

Oracle® Solaris 11.1 システムのインストール

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

はじめに	17
パート I Oracle Solaris 11.1 のインストールオプション	23
1 インストールオプションの概要	25
インストールオプションの比較	25
シンプルで、事前設定されたインストール	26
サーバーの設定を必要とするインストール	27
その他のオプション	28
パート II インストールメディアを使用したインストール	29
2 インストールの準備	31
Live Media およびテキストインストールのシステム要件	31
複数のオペレーティングシステムをインストールするためのブート環境の準備	32
システムのパーティション分割	33
インストール前にシステムをパーティション分割するためのガイドライン	33
対話型インストール中にシステムをパーティション分割するためのガイドライン	34
適切なデバイスドライバがあることの確認	38
▼ Oracle Device Detection Tool を使用する方法	39
Oracle Configuration Manager の使用	39
3 Live Media の使用	43
GUI インストーラを使用したインストール	43
GUI インストーラでのデフォルト設定	44

▼ GUI インストールを準備する方法	45
▼ GUI インストールを実行する方法	46
システムがコンソールモードでブートした場合に実行する操作	50
▼ システムがコンソールモードでブートした場合に Live Media から Oracle Solaris を インストールする方法	51
Live Media インストール後のソフトウェアの追加	52
4 テキストインストーラの使用	55
テキストインストーラを使用したインストール	55
テキストインストーラを使用したネットワーク構成	57
▼ テキストインストールを準備する方法	57
▼ テキストインストールを実行する方法	58
テキストインストール後のソフトウェアの追加	65
ネットワーク経由でのテキストインストールの実行	66
▼ ネットワーク経由でテキストインストールを実行する方法	66
5 メディアからブートする自動インストール	69
AI メディアを使用するインストールの概要	69
AI メディアを使用したインストール	69
AI メディアを使用してインストールするためのシステム要件	70
▼ AI メディアを使用してインストールする方法	71
カスタム AI マニフェストの作成	72
AI メディアからの SPARC システムのブート	72
AI メディアからの x86 システムのブート	73
インストールのログファイルの表示	75
6 Oracle Solaris インスタンスの構成解除または再構成	77
機能グループ	77
sysconfig ユーティリティの使用	78
Oracle Solaris インスタンスの構成解除	79
システムの構成	79
▼ SCI Tool を使用して再構成する方法	80
SCI Tool を使用した構成プロファイルの作成	84

パート III	インストールサーバーを使用したインストール	85
7	複数のクライアントへの自動インストール	87
	自動インストールとは	87
	自動インストーラの使用方法	88
	自動インストーラの使用事例	90
	AIを使用するための最小要件	90
	インストール手順のカスタマイズ	92
	システム構成手順の指定	93
	ローカル IPS パッケージリポジトリの指定	95
	カスタムの初回ブートスクリプトの指定	96
	追加の AI インストールサービスの指定	98
8	インストールサーバーの設定	101
	AI サーバー設定のタスクマップ	101
	インストールサーバーの要件	102
	AI サーバーのハードウェア要件	102
	インストールサービスの操作権限	102
	AI サーバーのソフトウェア要件	103
	AI インストールツールのインストール	104
	インストールサーバーの構成	106
	マルチキャスト DNS の有効化	106
	マルチホームインストールサーバーの構成	107
	Web サーバーホストのポートの構成	107
	デフォルトのイメージパスの構成	108
	ISC DHCP 構成の自動更新	108
	AI インストールサービスの作成	109
	DHCP を設定しないインストールサービスの作成	111
	ローカル DHCP 設定を含むインストールサービスの作成	114
	インストールサーバーの維持	117
	インストールサービスの追加、変更、または削除	118
	インストールサービスへのクライアントの関連付け	123
	インストールサービスへのクライアント固有のインストール手順の関連付け	124
	インストールサービスへのクライアント固有の構成手順の関連付け	127
	AI マニフェストまたはシステム構成プロファイルのエクスポート	130

AI マニフェストまたはシステム構成プロファイルの条件の変更	130
インストールサービスの情報の表示	131
カスタマイズしたインストールの情報の表示	133
AI SMF サービスを管理する	135
9 インストールのカスタマイズ	137
クライアントとインストール手順および構成手順との対応付け	137
AI マニフェストの選択	138
システム構成プロファイルの選択	139
選択条件	140
デフォルトの AI マニフェスト	144
10 クライアントシステムのプロビジョニング	147
XML AI マニフェストファイルのカスタマイズ	148
▼XML AI マニフェストファイルのカスタマイズする方法	148
クライアントインストール時の AI マニフェストの作成	150
▼派生マニフェストスクリプトを作成して適用する方法	150
派生マニフェストスクリプトの作成	151
インストールサービスへの派生マニフェストスクリプトの追加	166
AI マニフェストの例	167
iSCSI ターゲットデバイスの指定	167
RAID 構成の指定	168
SVR4 パッケージのインストール	169
11 クライアントシステムの構成	171
構成プロファイルの指定	171
システム構成プロファイルの作成	172
システム構成プロファイルの検証	172
インストールサービスへのシステム構成プロファイルの追加	173
システム構成プロファイル内での構成の指定	174
root アカウントとユーザーアカウントの構成	175
システム識別情報の設定	178
タイムゾーンとロケールの設定	179
端末タイプとキーボード配列の設定	181

静的なネットワークの構成	182
ネームサービスの構成	184
Oracle Configuration Manager と Oracle Auto Service Request の設定	185
システム構成プロファイルテンプレートの使用	185
システム構成プロファイルの例	187
システム構成プロファイルの例	187
静的なネットワーク構成の指定	189
ネームサービス構成の指定	191
12 ゾーンのインストールと構成	201
AIによる非大域ゾーンのインストール方法	201
大域ゾーンの AI マニフェストでの非大域ゾーンの指定	202
非大域ゾーンの構成およびインストールデータ	203
非大域ゾーンの AI マニフェスト	205
非大域ゾーンの構成プロファイル	207
13 初回ブート時のカスタムスクリプトの実行	209
初回ブート時に1度だけ実行する制御の実装	209
▼初回ブート時に1度だけ実行されるようにする方法	210
初回ブート時に実行するスクリプトの作成	211
SMF マニフェストファイルの作成	213
マニフェスト作成ツールの使用	213
生成されたマニフェストのカスタマイズ	215
スクリプトとサービス用の IPS パッケージの作成	217
▼IPS パッケージを作成して公開する方法	217
AIクライアントへの初回ブートパッケージのインストール	219
▼IPS パッケージのインストール方法	219
初回ブートサービスのテスト	220
▼スクリプトまたはサービスを更新する方法	222
14 クライアントシステムのインストール	223
クライアントのインストール方法	223
SPARC および x86 クライアントシステムの要件	224
インストールクライアントを設定する	225

SPARC クライアントの設定	225
x86 クライアントの設定	226
サービスからのクライアントの削除	226
クライアントのインストール	227
Secure Shell を使用したリモートからのインストールの監視	227
SPARC クライアントのインストール	228
x86 クライアントのインストール	229
クライアントインストールのメッセージ	230
15 自動インストールのトラブルシューティング	233
クライアントインストールが失敗する	233
インストールログおよび手順の確認	233
DNS の確認	233
クライアントブートエラーの確認	234
インストールを開始しないでインストール環境をブートする	242
インストールを開始しないでブートしたあとにインストールを開始する	243
パート IV 関連タスクの実行	245
A Oracle Configuration Manager の操作	247
Oracle Configuration Manager の概要	247
Oracle Configuration Manager の中央コレクタについて	249
Oracle Configuration Manager の管理 (タスク)	250
▼ Oracle Configuration Manager サービスを有効にする方法	250
▼ Oracle Configuration Manager サービスを無効にする方法	250
▼ システムを Oracle リポジトリに手動で登録する方法	251
▼ Oracle Configuration Manager のデータ収集の時間と頻度を変更する方法	251
B デバイスドライバユーティリティの使用	253
デバイスドライバユーティリティの概要	253
▼ デバイスドライバユーティリティを起動する方法	253
▼ 不足しているドライバをインストールする方法	254
▼ HCL にシステムを一覧表示させる方法	256

索引 259

目次

図 5-1	メディアを使用する AI インストール	70
図 7-1	AI を使用するための最小要件	91
図 7-2	インストール手順のカスタマイズ	92
図 7-3	システム構成手順の指定	94
図 7-4	ローカル IPS パッケージリポジトリの指定	95
図 7-5	カスタムの初回ブートスクリプトの指定	97
図 7-6	追加の AI インストールサービスの指定	98

表目次

表 1-1	インストールオプション	25
表 2-1	複数のオペレーティングシステム環境	32
表 2-2	対話型インストール中にディスクをパーティション分割するためのオプション	36
表 2-3	テキストインストール中のVTOCスライス変更のオプション	38
表 5-1	AIメディアを使用するインストールのためのシステム要件	70
表 6-1	機能グループ	77
表 8-1	AIサーバー設定のタスマップ	101
表 9-1	条件のキーワードと条件の階層	141
表 10-1	クライアント属性の環境変数	152
表 11-1	root_account プロパティグループのプロパティ	176
表 11-2	user_account プロパティグループのプロパティ	177
表 11-3	config プロパティグループのプロパティ	178
表 11-4	timezone プロパティグループのプロパティ	179
表 11-5	environment プロパティグループのプロパティ	180
表 11-6	install_ipv4_interface プロパティグループのプロパティ	182
表 11-7	install_ipv6_interface プロパティグループのプロパティ	183
表 11-8	config プロパティグループのプロパティ	184
表 11-9	システム構成テンプレートプロファイルの変数	186

例目次

例 8-1	AI SMF サービスの有効化	135
例 8-2	AI SMF サービスの無効化	135
例 9-1	クライアントと AI マニフェストとの照合	139
例 10-1	ディスクサイズに基づいたディスクパーティション分割の指定	155
例 10-2	追加ディスクの存在に基づいたルートプールレイアウトの指定	157
例 10-3	特定のサイズのディスクが2つ以上ある場合のミラー化構成の指定	158
例 10-4	IP アドレスに基づくインストールするパッケージの指定	161
例 10-5	特定のサイズ以上のディスクのみをターゲットディスクにする指定	162
例 10-6	マニフェスト仕様が正しくないスクリプト	163
例 11-1	パスワードが必ず期限切れになる root アカウントの構成	176
例 11-2	ホスト名の構成	179
例 11-3	ノード名マッピングの無効化	179
例 11-4	タイムゾーンの構成	180
例 11-5	ロケールの構成	180
例 11-6	端末タイプの構成	181
例 11-7	キーボード配列の構成	181
例 11-8	指定されたドメインでの NIS の有効化	192
例 11-9	NIS の構成および DNS の無効化	192
例 11-10	NIS の構成	193
例 11-11	指定されたドメインでの NIS と DNS の有効化	194
例 11-12	検索リストを使用した DNS の構成	195
例 11-13	LDAP および LDAP 検索ベースの構成	196
例 11-14	セキュアな LDAP サーバーを使用した LDAP の構成	197
例 12-1	ゾーンのデフォルトの AI マニフェスト	205
例 13-1	テンプレート初回ブートスクリプト	211
例 13-2	複数の IP インタフェースを構成する初回ブートスクリプト	212
例 13-3	生成された SMF サービスマニフェスト	213
例 13-4	カスタマイズされた SMF サービスマニフェスト	215

はじめに

Oracle Solaris 11.1 システムのインストールでは、次のいずれかの方法を使用して Oracle Solaris オペレーティングシステム (OS) をインストールおよび構成する手順について説明します。

- Oracle Solaris Live Media インストーラ
- Oracle Solaris 対話型テキストインストーラ
- Oracle Solaris 自動インストーラ (AI) 機能
- Oracle Solaris SCI ツールの対話型システム構成ツール
- sysconfig(1M) コマンド行のシステム構成ツール

いずれの場合も、インストールを完了するには、ネットワーク上のパッケージリポジトリにアクセスできることが必要です。

対象読者

このドキュメントは、Oracle Solaris 11.1 OS をインストールするシステム管理者を対象としています。

内容の紹介

このドキュメントは、次に示すパートと章で構成されています。

パート I 「[Oracle Solaris 11.1 のインストールオプション](#)」では、ユーザーのニーズにもっとも適した方法を選択するのに役立つ代替インストール方式について説明します。

パート II 「[インストールメディアを使用したインストール](#)」:

- 第 2 章 「[インストールの準備](#)」
- 第 3 章 「[Live Media の使用](#)」
- 第 4 章 「[テキストインストーラの使用](#)」
- 第 5 章 「[メディアからブートする自動インストール](#)」
- 第 6 章 「[Oracle Solaris インスタンスの構成解除または再構成](#)」

パート III 「インストールサーバーを使用したインストール」では、自動インストールとそれに関連するプロセスやツールについて説明します。

- 第7章「複数のクライアントへの自動インストール」では、AIによって1つのネットワーク内の複数の SPARC および x86 クライアントシステムで手入力なしのインストールが実行される方法について説明します。
- 第8章「インストールサーバーの設定」では、クライアントインストールを管理するために個別のシステムを設定する方法について説明します。
- 第9章「インストールのカスタマイズ」では、クライアントの選択条件をさまざまなインストール手順やシステム構成手順に適用して、それぞれのクライアントシステムが異なる方法でインストールおよび構成されるようにする方法について説明します。
- 第10章「クライアントシステムのプロビジョニング」では、各種クライアント用にカスタムのインストール手順を作成する方法について説明します。
- 第11章「クライアントシステムの構成」では、インストール後にクライアントシステムの構成に必要な情報の指定方法について説明します。
- 第12章「ゾーンのインストールと構成」では、AI クライアントインストールの一環として非大域ゾーンのインストールと構成を指定する方法について説明します。
- 第13章「初回ブート時のカスタムスクリプトの実行」では、クライアントシステムの追加のインストールまたは構成を行うために初回ブート時に実行されるスクリプトの作成方法について説明します。
- 第14章「クライアントシステムのインストール」では、AI クライアントのシステム要件を提供し、各クライアントを適切なネットイメージやインストールおよび構成手順に関連付ける方法について説明します。
- 第15章「自動インストールのトラブルシューティング」では、いくつかの考えられる障害とその回復方法について説明します。

パート IV 「関連タスクの実行」では、インストールに関連する領域について説明します。

- 付録 A 「Oracle Configuration Manager の操作」
- 付録 B 「デバイスドライバユーティリティの使用」

関連情報

『Oracle Solaris 11.1 カスタムインストールイメージの作成』では、Oracle Solaris ディストリビューションコンストラクタツールを使用してインストールイメージをカスタマイズする方法について説明します。

『Oracle Solaris 11.1 ブート環境の作成と管理』では、Oracle Solaris システムで複数のブート環境 (非大域ゾーンを含む) を管理する方法について説明します。

『Oracle Solaris 11.1 でのサービスと障害の管理』では、Oracle Solaris サービス管理機能 (SMF) 機能について説明しています。SMF プロファイルを使用して、システムを構成できます。

『Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新』では、Oracle Solaris Image Packaging System (IPS) 機能および IPS パッケージの検索とインストール方法について説明しています。pkg(5) のマニュアルページでは、Image Packaging System について詳しく説明しています。pkg(1) のマニュアルページでは、IPS パッケージの検索、インストール、更新、確認の方法について詳しく説明しています。

『Oracle Solaris 11.1 パッケージリポジトリのコピーおよび作成』では、Oracle Solaris IPS パッケージリポジトリのローカルコピーの作成方法、または独自のカスタムリポジトリの作成方法について説明しています。

Oracle Solaris 11.1 システムを管理する方法の詳細は、Oracle Solaris 11.1 システム管理ドキュメントを参照してください。

DHCP の構成については、次のリソースを参照してください。

- 『Oracle Solaris 11.1 での DHCP の作業』
- dhcpd.conf(5) のマニュアルページ
- ISC (Internet Systems Consortium) Web サイトの DHCP セクション

『Oracle Solaris 10 JumpStart から Oracle Solaris 11.1 自動インストーラへの移行』では、JumpStart から AI に移行する際に役立つ情報を提供します。どちらも、Oracle Solaris の自動インストール機能です。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照してください。聴覚に障害をお持ちの場合は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

表記上の規則

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

表P-1 表記上の規則

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

Oracle Solaris OS に含まれるシェルで使用する、UNIX のデフォルトのシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例に示されるデフォルトのシステムプロンプトは、Oracle Solaris のリリースによって異なります。

