, 594  
    構文, 606  
    信頼できる証明書の挿入, 546, 594  
    ハッシュキーの値の表示, 570, 578, 601  
    ハッシュキーの作成, 548, 595  
wanboot プログラム  
    wanboot.conf ファイルに指定, 610  
    WAN ブートインストール時に実行される作業, 515  
    WAN ブートインストール用のキーのインストール, 579

- wanboot プログラム (続き)
  - WAN ブートサーバーへのインストール, 539, 592
  - 説明, 512
  - ドキュメントルートディレクトリへの保存, 523
- WAN ブートインストール
  - bootlog-cgi プログラム, 544
  - wanboot.conf ファイルに指定, 612
  - DHCP サポートによる, 581
  - /etc/netboot ディレクトリ
    - アクセス権の設定, 541
    - 作成, 540
    - 説明, 524
    - 例, 526
  - ickey コマンド, 572
  - OBP での対応, 567
  - OBP のデバイス別名, 568
  - OBP のデバイス別名の確認, 568
  - rules ファイルのチェック, 555
  - set-security-key コマンド, 570
  - wanboot-cgi プログラム, 543
    - wanboot.conf ファイルに指定, 610
    - WAN ブートサーバーへのコピー, 543
  - wanboot-cgi プログラムのコピー, 543
  - wanboot-cgi プログラムの保存, 527
  - wanboot.conf ファイル
    - 構文, 610
    - 作成, 559
    - 妥当性検査, 560
    - パラメータ, 610
  - wanbootutil コマンド
    - 暗号化鍵の値の表示, 570, 572
    - 暗号化鍵の作成, 548, 595
    - 信頼できる証明書の作成, 546
    - ハッシュキーの値の表示, 570, 572
    - ハッシュキーの作成, 548, 595
    - 非公開鍵の作成, 546
  - wanboot プログラム
    - wanboot.conf ファイルに指定, 610
    - インストール, 539
    - 説明, 512
    - ドキュメントルートディレクトリへの保存, 523
  - wanboot プログラムのインストール, 539
  - WAN ブートミニルート
    - wanboot.conf ファイルに指定, 611
    - 作成, 536
- WAN ブートインストール, WAN ブートミニルート (続き)
  - 説明, 512
  - ドキュメントルートディレクトリへの保存, 523
  - Web サーバーの要件, 521
- 暗号化鍵
  - wanboot.conf ファイルに指定, 611
  - 値の表示, 569
  - インストール, 569
  - 作成, 548
  - 暗号化鍵のインストール, 569
    - ickey コマンドによる, 572
    - set-security-key コマンドによる, 570
    - クライアントの OBP に, 570
    - 動作中のクライアントに, 572
  - 暗号化鍵の機密性の問題, 528
  - イベントの順序, 513
  - インストールに必要な情報, 528
  - クライアント OBP での対応の確認, 567
  - クライアント認証
    - wanboot.conf ファイルに指定, 612
    - 構成, 546
    - 要件, 517
  - クライアントのインストール
    - インストール方法, 574
    - 必要な作業, 565
  - クライアントの要件, 520
  - 計画
    - /etc/netboot ディレクトリ, 524
    - インストールファイルの保存, 522
    - 構成ファイルとセキュリティファイルの共有, 524
    - 構成ファイルとセキュリティファイルの保存, 524
    - サーバーレイアウト, 521
    - システム要件, 519
    - ドキュメントルートディレクトリ, 522
  - 構成
    - DHCP サービスのサポート, 563
    - WAN ブートサーバー, 535
    - クライアント認証とサーバー認証, 546, 548, 594
    - ログサーバー, 544
  - 構成ファイルとセキュリティファイル, 説明, 525

- WAN ブートインストール (続き)
  - 構成ファイルとセキュリティファイルの共有
    - サブネット全体, 524, 541
    - 特定のクライアント, 524, 541
    - ネットワーク全体, 524, 541
  - コマンド, 605
  - サーバー構成、説明, 521
  - サーバー認証
    - wanboot.conf ファイルに指定, 611
    - 構成, 546
    - 要件, 517
  - サービス妨害攻撃, 528
  - 作成
    - begin スクリプト, 556
    - finish スクリプト, 556
    - rules ファイル, 554
    - sysidcfg ファイル, 552
    - wanboot.conf ファイル, 559
    - 暗号化鍵, 548
    - システム構成ファイル, 557
    - ハッシュキー, 548
    - フラッシュアーカイブ, 551
    - プロファイル, 553
  - システム構成ファイル
    - wanboot.conf ファイルに指定, 613
    - 構文, 609
    - 作成, 557
  - システム要件, 519
  - 自動インストール, 575, 602
  - 使用する場合, 512
  - セキュリティ構成、説明, 517
  - セキュリティの問題, 528
  - セキュリティ保護された構成
    - インストール作業, 532
    - 説明, 517
    - 要件, 517
  - セキュリティ保護されていない構成
    - インストール作業, 534
    - 説明, 518
    - 要件, 518
  - 説明, 511
  - 対話式インストール, 577
  - データの暗号化
    - HTTPS による, 516, 545
    - 暗号化鍵による, 516
  - データの保護
    - 暗号化鍵による, 516
    - デジタル証明書による, 516
- WAN ブートインストール, データの保護 (続き)
  - ハッシュキーによる, 516
  - デジタル証明書
    - 書式, 546
    - 挿入, 546
    - 分割, 546
    - 要件, 527
  - デジタル証明書の準備, 546
  - ドキュメントルートディレクトリ
    - 説明, 522
    - ファイル, 522
    - 例, 522
  - バイナリの破壊, 528
  - ハッシュキー
    - wanboot.conf ファイルに指定, 611
    - 値の表示, 569
    - インストール, 569
    - 作成, 548
  - ハッシュキーのインストール, 569
    - ickey コマンドによる, 572
    - set-security-key コマンドによる, 570
    - クライアントの OBP に, 570
    - 動作中のクライアントに, 572
  - ハッシュキーの機密性の問題, 528
  - 非公開鍵
    - keystore ファイルへの挿入, 546
    - PKCS#12 ファイルからの抽出, 546
    - クライアント認証に使用, 546
  - 要件
    - DHCP サービス, 520
    - SSL バージョンのサポート, 521
    - WAN ブートサーバー, 519
    - Web サーバー, 521
    - Web サーバーのオペレーティングシステム, 521
    - Web プロキシ, 521
    - インストールサーバーのディスク容量, 520
    - クライアントの CPU, 520
    - クライアントの OBP, 520
    - クライアントのディスク容量, 520
    - クライアントのメモリー, 520
    - デジタル証明書, 527
    - ログサーバー, 521
  - 例
    - DHCP サービスを使ったインストール, 582