- 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]
```

- Bash シェル、Korn シェル、および Bourne シェル

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

- Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー

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

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

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

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

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

パート I

Oracle Solaris 11.1 のインストールオプション

◆◆◆ 第 1 章

インストールオプションの概要

Oracle Solaris ソフトウェアは、ユーザーのニーズに応じていくつかの異なる方法でインストールできます。この章では、Oracle Solaris のインストールオプションについて説明します。

インストールオプションの比較

次の表では、各種インストールオプションの機能を比較しています。

表1-1 インストールオプション

インストールオプション	最低限の準備	サーバーが必要	単一または複数のシステムへのインストール	パッケージリポジトリからのパッケージのインストール
x86 のみ: 第 3 章「Live Media の使用」	はい	いいえ、メディアからインストールします	単一	いいえ
第 4 章「テキストインストーラの使用」	はい	いいえ、メディアからインストールします	単一	いいえ
66 ページの「ネットワーク経由でのテキストインストールの実行」	いいえ	はい、サーバーからインストールイメージを取得します	単一	はい
第 5 章「メディアからブートする自動インストール」	いいえ	インストールメディアをカスタマイズする場合はサーバーが必要です が、インストールには必要ありません	単一	はい
第 7 章「複数のクライアントへの自動インストール」	いいえ	はい、サーバーが必要です	単一または複数	はい

さらに、カスタム Live Media イメージ、テキストインストーライメージ、自動インストールイメージなど、『Oracle Solaris 11.1 カスタムインストールイメージの作成』のオプションもあります。

シンプルで、事前設定されたインストール

Live Media 上の GUI インストーラやテキストインストーラは、シンプルで、事前設定されたインストーラ方式です。

- どちらのインストーラを使用しても、Oracle Solaris を x86 プラットフォームにインストールできます。テキストインストーラは、SPARC プラットフォームに Oracle Solaris をインストールする場合にも使用できます。
- どちらのインストーラも、最小限のメモリーで実行できます。メモリー要件を確認するには、『Oracle Solaris 11.1 ご使用にあたって』を参照してください。
- どちらのインストーラを使用しても、インストール中にディスクのパーティションを選択、作成、または変更できます。

Live Media には、デスクトップまたはノートパソコンに適したソフトウェアセットが含まれています。テキストインストーラは、汎用的なサーバーシステムにより適した小さなソフトウェアセットをインストールします。

テキストインストーラには、GUI インストーラにはない次の利点があります。

- オペレーティングシステムを SPARC または x86 ベースのシステムにインストールできる。
- グラフィックスカードを搭載していない、または必要としないシステムで使用できる。
- システムの仕様に応じ、要求されるメモリーが GUI インストーラよりも少なく済む。
- ネットワークやネームサービスを手動で構成できる。
- ネットワークが自動インストールを実行するように設定されている場合は、ネットワーク上にインストールサービスを設定し、クライアントシステムがブートしたときにテキストインストールを選択することで、ネットワーク経由でテキストインストールを実行できます。

注-テキストインストーラによってインストールされるパッケージセットは solaris-large-server パッケージセットです。ただし、ネットワーク経由でテキストインストーラを使用する場合は、これとは別の、より小さなパッケージセット solaris-auto-install がインストールされます。インストールされたシステムをブートしたあとで、solaris-large-server パッケージセットをインストールするようにしてください。

- テキストインストーラでは、パーティションの変更に加え、Solaris パーティション内で VTOC スライスを作成および変更を行うこともできます。

シンプルインストールの実行の詳細は、[パート II 「インストールメディアを使用したインストール」](#)を参照してください。

サーバーの設定を必要とするインストール

自動インストーラ (AI) 機能を使用すると、単一または複数のクライアントシステムで Oracle Solaris ソフトウェアの「手入力なし」のインストールを実行できます。

注-インストール処理ではネットワーク上のリポジトリからパッケージを取得するため、各システムでネットワークにアクセスする必要があります。

AI を使用するには、まずネットワーク上にサーバーを設定する必要があります。クライアントシステムがブートしたときに、システムはそのサーバーからインストール仕様を入手し、Oracle Solaris パッケージリポジトリからソフトウェアパッケージを取得します。その後、ソフトウェアがクライアントシステムにインストールされます。

AI は、x86 ベースおよび SPARC ベースの両方のクライアントシステムで「手入力なし」の自動ネットワークインストールを実行できます。インストールクライアントのアーキテクチャー、ディスクおよびメモリの容量、およびその他の特性は異なっても構いません。インストールのネットワーク構成、インストールするパッケージ、およびその他の仕様は、さまざまなものが可能です。

詳細は、[パート III 「インストールサーバーを使用したインストール」](#)を参照してください。

AI サーバーが設定されていると、「手入力なし」のネットワークインストール以外に追加のインストールオプションを 2 つ使用できます。

- ネットワーク経由で対話型テキストインストールを実行するオプションが用意されています。対話型インストールでは、特定のシステムに対してインストール仕様をさらにカスタマイズできます。

詳細は、[66 ページの「ネットワーク経由でのテキストインストールの実行」](#)を参照してください。

- AI の設定には、AI イメージのダウンロードやそれらのネットワーク上またはローカルでの格納などがあります。イメージは、CD、DVD、USB フラッシュドライブ (x86 でのインストールの場合) などのリムーバブルメディアに書き込むことができます。その後、各システムで AI メディアを直接ブートして、自動インストールを開始できます。AI メディアを使用するインストールは対話型ではありません。

手順については、第5章「メディアからブートする自動インストール」を参照してください。

その他のオプション

すでに説明したインストールオプションのほかに、Oracle Solaris オペレーティングシステムのインストールと変更用に次のオプションが用意されています。

カスタムインストールイメージの作成

ディストリビューションコンストラクタツールを使用して、事前構成済みの Oracle Solaris インストールイメージを作成できます。このツールは、カスタマイズされた XML マニフェストファイルを入力として受け取り、そのマニフェストファイルに指定されているパラメータに基づいてインストールイメージを構築します。いずれかのデフォルトインストールイメージに基づいてカスタムイメージを構築できます。たとえば、カスタムのテキストインストーライメージやカスタムの GUI インストーライメージを構築できます。詳細は、『[Oracle Solaris 11.1 カスタムインストールイメージの作成](#)』を参照してください。

インストール済みの Oracle Solaris システムの更新

インストーラを使用して、既存の Oracle Solaris がインストールされているシステムを更新することはできません。代わりに、pkg ユーティリティを使用してパッケージリポジトリにアクセスし、システム用の新規または更新されたソフトウェアパッケージをダウンロードする必要があります。詳細は、『[Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新](#)』を参照してください。

パート II

インストールメディアを使用したインストール

GUI インストーラまたはテキストインストーラのいずれかを使用すると、最低限の準備で Oracle Solaris オペレーティングシステムを単一のシステムにインストールできます。テキストインストールはローカルでもネットワーク経由でも実行できます。さらに、自動インストーラ (AI) 機能を使用する場合は、自動インストールイメージを作成し、それをメディアに書き込み、そのメディアを使用して単一のシステムをインストールできます。また、インストール済みのシステムを構成解除したり、再構成したりするオプションも用意されています。

次の情報を参照してください。

- 第2章 「インストールの準備」
- 第3章 「Live Media の使用」
- 第4章 「テキストインストーラの使用」
- 66 ページの「ネットワーク経由でのテキストインストールの実行」
- 第5章 「メディアからブートする自動インストール」
- 第6章 「Oracle Solaris インスタンスの構成解除または再構成」

インストールの準備

システムをインストールする前に、次の情報を確認してください。

- 31 ページの「Live Media およびテキストインストールのシステム要件」
- 32 ページの「複数のオペレーティングシステムをインストールするためのブート環境の準備」
- 33 ページの「システムのパーティション分割」
- 38 ページの「適切なデバイスドライバがあることの確認」
- 39 ページの「Oracle Configuration Manager の使用」

Live Media およびテキストインストールのシステム要件

次の表に、Live Media インストールイメージまたはテキストインストールイメージを使用して Oracle Solaris 11.1 リリースをインストールするための要件の概要を示します。

要件	説明
メモリー	最新リリースの最小メモリー要件を確認するには、 『Oracle Solaris 11.1 ご使用にあたって』 を参照してください。 注-テキストインストーラに必要なメモリーはLive Media インストーラよりも少なくなります。正確な最小要件は、システムの仕様によって変わります。GUI インストーラを実行するための十分なメモリーがシステムに搭載されていない場合は、代わりにテキストインストーラを使用してください。
ディスク容量	最新リリースのディスク容量要件を確認するには、 『Oracle Solaris 11.1 ご使用にあたって』 を参照してください。

複数のオペレーティングシステムをインストールするためのブート環境の準備

Oracle Solaris を複数のブート環境の一部としてインストールする場合は、各種オペレーティングシステムに関する次の仕様を確認してください。

表 2-1 複数のオペレーティングシステム環境

既存のオペレーティングシステム	説明
Microsoft Windows	<p>Oracle Solaris リリースをインストールするための十分なディスク容量を設定します。このリリースでは、x86 プラットフォーム用の Oracle Solaris は GRand Unified Bootloader の新しいバージョン (GRUB 2) を使用します。Oracle Solaris は Windows を認識して、インストール中に Windows パーティションを変更しないようにします。インストールが完了すると、システムがリブートし、GRUB 2 メニューに Windows と Oracle Solaris の両方のブートエントリが表示されます。</p> <p>GRUB 2 の詳細は、『Oracle Solaris 11.1 システムのブートおよびシャットダウン』の「GRUB 2 の紹介」を参照してください。</p>
Solaris 10 OS	<p>Live Media インストーラを使用して、Oracle Solaris オペレーティングシステムの複数のインスタンスをインストールすることはできません。一方で、テキストインストーラは同じパーティションにある Oracle Solaris オペレーティングシステムの複数のインスタンスをサポートします (それらのインスタンスが異なるスライス上にある場合に限る)。Live Media インストーラとテキストインストーラを使用して、Oracle Solaris の複数のインスタンスがインストールされている既存のシステム上の Solaris 10 1/06 以降のリリースを置き換えることは可能です。</p> <p>注 - 現在のオペレーティングシステムで特定の Solaris VTOC (Volume Table of Contents) スライスを保持する必要がある場合は、テキストインストーラを使用します。</p>
拡張パーティション	<p>拡張パーティション上に別のオペレーティングシステムが存在する場合、インストール中に既存の拡張パーティションを変更する必要はありません。Live Media の GUI インストーラ、テキストインストーラ、または自動インストーラのいずれかを使用して Oracle Solaris をインストールするときに、拡張パーティションを作成、サイズ変更、または削除できます。拡張パーティション内部の論理パーティションに Oracle Solaris をインストールすることも選択できます。</p>

システムのパーティション分割

このセクションでは、インストール前または対話型インストール中にシステムをパーティション分割するためのガイドラインについて説明します。

ディスク全体またはフォーマットされていないディスクにインストールする場合、インストーラはGPTフォーマットを使用します。ただし、既存のGPTパーティションまたはDOSパーティションがデフォルトで保持され、インストーラに表示されるため、既存のパーティションを保持してそこにインストールできません。



注意 - GPT フォーマットは、SPARC プラットフォームでは現在使用できません。

このセクションでは、Solaris VTOC スライスを設定する方法についても説明します。

インストール前にシステムをパーティション分割するためのガイドライン

Live Media ISO イメージまたはテキストインストーライメージから Oracle Solaris をインストールする場合は、ディスク全体を使用することも、パーティションにオペレーティングシステムをインストールすることもできます。また、テキストインストーラをスライスにインストールすることもできます。

インストールの前に、市販の製品またはオープンソースのツールを使用して、Oracle Solaris をインストールするためのパーティションを作成できます。または、Oracle Solaris のインストール中にパーティションを作成できます。x86 ベースのシステムでは、Oracle Solaris インストーラは1つ以上のドライブ上にある複数のオペレーティングシステムのブートをサポートする GRUB 2 を使用します。各種オペレーティングシステムのパーティション分割とインストールを行なったあとは、ブート時に GRUB 2 メニューで適切なメニューエントリを選択することで、いずれかのオペレーティングシステムを配備できます。

GRUB 2 の詳細は、『[Oracle Solaris 11.1 システムのブートおよびシャットダウン](#)』の「[GRUB 2 の紹介](#)」を参照してください。

注 - Linux スワップパーティションを作成する場合は、Linux スワップが Oracle Solaris で使用するのと同じパーティション ID を使用することに注意してください。インストール中に、ディスクパーティション分割の段階で、Linux スワップパーティションを Oracle Solaris パーティションに変更できます。



注意-ハードディスクのパーティション分割を行う前に必ずシステムのバックアップを取ってください。

対話型インストール中にシステムをパーティション分割するためのガイドライン

x86 ベースのシステムでは、GUI インストールまたはテキストインストール中にパーティションを選択、作成、または変更できます。ディスク全体またはフォーマットされていないディスクにインストールする場合、インストーラは GPT フォーマットを使用します。ただし、既存の GPT パーティションまたは DOS パーティションがデフォルトで保持され、インストーラに表示されるため、既存のパーティションを保持してそこにインストールできます。また、テキストインストーラにかぎり、対話型インストール中に VTOC スライスを選択、作成、または変更できます。



注意-GPT フォーマットは、SPARC プラットフォームでは現在使用できません。

Oracle Solaris をインストールするときは、ディスクパーティション分割に関する次の重要な情報に留意してください。

- パーティション分割の次の仕様に注意してください。
 - ディスクに既存の DOS パーティションが含まれている場合は、最大 4 つの DOS プライマリパーティションが表示されます。DOS 拡張パーティションが存在する場合は、その論理パーティションも、拡張パーティション内のディスクの配置順に表示されます。Solaris パーティションは 1 つしか許可されないため、その Solaris パーティションをインストールに使用する必要があります。Solaris パーティションは、拡張パーティション内の論理パーティションである場合があります。
 - ディスクに既存の GPT パーティションが含まれている場合は、それらの GPT パーティションが表示されます。最大 7 つの GPT パーティションがサポートされます。インストール中に 1 つまたは複数の Solaris パーティションを作成できますが、インストール先として 1 つの Solaris パーティションを選択する必要があります。既存の Solaris GPT パーティションが複数存在する場合、デフォルトでは、最初の適切な Solaris GPT パーティションがインストール先として選択されます。
- 次のいずれかに当てはまる場合、インストールによってディスクレイアウト全体が上書きされます。
 - ディスクテーブルが読み込めない。
 - ディスクが事前にパーティション分割されていない。

- インストール用にディスク全体を選択する。
- 既存の Solaris パーティションが存在し、ユーザーがほかの既存のパーティションのいずれにも変更を加えない場合、インストールのデフォルトでは Solaris パーティションのみが上書きされます。そのパーティションは、既存の拡張パーティション内にある論理パーティションである可能性があります。ほかの既存パーティションは変更されません。
- インストールには Solaris パーティションを使用する必要があります。
- ディスクのパーティション分割またはスライスの変更は、インストーラパネルで選択を完了し、インストールが開始するまでは有効になりません。インストール前であればいつでも、変更を取り消して元の設定に戻すことができます。
- 既存のパーティションテーブルを読み取れない場合は、提案されたパーティション分割の情報が表示されます。



注意- この場合は、インストール中にディスク上の既存のデータがすべて破棄されます。

- インストール中、「ディスクをパーティション分割する」オプションを選択した場合は、選択されたディスクの既存のパーティションがディスク上の配置と同じ順序でパネルに表示されます。これらのパーティションの未使用のディスク領域が表示されます。各パーティションのパーティションタイプ、現在のサイズ、および利用可能なディスク容量の最大値も表示されます。拡張パーティションが存在する場合、その論理パーティションも、拡張パーティション内のディスクと同じ順番で表示されます。
- インストールを成功させるために十分な容量のないディスクやパーティションには、そのことを示すラベルが付けられます。

x86: 対話型インストール中のパーティションの設定

x86 プラットフォームへのインストールでは、インストール画面のエントリを直接編集することで、ディスクのパーティション分割を変更できます。インストール処理中に、ソフトウェアをインストールするための最小サイズと推奨される最小サイズも表示されます。

次の表に、ディスクのパーティション分割オプションを示します。この表は、ニーズにもっとも適したオプションを決定するのに役立ちます。

表2-2 対話型インストール中にディスクをパーティション分割するためのオプション

パーティション分割オプション	説明およびユーザーの処理(必要な場合)
既存の Solaris パーティションを使用する。	このオプションでは、既存の Solaris パーティションの現在のサイズを使用して、そのパーティションに Oracle Solaris オペレーティングシステムをインストールします。「ディスクをパーティション分割する」オプションを選択します。ほかの変更は必要ありません。
Solaris パーティションが存在しない場合は、新しい Solaris パーティションを作成する必要がある。	現在、システム上に Solaris パーティションが存在しない場合は、新しい Solaris パーティションを作成する必要があります。そのためには、プライマリパーティションまたは論理パーティションを選択してから、そのタイプを「Solaris」に変更します。インストールの途中で、この変更は既存のパーティションの内容を消去します。
Solaris パーティションに割り当てる領域を増やし、そのパーティションにインストールする。	十分なディスク領域を利用できる場合、Solaris パーティションにソフトウェアをインストールする前に、そのパーティションに割り当てるサイズを増やすことができます。利用可能な領域とは、選択されたパーティションの前後に隣接する未使用の領域を指します。パーティションを拡大する場合、そのパーティションの後ろにある未使用の領域が最初に使用されます。次に、そのパーティションの前にある未使用の領域が使用され、選択されたパーティションの開始シリンダが変わります。
別の Solaris パーティションに Oracle Solaris オペレーティングシステムをインストールする。	別の Solaris パーティションにオペレーティングシステムをインストールできます。別のパーティションを選択し、そのタイプを「Solaris」に変更します。インストール中に、この変更によって以前の Solaris パーティションと新しい Solaris パーティションの両方で、パーティションの既存の内容が消去されます。 注-システムに既存の DOS パーティションがある場合、Solaris パーティションは1つしか許可されません。最初に既存の Solaris パーティションのタイプを「未使用」に変更してから、新しい Solaris パーティションを作成する必要があります。

表 2-2 対話型インストール中にディスクをパーティション分割するためのオプション (続き)

パーティション分割オプション	説明およびユーザーの処理 (必要な場合)
拡張パーティション内に新しい Solaris パーティションを作成する。	<p>拡張パーティション内に新しい Solaris パーティションを作成できます。パーティションのタイプを「拡張」に変更します。拡張パーティションの大きさを変更してから、拡張パーティション内のいずれかの論理パーティションを Solaris パーティションに変更することができます。また論理パーティションを、その論理パーティションを含む拡張パーティションの大きさにまで拡張できます。</p> <p>注 - システムに既存の DOS パーティションがある場合、Solaris パーティションは1つしか許可されません。最初に既存の Solaris パーティションのタイプを「未使用」に変更してから、拡張パーティション内に Solaris パーティションを作成する必要があります。</p>
既存のパーティションを削除する。	<p>既存のパーティションのタイプを「未使用」に変更することによってそのパーティションを削除できます。インストールの途中で、そのパーティションは破棄され、隣接するパーティションのサイズ変更時にその領域を使用できるようになります。</p>

テキストインストール中の VTOC スライスの設定

SPARC プラットフォームへのテキストインストールでは、インストール中に VTOC スライスを変更できます。x86 プラットフォームへのテキストインストールでは、インストール中にパーティションがまだ変更されていない場合、そのパーティションの内部のスライスを変更できます。

VTOC スライスを設定するときは、次のことに留意してください。

- インストーラは既存のスライスを表示します。スライスはその配置と同じ順番で表示されます。各スライスの現在のサイズと、最大の利用可能サイズも表示されます。
- Oracle Solaris は、ZFS ルートプールにインストールする必要があります。デフォルトでは、ルートプールを含むスライスにはインストーラによって `rpool` というラベルが付けられます。ルートプールを含まないスライスにオペレーティングシステムをインストールする場合は、インストーラでそのスライスのタイプを `rpool` に変更します。インストールの途中で、そのスライス上に ZFS ルートプールが作成されます。

注 - `rpool` という名前は1つの ZFS プールにしか付けられないため、デバイス上に `rpool` がすでに存在する場合、インストーラは `rpool#` という形式を使用して新しいプールに名前を付けます。

- スライスのサイズは、最大の利用可能サイズにまで増やすことができます。利用できる領域を増やすには、隣接するスライスを「未使用」に変更し、それによって隣接するスライスでその領域を利用できるようにします。
- スライスを明示的に変更しない場合、インストール中にスライスの内容は保持されます。

次の表は、テキストインストール中のスライス変更に関するオプションについての説明です。

表 2-3 テキストインストール中の VTOC スライス変更のオプション

オプション	説明およびユーザーの処理 (必要な場合)
既存のスライスを使用する	このオプションでは、既存の VTOC スライスの現在のサイズを使用して、そのスライスに Oracle Solaris オペレーティングシステムをインストールします。ターゲットスライスを選択して、そのタイプを「rpool」に変更します。
スライスサイズを変更する	新しく作成する rpool スライスにかぎり、そのサイズを変更できません。新しいサイズをフィールドに入力します。
新しいスライスを作成する	未使用のスライスを選択し、そのタイプを変更します。たとえば、「未使用」から「rpool」に変更します。
既存のスライスを削除する	スライスのタイプを「未使用」に変更します。インストール中にそのスライスは破棄され、その領域が隣接するスライスのサイズ変更で使用できるようになります。

適切なデバイスドライバがあることの確認

Oracle Solaris OS をインストールする前に、システムのデバイスがサポートされているかどうかを調べる必要があります。<http://www.oracle.com/webfolder/technetwork/hcl/index.html> のハードウェア互換リスト (HCL) を確認してください。HCL は、Oracle Solaris オペレーティングシステムと連携して動作することが認証または報告されているハードウェアに関する情報を提供します。

また、インストールの前またはあとで **Oracle Device Detection Tool** を使用して、デバイスドライバが使用可能かどうかを調べることもできます。Oracle Device Detection Tool では、使用しているシステムで検出されたデバイスが、最新のリリースによりサポートされているかどうか報告されます。このツールは、各種の Solaris 10 リリース、Windows、Linux、Mac OS X、および FreeBSD など、多くのシステムで動作します。HCL (<http://www.oracle.com/webfolder/technetwork/hcl/index.html>) には、Oracle Device Detection Tool へのリンクが用意されています。

Oracle Device Detection Tool の使用手順については、39 ページの「[Oracle Device Detection Tool を使用する方法](#)」を参照してください。

注-インストール後は、デバイスドライバユーティリティーを使用して同様のタスクを実行できます。デバイスドライバユーティリティーの詳細は、[付録 B 「デバイスドライバユーティリティーの使用」](#)を参照してください。

▼ Oracle Device Detection Tool を使用する方法

インストールを実行する前またはあとで Oracle Device Detection Tool を使用して、システム上のすべてのデバイスに対するドライバが現在のリリースに含まれているかどうかを調べることができます。

- 1 Web ブラウザで、http://www.oracle.com/webfolder/technetwork/hcl/hcts/device_detect.html にアクセスします。
- 2 「Using Oracle Device Detection Tool」セクションで、「Start Oracle Device Detection Tool」オプションをクリックします。
- 3 ライセンス契約に同意します。
- 4 `ddtool` のダウンロードリンクをクリックします。
- 5 「JavaWS で開く」オプションを選択し、「実行」を選択します。
ツールが実行されますが、システムにはインストールされません。
- 6 必要なドライバが利用可能かどうかを調べる対象の「ターゲットオペレーティングシステム」を選択します。

ヒント-追加情報を表示するには、「ヘルプ」ボタンをクリックします。

Oracle Configuration Manager の使用

この Oracle Solaris リリースでは、Oracle Configuration Manager および Oracle Auto Service Request ユーティリティーがシステムにインストールされる場合は、インストールされるシステムのためにこれらのサービスを構成するように求めるプロンプトが対話型インストール中に表示されます。

- Oracle Configuration Manager は、システムのソフトウェア構成を記述した定期的なデータを Oracle サポート組織に送信します。
- Oracle Auto Service Request は、ハードウェアまたはソフトウェアの問題を示す障害管理アーキテクチャー (FMA) イベントが発生したときに、Oracle サポート組織にデータを送信します。

注- すべてのデータはセキュアモードで送信されます。

対話型インストールを実行するときは、次のオプションが用意されています。

- デフォルトの「サポート登録」インストーラパネルには、匿名の登録アドレスが表示されます。この匿名のアドレスまたは別の電子メールアドレスをパスワードなしで使用すると、匿名のシステム構成が Oracle サポート組織にアップロードされます。ただし、My Oracle Support のログイン情報を指定しなかったため、インストールされたシステムの構成に関する情報が送信されたときに、My Oracle Support は顧客識別情報を一切受け取りません。
- 「サポートの構成」パネルの匿名の電子メールアドレスを My Oracle Support のログイン ID に置き換えて、My Oracle Support のパスワードを追加できます。My Oracle Support で顧客情報を参照し、セキュリティー更新を受け取る場合は、このオプションを使用します。このオプションを使用すると、Oracle Auto Service Request も起動されます。

顧客の構成データが定期的にアップロードされると、顧客サポート担当者はこのデータを分析して、よりよいサービスを提供できます。たとえば、サービスリクエストを登録すると、サポート担当者は構成データを直接そのサービスリクエストに関連付けることができます。そして、顧客サポート担当者はシステムの一覧を表示し、それに従って問題を解決できます。
- 「サポートの構成」パネルの匿名の電子メールアドレスを削除し、そのフィールドを空白のままにした場合は、Oracle Configuration Manager が切断モードで起動します。My Oracle Support にデータは送信されません。このモードでも、データを送信するために Oracle Configuration Manager を手動でアクティブ化できます。たとえば、技術サポート担当者にシステム上のデータを提供するように求められた場合は、Oracle Configuration Manager を手動で使用して、そのデータを提供できます。

Oracle Configuration Manager が切断モードになっていない場合は、初回リブート中に Oracle Configuration Manager サービスが実行され、登録サーバーへのシステムの登録が試みられます。この登録に成功すると、構成情報のアップロードが実行されます。また、登録に成功した時点で内部スケジューラが開始されます。その後、スケジューラの管理のもとで構成データがアップロードされます。それ以降のリブートでは、サービスのブートの一環として構成データが送信されることはありません。サービスは、システムがすでに登録されていることを認識し、単にスケジューラを起動します。スケジューリングは、`/usr/sbin/emCCR` を使用して調整できます。`emCCR(1M)` のマニュアルページと『[Oracle Configuration Manager インストールレーションおよび管理ガイド](#)』を参照してください。

登録の許可を選択したかどうかに関係なく、今後のサポートを容易にするため、あとで Oracle Configuration Manager でシステムを登録または再登録できます。

次のような状況で登録または再登録を選択できます。

- 以前に匿名で登録しました。
- 以前に Oracle Configuration Manager を切断しました。
- My Oracle Support 資格が入力されたときに、Oracle に接続できなかったため、資格を検証できませんでした。たとえば、ネットワークプロキシの要件のため、自動登録を完了できませんでした。

登録または再登録するには、対話型モードで `configCCR` ユーティリティー (`/usr/sbin/configCCR`) を使用します。たとえば、次のコマンドを実行して既存の構成仕様を削除します。

```
# /usr/lib/ocm/ccr/bin/configCCR -r
```

そして、次のコマンドを実行して Oracle Configuration Manager を手動で構成します。

```
# /usr/lib/ocm/ccr/bin/configCCR -a
```

登録完了後は、次のようにサービスを有効にできます。

```
# svcadm enable system/ocm
```

サービスが有効になると、システムがリブートしたときに Oracle Configuration Manager クライアントが再起動されます。

Oracle Configuration Manager および Oracle Auto Service Request の詳細は次を参照してください。

- [付録 A 「Oracle Configuration Manager の操作」](#)
- [configCCR\(1M\) のマニュアルページ](#)
- 『[Oracle Configuration Manager インストレーションおよび管理ガイド](#)』
- <http://www.oracle.com/support/policies.html>
- <http://www.oracle.com/asr> にある Oracle Auto Service Request のドキュメント

Live Media の使用

この章では、Live Media イメージを使用してインストールを行う方法について説明します。

GUI インストーラを使用したインストール

Oracle Solaris ソフトウェアをインストールするときは、次の情報を考慮してください。

- 31 ページの「[Live Media およびテキストインストールのシステム要件](#)」を参照してください。
- Live Media ISO イメージ上のインストーラは、x86 プラットフォーム専用です。
- 複数のオペレーティングシステムがインストールされる予定のシステムに Oracle Solaris をインストールする場合は、インストールの処理中にディスクをパーティション分割できます。

次に注意してください。

- ディスク全体またはフォーマットされていないディスクにインストールする場合、インストーラは GPT フォーマットを使用します。ただし、既存の GPT パーティションまたは DOS パーティションがデフォルトで保持され、インストーラに表示されるため、既存のパーティションを保持してそこにインストールできます。詳細は、34 ページの「[対話型インストール中にシステムをパーティション分割するためのガイドライン](#)」を参照してください。



注意 - GPT フォーマットは、SPARC プラットフォームでは現在使用できません。

- このリリースでは、x86 プラットフォーム用の Oracle Solaris は GRand Unified Bootloader の新しいバージョン (GRUB 2) をインストールします。GRUB 2 については、『Oracle Solaris 11.1 システムのブートおよびシャットダウン』の「GRUB 2 の紹介」を参照してください。

必要な場合は、インストールの前にサードパーティーまたはオープンソースのパーティション分割ツールを使用して、新しいパーティションを作成したり、既存のパーティションを調整したりできます。33 ページの「インストール前にシステムをパーティション分割するためのガイドライン」を参照してください。

- このリリースでは、iSCSI ターゲットがブートディスクとして機能でき、iSCSI ブートに必要なサポートがシステムに存在する場合は、GUI インストーラを使用して iSCSI ターゲットに Oracle Solaris オペレーティングシステムをインストールできます。

システムで iSCSI ディスクの自動検出がサポートされている場合は、インストーラによってそのオプションが提供されます。あるいは、インストール画面で値を手動入力して iSCSI ターゲットを指定することもできます。

詳細は、この章のインストール手順を参照してください。iscsiadm(1M) のマニュアルページも参照してください。

- GUI インストーラはオペレーティングシステムをアップグレードできません。ただし、Oracle Solaris オペレーティングシステムのインストール後に Image Packaging System を使用すると、システム上のパッケージのうち、利用可能な更新があるすべてのパッケージを更新できます。『Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新』を参照してください。
- GUI インストーラは、ディスク全体またはディスク上の Oracle Solaris x86 パーティションに対して初期インストールを実行できます。



注意-インストールによって、ターゲットデバイス上のソフトウェアとデータがすべて上書きされます。

GUI インストーラでのデフォルト設定

Live Media 上の GUI インストーラによって使用されるデフォルトのネットワーク設定とセキュリティー設定は次のとおりです。

- Oracle Solaris のネットワークは、DHCP を使用して自動的に設定され、DNS (Domain Name System) の解決が行われます。
DNS ドメインとサーバーのインターネットプロトコル (Internet Protocol, IP) アドレスは DHCP サーバーから取得されます。
- 自動ネットワーク接続によって、アクティブなインタフェース上での IPv6 自動構成が有効になります。
- NFSv4 ドメインは動的に派生されます。

▼ GUI インストールを準備する方法

GUI インストールを実行する前に、次のタスクを完了します。

- 1 **Live Media** がない場合は、**Live Media ISO** イメージをダウンロードします。
Oracle Solaris の Live Media ISO イメージをダウンロードするには、<http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html> にアクセスします。

注-あるいは、イメージを USB フラッシュドライブに書き込む場合は、USB イメージをダウンロードします。

イメージをダウンロードしたら、イメージを CD、DVD、USB フラッシュドライブなどのリムーバブルメディアにコピーします。

注-USB イメージの場合は、イメージを USB フラッシュドライブにコピーするために `usbcopy` ユーティリティが必要です。このユーティリティは、`pkg:/install/distribution-constructor` パッケージをインストールすることによって、システムに追加できます。

- 2 使用しているシステムでインストーラを実行するための要件と制限事項を確認します。
 - a. すべての必要なシステム要件が満たされていることを確認します。
[31 ページの「Live Media およびテキストインストールのシステム要件」](#) を参照してください。
 - b. 必要なデバイスドライバがすべて揃っていることを確認します。
[38 ページの「適切なデバイスドライバがあることの確認」](#) を参照してください。
- 3 **Oracle Solaris** オペレーティングシステムのインストールに関する次のいずれかのオプションを選択します。
 - このインストールによってシステム上の唯一のオペレーティングシステムが提供される場合は、[43 ページの「GUI インストーラを使用したインストール」](#) を参照してください。
 - 複数のオペレーティングシステムのインストールをサポートする環境を設定する場合は、次の手順を実行します。
 - a. [32 ページの「複数のオペレーティングシステムをインストールするためのブート環境の準備」](#) に記載されている仕様を確認します。

- b. システムをバックアップします。
- c. インストールの前にシステムをパーティション分割する場合は、[33 ページ](#)の「[システムのパーティション分割](#)」を参照してください。

▼ GUI インストールを実行する方法

- 1 インストールメディアを挿入して、システムをブートします。

Live Media では、GRUB2 メニューが表示されたときに、別のオプションを選択しないかぎり、デフォルトのエントリが自動的に使用されます。

注-システムのグラフィックカードが Live Media でサポートされていないか、またはシステムにグラフィックカードが搭載されていない場合、Live Media を挿入するとシステムはコンソールモードでブートします。この場合、GUI インストールは実行できません。[50 ページ](#)の「[システムがコンソールモードでブートした場合に実行する操作](#)」を参照してください。

- Live Media へのログインのプロンプトが表示されたら、ユーザー名とパスワードの両方に `jack` を指定してください。
- `root` パスワードは `solaris` です。

- 2 キーボードと言語を選択するか、またはデフォルトの英語オプションを受け入れます。

注-言語とキーボードの選択により、インストーラとインストールされたシステムのデフォルトが設定されます。ロケールは、インストールされたシステムのログインパネルで変更できます。

- 3 インストールに必要な、不足しているドライバをすべてインストールします。

Live Media からブートした場合、不足しているドライバが存在すると、プロンプトが表示されます。デバイスドライバユーティリティの使用方法に従い、インストールに必要なドライバをすべて見つけて、インストールします。

- 4 Live Media のデスクトップで、「Oracle Solaris のインストール」アイコンをダブルクリックして GUI インストーラを起動します。

- 5 開始パネルで、「次へ」を選択します。

- 6 「ディスク検出」パネルで、インストーラが検出するディスクのタイプを選択します。

- 「ローカルディスク」 - これは、コンピュータに接続されているディスク (内蔵および外付けハードディスクを含む) のデフォルトのオプションです。
- 「iSCSI」 - iSCSI 標準を使用してネットワーク経由でアクセスできるリモートディスクをインストーラで検索する場合は、このオプションを選択します。追加のフィールドが次のように表示されます。
 - 「DHCP 自動検出を使用」 - システムで iSCSI ディスクの自動検出がサポートされている場合は、このオプションが有効になります。このオプションを選択すると、条件フィールドに自動検出から返された値が取り込まれます。「検索条件の指定」オプションを選択すると、これらの値をさらに細かく調整できます。
 - 「検索条件の指定」 - このオプションを選択して、iSCSI 検索の値を手動で設定できます。

ターゲット IP	iSCSI ターゲットの IP アドレス。0-255 の範囲の 4 つの数字を入力する必要があります。この IP アドレスのシステムがオンラインであり、このシステムからアクセス可能である必要があります。これらのフィールドは必須です。
LUN	指定した IP アドレスに配置されている iSCSI デバイスの論理ユニット番号。LUN は多くの場合、「0」や「1」などの数値です。このフィールドは省略可能です。
ターゲット名	iSCSI ターゲットの iSCSI 修飾名 (IQN) 形式の名前。このフィールドは省略可能です。
ポート	iSCSI デバイスを検出するために、指定した IP アドレスと組み合わせて使用されるポート番号。デフォルト値の「3260」は、iSCSI に通常使用されるポートです。このフィールドは省略可能です。
イニシエータ名	iSCSI 検出セッション用に設定されるイニシエータノード名。iSCSI ブートの場合は、イニシエータノード名を変更できないため、このフィールドは非表示になっています。このフィールドは省略可能です。
CHAP の使用	CHAP (チャレンジハンドシェイク認証プロトコル) 認証の詳細を入力する場合は、このオプションを選択します。
名前	認証に使用される CHAP 名。このフィールドは省略可能です。
パスワード	認証用の CHAP シークレットの値。入力する場合は、この値の長さを 12 から 16 文字にする必要があります。このフィールドは省略可能です。

「iSCSI」オプションを選択した場合は、「次へ」を選択したときに入力した詳細が検証され、遅延が発生する可能性があります。iSCSI LUN を検出できない場合は、エ

ラーが表示されます。有効な条件を入力するか、または「iSCSI」の選択を解除するかどちらかによって問題が解決されるまで、処理を続行できません。

- 7 「ディスク選択」パネルで、複数のインストール先が表示されている場合は、インストール先を選択するか、またはデフォルトを受け入れます。次に、オペレーティングシステムをディスク全体にインストールするか、ディスク上のパーティションにインストールするかを指定します。

ディスク全体またはフォーマットされていないディスクにインストールする場合、インストーラは GPT フォーマットを使用します。ただし、既存の GPT パーティションまたは DOS パーティションがデフォルトで保持され、インストーラに表示されるため、既存のパーティションを保持してそこにインストールできません。



注意 - GPT フォーマットは、SPARC プラットフォームでは現在使用できません。

次に注意してください。

- ディスクに既存の DOS パーティションが含まれている場合は、最大 4 つの DOS プライマリパーティションが表示されます。DOS 拡張パーティションが存在する場合は、その論理パーティションも、拡張パーティション内のディスクの配置順に表示されます。Solaris パーティションは 1 つしか許可されないため、その Solaris パーティションをインストールに使用する必要があります。Solaris パーティションは、拡張パーティション内の論理パーティションである場合があります。
- ディスクに既存の GPT パーティションが含まれている場合は、それらの GPT パーティションが表示されます。最大 7 つの GPT パーティションがサポートされます。インストール中に 1 つまたは複数の Solaris パーティションを作成できますが、インストール先として 1 つの Solaris パーティションを選択する必要があります。既存の Solaris GPT パーティションが複数存在する場合、デフォルトでは、最初の適切な Solaris GPT パーティションがインストール先として選択されます。

パーティションのレイアウトを変更することもできます。手順については、[34 ページの「対話型インストール中にシステムをパーティション分割するためのガイドライン」](#)を参照してください。

インストールのこのフェーズでは、いつでも設定を元に戻すことができます。



注意 - 既存のパーティションテーブルを読み取れない場合は、提案されたパーティション分割がパネルに表示されます。この場合は、インストール中にディスク上のデータがすべて破棄されます。

- 8 ターゲットのタイムゾーンを選択します。次に、現在の現地時間に一致するように日付と時間を調整します。

可能な場合、インストーラは、システム内に設定されているタイムゾーンを初期のデフォルトとして使用します。地図で所在地を選択すると、インストーラはその情報を使用して日付、時間、およびタイムゾーンを設定します。

- 9 ユーザー設定を完了します。

- ユーザー名とパスワードを入力します。

ユーザーアカウント設定を完了するには、ログイン名とパスワードが必要です。ログイン名は英文字で始まる必要があり、英文字と数字のみを含めることができます。

注-作成したユーザーアカウントには管理者権限が割り当てられます。

インストールされたシステムでは、デフォルトで初期の root パスワードが、ここで入力したユーザーアカウントパスワードになります。root パスワードの初回使用時に、パスワードの変更を求めるプロンプトが表示されます。

- コンピュータ名を入力するか、デフォルトを使用します。このフィールドを空白にすることはできません。

- 10 「サポートの構成」パネルで、**OCM** および **ASR** の匿名登録を受け入れることができます。

デフォルトの「サポートの構成」インストーラパネルには、匿名の登録アドレスが表示されます。パスワードなしでこの匿名アドレスを使用した場合は、Oracle サポート組織にシステム構成がアップロードされたときに、My Oracle Support (MOS) はインストール済みシステムの構成に関する情報を受け取りますが、顧客情報は一切受け取りません。

あるいは、次のようにしてセキュリティ更新に登録したり、OCM を切断したりすることもできます。

- 「サポートの構成」パネルの匿名の電子メールアドレスを My Oracle Support のログイン ID に置き換えて、My Oracle Support のパスワードを追加できます。My Oracle Support で顧客情報を参照し、セキュリティ更新を受け取る場合は、このオプションを使用します。このオプションを使用すると、ASR も起動します。
- 「サポートの構成」パネルの匿名の電子メールアドレスを削除し、そのフィールドを空白のままにした場合は、OCM が切断モードで起動します。My Oracle Support にデータは送信されません。匿名の電子メールアドレスを削除し、それを MOS のログイン ID 以外の別の電子メールアドレスで置き換えた場合、OCM は未認証モードで Oracle サポートにデータを送信します。

詳細は、39 ページの「Oracle Configuration Manager の使用」を参照してください。

- 11 インストール仕様を確認します。
「インストールのサマリー」パネルで、仕様を確認します。インストールを開始する前に、必要に応じて以前の手順に戻って変更を加えます。
- 12 指定した仕様を使ってシステムをインストールします。
Oracle Solaris のインストール処理が開始します。



注意-インストールの進行中に処理を中断しないでください。インストールを完了しないと、ディスクが未確定な状態のままになります。

- 13 インストールログを確認します。
「完了」パネルで、インストールログを表示および確認できます。
- 14 システムをリブートするか、またはインストーラを終了してシステムをシャットダウンします。
インストールが成功したら、システムをリブートするか、インストーラを終了してシステムをシャットダウンします。

次回システムのブートを開始するとき Live Media を取り出してください。または、GRUB メニューの「Boot from Hard Disk」オプションを選択してください。

インストールに失敗した場合は、インストールログを表示して、インストーラを終了できます。

システムがコンソールモードでブートした場合に実行する操作

システムのグラフィックカードが Live Media でサポートされていないか、またはシステムにグラフィックカードが搭載されていない場合、Live Media を挿入するとシステムはコンソールモードでブートします。この場合、GUI インストールは実行できません。

次の2つの方法を選択できます。

- Live Media ISO イメージの代わりにテキストインストーライメージを使用します。
ローカルコンソールで、ネットワークにアクセスせずにテキストインストーラを実行できます。第4章「テキストインストーラの使用」を参照してください。

- 51 ページの「システムがコンソールモードでブートした場合に **Live Media** から **Oracle Solaris** をインストールする方法」の説明に従って、リモートインストールを実行します。

注-この方法を使用する場合、テキストインストーライメージをダウンロードする必要はありません。ただし、リモート `ssh` アクセスと、ターゲットシステムで `X` サーバーが動作していることが必要です。

▼ システムがコンソールモードでブートした場合に **Live Media** から **Oracle Solaris** をインストールする方法

始める前に この手順では、**Live Media** がブートされたシステム (ターゲットシステム) とインストールの実行元となるリモートシステムの2つのシステムがネットワーク上に必要です。両方のシステムをネットワークにアクセスできるようにします。2つのシステムが同じサブネット上に存在する必要はありません。ただし、ターゲットシステムがリモートシステムから到達可能である必要があります。また、リモートシステムでは、グラフィカルデスクトップをサポートする OS を実行している必要があります。

- 1 インストール先のシステムに **Live Media** を挿入してから、システムをブートします。
- 2 コンソールログインで、デフォルトのログインおよびパスワードを入力します。
Oracle Solaris のデフォルトのユーザーログインおよびパスワードは `jack` です。

- 3 **root** ユーザーになります。