- WAN ブートインストール, 例 (続き)
    - /etc/netboot ディレクトリ, 526
    - /etc/netboot ディレクトリの作成, 542, 592
    - net デバイス別名の確認, 569, 601
    - net デバイス別名の設定, 569
    - OBP への暗号化鍵のインストール, 571, 601
    - OBP へのハッシュキーのインストール, 571, 601
    - rules ファイルの作成, 597
    - sysidcfg ファイル, 552
    - sysidcfg ファイルの作成, 596
    - wanboot.conf ファイル, 560, 562, 599
    - WAN ブートミニルートの作成, 591
    - 暗号化鍵の作成, 550, 595
    - 暗号化の使用, 595
    - インストール wanboot プログラム, 592
    - カスタム JumpStart プロファイル, 554
    - カスタム JumpStart プロファイルの作成, 597
    - クライアント OBP での対応の確認, 567, 600
    - クライアント認証の有効化, 594
    - クライアントの証明書の挿入, 547, 594
    - クライアントの非公開鍵の挿入, 547, 594
    - コピー wanboot-cgi プログラム, 593
    - サーバー認証の有効化, 547, 594
    - システム構成ファイル, 558
    - システム構成ファイルの作成, 598
    - 自動インストール, 577, 602
    - 信頼できる証明書の挿入, 547, 594
    - 対話式インストール, 580
    - デジタル証明書の準備, 594
    - 動作中のクライアントに対する暗号化鍵のインストール, 573
    - 動作中のクライアントに対するハッシュキーのインストール, 573
    - ドキュメントルートディレクトリ, 591
    - ネットワークのセットアップ, 590
    - ハッシュキーの作成, 550, 595
    - フラッシュアーカイブの作成, 596
    - ローカル CD メディアからのインストール, 585
    - ログサーバーの構成, 544, 593
    - ローカルの CD メディアによる, 583
    - ログサーバー
      - wanboot.conf ファイルに指定, 612
  - WAN ブートインストール時のデータの暗号化
    - HTTPS による, 545
    - デジタル証明書による, 546, 594
    - 非公開鍵による, 546, 594
  - WAN ブートインストール時のデータの保護
    - HTTPS による, 516
    - 暗号化鍵による, 516, 548
    - ハッシュキーによる, 516, 548
  - WAN ブートインストールでの機密性の問題, 528
  - WAN ブートインストールのセキュリティの問題, 528
  - WAN ブートインストール用の boot コマンドの構文, 608
  - WAN ブートサーバー
    - wanboot-cgi プログラムのコピー, 543
    - Web サーバーの要件, 521
    - 構成, 535
    - 説明, 519
    - 要件, 519
  - WAN ブートファイルシステム, 説明, 512
  - WAN ブートミニルート
    - wanboot.conf ファイルに指定, 611
    - 作成, 536, 591
    - 説明, 512
    - ドキュメントルートディレクトリへの保存, 523
  - WARNING: CHANGE DEFAULT BOOT DEVICE, 652
  - WARNING: clock gained xxx days メッセージ, 645
  - Web プロキシ, WAN ブートインストールの要件, 521
  - Web プロキシの事前設定, 65
- あ
- アーカイブ
    - 「スクリプト」も参照
    - flar create コマンド, 246
    - JumpStart プロファイルの例, 278, 279, 280
    - WAN ブートインストール用のドキュメントルートディレクトリへの保存, 523
    - WAN ブートプロファイルの例, 554
    - アーカイブの作成
      - WAN ブートインストール, 551
      - 差分, 225, 228