```
$ su root  
Password: solaris
```

`root` パスワードは `solaris` です。

- 4 `ssh` リモートログインプログラムのサービスを使用可能にします。

```
# svcadm enable ssh:default
```
- 5 **DHCP** によってターゲットシステムに割り当てられている **IP** アドレスを表示します。

```
# ifconfig -a
```

- 6 リモートシステムで端末ウィンドウを開き、次のように入力します。

```
$ ssh -X IP-address-of-target -l jack
```

IP-address-of-target は、ターゲットシステムで実行した `ifconfig -a` コマンドの出力です。

このコマンドをリモートシステムで実行すると、Secure Shell が開かれ、ターゲットシステムにアクセスして GUI インストーラを使用できるようになります。

7 root 役割になります。

```
$ su root
Password: solaris
```

注-インストール前のデフォルトの root パスワードは「solaris」です。

8 GUI インストーラを実行します。

```
# /usr/bin/gui-install
```

注-この方法を使用すると、インストーラのグラフィック表示が不完全な場合があります。

9 インストールが完了したら、ターゲットシステムをリブートします。

Live Media インストール後のソフトウェアの追加

オペレーティングシステムをインストールしたあとでソフトウェアパッケージを追加するには、`pkg(1)` のマニュアルページの説明に従って `pkg` コマンドを使用します。または、Oracle Solaris パッケージマネージャーの GUI ツールを使用して追加のソフトウェアをインストールできます。デスクトップメニューで、「システム」⇒「管理」⇒「パッケージマネージャー」の順に選択します。

注-パッケージのインストール、更新、およびアンインストールには強力な権限が必要です。詳細は、『[Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新](#)』の「インストール権限」を参照してください。

`pkg` コマンドまたはパッケージマネージャーツールを使用して、インストールするパッケージの名前を見つけ、パッケージの詳細情報を入手して、パッケージをインストールします。

必要に応じて、新しいインストールで問題が発生した場合でも現在のイメージを使い続けることができるように、新しいブート環境にインストールすることもできます。

`pkg install` コマンドでは、最初に `-nv` オプションを使用して、実際にパッケージをインストールする前にパッケージのインストールがどのように行われるのかを確認するようにしてください。インストールするパッケージを特定し、`-nv` オプションを指定した `pkg install` コマンドからの出力を確認したあと、次のようなコマンドを発行して追加のソフトウェアをインストールします。

```
$ pfexec pkg install --be-name new-BE-name package-name
```

このコマンド例には、新しいブート環境の作成を求めるオプションが含まれ、インストールするパッケージが指定されています。

GUI デスクトップがなく、Oracle Solaris デスクトップをインストールする場合は、`solaris-desktop` パッケージをインストールします。

テキストインストーラの使用

個々の SPARC および x86 クライアントシステムで対話型のテキストインストールを実行できます。さらに、自動インストール用にネットワークを設定している場合、ネットワーク経由でテキストインストールを実行できます。

テキストインストーラを使用したインストール

Oracle Solaris オペレーティングシステムをインストールする場合は、次の情報を考慮してください。

- 31 ページの「[Live Media およびテキストインストールのシステム要件](#)」を参照してください。
- 複数のオペレーティングシステムがインストールされる予定の x86 ベースシステムに Oracle Solaris をインストールする場合は、インストールの処理中にディスクをパーティション分割できます。

次に注意してください。

- ディスク全体またはフォーマットされていないディスクにインストールする場合は、インストーラは GPT フォーマットを使用します。ただし、既存の GPT パーティションまたは DOS パーティションがデフォルトで保持され、インストーラに表示されるため、既存のパーティションを保持してそこにインストールできます。詳細は、34 ページの「[対話型インストール中にシステムをパーティション分割するためのガイドライン](#)」を参照してください。



注意 - GPT フォーマットは、SPARC プラットフォームでは現在使用できません。

- このリリースでは、Oracle Solaris インストーラは x86 システムに対して GRUB 2 を使用します。GRUB 2 は、1 つ以上のドライブ上にある複数のオペレーティングシステムのブートをサポートします。GRUB 2 については、『Oracle Solaris 11.1 システムのブートおよびシャットダウン』の「GRUB 2 の紹介」を参照してください。

インストールの前にオープンソースまたはサードパーティーのパーティション分割ツールを使用して、新しいパーティションを作成したり、既存のパーティションを調整したりすることもできます。33 ページの「インストール前にシステムをパーティション分割するためのガイドライン」を参照してください。

- Oracle Solaris インストーラでオペレーティングシステムのアップグレードを行うことはできません。ただし、Oracle Solaris オペレーティングシステムのインストール後に Image Packaging System を使用すると、システム上のパッケージのうち、利用可能な更新があるすべてのパッケージを更新できます。『Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新』を参照してください。
- このリリースでは、iSCSI ターゲットがブートディスクとして機能し、iSCSI ブートに必要なサポートがシステムに存在する場合は、テキストインストーラを使用して iSCSI ターゲットに Oracle Solaris オペレーティングシステムをインストールできます。

システムで iSCSI ディスクの自動検出がサポートされている場合は、インストーラによってそのオプションが提供されます。あるいは、インストール画面で値を手動入力して iSCSI ターゲットを指定することもできます。

詳細は、この章のインストール手順を参照してください。iscsiadm(1M) のマニュアルページも参照してください。

- テキストインストーラは、ディスク全体、Oracle Solaris x86 パーティション、または SPARC スライスに対して初期インストールを実行できます。



注意-インストールによって、ターゲットデバイス上のソフトウェアとデータがすべて上書きされます。

- Live Media には、デスクトップまたはノートパソコンに適したソフトウェアセットが含まれています。テキストインストーラは、汎用的なサーバーシステムにより適した小さなソフトウェアセットをインストールします。具体的には、テキストインストーラでは GNOME デスクトップはインストールされません。テキストインストーラを使用したインストール後に追加のパッケージをインストールする場合は、65 ページの「テキストインストール後のソフトウェアの追加」を参照してください。

テキストインストーラを使用したネットワーク構成

テキストインストーラのネットワーク用パネルでは、ユーザーに対して次のオプションが用意されています。

- 自動 - Live Media インストーラの方法と同様に、自動 NCP を使用してターゲットシステムを構成します。
- 手動 - 「DefaultFixed」 NCP を選択し、1つのネットワークインタフェース (NIC) の静的な IPv4 構成を提供します。その選ばれた NIC に対して、IPv4 のデフォルトルートと IPv6 の自動構成が有効になります。このオプションは、DNS、NIS、および LDAP ネームサービスの手動構成も提供します。
- なし - 「DefaultFixed」 NCP を選択し、ループバックインタフェースのみを構成します。

▼ テキストインストーラを準備する方法

テキストインストーラを実行する前に、次のタスクを完了します。

- 1 テキストインストーライメージがない場合は、そのイメージをダウンロードします。

Oracle Solaris テキストインストーラの ISO イメージをダウンロードするには、<http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html> にアクセスします。

注 - イメージを USB フラッシュドライブに書き込む場合は、USB イメージをダウンロードします。

イメージをダウンロードしたら、イメージを CD、DVD、USB フラッシュドライブなどのリムーバブルメディアにコピーします。

注 - USB イメージの場合は、イメージを USB フラッシュドライブにコピーするために `usbcopy` ユーティリティーが必要です。このユーティリティーは、`pkg:/install/distribution-constructor` パッケージをインストールすることによって、システムに追加できます。

- 2 使用しているシステムでインストーラを実行するための要件と制限事項を確認します。
 - a. すべての必要なシステム要件が満たされていることを確認します。
31 ページの「[Live Media およびテキストインストールのシステム要件](#)」を参照してください。
 - b. 必要なデバイスドライバがすべて揃っていることを確認します。
38 ページの「[適切なデバイスドライバがあることの確認](#)」を参照してください。
- 3 複数のオペレーティングシステムのインストールをサポートする環境を設定する場合は、次の手順を実行します。
 - a. 32 ページの「[複数のオペレーティングシステムをインストールするためのブート環境の準備](#)」に記載されている仕様を確認します。
 - b. システムをバックアップします。
 - c. インストールの前にシステムをパーティション分割する場合は、[第2章「インストールの準備」](#)のガイドラインを確認してください。
特に、Oracle Solaris をパーティションまたはスライスに設定およびインストールすることを計画していて、まだ実行していない場合は、33 ページの「[インストール前にシステムをパーティション分割するためのガイドライン](#)」の情報を確認してください。

▼ テキストインストーラを実行する方法

- 1 インストールメディアを挿入し、システムをブートしてから、予備的なキーボードと言語の選択を行います。

注- 言語とキーボードの選択により、インストーラとインストールされたシステムのデフォルトが設定されます。

- 2 (オプション) 必要なドライバをインストールするには、インストールメニューでオプション2を選択します。
デバイスドライバユーティリティーの使用手順については、253 ページの「[デバイスドライバユーティリティーを起動する方法](#)」を参照してください。ドライバのインストールが完了したら、テキストインストーラを再起動して、インストールメニューに戻ります。

- 3 インストールメニューの最初のオプションを選択して、インストールを開始します。

```
Welcome to the Oracle Solaris 11.1 installation menu
```

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun-color)
5 Reboot
```

```
Please enter a number [1]:
```

注- インストーラパネル間の移動にはキーボードを使用してください。マウスは使用できません。各パネルに表示された主要なコマンドを参照してください。詳細は、オンラインヘルプを参照してください。

- 4 開始パネルを過ぎて先に進みます。
- 5 「ディスク検出」パネルで、インストーラが検出するディスクのタイプを選択できません。
- 「ローカルディスク」 - これは、コンピュータに接続されているディスク (内蔵および外付けハードディスクを含む) のデフォルトのオプションです。
 - 「iSCSI」 - iSCSI 標準を使用してネットワーク経由でアクセスできるリモートディスクをインストーラで検索する場合は、このオプションを選択します。追加のフィールドが次のように表示されます。
 - 「DHCP 自動検出を使用」 - システムで iSCSI ディスクの自動検出がサポートされている場合は、このオプションが有効になります。このオプションを選択すると、条件フィールドに自動検出から返された値が取り込まれます。「検索条件の指定」オプションを選択すると、これらの値をさらに細かく調整できます。
 - 「検索条件の指定」 - このオプションを選択して、iSCSI 検索の値を手動で設定できます。

ターゲット IP iSCSI ターゲットの IP アドレス。0-255 の範囲の 4 つの数字を入力する必要があります。この IP アドレスのシステムがオンラインであり、このシステムからアクセス可能である必要があります。これらのフィールドは必須です。

ターゲット LUN 指定した IP アドレスに配置されている iSCSI デバイスの論理ユニット番号。LUN は多くの場合、「0」や「1」などの数値です。このフィールドは省略可能です。

ターゲット名 iSCSI ターゲットの iSCSI 修飾名 (IQN) 形式の名前。このフィールドは省略可能です。

ポート	iSCSI デバイスを検出するために、指定した IP アドレスと組み合わせで使用されるポート番号。デフォルト値の「3260」は、iSCSI に通常使用されるポートです。このフィールドは省略可能です。
イニシエータ名	iSCSI 検出セッション用に設定されるイニシエータノード名。iSCSI ブートの場合は、イニシエータノード名を変更できないため、このフィールドは非表示になっています。このフィールドは省略可能です。
CHAP の使用	CHAP (チャレンジハンドシェイク認証プロトコル) 認証の詳細を入力する場合は、このオプションを選択します。
名前	認証に使用される CHAP 名。このフィールドは省略可能です。
パスワード	認証用の CHAP シークレットの値。入力する場合は、この値の長さを 12 から 16 文字にする必要があります。このフィールドは省略可能です。

「iSCSI」オプションを選択した場合は、「次へ」を選択したときに入力した詳細が検証され、遅延が発生する可能性があります。iSCSI LUN を検出できない場合は、エラーが表示されます。有効な条件を入力するか、または「iSCSI」の選択を解除するかのどちらかによって問題が解決されるまで、処理を続行できません。

- 6 「ディスク選択」パネルで、複数のターゲットディスクが表示されている場合は、ターゲットディスクを選択するか、またはデフォルトを受け入れます。
- 7 オペレーティングシステムをディスク全体にインストールするか、ディスク上のパーティションまたはスライスにインストールするかを選択します。
 - ディスク全体
 - x86 パーティション
 - SPARC スライス
- 8 (省略可能)一連のターゲット選択パネルで、パーティションまたはスライスのレイアウトを変更することもできます。
インストールパネルを進めているときはいつでも、元の設定に戻すことができます。



注意-既存のパーティションテーブルを読み取れない場合は、提案されたパーティション分割がパネルに表示されます。この場合は、インストール中にディスク上のデータがすべて破棄されます。

ディスク全体またはフォーマットされていないディスクにインストールする場合、インストーラは GPT フォーマットを使用します。ただし、既存の GPT

パーティションまたはDOSパーティションがデフォルトで保持され、インストーラに表示されるため、既存のパーティションを保持してそこにインストールできます。



注意-GPTフォーマットは、SPARCプラットフォームでは現在使用できません。

次に注意してください。

- ディスクに既存のDOSパーティションが含まれている場合は、最大4つのDOSプライマリパーティションが表示されます。DOS拡張パーティションが存在する場合は、その論理パーティションも、拡張パーティション内のディスクの配置順に表示されます。Solarisパーティションは1つしか許可されないため、そのSolarisパーティションをインストーラに使用する必要があります。Solarisパーティションは、拡張パーティション内の論理パーティションである場合があります。
- ディスクに既存のGPTパーティションが含まれている場合は、それらのGPTパーティションが表示されます。最大7つのGPTパーティションがサポートされます。インストール中に1つまたは複数のSolarisパーティションを作成できますが、インストール先として1つのSolarisパーティションを選択する必要があります。既存のSolaris GPTパーティションが複数存在する場合、デフォルトでは、最初の適切なSolaris GPTパーティションがインストール先として選択されます。

パーティション分割の詳細な手順については、[34 ページの「対話型インストール中にシステムをパーティション分割するためのガイドライン」](#)を参照するか、インストーラのオンラインヘルプを参照してください。

- 9 ネットワーク上でシステムを識別するためのコンピュータ名を指定します。
- 10 次のいずれかのオプションを選択して、有線のEthernetネットワーク接続がどのように構成されるかを指定します。
 - DHCPを使用してネットワーク接続を構成するには、「自動」を選択します。インストーラでは続けて「タイムゾーン」パネルが表示されます。
 - ネットワーク仕様を指定するには、「手動」を選択し、次のように続けます。
 - a. 複数のインタフェースがある場合は、構成される接続を選択します。
 - b. 接続設定を入力するか、またはインストーラで検出され提供されたデフォルト情報を受け入れます。

注-IPアドレスとネットマスクは必須フィールドです。ルーターは省略可能なフィールドです。

- c. システムで**DNS** ネームサービスを使用するかどうかを指定します。
- d. 「**DNSの構成**」を選択した場合は、次を実行します。
 - i. システムによって使用される**DNS** サーバーの**IP**アドレスを1つ以上入力します。
 - ii. **DNS** クエリーが行われる際に検索されるドメイン名を1つ以上入力します。
- e. システムが**LDAP** ネームサービス、**NIS** ネームサービス、または「なし」のどれを使用するかを指定します。
 - 前の手順で**DNS** を選択した場合は、**DNS** に加えて**LDAP** または**NIS** が代替ネームサービスとして設定されます。
 - 前の手順で**DNS** を選択しなかった場合は、**LDAP** または**NIS** が唯一のネームサービスとして設定されます。
 - **LDAP** プロファイルのないシステムで**LDAP** を構成する場合は、**LDAP** を選択するのではなく「なし」を選択します。その後、インストールが完了したあとで**LDAP** を手動で構成します。

注-ネットワークのネームサービスが選択されていない場合は、`/etc/hosts(4)` などの標準の名前ソースファイルを使用してネットワーク名を解決できます。詳細は、`nsswitch.conf(4)` のマニュアルページを参照してください。

- f. 選択した代替ネームサービス用に、システムが属するドメインを指定します。

注-ドメイン名を確認するには、システム管理者に問い合わせてください。または、以前にインストールされたシステム上で `domainname` コマンドを使用してください。

- g. **LDAP** を唯一のネームサービスとして、または**DNS** とともに追加のネームサービスとして選択した場合は、次のように**LDAP** 仕様を指定します。
 - i. システムで**LDAP** ネームサービスを構成するために使用される**LDAP** プロファイルを指定します。

- ii. LDAP プロファイルサーバーの IP アドレスを入力します。
- iii. LDAP 検索ベースを指定するか、またはデフォルトの検索ベースを受け入れます。
- iv. LDAP プロキシバインド情報を入力するかどうかを指定します。

注-プロファイルによってプロキシ資格レベルが指定され、認証方法が「なし」でない場合は、プロキシバインド情報を入力する必要があります。その情報を省略した場合、LDAP は初期化されません。

- v. 必要に応じて、LDAP プロキシ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します。
- h. NIS を唯一のネームサービスとして、または DNS とともに追加のネームサービスとして選択した場合は、NIS 仕様を指定します。
ソフトウェアにネームサーバーを検索させることも、自分でネームサーバーを指定することもできます。次の2つの選択肢のどちらかを選択します。

- 「1つ検索」を選択します。

注-ソフトウェアは、ネームサーバーがローカルのサブネット上にある場合にのみ検出できます。

- 「1つ指定」を選択して、サブパネルにネームサーバーのホスト名または IP アドレスを入力します。

一連のネットワーク構成パネルが完了したら、インストーラによって一連のタイムゾーンパネルおよび「日付と時間」パネルが表示されます。

- インストール中にネットワークを構成しないことを指定するには、「なし」を選択します。
インストールでは続けて「タイムゾーン」パネルが表示されます。

- 11 一連のタイムゾーンパネルで、最初にタイムゾーンを選択してから、現地時間に一致するように日付と時間を調整します。

注-デフォルトでは、GMT タイムゾーンが構成されます。

12 「ユーザー」パネルを完了します。

ユーザーアカウントを作成する必要はありませんが、root パスワードを作成する必要があります。

- このパネルでユーザーアカウントを作成する場合は、ユーザーのパスワードと root パスワードの両方を指定する必要があります。

この場合、root はユーザーに割り当てられた役割になります。

ユーザーアカウントを作成するには、ユーザー名とパスワードを入力します。この名前は英文字で始まる必要があります、英文字と数字のみを含めることができます。

- ユーザーアカウントを作成しない場合でも、root パスワードは指定する必要があります。

この場合、root は通常のユーザーとなります。

13 「サポートの構成」パネルで、OCM および ASR の匿名登録を受け入れることができます。

デフォルトの「サポートの構成」インストーラパネルには、匿名の登録アドレスが表示されます。パスワードなしでこの匿名アドレスを使用した場合は、Oracle サポート組織にシステム構成がアップロードされたときに、My Oracle Support (MOS) はインストール済みシステムの構成に関する情報を受け取りますが、顧客情報は一切受け取りません。

あるいは、次のようにしてセキュリティ更新に登録したり、OCM を切断したりすることもできます。

- 「サポートの構成」パネルの匿名の電子メールアドレスを My Oracle Support のログイン ID に置き換えて、My Oracle Support のパスワードを追加できます。My Oracle Support で顧客情報を参照し、セキュリティ更新を受け取る場合は、このオプションを使用します。このオプションを使用すると、ASR も起動します。
- 「サポートの構成」パネルの匿名の電子メールアドレスを削除し、そのフィールドを空白のままにした場合は、OCM が切断モードで起動します。My Oracle Support にデータは送信されません。匿名の電子メールアドレスを削除し、それを MOS のログイン ID 以外の別の電子メールアドレスで置き換えた場合、OCM は未認証モードで Oracle サポートにデータを送信します。

詳細は、39 ページの「Oracle Configuration Manager の使用」を参照してください。

14 インストール仕様を確認します。

「インストールのサマリー」パネルで、仕様を確認します。インストールを開始する前に、必要に応じて以前の手順に戻って変更を加えます。

- 15 指定した仕様を使ってシステムをインストールします。
Oracle Solaris のインストール処理が開始します。



注意-インストールの進行中に処理を中断しないでください。インストールを完了しないと、ディスクが未確定な状態のままになります。

- 16 インストールログを確認します。
「完了」パネルで、インストールログを表示および確認できます。
- 17 リブートするか、またはシェルにアクセスしてシステムをシャットダウンします。

テキストインストール後のソフトウェアの追加

オペレーティングシステムをインストールしたあとでソフトウェアパッケージを追加するには、`pkg(1)` のマニュアルページの説明に従って `pkg` コマンドを使用します。

`pkg` コマンドまたはパッケージマネージャーツールを使用して、インストールするパッケージの名前を見つけ、パッケージの詳細情報を入手して、パッケージをインストールします。

注-パッケージのインストール、更新、およびアンインストールには強力な権限が必要です。詳細は、『[Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新](#)』の「インストール権限」を参照してください。

必要に応じて、新しいインストールで問題が発生した場合でも現在のイメージを使い続けることができるように、新しいブート環境にインストールすることもできます。

`pkg install` コマンドでは、最初に `-nv` オプションを使用して、実際にパッケージをインストールする前にパッケージのインストールがどのように行われるのかを確認するようにしてください。インストールするパッケージを特定し、`-nv` オプションを指定した `pkg install` コマンドからの出力を確認したあと、次のようなコマンドを発行して追加のソフトウェアをインストールします。

```
$ pfexec pkg install package-name
```

`package-name` 変数は、インストールするパッケージの名前に置き換えます。

別の方法として、次のサンプルコマンドを使用して新しいバックアップブート環境を作成し、インストールするパッケージを指定することができます。

```
$ pfexec pkg install --be-name new-BE-name package-name
```

GUI デスクトップがなく、Oracle Solaris デスクトップをインストールする場合は、`solaris-desktop` パッケージをインストールします。

ネットワーク経由でのテキストインストールの実行

ネットワーク経由で自動インストールを実行するようにシステムを設定してある場合は、ネットワーク経由で対話型テキストインストールを実行することもできます。このオプションでは一度に1つのシステムしかインストールできませんが、対話型の選択を使用してインストール仕様を変更することで、各インストールをカスタマイズする機会を得られます。

▼ ネットワーク経由でテキストインストールを実行する方法

- 1 AI クライアントイメージをダウンロードし、そのイメージに基づいてインストールサービスを作成します。

手順については、[パート III 「インストールサーバーを使用したインストール」](#) を参照してください。

- 2 ネットワーク経由でクライアントシステムをブートします。

- SPARC クライアントの場合、OBP プロンプトで次のように入力します。

```
# boot net:dhcp
```

- x86 クライアントの場合、インストールメニューから「1」を選択します。

```
Welcome to the Oracle Solaris 11.1 installation menu
```

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun-color)
5 Reboot
```

```
Please enter a number [1]:
```

- 3 クライアントシステムのテキストインストールを完了します。

手順については、[58 ページの「テキストインストールを実行する方法」](#) を参照してください。

注-テキストインストーラによってインストールされるパッケージセットは `solaris-large-server` パッケージセットです。ただし、ネットワーク経由のテキストインストーラは実際には自動インストールです。自動インストールでは、できるだけ多くの必要なソフトウェアが IPS リポジトリからダウンロードされます。ネットワーク経由でテキストインストーラを使用すると、より小さなパッケージセット `solaris-auto-install` がデフォルトでインストールされます。

インストールされるこのシステムは、きわめて最低限のものです。インストールされたシステムをブートしたあと、場合によっては次に示すように `solaris-large-server` パッケージセットをインストールし、オプションとしてデスクトップをインストールするようにしてください。

パッケージのインストール、更新、およびアンインストールには強力な権限が必要です。詳細は、『[Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新](#)』の「インストール権限」を参照してください。

```
$ pfexec pkg install solaris-desktop
```

```
$ pfexec pkg install solaris-large-server
```

メディアからブートする自動インストール

SPARC システムまたは x86 システムへの Oracle Solaris OS の自動インストールは、ネットワーク経由のブートではなく、メディア上の AI イメージのブートによって開始できます。この章では、メディアから AI クライアントをブートする理由と、そのモードでインストールを実行する方法について説明します。

AI メディアを使用するインストールの概要

AI メディアを使用するインストールによって、次のタスク (オプション) を実行できます。

- AI インストールサーバーになるシステムをインストールします。
- WAN ブートの機能がない SPARC システムをインストールします。
- 障害が発生しているシステムのトラブルシューティングを行います。リムーバブルメディアからシステムをブートしてから、インストールしたシステムを調べ、診断を実行します。

AI メディアを使用するインストールには、次の特徴があります。

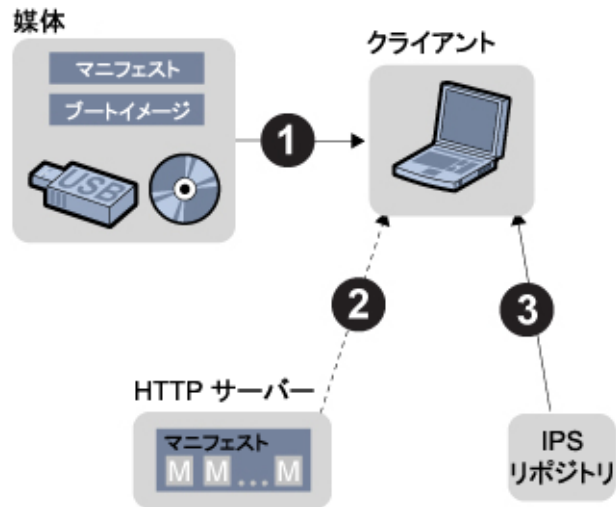
- インストールサーバーまたはインストールサービスを設定する必要がありません。
- システムをネットワーク経由でブートする必要がありません。

AI メディアを使用したインストール

CD、DVD、または USB デバイスから AI イメージをブートして、そのシステムだけの手入力なしのインストールを開始できます。AI マニフェストは、インストール手順を提供します。インストールするシステムにはネットワークアクセスが必要です。インストールを完了するために、インターネット上またはローカルネット

ワーク上のIPSリポジトリからソフトウェアパッケージが取得されます。[72ページ](#)の「[カスタムAIマニフェストの作成](#)」で説明されているデフォルトのAIマニフェストを確認してください。

図 5-1 メディアを使用するAIインストール



AIメディアを使用してインストールするためのシステム要件

SPARC システムも x86 システムも、次の要件を満たしている必要があります。

表 5-1 AIメディアを使用するインストールのためのシステム要件

要件	仕様
メモリー	最新リリースの最小メモリー要件を確認するには、 『Oracle Solaris 11.1 ご使用にあたって』 を参照してください。
ディスク容量	最新リリースのディスク容量要件を確認するには、 『Oracle Solaris 11.1 ご使用にあたって』 を参照してください。

表 5-1 AIメディアを使用するインストールのためのシステム要件 (続き)

要件	仕様
ネットワークアクセス	インストールされるシステムは、クライアントシステムにインストールするパッケージが含まれるIPSリポジトリにアクセスできる必要があります。また、カスタムAIマニフェストを作成する場合、システムはHTTPサーバー上のそのマニフェストにアクセスできる必要があります。

▼ AIメディアを使用してインストールする方法

1 AIブートイメージをダウンロードします。

AIブートイメージをダウンロードするには、インターネットで次の場所にアクセスします。

[http://www.oracle.com/](http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html)

[technetwork/server-storage/solaris11/downloads/index.html](http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html)

- **SPARCシステム** - **SPARC AI .iso** ファイルをダウンロードします。
- **x86システム** - **x86 AI .iso** ファイルまたは **x86 AI .usb** ファイルをダウンロードします。

2 デフォルトのAIマニフェストを確認します。

AIイメージに用意されているデフォルトのマニフェストを使用するか、またはカスタムマニフェストを作成して、クライアントをブートするときそのカスタムマニフェストの場所を指定できます。[72 ページの「カスタム AI マニフェストの作成」](#)を参照してください。

3 ブート可能なメディアを作成します。

- **SPARC** および **x86 ISO** イメージ - **.iso** ファイルを **CD** または **DVD** に書き込みます。
- **x86 USB** イメージ - イメージを **USB** フラッシュドライブにコピーするために **usbcopy** ユーティリティを使用します。

注-このユーティリティは、`pkg:/install/distribution-creator` パッケージをインストールすることによって、システムに追加できます。

4 メディアからブートします。

ブートイメージが含まれているデバイスからシステムをブートします。デフォルトのAIマニフェストまたはカスタムAIマニフェストを指定する方法に関する手順につ

いては、72 ページの「AIメディアからの SPARC システムのブート」および73 ページの「AIメディアからの x86 システムのブート」を参照してください。

「手入力なし」のインストールが実行されます。インストールの終了後、SCI Tool が起動し、システムの構成情報の入力を求められます。

- 5 **SCI Tool** パネルで構成情報を入力します。
84 ページの「SCI Tool を使用した構成プロファイルの作成」を参照してください。

カスタム AI マニフェストの作成

AI ブートイメージに用意されている AI マニフェスト内のインストール仕様を使用してシステムをインストールするか、カスタムのインストール仕様を作成できます。カスタム AI マニフェストを作成する場合は、そのマニフェストを HTTP サーバーに格納して、インストールするシステムをブートするときそのマニフェストの場所を指定します。

.iso AI イメージをダウンロードする場合は、次のサンプルコマンドを使用して、そのイメージ内の AI マニフェストを調べることができます。この例では、/tmp は AI イメージをダウンロードしたディレクトリで、/home/username は AI マニフェストをコピーおよび編集するディレクトリです。AI マニフェストは、イメージ内の auto-install/default.xml にあります。