- アーカイブ、アーカイブの作成 (続き)
  - 初期インストール, 220
  - プラットフォームの要件, 205
  - 例, 221, 227
- 圧縮, 211
- インストール
  - Solaris suninstall プログラム、SPARC, 188
  - Solaris suninstall プログラム、x86, 194
  - Live Upgrade ブート環境, 459
  - WAN ブートによる, 574
  - Solaris Web Start、SPARC, 173
  - Solaris Web Start、x86, 181
  - インストール方法, 211, 233
  - カスタム JumpStart, 315
  - 説明, 199
- カスタマイズ
  - スクリプトの使用, 209
  - 説明, 208
- 管理, 234
- キーワード
  - section\_begin と section\_end, 241
  - カスタム JumpStart, 342
  - 識別セクション, 242
  - 説明, 241
  - ユーザー定義, 246
- クローンの更新
  - 説明, 201
- 計画
  - アーカイブのインストール, 211
  - アーカイブの作成, 207
  - 差分アーカイブの作成, 207
  - マスターシステム, 204
- 作業マップ, 213
- 情報の取得, 236
- セクション
  - アーカイブ cookie、説明, 239
  - アーカイブ識別、説明, 240
  - アーカイブファイル、説明, 241
  - 作成, 217
  - サマリー、説明, 240
  - 説明, 210, 239
  - マニフェスト、説明, 240
  - ユーザー定義、説明, 240
- 説明, 199
- ほかのインストール方法との比較, 37
- アクセス権
  - begin スクリプト, 292
  - /etc/netboot ディレクトリ, 542
  - finish スクリプト, 293
  - JumpStart ディレクトリ, 262, 265
- アップグレード
  - Solaris Live Upgrade
    - アップグレード障害の回復, 468
    - ガイドライン, 449
    - 作業, 448, 450, 451, 458, 459, 461, 462, 463
    - 説明, 390
    - 例, 491, 494, 498, 499
  - Solaris Update リリースへ, 673
  - カスタム JumpStart インストール, 313
  - カスタム JumpStart による, 90
  - クリーンアップ, 182, 196
  - 作業の概要, 27
  - システム要件, 43
  - 初期インストールとの比較, 29, 89
  - ディスク容量の再配置, 90
  - ディスク容量の推奨, 44
  - プロファイルキーワード, 353, 361, 367
  - 方法, 30, 89
  - 方法の比較, 35
  - ログファイル, 175, 188, 189
- アップグレードインストール、アップグレードの失敗, 655
- アップグレード障害、Solaris Live Upgrade による回復, 468
- アップグレードの失敗
  - リポート時の問題, 656
  - リポートの問題, 655
- アップグレードのための準備
  - アップグレードする前に必要な情報, 59
  - システムの準備, 57
- 暗号化鍵
  - wanboot.conf ファイルに指定, 611
  - WAN ブートインストール時のデータの暗号化, 516
  - WAN ブートインストールでの使用, 548
- インストール
  - ickey コマンドによる, 572
  - set-security-key コマンドによる, 570
  - wanboot プログラムによる, 579
  - インストール方法, 569
  - クライアントの OBP に, 570
  - 動作中のクライアントに, 572

暗号化鍵, インストール (続き)

例, 571, 573, 601

作成, 548, 595

説明, 516

アンパサンド (&&) ルールフィールド, 272

い

インストール

WAN ブート、説明, 511

アップグレードとの比較, 29

作業の概要, 27

システム要件, 43

ディスク容量の推奨, 44

フラッシュアーカイブ、説明, 199

方法の比較, 35

インストールサーバー

CD メディアを使用する作成, 137, 142

CD メディアを使用する作成、例, 141, 146

CD メディアを使用するネットワークイン  
ストール設定, 162

DVD による作成, 122

DVD メディアを使用する作成, 119

DVD メディアを使用する作成の例, 121

DVD メディアを使用するネットワークイン  
ストールの設定, 129

DVD を使用する作成の例, 122, 124, 125

PXE インストール用に作成, 682

PXE インストール用に作成、例, 684

WAN ブートインストールの要件, 520

クロスプラットフォーム CD メディアを使用  
して作成, 151, 157

クロスプラットフォーム CD メディアを使用  
して作成、例, 152, 157

クロスプラットフォームメディアを使用して  
作成, 147

サブネット上の, 121

適切なシステムの種類, 113

インストールの準備

WAN ブートインストール, 531

WAN ブートインストール用のクライアン  
ト, 566

カスタム JumpStart インストール, 258

システム構成情報の事前設定

方法, 64

利点, 63

インストールのための準備

インストールする前に必要な情報, 52

システムの準備, 51

え

エンドユーザーシステムサポート, 352

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

サイズ, 46

説明, 45

か

開始

check スクリプト, 289, 290

開発者システムサポート, 352

開発者システムサポートソフトウェアグルー  
プ

サイズ, 46

説明, 45

カスタム JumpStart

準備, 315

フラッシュアーカイブのインストール、イン  
ストールの準備, 315

カスタム JumpStart インストール, 313

tip ライン接続, 316, 320

WAN ブートインストールでの, 550

いつアップグレードするか, 90

オプション機能, 291, 293, 306

begin スクリプト, 293

finish スクリプト, 297

概要, 257

準備, 258, 290

説明, 258

ブートとインストール, 313

プロファイルキーワード, 340

ほかのインストール方法との比較, 36

例, 254, 257, 325, 326, 328, 329, 330, 331, 332,  
333, 334

check スクリプト, 331

JumpStart ディレクトリ, 329

WAN ブートインストールのプロファイ  
ル, 554

WAN ブートインストールプロファイ  
ル, 279

フラッシュプロファイル, 278, 280

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, 332  
ミラー化されたファイルシステムプロ  
ファイル, 281, 282  
画面サイズの事前設定, 65  
カラー深度の事前設定, 65  
代替りのインストールプログラム, 306  
!(感嘆符) ルールフィールド, 272  
感嘆符 (!) ルールフィールド, 272

## き

キー, 「暗号化鍵、ハッシュキー」を参照  
キーボード言語、配置の事前設定, 65  
キーワード  
Solaris Live Upgrade  
プロファイル, 454, 455  
ボリューム, 440  
sysidcfg ファイル, 67  
フラッシュアーカイブ, 241  
フラッシュアーカイブ、カスタム  
JumpStart, 342  
プローブ, 307

## 起動

rpld デーモン, 652  
tftpd デーモン, 651, 652  
共有, WAN ブート構成情報, 525  
共有、JumpStart ディレクトリの, 262  
共有可能なファイルシステム、定義, 380  
共有コマンド、JumpStart ディレクトリの共  
有, 262

## く

クライアント、WAN ブートインストールの要  
件, 520  
クライアント認証、WAN ブートインストール用  
に構成, 546  
クライアント認証とサーバー認証  
WAN ブートインストール用に構成, 546,  
548, 594  
クラスタのプロファイルキーワード  
説明と値, 353, 454  
グラフィックスカードの事前設定, 65  
クリティカルなファイルシステム、定義, 380