```
# /usr/sbin/mount -o ro -F hfsfs /home/username/sol-11_1-20-ai-x86.iso /mnt
# cp /mnt/auto_install/manifest/default.xml /home/username/custom.xml
# umount /mnt
```

デフォルトのマニフェストファイル(この例の /home/username/custom.xml)のコピーを確認し、それらの仕様がこのインストールに適しているかどうかを判断します。

または、144 ページの「デフォルトの AI マニフェスト」に示したマニフェストをベースにして、カスタムマニフェストを作成することもできます。

インストールするターゲットディスクや追加パッケージなどのインストール仕様を変更する方法については、ai_manifest(4)のマニュアルページを参照してください。

AI マニフェストの変更が終わったら、カスタムマニフェストを HTTP サーバーにコピーします。カスタム AI マニフェストの URL を書き留めて、インストールするシステムをブートするときその URL を指定できるようにします。URL は、たとえば http://example.com/custom.xml のようになります。

AIメディアからの SPARC システムのブート

AIメディアからシステムをブートするとき、デフォルトの AI マニフェストまたはカスタム AI マニフェストを指定できます。

デフォルトのAI マニフェストの使用

AI ブートイメージ内にあるデフォルトの AI マニフェストを使用するには、OBP プロンプトで次のコマンドを入力します。

```
ok> boot cdrom - install
```

デフォルトのマニフェスト内の仕様を使って自動インストールが進行します。

カスタム AI マニフェストの使用

カスタム AI マニフェストを使用するには、OBP プロンプトで次のコマンドを入力します。

```
ok> boot cdrom - install aimanifest=prompt
```

次のプロンプトが表示されます。

Enter the URL for the AI manifest [HTTP, default]:

カスタムマニフェストの URL を入力します。たとえば、`http://example.com/custom.xml` と入力します。

カスタムマニフェスト内の仕様を使って自動インストールが進行します。

SPARC イメージをブートするときにインストールを行わない

メディアからブートするときにインストールを行いたくない場合があります。たとえば、システムのトラブルシューティングを行ったり、システムを調べたりする場合があります。

AI イメージをブートするときに自動インストールを開始しないようにするには、次のコマンドを使用します。

```
ok> boot cdrom
```

システムがブートされ、ログインパネルが表示されますが、インストールは開始されません。

AI メディアからの x86 システムのブート

x86 システムでは、GRUB メニューから自動インストールオプションを選択します。使用する GRUB メニューの選択項目またはブートコマンドによって、インストールでメディアのデフォルトのマニフェストを使用するか、HTTP サーバーに格納したカスタムマニフェストを使用するかが指定されます。

GRUB メニューの選択項目は、次の例のように表示されます。

```
GNU GRUB version 1.99.5.11.0.175.1.0.0.20.0
```

```
Oracle Solaris 11.1 Automated Install custom
Oracle Solaris 11.1 Automated Install
Oracle Solaris 11.1 Automated Install custom ttya
Oracle Solaris 11.1 Automated Install custom ttyb
Oracle Solaris 11.1 Automated Install ttya
Oracle Solaris 11.1 Automated Install ttyb
Boot from Hard Disk
```

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

デフォルトのAIマニフェストの使用

AIブートイメージ内にあるデフォルトのAIマニフェストを使用するには、矢印キーを使用して、次のオプションのいずれかを選択します。

```
Oracle Solaris 11.1 Automated Install
Oracle Solaris 11.1 Automated Install ttya
Oracle Solaris 11.1 Automated Install ttyb
```

ttya オプションは、インストール時の画面出力をシリアルコンソール ttya (COM1) に送信します。ttyb オプションは、インストール時の画面出力をシリアルコンソール ttyb (COM2) に送信します。

デフォルトのマニフェスト内の仕様を使って自動インストールが進行します。

カスタムAIマニフェストの使用

カスタムAIマニフェストを使用するには、次のオプションのいずれかを選択します。

```
Oracle Solaris 11.1 Automated Install custom
Oracle Solaris 11.1 Automated Install custom ttya
Oracle Solaris 11.1 Automated Install custom ttyb
```

これらのカスタムオプションのいずれかを選択すると、次のプロンプトが表示されます。

Enter the URL for the AI manifest [HTTP, default]:

カスタムマニフェストのURLを入力します。たとえば、<http://example.com/custom.xml> と入力します。

カスタムマニフェスト内の仕様を使って自動インストールが進行します。

x86 イメージをブートするときにインストールを行わない

メディアからブートするときにインストールを行いたくない場合があります。たとえば、システムのトラブルシューティングを行ったり、システムを調べたりする場合です。

一般に、使用する GRUB2 エントリの「\$multiboot」で始まる行に `install=true` が指定されている場合は、自動的にインストールが開始されます。x86 システムのブート時にすぐに自動インストールを開始しない場合は、選択する予定の GRUB2 メニューエントリを確認します。その GRUB2 エントリのカーネル行に `install=true` が指定されている場合は、その行を編集して `install=true` を削除します。こうすると、そのオプションを選択したときにシステムがブートしてログイン画面が表示されますが、インストールは開始されません。

インストールのログファイルの表示

自動インストールが完了すると、インストールに成功したか失敗したかを示す出力が表示されます。

- インストールに失敗した場合、インストールログは `/system/volatile/install_log` で確認できます。
- インストールに成功した場合、そのログは、システムをリブートする前であれば `/system/volatile/install_log` に、リブートしたあとであれば `/var/sadm/system/logs/install_log` にあります。

Oracle Solaris インスタンスの構成解除または再構成

Oracle Solaris インスタンスは、インストール中に作成および構成されます。Oracle Solaris インスタンスは、大域ゾーン内または非大域ゾーン内のいずれかのブート環境として定義されます。この章では、Oracle Solaris インスタンスの構成解除と再構成の方法について説明します。

機能グループ

Oracle Solaris インスタンスの構成解除または再構成を行うときに、一部の定義済みサブシステムが影響を受けます。これらのサブシステムは機能グループと呼ばれます。

1つのインスタンスのグループ全体は「システム」と呼ばれます。

次の表に、Oracle Solaris インスタンスに存在する構成可能な機能グループの一覧を示します。

表 6-1 機能グループ

グループ	コンポーネント	未構成の状態
system	システム全体	「system」グループには、ほかのすべてのグループが含まれます。
identity	システムのノード名	不明
kbd_layout	キーボード	アメリカ英語
network	ネットワーク	ネットワークなし
location	タイムゾーン	UTC
	ロケール	C ロケール

表 6-1 機能グループ (続き)

グループ	コンポーネント	未構成の状態
users	root	空の root パスワード
	初期ユーザーアカウント	ユーザーアカウントの削除
naming_services	DNS、NIS、およびLDAP クライアント、nsswitch	ネットワークのネームサービスなし
support	OCM および ASR のサポート	デフォルト設定は、OCM および ASR への匿名登録です。

sysconfigユーティリティの使用

sysconfig ユーティリティを使用して、Oracle Solaris インスタンスに対して次の構成タスクを行うことができます。

- 大域ゾーンまたは非大域ゾーンの Oracle Solaris インスタンスを構成解除し、それを未構成の状態のままにするには、`sysconfig unconfigure` コマンドを使用します。
79 ページの「[Oracle Solaris インスタンスの構成解除](#)」を参照してください。
- 大域ゾーンまたは非大域ゾーンの Oracle Solaris インスタンスを再構成するには、`sysconfig configure` コマンドを使用します。
 - このコマンドで既存の構成プロファイルを指定した場合、非対話型の構成が実行されます。
 - このコマンドで既存の構成プロファイルを指定しなかった場合、対話型の SCI Tool が実行されます。SCI Tool を使用すると、その Solaris インスタンスの特定の構成情報を指定できます。

79 ページの「[システムの構成](#)」を参照してください。

- `sysconfig create-profile` コマンドを使用して、新しいシステム構成プロファイルを作成できます。84 ページの「[SCI Tool を使用した構成プロファイルの作成](#)」を参照してください。

`sysconfig` コマンドは、Solaris インスタンス内のすべての機能グループに影響を及ぼします。詳細な手順については、`sysconfig(1M)` のマニュアルページを参照してください。

注 - `sysconfig` ユーティリティを使用するには、`root` の役割になる必要があります。

Oracle Solaris インスタンスの構成解除

以前に構成した Solaris インスタンスを構成解除し、それを未構成の状態のままにする場合は、構成解除コマンドを使用します。すべての機能グループが構成解除されます。

次の例にあるように `sysconfig unconfigure` コマンドを使用します。

```
# sysconfig unconfigure -g system
```

この例では、インスタンスが構成解除されます。

注 --g オプションを指定しない場合、システムが構成解除される前に確認が求められます。

あるいは、次のようにシステムを構成解除し、システムをシャットダウンすることもできます。

```
# sysconfig unconfigure -s
```

詳細は、`sysconfig(1M)` のマニュアルページを参照してください。

システムの構成

`sysconfig configure` コマンドを使用して、大域ゾーンまたは非大域ゾーンで Oracle Solaris インスタンスを構成または再構成できます。この構成は、対話型でも非対話型でも実行できます。

- `sysconfig configure` ユーティリティーに `-c` オプションを使用して、既存のシステム構成プロファイルを指定できます。このユーティリティーをそのオプションで実行した場合、ユーティリティーは既存のプロファイル内の構成仕様を読み取り、それらの仕様を使って非対話型でシステムを構成します。

たとえば、次のコマンドはシステムが `myprofile.xml` という名前の既存の構成プロファイルを使って構成されることを指定しています。

```
# sysconfig configure -c myprofile.xml
```

注 --c オプションは、プロファイルまたはプロファイルのディレクトリを示しています。すべてのプロファイルに `.xml` ファイル拡張子を含める必要があります。

システム構成 (SC) プロファイルについては、[第 11 章「クライアントシステムの構成」](#)を参照してください。

- プロファイルを指定しないで `sysconfig configure` コマンドを呼び出した場合は、SCI Tool が自動的にアクティブ化されます。

SCI Tool は、新たにインストールしたシステムまたは未構成のシステムの構成をサポートしています。このツールを使用して、新たに作成した非大域ゾーンやその他の未構成システムのシステム構成を行うことができます。SCI Tool は、構成情報の入力を求める一連の対話型テキストパネルから構成されます。[80 ページの「SCI Tool を使用して再構成する方法」](#)を参照してください。

注 - SCI Tool の一連の画面はテキストインストールの一環としても自動的に実行されます。

あるいは、SCI Tool を実行して、SCI Tool 画面で入力される構成仕様に基づいて新しいシステム構成プロファイルを作成することもできます。[84 ページの「SCI Tool を使用した構成プロファイルの作成」](#)を参照してください。

▼ SCI Tool を使用して再構成する方法

- 1 `root` の役割になります。

注 - 非大域ゾーンで作業している場合は、次のように `root` の役割としてゾーンにログインします。

```
# zlogin -C -e ^ zonename
```

- 2 プロファイルを指定しないで `sysconfig configure` コマンドを実行します。

```
# sysconfig configure
```

SCI Tool が表示されます。以降の手順は、SCI Tool の一連の対話型パネルの実行手順を示しています。

注 - SCI Tool パネル間の移動には、ファンクションキーを使用してください。マウスは使用できません。必要に応じて、各パネルのファンクションキーリファレンスやオンラインヘルプを参照してください。

- 3 初期の開始パネルを過ぎて先に進みます。
- 4 ネットワーク上でシステムを識別するための名前を指定します。

- 5 次のいずれかのオプションを選択して、有線の **Ethernet** ネットワーク接続がどのように構成されるかを指定します。
 - **DHCP** を使用してネットワーク接続を構成するには、「自動」を選択します。
SCITool では続けて「タイムゾーン」パネルが表示されます。
 - ネットワーク仕様を指定するには、「手動」を選択し、次のように続けます。
 - a. 複数のインタフェースがある場合は、構成される接続を選択します。
 - b. 接続設定を入力するか、または **SCITool** で検出され提供されたデフォルト情報を受け入れます。

注-IP アドレスとネットマスクは必須フィールドです。ルーターは省略可能なフィールドです。

- c. システムで **DNS** ネームサービスを使用するかどうかを指定します。
- d. 「**DNS の構成**」を選択した場合は、次を実行します。
 - i. システムによって使用される **DNS** サーバーの **IP** アドレスを1つ以上入力します。
 - ii. **DNS** クエリーが行われる際に検索されるドメイン名を1つ以上入力します。
- e. システムが **LDAP** ネームサービス、**NIS** ネームサービス、または「なし」のどれを使用するかを指定します。
 - 前の手順で **DNS** を選択した場合は、**DNS** に加えて **LDAP** または **NIS** が代替ネームサービスとして設定されます。
 - 前の手順で **DNS** を選択しなかった場合は、**LDAP** または **NIS** が唯一のネームサービスとして設定されます。
 - **LDAP** プロファイルのないシステムで **LDAP** を構成する場合は、**LDAP** を選択するのではなく「なし」を選択します。その後、**SCITool** プロセスが完了したあとで **LDAP** を手動で構成します。

注-ネットワークのネームサービスが選択されていない場合は、`/etc/hosts(4)` などの標準の名前ソースファイルを使用してネットワーク名を解決できません。詳細は、`nsswitch.conf(4)` のマニュアルページを参照してください。

- f. 選択した代替ネームサービス用に、システムが属するドメインを指定します。

注-ドメイン名を確認するには、システム管理者に問い合わせてください。または、以前にインストールされたシステム上で domainname コマンドを使用してください。

- g. LDAP を唯一のネームサービスとして、または DNS とともに追加のネームサービスとして選択した場合は、次のように LDAP 仕様を指定します。

- i. システムで LDAP ネームサービスを構成するために使用される LDAP プロファイルを指定します。
- ii. LDAP プロファイルサーバーの IP アドレスを入力します。
- iii. LDAP 検索ベースを指定するか、またはデフォルトの検索ベースを受け入れます。
- iv. LDAP プロキシバインド情報を入力するかどうかを指定します。

注-プロファイルによってプロキシ資格レベルが指定され、認証方法が「なし」でない場合は、プロキシバインド情報を入力する必要があります。その情報を省略した場合、LDAP は初期化されません。

- v. 必要に応じて、LDAP プロキシ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します。

- h. NIS を唯一のネームサービスとして、または DNS とともに追加のネームサービスとして選択した場合は、NIS 仕様を指定します。

ソフトウェアにネームサーバーを検索させることも、自分でネームサーバーを指定することもできます。次の2つの選択肢のどちらかを選択します。

- 「1つ検索」を選択します。

注-ソフトウェアは、ネームサーバーがローカルのサブネット上にある場合にのみ検出できます。

- 「1つ指定」を選択して、サブパネルにネームサーバーのホスト名または IP アドレスを入力します。

一連のネットワーク構成パネルが完了したら、SCIToolによって一連のタイムゾーンパネルおよび「日付と時間」パネルが表示されます。

- インストール中にネットワークを構成しないことを指定するには、「なし」を選択します。
SCIToolでは続けて「タイムゾーン」パネルが表示されます。
- 6 一連のタイムゾーンパネルで、最初にタイムゾーンを選択してから、現地時間に一致するように日付と時間を調整します。

注-デフォルトでは、GMTタイムゾーンが構成されます。

- 7 「ユーザー」パネルを完了します。
ユーザーアカウントを作成する必要はありませんが、rootパスワードを作成する必要があります。
- このパネルでユーザーアカウントを作成する場合は、ユーザーのパスワードと **root** パスワードの両方を指定する必要があります。
この場合、root はユーザーに割り当てられた役割になります。
ユーザーアカウントを作成するには、ユーザー名とパスワードを入力します。この名前は英字で始まる必要があります、英字と数字のみを含めることができます。
 - ユーザーアカウントを作成しない場合でも、**root** パスワードは指定する必要があります。
この場合、root は通常のユーザーとなります。
- 8 「サポートの構成」パネルで、**OCM** および **ASR** の匿名登録を受け入れることができます。
デフォルトの「サポートの構成」パネルには、匿名の登録アドレスが表示されません。パスワードなしでこの匿名アドレスを使用した場合は、Oracle サポート組織にシステム構成がアップロードされたときに、My Oracle Support (MOS) はシステムの構成に関する情報を受け取りませんが、顧客情報は一切受け取りません。
あるいは、次のようにしてセキュリティー更新に登録したり、OCM を切断したりすることもできます。
- 「サポートの構成」パネルの匿名の電子メールアドレスを My Oracle Support のログイン ID に置き換えて、My Oracle Support のパスワードを追加できます。My Oracle Support で顧客情報を参照し、セキュリティー更新を受け取る場合は、このオプションを使用します。このオプションを使用すると、ASR も起動します。
 - 「サポートの構成」パネルの匿名の電子メールアドレスを削除し、そのフィールドを空白のままにした場合は、OCM が切断モードで起動します。My Oracle Support にデータは送信されません。匿名の電子メールアドレスを削除し、それを MOS のログイン ID 以外の別の電子メールアドレスで置き換えた場合、OCM は未認証モードで Oracle サポートにデータを送信します。

詳細は、[39 ページの「Oracle Configuration Manager の使用」](#)を参照してください。

- 9 構成設定を確認します。
 - それらの設定が正しい場合は、構成をシステムに適用します。
 - それらの設定が正しくない場合は、必要なだけ「戻る」ボタンを押して正しくない情報が含まれるパネルに戻り、変更を行なって、再度パネルを進みます。

SCIToolを使用した構成プロファイルの作成

SCIToolを実行して、SCIToolパネルで入力された構成仕様に基づいて新しいシステム構成プロファイルを生成できます。新しいプロファイルのデフォルトの場所は `/system/volatile/profile/sc_profile.xml` です。

新しい構成プロファイルを作成するには、`sysconfig create-profile` コマンドを使用します。プロファイルは作成されますが、構成はシステムに適用されません。

SCIToolは、SCIToolパネルで入力された仕様に基づいて新しい構成プロファイルを作成します。新しいプロファイルはデフォルトの場所に格納されます。次の例に示すように、その新しいプロファイルを使用してシステムを構成できます。

```
# sysconfig configure -g system -c /system/volatile/profile/sc_profile.xml
```

`-g` オプションは、構成される特定の機能グループを指定するために使用します。この例では、システム全体が構成されます。機能グループの一覧については、[表 6-1](#) を参照してください。

次の例では、`sysconfig create-profile -o` オプションを使って、プロファイルの作成時に別の出力ファイル場所を指定しています。その後、`sysconfig configure -c` オプションでそのプロファイル場所を指し示して、システムを再構成します。

```
# sysconfig create-profile -o /tmp/myprofile.xml  
# sysconfig configure -g system -c /tmp/myprofile.xml
```

注-そのプロファイルを再構成に正しく使用するためには、構成プロファイルに `.xml` 拡張子を含める必要があります。

詳細は、`sysconfig(1M)` のマニュアルページを参照してください。第 11 章「クライアントシステムの構成」も参照してください。

パート III

インストールサーバーを使用したインストール

このセクションでは、ネットワークを介したクライアントシステムの自動インストールについて説明します。

複数のクライアントへの自動インストール

自動インストーラ (AI) を使用して、Oracle Solaris 11 オペレーティングシステム (OS) または Oracle Solaris 11.1 OS をネットワーク上の複数のクライアントシステムにインストールします。AI は、SPARC と x86 の両方のシステムで手入力なしのインストールを実行します。すべてのインストールで、ネットワーク上のソフトウェアパッケージリポジトリにアクセスできる必要があります。

自動インストールとは

AI は、ネットワーク経由で SPARC および x86 クライアントへの Oracle Solaris 11 OS のインストールを自動化します。クライアントは、ディスクレイアウトやソフトウェア選択などのインストールパラメータを使って、またホスト名、ネットワーク構成、ユーザーアカウントなどのシステム構成パラメータを使ってカスタマイズできます。カスタマイズはクライアントごとに行うことができ、大規模な環境にも対応できます。

ネットワーク経由でのクライアントの自動インストールは、次の概要レベルの手順で構成されます。

1. クライアントシステムがネットワーク経由でブートし、そのネットワーク構成とインストールサーバーの場所を DHCP サーバーから取得します。SPARC クライアントは、必要に応じて、オープンブート PROM (OBP) に設定された `network-boot-arguments` 変数からネットワーク構成とインストールサーバーの場所を取得できます。
2. インストールサーバーがブートイメージをクライアントに提供します。
3. クライアントの特性によって、クライアントのインストールに使用するインストール手順とシステム構成手順が決まります。
4. AI インストールサービスのインストール手順で指定されているパッケージリポジトリからパッケージを取得し、Oracle Solaris 11 OS がクライアントにインストールされます。

自動インストーラの使用法

AIを使用してネットワーク経由でクライアント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上のAIインストールサービスを設定する必要があります。第8章「インストールサーバーの設定」を参照してください。AIは、DHCPを使用して、インストールされるクライアントマシンにIPアドレス、サブネットワーク、ルーター、ネームサービスサーバー、およびインストールサーバーの場所を提供します。SPARCクライアントは、必要に応じて、OBPに設定された `network-boot-arguments` 変数からネットワーク構成とインストールサーバーの場所を取得できます。`network-boot-arguments` 変数の使用例については、228ページの「SPARCクライアントのインストール」を参照してください。DHCPサーバーとAIインストールサーバーは、同じマシンにすることも、2つの別々のマシンにすることもできます。

インストール対象のクライアントマシンは、Oracle Solaris Image Packaging System (IPS) ソフトウェアパッケージリポジトリにアクセスする必要があります。リポジトリは、ソフトウェアパッケージを取得する場所です。場所はURI (Universal Resource Identifier) によって指定されます。IPSパッケージリポジトリは、インストールサーバー上、ローカルネットワークの別のサーバー上、またはインターネット上に配置できます。パッケージリポジトリへのアクセスについては、『Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新』の「発行元の構成」を参照してください。

AIインストールサービスには、SPARCまたはx86ネットワークブートイメージ(ネットイメージ)、1つ以上のインストール手順ファイル(AIマニフェスト)、および0個以上のシステム構成SMFプロファイルファイルが含まれています。AIネットイメージは完全なインストールではありません。クライアントマシンは、インストールを完了するためにIPSパッケージリポジトリにアクセスする必要があります。各クライアントでは、AIマニフェストを1つのみ使用します。それぞれのクライアントが別々のAIマニフェストを使用できます。

AIマニフェストによって、クライアントがインストールの完了に必要なパッケージを取得する1つ以上のIPSパッケージリポジトリが指定されます。AIマニフェストには、インストールする追加パッケージの名前と、ターゲットのインストールデバイスやパーティション情報などの情報も含まれています。クライアントのブート前に、またはクライアントのインストール時に動的にAIマニフェストをカスタマイズする方法については、第10章「クライアントシステムのプロビジョニング」を参照してください。クライアントの構成手順を指定することもできます。システム構成プロファイルについては、第11章「クライアントシステムの構成」を参照してください。クライアントの初回ブート時にインストールと構成をさらに行う方法については、第13章「初回ブート時のカスタムスクリプトの実行」を参照してください。

2台のクライアントマシンのアーキテクチャーが異なる場合、または異なるバージョンのOracle Solaris 11 OSをインストールする必要がある場合は、2つのAIインストールサービスを作成して、それぞれのインストールサービスを、インス

ツール対象のアーキテクチャーとOSバージョンの適切なイメージソースに関連付けます。特定のアーキテクチャーの1番目のインストールサービスがインストールサーバーに作成されると、そのサービスのコピー `default-i386` または `default-sparc` が自動的に作成されます。このデフォルトのサービスは、そのアーキテクチャーのクライアントのうち、`create-client` サブコマンドを使って別のインストールサービスに明示的に関連付けられていなかったクライアントへのすべてのインストールで使用されます。

2台のクライアントマシンに同じバージョンの Oracle Solaris 11 OS をインストールする必要があるときに、異なる方法でインストールする必要がある場合は、AIインストールサービスに対して2つのAIマニフェストを作成します。AIマニフェストを別にすることによって、たとえば、異なるパッケージをインストールするように指定したり、異なるスライスをインストールターゲットとして指定したりすることができます。

クライアントシステムに異なる構成を適用する必要がある場合は、インストールサービスに対して複数のシステム構成プロファイルを作成します。システム構成プロファイルを別にするによって、たとえば、異なるネットワークまたはロケール設定を指定したり、独自のホスト名やIPアドレスを指定したりできます。

インストールは、クライアントをブートしたときに開始します。クライアントをブートすると、クライアントはAIインストールサーバーに接続され、正しいインストールサービスとそのサービスに含まれている正しいAIマニフェストおよびシステム構成プロファイルにアクセスします。第14章「クライアントシステムのインストール」では、クライアントを特定のインストールサービスに関連付ける方法について説明します。第9章「インストールのカスタマイズ」では、使用する正しいAIマニフェストとシステム構成プロファイルをクライアントで特定する方法について説明します。

適切なシステム構成手順が指定されていない場合、インストール後の初回ブート時に対話型ツールによって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す。システム構成プロファイルの情報と例については、第11章「クライアントシステムの構成」を参照してください。対話型構成ツールについては、79ページの「システムの構成」を参照してください。

非大域ゾーンのインストールを指定した場合は、インストール後の初回ブート時にそのゾーンが構成されインストールされます。AIクライアントインストールの一環として非大域ゾーンの構成およびインストールを指定する方法については、第12章「ゾーンのインストールと構成」を参照してください。

自動インストーラの使用事例

次の使用事例では、AIのもっとも明確な使用法について説明します。これらの使用事例は、相互に依存するものではありません。代わりに、各事例ではAIの個別の機能について説明し、その機能に含まれない動作はすべて最小事例に含まれているものと同じになります。おそらく、これらの使用事例に記載されている機能を組み合わせて使うことになります。

AIを使用するための最小要件

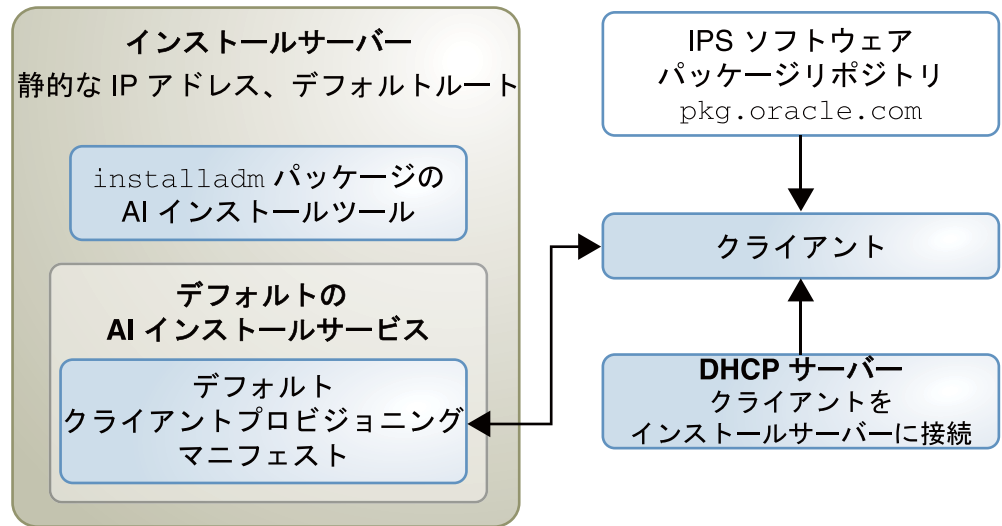
AIを使用するために最低限必要なことは、インストールサービスを1つ作成することです。この最小のシナリオでは、すべてのクライアントのアーキテクチャーが同じであり、同じバージョンの Oracle Solaris OS をインストールします。インストールではデフォルトの AI マニフェストが使用されます。Oracle Solaris 11.1 リリースでは、デフォルトの AI マニフェストに、<http://pkg.oracle.com/solaris/release> IPS パッケージリポジトリから入手可能な最新バージョンの Oracle Solaris 11.1 リリースが指定されています。

1. インストールサーバーに静的な IP アドレスとデフォルトルートが指定されていることを確認します。
2. インストールツールパッケージ `install/installadm` をインストールします。
3. `installadm create-service` コマンドを実行します。

特定のアーキテクチャーの1番目のインストールサービスがインストールサーバーに作成されると、そのサービスのコピー `default-i386` または `default-sparc` が自動的に作成されます。このデフォルトのサービスは、そのアーキテクチャーのクライアントのうち、`create-client` サブコマンドを使って別のインストールサービスに明示的に関連付けられていなかったクライアントへのすべてのインストールで使用されます。

4. クライアントが DHCP サーバーにアクセスでき、クライアントがそのサービスをブートするために必要な情報が DHCP 構成で使用できることを確認します。SPARC クライアントの場合は、[228 ページの「SPARC クライアントのインストール」](#) に示すように、必要に応じて OBP に `network-boot-arguments` 変数を設定してサービスをブートできます。
5. クライアントが IPS ソフトウェアパッケージリポジトリにアクセスできることを確認します。デフォルトの IPS パッケージリポジトリを使用するには、クライアントがインターネットにアクセスする必要があります。
6. クライアントをネットワーク経由でブートします。

図 7-1 AIを使用するための最小要件



このシナリオでは、クライアントをネットワーク経由でブートすると、次の手順が実行されます。

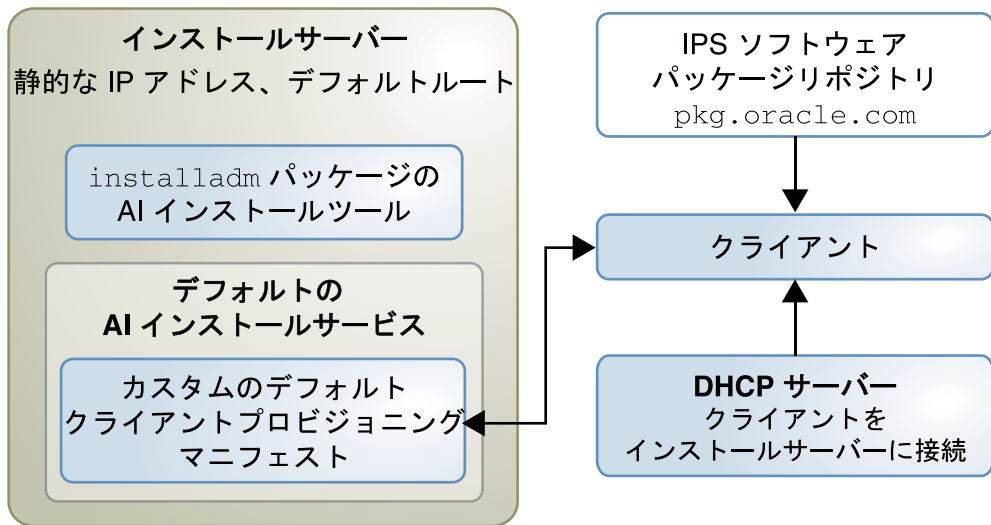
1. クライアントはDHCPサーバーからインストールサーバーアドレスを取得します。SPARCクライアントは、必要に応じて、OBPの `network-boot-arguments` 変数からインストールサーバーのアドレスを取得できます。
2. アーキテクチャーが一致する場合、クライアントは `default-arch` インストールサービスを使用します。
3. クライアントは `default-arch` インストールサービスのデフォルトのAIマニフェストを使用して、ネットワーク経由でIPSパッケージリポジトリのソフトウェアパッケージをインストールします。
4. インストール後にクライアントがブートすると、システム構成プロファイルが指定されていないため、対話型ツールによって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す。

インストール手順のカスタマイズ

インストールのターゲットディスク、パーティションまたはミラー構成、インストールする追加のソフトウェアパッケージなどのインストールパラメータを指定するには、カスタマイズされた AI マニフェストを指定します。最低限必要な手順に加え、クライアントをブートする前に次の手順も実行してください。

1. 新しい AI マニフェストを作成するか、またはクライアントのインストール時に動的にカスタム AI マニフェストを作成するスクリプトを記述します。第 10 章「クライアントシステムのプロビジョニング」を参照してください。
2. `installadm create-manifest` コマンドを実行して、新しいマニフェストまたはスクリプトを `default-arch` インストールサービスに追加します。クライアントでこのマニフェストまたはスクリプトを選択するための条件を指定します。

図 7-2 インストール手順のカスタマイズ



このシナリオでは、クライアントをネットワーク経由でブートすると、次の手順が実行されます。

1. クライアントは DHCP サーバーからインストールサーバーアドレスを取得します。SPARC クライアントは、必要に応じて、OBP の `network-boot-arguments` 変数からインストールサーバーのアドレスを取得できます。
2. アーキテクチャーが一致する場合、クライアントは `default-arch` インストールサービスを使用します。

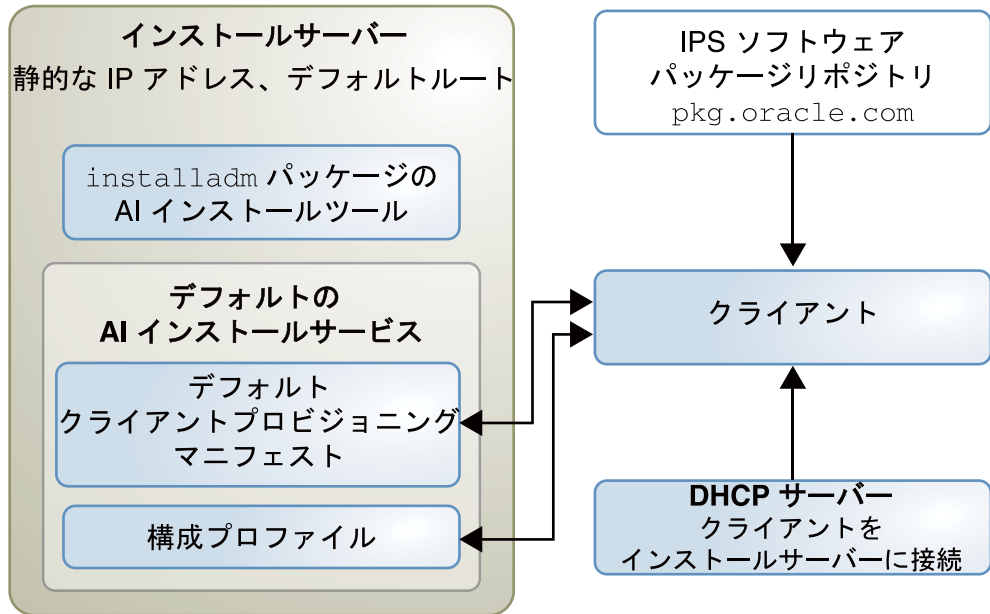
3. `create-manifest` で指定された条件によってクライアントは正しい AI マニフェストにアクセスします。条件が一致しない場合、クライアントはデフォルトのマニフェストをこのサービスに使用します。
4. 選択された AI マニフェストに従ってクライアントがプロビジョニングされます。