クローンシステム  
「アーカイブ」も参照  
説明, 199  
クローンシステムのインストール  
更新, 201  
初期インストール, 199  
クローンシステムの更新, 説明, 201

## け

### 計画

Solaris Live Upgrade, 397  
WAN ブートインストール  
wanboot-cgi プログラムの保存, 527  
Web サーバーの要件, 521  
インストールに必要な情報, 528  
インストールファイルの保存, 522  
構成ファイルとセキュリティファイルの  
保存, 524  
サーバーレイアウト, 521  
システム要件, 519  
ファイルとセキュリティファイルの共  
有, 525  
インストール方法の比較, 35  
作業の概要, 27  
システム要件, 43  
初期インストールとアップグレードの比  
較, 29  
ディスク容量, 44  
ネットワークからインストール, 31  
フラッシュインストール, 204

### 検査

rules ファイルの妥当性検査, 289  
プロファイル, 285  
検証, プロファイル, 285

## こ

コアシステムサポート, 352  
サイズ, 46  
説明, 45  
構成  
Solaris Live Upgrade のファイルシステ  
ム, 420  
WAN ブートインストール用の DHCP サー  
ビス, 563

## 構成 (続き)

- WAN ブートインストール用のログサーバー, 544
- WAN ブートサーバー, 535
- ディスク構成ファイルの作成, 301

## コピー

- JumpStart インストールファイル, 263, 267, 269
- JumpStart ディレクトリのファイル, 294
- Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版), 268
- Solaris Live Upgrade のファイルシステム, 478

## コメント

- rules ファイル内の, 271
- wanboot.conf ファイル内の, 610
- プロファイル内の, 275

## さ

### サーバー

- CD メディアを使用したネットワークインストールの設定
  - PXE インストール, 684
- CD メディアを使用するネットワークインストールの設定
  - スタンドアロンインストール, 161
- DVD メディアを使用するネットワークインストールの設定
  - スタンドアロンインストール, 128
- JumpStart ディレクトリの作成, 262
- WAN ブートインストール
  - Web サーバーソフトウェアの要件, 521
  - 構成オプション, 521
  - 説明, 519
  - 要件, 519
- ネットワークインストールの要件, 113
- ルート領域, 351
- ログ (WAN ブートインストール), 544
- サーバー認証, WAN ブートインストール用に構成, 546
- サービス妨害攻撃, WAN ブートインストール, 528
- 再起動スクリプト, 作成, 219
- サイズ
  - tip ウィンドウの大きさ, 316, 320

## サイズ (続き)

- スワップ空間
  - 最大サイズ, 352
  - プロファイルの例, 258
- スワップ領域
  - Solaris Web Start 用の最小サイズ, 47
  - ディスクレスクライアント, 351
- ハードディスク, 120, 336, 340, 374, 375
  - プロンプキーワード, 374
  - ルート領域, 351
- メモリー, 337, 374
- 再設定する、入出力割り込みの後にディスプレイと端末を, 168
- サイト固有のインストールプログラム, 306
- 削除
  - Live Upgrade ブート環境, 482
  - クラスタ、アップグレード時の, 353
  - ソフトウェアグループからパッケージを, 367

## 作成

- CD メディアを使用するインストールサーバーの, 136, 137, 142
- custom\_probes.ok ファイル, 311
- DVD メディアを使用するインストールサーバーの, 117, 119
- DVD メディアを使用するサブネット上でのブートサーバーの, 126
- /etc/locale ファイル, 73
- JumpStart ディレクトリ, 262
- PXE インストール用インストールサーバー, 682
- rules.ok ファイル, 311
- rules.ok ファイル, 288, 289
- rules ファイル, 270
- Solaris Live Upgrade
  - ブート環境、作業, 418, 424, 426, 428, 430, 431, 433, 435, 436, 438, 499
  - ブート環境、説明, 380
  - ブート環境の内容のカスタマイズ、作業, 444
- sysidcfg ファイル, 70, 552
- UFS, 266
- WAN ブート
  - /etc/netboot ディレクトリ, 540
  - rules ファイル, 554
  - wanboot.conf ファイル, 559
  - WAN ブートミニルート, 536
  - 暗号化鍵, 548



- 作成, WAN ブート (続き)
  - インストールファイル, 550
  - カスタム JumpStart ファイル, 550
  - クライアントの非公開鍵, 546
  - システム構成ファイル, 557
  - 信頼できる証明書, 546
  - デジタル証明書, 546
  - ドキュメントルートディレクトリ, 536
  - ハッシュキー, 548
  - フラッシュアーカイブ, 551
- インストールサーバー, 122
- クロスプラットフォーム CD メディアを使用したインストールサーバー, 151, 157
- サブネット上でブートサーバーを
  - CD メディアを使用して, 159
  - 作業, CD メディア, 136
  - 作業, DVD メディア, 118
- ディスク構成ファイル, 301
- フラッシュアーカイブ
  - カスタマイズ, 208
  - 計画, 207
  - 作業, 220
  - 作業マップ, 213
  - プラットフォームの要件, 205
- プロファイル
  - 説明, 274
  - 動的, 292
- ミラー化されたファイルシステム, 359
- ローカルファイルシステム, 357
- 作成前スクリプト, 作成, 216
- サブネット
  - CD メディアを使用するブートサーバー作成, 159
  - DVD メディアを使用するブートサーバーの作成, 126
  - インストールサーバー, 124, 151, 157
- サブミラー, 100
  - ガイドライン, 108
- サブミラー、説明, 386
- 差分アーカイブ
  - 「アーカイブ」も参照
- Solaris Live Upgrade で作成, 228
  - 計画, 207
  - 作成, 225
  - 説明, 201
- し
- 時間帯の事前設定, 65
- システム構成情報の事前設定
  - DHCP による, 75
  - NIS+ によるロケール, 74
  - NIS による言語またはロケール, 72
  - 電源管理, 85
  - ネームサービスによる, 65, 72
  - 方法の選択, 64
  - 利点, 63
- システム構成ファイル
  - SjumpsCF 設定, 609
  - SsysidCF 設定, 609
  - wanboot.conf ファイルに指定, 613
  - WAN ブートインストール用に作成, 557, 598
  - 構文, 609
  - 説明, 525
  - 例
    - セキュリティ保護された WAN ブートインストール, 558, 598
    - セキュリティ保護されていない WAN ブートインストール, 558
- システム情報の表示, 168
- 事前設定、システム構成情報の, sysidcfg ファイルを使用する, 65
- 出力ファイル
  - begin スクリプトのログ, 292
  - finish スクリプトのログ, 293
  - WAN ブートインストール用の bootlog ファイル, 544
  - アップグレードログ, 175, 188, 189
- 準備、インストールする, カスタム JumpStart インストール, 290
- 照合
  - rootdisk の値, 369
  - 動的プロファイル, 292
  - ルールでの順序, 273
  - ルールの順序, 313, 319
- 状態データベース
  - Solaris Live Upgrade の要件, 399
  - 概念的な概要, 99
  - 説明, 385
  - 定義, 99
- 状態データベースの複製, 99
  - 基本操作, 100
  - 最小数, 107
  - 使用方法, 100
  - 単一スライス上に複数作成, 106

状態データベースの複製 (続き)  
定義, 99  
場所, 107, 108  
証明書, 「デジタル証明書」を参照  
シリアルコンソール  
  add\_install\_client コマンドで指  
  定, 132, 165, 687  
シングルユーザーモードでのブート, 110  
信頼できる証明書  
  PKCS#12 ファイル形式で作成, 546  
  truststore ファイルへの挿入, 546, 594  
  サーバー認証に使用, 546

す

スクリプト  
  begin スクリプト, 291, 293, 306  
  finish スクリプト, 293, 297, 306  
  フラッシュアーカイブ  
    ガイドライン, 209  
    カスタマイズ, 209  
    再起動, 219  
    作成, 215, 219  
    配置前の作成, 218  
  ルールフィールド内の Bourne シェルスクリ  
  プト, 272  
スタンドアロンシステム  
  カスタム JumpStart インストールの例, 254  
  プロファイルの例, 276  
ステータス, ブート環境を表示する, 476  
スライス, 276  
  Solaris Live Upgrade  
    選択するための指針, 403  
    ファイルシステムのカスタマイズ, 420  
  プロファイルの例, 276  
  ルートスライスの位置, 646  
  ルールキーワード, 374  
スワップファイルシステム  
  Solaris Live Upgrade  
    カスタマイズ, 422, 430  
    スライスを選択するための指針, 407  
  スワップの割り当て, 46  
  ディスクレスクライアントのスワップ領  
  域, 351  
  プロファイルの例, 258

せ

セキュリティ  
  root パスワード, 297  
  WAN ブートインストール  
    説明, 515  
セキュリティポリシー, 事前設定, 65  
全体ディストリビューション, 352  
全体ディストリビューションソフトウェアグ  
ループ  
  サイズ, 46  
  説明, 45  
全体ディストリビューションと OEM サポー  
ト, 352  
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  
  サイズ, 46  
  説明, 45

そ

ソフトウェアグループ  
  アップグレード, 31, 89, 353  
  サイズ, 46  
  説明, 46  
  パッケージの指定, 367  
  プロファイルの, 352, 353  
  プロファイルの例, 276

た

多数決アルゴリズム, 100  
妥当性検査  
  custom\_probes ファイル  
    check を使用する, 312  
    テスト, 312  
  rules ファイル, 289  
    check の使用, 288, 289, 312  
    WAN ブートインストール用の, 555  
    カスタム JumpStart の例, 331  
    動的プロファイルと, 293  
  wanboot.conf ファイル, 559, 560  
端末タイプの事前設定, 65

## つ

### 追加

- finish スクリプトによりパッケージとパッチを, 294
- locale.org\_dir テーブルエントリ, 74
- rules ファイルにルールを, 271
- インストールサーバー構成
  - CD メディアを使用する, 162
- インストールサーバーの構成
  - DVD メディアを使用する, 129
- クラスタ、アップグレード時の, 353
- ソフトウェアグループ内のパッケージ, 367
- データレスクライアント
  - DVD メディアを使用する, 128
  - PXE インストール用, 684
- データレスクライアントを
  - CD メディアを使用して, 161
- ネットワーク上でシステムを, 118, 136
- パッケージをソフトウェアグループから, 454
- ファイルを finish スクリプトで, 294

## て

### ディスク構成ファイル

#### 作成

- SPARC システム, 301
- x86 システム, 302

説明, 284, 301

### ディスク容量

- WAN ブートインストールの要件, 520
- 計画, 44
- 要件

- Solaris Live Upgrade の, 399
- ソフトウェアグループの, 46

### ディスクレスクライアント

- スワップ領域, 351
- プラットフォーム, 351

### ディレクトリ

- /etc/netboot
  - 暗号化鍵の挿入, 548
  - クライアント間での構成ファイルとセキュリティファイルの共有, 524
  - クライアント認証とサーバー認証の構成, 546
  - クライアントの非公開鍵の挿入, 546

### ディレクトリ, /etc/netboot (続き)

- 構成ファイルとセキュリティファイル、説明, 525
- 構成ファイルとセキュリティファイルの保存, 524
- 信頼できる証明書の挿入, 546
- 説明, 524
- デジタル証明書の挿入, 546
- ハッシュキーの挿入, 548
- ファイルとセキュリティファイルの共有, 525
- 例, 526
- /etc/netboot ディレクトリ, 540
- JumpStart, 262, 265, 271, 294, 329
  - アクセスの許可, 263
  - インストールファイルのコピー, 263, 267
  - インストールファイルをコピー, 269
  - ディレクトリの共有, 262, 329
  - ファイルのコピー, 294
  - ファイルの追加, 294
- ドキュメントルート
  - 作成, 536, 591
  - 説明, 522
  - 例, 522, 591
- 変更, 289, 311
  - ローカルディスク上の Solaris 9 x86 Platform Edition ソフトウェアのイメージへ, 269
  - ローカルディスク上の Solaris 9 SPARC Platform Edition ソフトウェアのイメージ, 267
  - ローカルディスク上の Solaris 9 ソフトウェアのイメージに, 263
- ディレクトリの変更
  - ローカルディスク上のソフトウェアイメージ, 267
  - ローカルディスクの Solaris 9 x86 Platform Edition ソフトウェアのイメージへ, 269
- デーモン, 651
- デジタル証明書
  - WAN ブートインストール時のデータの保護, 516
  - WAN ブートインストールの準備, 546, 594
  - WAN ブートインストールの要件, 527
  - 説明, 516, 527
- テスト
  - custom\_probes ファイルの妥当性検査 check の使用, 311

テスト, custom\_probes ファイルの妥当性検査 (続き)

    custom\_probes のテスト, 312

    rules ファイルの妥当性検査

        check の使用, 312

    rules ファイルの妥当性検査

        check の使用, 288, 289

        カスタム JumpStart の例, 331

        動的プロファイルと, 293

    Solaris Live Upgrade、プロファイル, 457

    WAN ブート

        rules ファイル, 555

        wanboot.conf ファイル, 560

    プロファイル, 284, 288

デフォルト

    partitioning, 367

        使用しないディスク, 354

        ディスクの指定, 370

    SI\_CONFIG\_DIR 変数, 294

    インストールされるソフトウェアグループ, 353

    動的プロファイル名, 293

電源管理, 65, 85

と

動的プロファイル, 292, 293

トークンリングカードでのブートエラー, 650

ドキュメントルートディレクトリ

    作成, 536

    説明, 522

    例, 522, 591

ドメイン

    プローブキーワード, 374

    ルールキーワード, 336, 374

ドメイン名の事前設定, 64

トラストアンカー, 「信頼できる証明書」を参照

トラブルシューティング, インストールの一般的な問題, 646

な

名前/名前の指定

    rules ファイル, 271

    rules ファイル, 271

名前/名前の指定 (続き)

    WAN ブートインストール用のシステム構成

        ファイル, 557

        ソフトウェアグループ, 353

名前/名前の割り当て, ホスト名, 164

名前/命名

    ホスト名, 336, 374

名前/指定

    custom\_probes ファイル, 308

    システムプラットフォーム名, 168

名前/割り当て

    システムのモデル名, 338, 374

    動的プロファイル名, 293

に

認証, WAN ブートインストール用に構成, 546

ね

    ネームサーバーの事前設定, 64

    ネームサービス, 事前設定, 64

ネットワークインストール

    「WAN ブートインストール」も参照

    CD メディアを使用, 137, 142, 159

    DHCP を使用, 116

    DVD メディアを使用する, 118, 126

    PXE を使用, 33, 679

    WAN ブートインストールの例, 589

    カスタム JumpStart インストール, 257

    クロスプラットフォームメディアを使用, 147

    準備, 113

    説明, 113

    要件, 113

ネットワークインタフェースの事前設定, 64

ネットワーク番号, 338, 374

は

ハードディスク

    fdisk パーティションの要件, 47

    IDE ドライブの表面検査, 652

    partitioning

        partitioning default に含めない, 354

- ハードディスク, partitioning (続き)
  - partitioning default の指定, 370
  - プロファイルキーワード, 367
  - 例, 276
- rootdisk の値, 369
- サイズ, 120, 336, 340, 374, 375
  - プローブキーワード, 374
  - ルート領域, 351
- スワップ空間
  - 最大サイズ, 352
  - プロファイルの例, 258, 276
- スワップ領域
  - Solaris Web Start に必要な容量の要件, 47
  - スワップの割り当て, 46
  - ディスクレスクライアント, 351
- マウント, 356
- 配置後スクリプト, 作成, 219
- 配置前スクリプト
  - 作成, 218
  - 説明, 240
- バイナリの破壊、WAN ブートインストール, 528
- パス
  - check スクリプト, 289, 311
- パスワード、root の, 297
- バックスラッシュrules ファイル内の, 271
- パッケージ
  - chroot により追加, 296
  - finish スクリプトによる追加, 294
  - Solaris Live Upgrade
    - 追加, 401
    - 要件, 665, 666, 668
  - カスタム JumpStart を使用するときの要件, 665
  - 管理ファイル, 291
- ハッシュキー
  - wanboot.conf ファイルに指定, 611
  - WAN ブートインストール時のデータの保護, 516
  - WAN ブートインストールでのデータの保護, 548
- インストール
  - ickey コマンドによる, 572
  - set-security-key コマンドによる, 570
  - wanboot プログラムによる, 579
  - インストール方法, 569
  - クライアントの OBP に, 570

- ハッシュキー, インストール (続き)
  - 動作中のクライアントに, 572
  - 例, 601
  - 作成, 548, 595
  - 説明, 516
- パッチ
  - chroot により追加, 296
  - finish スクリプトによる追加, 294
  - 追加
    - Solaris Live Upgrade による, 401
  - パッチレベルの確認, 402
- パッチアナライザ, 673

## ひ

### 非公開鍵

- keystore ファイルへの挿入, 546
- wanbootutil コマンドによる作成, 546
- WAN ブートインストールの準備, 546
- クライアント認証に使用, 546

日付の事前設定, 65

表示

- Live Upgrade ブート環境の名前, 483
- tip ライン接続とカスタム JumpStart インストール, 316, 320
- 暗号化鍵の値をwanbootutil コマンドで, 570
- 共有ファイルシステム, 168
- システム情報, 168
- ハッシュキーの値を wanbootutil コマンドで, 570
- プラットフォーム名, 168
- マウントされたファイルシステム, 168

表示解像度の事前設定, 65

表面検査、IDE ドライブの, 652

## ふ

### ファイルとファイルシステム

- begin スクリプトの出力, 292
- finish スクリプトの出力, 293
- Solaris Live Upgrade
  - RAID-1 ボリューム (ミラー) の作成、説明, 384
  - カスタマイズ, 420

- ファイルとファイルシステム, Solaris Live Upgrade (続き)
  - 共有可能なファイルシステムのコピー、作業, 433
  - サイズの見積もり, 399
  - 作成のための指針, 402
  - スライスを選択するための指針, 403
  - 説明, 380
  - ブート環境間でのファイルシステムの共有, 407
- UFS の作成, 266
- wanboot.conf
  - 構文, 610
  - 作成, 559
  - 説明, 559, 610
- WAN ブートファイルシステム, 512
- 共有ファイルシステムの表示, 168
- コピー
  - JumpStart インストールファイル, 263, 267, 269
  - JumpStart ディレクトリのファイルを finish スクリプトで, 294
  - Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版), 268
- 作成
  - ミラー化されたファイルシステム, 359
  - ローカルファイルシステム, 357
- システム構成
  - WAN ブート用に作成, 557
  - 構文, 609
  - 説明, 557
- マウントされたファイルシステムの表示, 168
- リモートファイルシステムのマウント, 356
- ファクトリ JumpStart, ほかのインストール方法との比較, 39
- ブート環境、Solaris Live Upgrade
  - 構成の表示, 488
  - 失敗、説明, 394
- ブート環境のアクティブ化
  - 作業, 464
  - 失敗、説明, 394
  - 説明, 392
  - ファイルの、説明, 409
- ブートサーバー
  - DVD を使用してサブネット上に作成, 124
  - DVD を使用する例, 127
- ブートサーバー (続き)
  - サブネット上で CD メディアを使用して作成, 159
  - サブネット上での作成
    - DVD メディアを使用する, 126
    - 説明, 114
    - ネットワークインストールの要件, 114
  - ブート、システムの、端末とディスプレイを再設定する, 168
  - ブートディスクパーティションレイアウト、新規デフォルト (x86 システム), 48
  - ブートデバイス
    - add\_install\_client コマンドで指定, 133, 166, 688
  - 複数行のルールrules ファイル内の、複製, 99
  - プライマリドキュメントディレクトリ、「ドキュメントルートディレクトリ」を参照
  - フラッシュ、「アーカイブ」を参照
  - フラッシュアーカイブのカスタマイズ
    - スクリプトの使用, 209
    - マスターシステム, 204
  - フラッシュアーカイブの結合, 235
  - フラッシュアーカイブの分割, 234
  - プラットフォーム
    - インストールサーバーの設定, 164, 687
    - システム属性とプロファイルの一致, 313, 319
    - システム属性とプロファイルの照合, 273
    - システムのモデル名, 338, 374
    - ディスクレスクライアント, 351
    - 名前の決定, 168
    - プロンプトキーワード, 374
    - ルールキーワード, 337, 374
  - 不良ブロック, 652
  - プロンプトキーワード
    - arch, 374
    - disks, 374
    - domainname, 374
    - hostaddress, 374
    - hostname, 374
    - installed, 374
    - karch, 374
    - memsize, 374
    - model, 374
    - network, 374
    - osname, 374
    - rootdisk, 374

- プロープキーワード (続き)
  - totaldisk, 375
- プロセッサ
  - WAN ブートインストールの要件, 520
  - プロープキーワード, 374
  - ルールキーワード, 335, 374
- フロッピーディスク
  - JumpStart ディレクトリ, 265
  - JumpStart ディレクトリのアクセス, 264
  - Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版) のコピー, 268
  - フォーマット, 266, 269, 678
- フロッピーディスクのフォーマット, 266
- プロファイル
  - Solaris Live Upgrade
    - 作成, 453
    - 差分アーカイブ用の例, 456
    - テスト, 457
    - 例, 456
  - WAN ブートインストールの, 553
  - 検査, 285
  - コメント, 275
  - 作成, 274
  - システムの照合, 273, 313, 319
  - 説明, 274
  - テスト, 288
  - 動的プロファイル, 292, 293
  - 名前の指定, 275, 553
  - 要件, 271, 274
  - ルールフィールド, 272
  - 例, 276, 329, 330
  - WAN ブートインストール, 279, 554
  - フラッシュ, 278, 280
- プロファイルキーワード, 340
  - archive\_location, 342
  - backup\_media, 347
  - boot\_device, 349
  - bootenv createbe, 350
  - client\_arch, 351
  - client\_root, 351
  - client\_swap, 351
  - cluster
    - 説明と値, 352
    - 例, 276
  - dontuse
    - usedisk と, 370
    - 説明と値, 354
- プロファイルキーワード (続き)
  - fdisk
    - 説明と値, 354
    - 例, 276
  - filesystem
    - 説明と値, 356
    - ミラー化されたファイルシステム, 359
    - リモートファイルシステム, 356
    - 例, 276
    - ローカルファイルシステム, 357
  - forced\_deployment
    - 説明と値, 360, 455
  - geo
    - 説明と値, 360
  - install\_type, 274
    - 説明と値, 361
    - 要件, 276
    - 例, 276
  - isa\_bits
    - 説明と値, 362
  - layout\_constraint
    - 説明と値, 362
  - local\_customization
    - 説明と値, 364, 454, 455
  - locale
    - 説明と値, 365
  - metadb
    - 説明と値, 365
    - 例, 276
  - no\_master\_check
    - 説明と値, 366
  - num\_clients, 366
  - package
    - 説明と値, 367
  - partitioning
    - 使用しないディスク, 354
    - 説明と値, 367
    - ディスクの指定, 370
    - 例, 276
  - root\_device, 368
  - system\_type
    - 説明と値, 370
    - 例, 276
  - usedisk
    - 説明と値, 370
  - 大文字と小文字の区別, 340
  - クラスター
    - 説明と値, 353, 454

プロファイルキーワード (続き)  
状態データベースの複製 (metadbs) の作成, 365  
プロファイルフィールドの等号(=), 292

## へ

変更、ディレクトリの  
JumpStart ディレクトリへ, 289, 311  
ローカルディスク上の Solaris 9 ソフトウェアのイメージに, 263

## 変数

SI\_CONFIG\_DIR, 294  
SI\_PROFILE, 293  
SYS\_MEMSIZE, 286

## 編成

Solaris DVD (SPARC 版), 619  
Solaris DVD (x86 版), 625  
Solaris INSTALLATION CD (SPARC 版), 620  
Solaris INSTALLATION CD (x86 版), 627  
Solaris LANGUAGES CD (SPARC 版), 622  
Solaris LANGUAGES CD (x86 版), 630  
Solaris SOFTWARE (SPARC 版) CD, 621  
Solaris SOFTWARE (x86 版) CD, 628

## ほ

ポインティングデバイスの事前設定, 65  
ホスト名の事前設定, 64  
ボリューム

RAID-0、説明, 386  
RAID-1、説明, 386  
概念的な概要, 97  
仮想ディスク, 96  
使用, 96  
命名規則, 106

## ボリュームマネージャ

「Solaris ボリュームマネージャ」も参照  
コピー, 266, 269  
Solaris Device Configuration Assistant (デバイス構成用補助) (x86 版), 268  
Solaris Device Configuration Assistant (デバイス構成用補助) イメージ, 269

ボリューム命名規則, 106

## ま

マイクロプロセッサ  
プローブキーワード, 374  
ルールキーワード, 335, 374  
マウント  
begin スクリプトに関する注意事項, 292  
Solaris 9 のインストールによる, 293  
マウントされたファイルシステムの表示, 168  
リモートファイルシステム, 356  
マスターシステム  
「アーカイブ」も参照  
インストールのカスタマイズ, 204  
インストール方法, 215  
周辺装置, 205  
説明, 204  
マッピング、IDE ドライブの不良ブロック, 652

## み

### ミラー

「RAID-1 ボリューム」を参照  
およびディスクのジオメトリ, 108  
ガイドライン, 108  
概念的な概要, 101  
作成情報, 108  
定義, 101

## め

メタデバイス、「ボリューム」を参照  
メモリー  
WAN, 520  
スワップ空間のサイズと, 352  
プローブキーワード, 374  
ルールキーワード, 337, 374

## も

モニタータイプの事前設定, 65  
問題の解決  
DHCP によるネットワークブート, 650  
一般的なインストールの問題, 651  
DHCP によるネットワークブート, 650  
間違ったサーバーからのブート, 651



## ゆ

ユーザー定義セクション  
フラッシュアーカイブ  
作成, 217

## よ

### 要件

custom\_probes ファイル, 308  
fdisk パーティション, 47  
Live Upgrade の使用, 397  
Solaris Web Start の使用, 47, 48  
WAN ブートインストール, 519  
スワップ領域, 47  
ディスク容量, 44  
ネットワークインストール  
サーバー, 113  
プロファイル, 271, 274  
メモリー, 43

## り

リモートファイルシステム, マウント, 356

## る

ルート (/) ファイルシステム  
JumpStart で設定される値, 369  
非アクティブブート環境のパッケージ要  
件, 666  
ルート (/) ファイルシステム, プロファイルの  
例, 258  
ルート環境を finish スクリプトでカスタマイズ  
する, 297  
ルートスライスの位置, 646  
ルール  
rootdisk 照合ルール, 369  
構文, 271  
照合順序, 273, 313, 319  
妥当性の検査, 289, 312  
動的プロファイル, 292, 293  
フィールドの説明, 271  
複数行のルール, 271  
例, 273  
ルールキーワード, 335

## ルールキーワード (続き)

any, 335, 374  
arch, 335, 374  
disksize, 336, 374  
domainname, 336, 374  
hostaddress, 336, 374  
hostname, 335, 336, 374  
installed, 337, 374  
karch, 337, 374  
memsize, 337, 374  
model, 338, 374  
network, 338, 374  
osname, 338, 374  
probe, 339  
totaldisk, 340, 375

## れ

### 連結

概念的な概要, 100  
作成情報, 108  
定義, 100

連結、説明, 386

連結ボリューム、「連結」を参照

## ろ

### ログサーバー

wanboot.conf ファイルに指定, 612  
WAN ブートインストールの要件, 521  
WAN ブートインストール用に構成, 544, 593  
説明, 521  
ログメッセージの場所, 544

### ログファイル

begin スクリプトの出力, 292  
finish スクリプトの出力, 293  
WAN ブートインストール用の, 544  
アップグレードインストール, 175, 188, 189

ロケールファイル, 73

論理積ルールフィールド, 272