5. インストール後にクライアントがブートすると、システム構成プロファイルが指定されていないため、対話型ツールによってシステム構成情報の入力を求めるプロンプトが表示されます。

システム構成手順の指定

タイムゾーン、ユーザーアカウント、ネットワーク接続などのシステム構成パラメータを指定するには、サービス管理機能 (SMF) のシステム構成プロファイルファイルを指定します。最低限必要な手順に加え、クライアントをブートする前に次の手順も実行してください。

1. [第 11 章「クライアントシステムの構成」](#) の説明に従ってシステム構成プロファイルを作成します。
2. `installadm create-profile` コマンドを使用してそのプロファイルを検証し、プロファイルを `default-arch` インストールサービスに追加してから、このシステム構成プロファイルを使用するクライアントを選択するための条件を指定します。条件を指定しない場合は、そのサービスのすべてのクライアントがこのプロファイルを使用します。

図7-3 システム構成手順の指定



このシナリオでは、クライアントをネットワーク経由でブートすると、次の手順が実行されます。

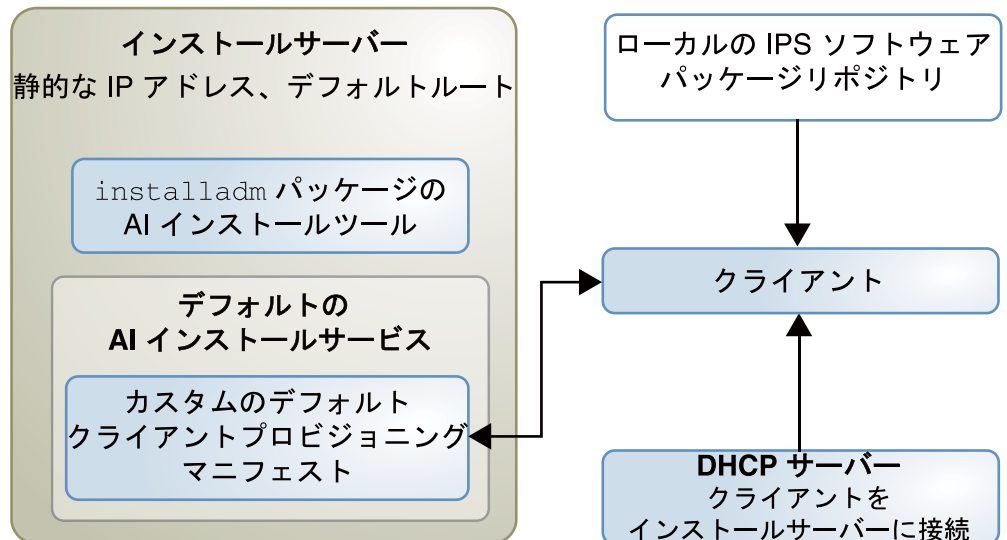
1. クライアントは DHCP サーバーからインストールサーバーアドレスを取得します。SPARC クライアントは、必要に応じて、OBP の `network-boot-arguments` 変数からインストールサーバーのアドレスを取得できます。
2. アーキテクチャーが一致する場合、クライアントは `default-arch` インストールサービスを使用します。
3. クライアントは `default-arch` インストールサービスのデフォルトの AI マニフェストを使用して、ネットワーク経由で IPS パッケージリポジトリのソフトウェアパッケージをインストールします。
4. `default-arch` インストールサービスの `create-profile` で指定された条件によって、クライアントは正しいシステム構成プロファイルにアクセスします。
5. 選択された構成プロファイルに従ってクライアントが構成されます。条件が一致しないために構成プロファイルが選択されない場合は、対話型構成ツールが起動します。

ローカルIPSパッケージリポジトリの指定

クライアントがインターネットにアクセスできないなどの理由で、インターネットパッケージリポジトリではなくローカルのパッケージリポジトリを使用してデータ転送のパフォーマンスを向上させることがあります。最低限必要な手順に加え、クライアントをブートする前に次の手順も実行してください。

1. IPSパッケージリポジトリのローカルコピーを作成し、クライアントシステムがそのリポジトリにアクセスできるようにします。手順については、『[Oracle Solaris 11.1 パッケージリポジトリのコピーおよび作成](#)』を参照してください。
2. デフォルトのAIマニフェストをカスタマイズして新しいリポジトリをソフトウェアのソースとして指定します。デフォルトのマニフェストをエクスポートおよび編集し、`installadm update-manifest` コマンドを実行して `default-arch` インストールサービス内のデフォルトのAIマニフェストを編集済みのマニフェストで置き換えます。手順については、[第10章「クライアントシステムのプロビジョニング」](#)を参照してください。

図7-4 ローカルIPSパッケージリポジトリの指定



このシナリオでは、クライアントをネットワーク経由でブートすると、次の手順が実行されます。

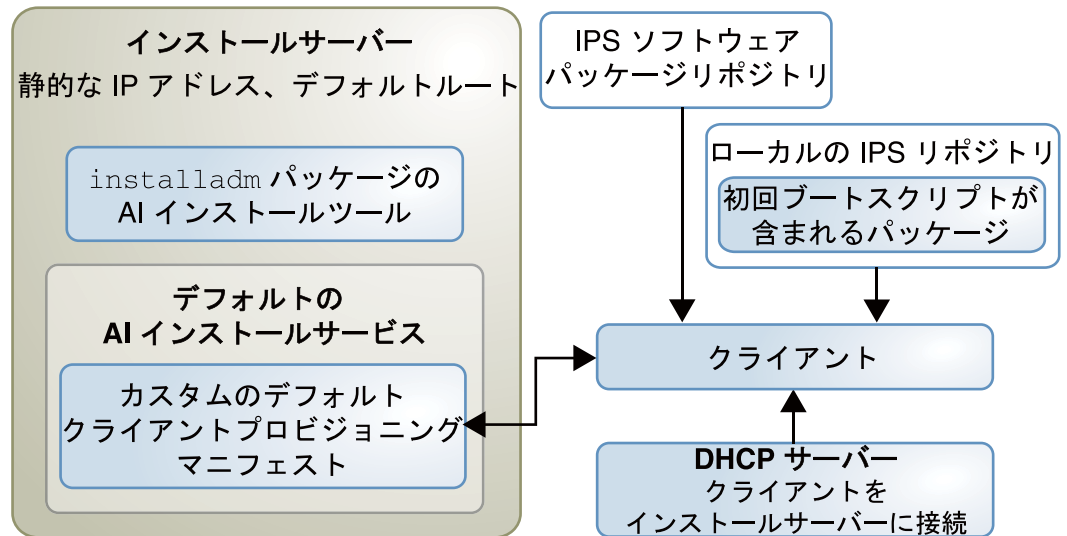
1. クライアントは DHCP サーバーからインストールサーバーアドレスを取得します。SPARC クライアントは、必要に応じて、OBP の `network-boot-arguments` 変数からインストールサーバーのアドレスを取得できます。
2. アーキテクチャーが一致する場合、クライアントは `default-arch` インストールサービスを使用します。
3. カスタマイズされた AI マニフェストに従い、ローカルの IPS パッケージリポジトリを使用して、クライアントがプロビジョニングされます。
4. インストール後にクライアントがブートすると、システム構成プロファイルが指定されていないため、対話型ツールによって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す。

カスタムの初回ブートスクリプトの指定

AI マニフェストまたはシステム構成プロファイルで表現できない構成を含める場合は、初回ブート時に実行されるスクリプトを追加できます。最低限必要な手順に加え、クライアントをブートする前に次の手順も実行してください。これらの手順の詳細は、[第 13 章「初回ブート時のカスタムスクリプトの実行」](#)を参照してください。

1. クライアントの初回ブート時に実行されるスクリプトを作成します。
2. このスクリプトを実行するための 1 度だけ実行する SMF サービスを作成します。
3. そのサービスとスクリプト用の IPS パッケージを作成し、そのパッケージをローカルの IPS リポジトリに追加します。
4. クライアントシステムがそのリポジトリにアクセスできるようにします。
5. デフォルトの AI マニフェストをカスタマイズして、新しいリポジトリをソフトウェアのソースとして指定し、新しいパッケージがインストールされるように指定します。デフォルトのマニフェストをエクスポートおよび編集し、`installadm update-manifest` コマンドを実行して `default-arch` インストールサービス内のデフォルトの AI マニフェストを編集済みのマニフェストで置き換えます。手順については、[第 10 章「クライアントシステムのプロビジョニング」](#)を参照してください。

図 7-5 カスタムの初回ブートスクリプトの指定



このシナリオでは、クライアントをネットワーク経由でブートすると、次の手順が実行されます。

1. クライアントは DHCP サーバーからインストールサーバーアドレスを取得します。SPARC クライアントは、必要に応じて、OBP の `network-boot-arguments` 変数からインストールサーバーのアドレスを取得できます。
2. アーキテクチャーが一致する場合、クライアントは `default-arch` インストールサービスを使用します。
3. 初回ブートスクリプトが含まれるカスタムパッケージのインストールを含む、カスタマイズされた AI マニフェストに従ってクライアントがプロビジョニングされます。
4. インストール後にクライアントがブートすると、システム構成プロファイルが指定されていないため、対話型ツールによって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す。
5. インストール後にクライアントがブートすると、カスタムの 1 度だけ実行する初回ブートサービスが実行され、カスタムスクリプトが実行されます。

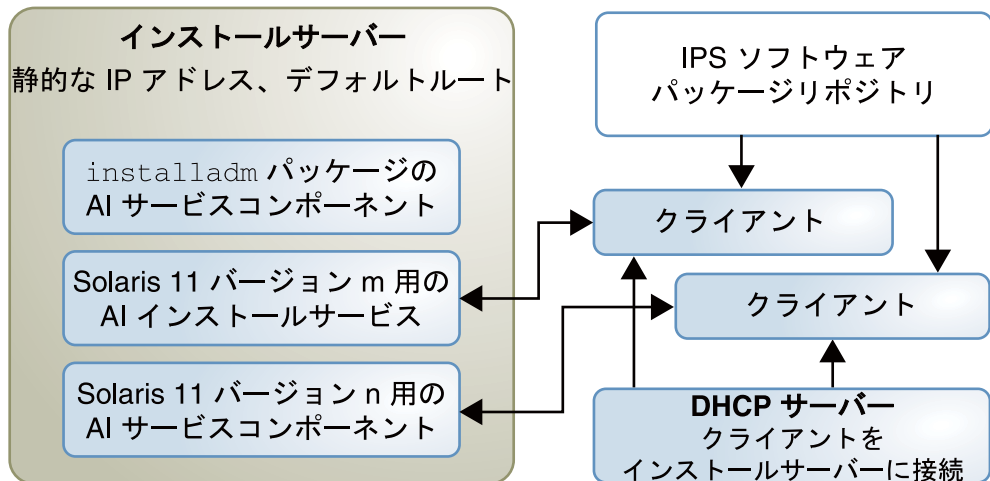
追加の AI インストールサービスの指定

異なるクライアントアーキテクチャーでインストールするには、または異なるバージョンの Oracle Solaris 11 OS をインストールするには、第 8 章「インストールサーバーの設定」の説明に従って追加の AI インストールサービスを作成します。最低限必要な手順に加え、クライアントをブートする前に次の手順も実行してください。

1. `installadm create-service` コマンドを実行し、インストール対象のアーキテクチャーと OS バージョンに対応するソースを指定します。
2. これが異なるアーキテクチャーの最初のインストールサービスである場合は、そのサービスのコピー `default-arch` が自動的に作成されます。このデフォルトのサービスは、そのアーキテクチャーのクライアントのうち、`create-client` サブコマンドを使って別のインストールサービスに明示的に関連付けられていなかったクライアントへのすべてのインストールで使用されます。

この新しいインストールサービスが既存のインストールサービスと同じアーキテクチャー用である場合は、`installadm create-client` コマンドを実行して、クライアントがこのアーキテクチャーのデフォルトのサービスではなく、この新しいインストールサービスにアクセスするように指定します。

図 7-6 追加の AI インストールサービスの指定



このシナリオでは、クライアントをネットワーク経由でブートすると、次の手順が実行されます。

1. クライアントはDHCPサーバーからインストールサーバーアドレスを取得します。SPARCクライアントは、必要に応じて、OBPの `network-boot-arguments` 変数からインストールサーバーのアドレスを取得できます。
2. クライアントは `create-client` によって新しいインストールサービスにアクセスするように指定されます。または、`create-client` がこのクライアントに対して実行されなかった場合は、クライアントはデフォルトのインストールサービスにアクセスします。
3. 選択されたインストールサービス用のデフォルトのAIマニフェストに従ってクライアントがプロビジョニングされます。
4. インストール後にクライアントがブートすると、システム構成プロファイルが指定されていないため、対話型ツールによって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す。

インストールサーバーの設定

ネットワーク経由でクライアントをインストールするには、AIではインストールサーバーとして機能する別個のシステムが必要です。インストールサーバーでは、AIインストールサービスを作成して、ネットイメージや、さまざまなクライアントに必要な Oracle Solaris 11 リリースをインストールするための手順を提供します。

AI サーバー設定のタスクマップ

次のタスクマップに、AIインストールサーバーを設定する手順の概要を示します。

表 8-1 AI サーバー設定のタスクマップ

タスク	リファレンス
サーバーを AI インストールサーバーにできる最小ハードウェア要件を満たしているかどうかを確認します。	102 ページの「AI サーバーのハードウェア要件」を参照してください。
静的な IP アドレスとデフォルトルートを使用するように AI インストールサーバーを構成します。オプションで、 <code>svc:/network/dns/multicast</code> SMF サービスを有効にします。AI インストールサーバーが IPS ソフトウェアパッケージリポジトリにアクセスできることを確認します。	103 ページの「AI サーバーのソフトウェア要件」を参照してください。
AI ツールセットをインストールします。	104 ページの「AI インストールツールのインストール」を参照してください。
インストールサービスを設定します。	109 ページの「AI インストールサービスの作成」を参照してください。インストールするアーキテクチャーごとに、またインストールする異なるバージョンのオペレーティングシステムごとに、個別のインストールサービスが必要です。

インストールサーバーの要件

ノートパソコン、デスクトップ、仮想マシン、エンタープライズサーバーなど、これらの要件を満たすシステムであれば、AIインストールサーバーとして使用できます。AIインストールサーバーには、x86マシンとSPARCマシンのどちらも使用できます。x86インストールサーバーはSPARCとx86の両方のクライアントをインストールでき、SPARCインストールサーバーもSPARCとx86の両方のクライアントをインストールできます。

AIサーバーのハードウェア要件

次の要件は、Oracle Solaris 11.1 OSがすでにインストールされていることを前提としています。AIインストールサーバーでOracle Solaris 11.1 OSのインストールまたは更新が必要な場合は、[第4章「テキストインストーラの使用」](#)および[第3章「Live Mediaの使用」](#)でメモリーとディスク容量の要件を確認してください。

- メモリー AIインストールサーバーとして動作するための最小メモリー要件は1Gバイトです。
- ディスク容量 AIインストールサーバーとして動作するために必要な追加のディスク容量は、設定するインストールサービスの数によって変わります。インストールを計画しているクライアントのアーキテクチャーごとに、またクライアントシステムへのインストールを計画しているOracle Solaris 11 OSのバージョンごとに個別のインストールサービスが必要になります。各ネットイメージは約300-400Mバイトです。

インストールサービスの操作権限

自動インストールで使用されるコマンドの多くには、強力な権限が必要です。権限を強めるには、次のいずれかの方法を使用します。

- 権利プロファイル `profiles` コマンドを使用して、自分に割り当てられている権利プロファイルを一覧表示します。

Software Installation

Software Installation 権利プロファイルがある場合、`pfexec` コマンドを使用して、パッケージをインストールおよび更新できます。

```
$ pfexec pkg install install/installadm
```

Install Service Management

Install Service Management 権利プロファイルがある場合、`pfexec` コマンドを使用して、インストールサービスの

作成や、インストールサービスへのシステム構成プロファイルの追加などができます。

```
$ pfexec installadm create-service
```

Service Management

Service Management 権利プロファイルがある場合、SMF サービスを構成して有効化できます。Service Management 権利プロファイルには pfexec は必要ありません。

```
$ svcadm refresh system/install/server:default
```

sudo

サイトのセキュリティーポリシーに応じて、自分のユーザーパスワードで sudo コマンドを使用し、特権コマンドを実行できる場合があります。

```
$ sudo pkg install install/installadm
```

役割

roles コマンドを使用して、自分に割り当てられている役割を一覧表示します。root 役割を持つ場合、root パスワードで su コマンドを使用して、root 役割になることができます。

AI サーバーのソフトウェア要件

オペレーティングシステム

Oracle Solaris 11.1 OS を AI サーバーにインストールします。Oracle Solaris 11.1 OS を AI サーバーにインストールするには、[パート II 「インストールメディアを使用したインストール」](#) を参照してください。

静的 IP アドレス

静的 IP アドレスを使用するように AI サーバーを構成します。『[Oracle Solaris 11.1 での固定ネットワーク構成を使用したシステムの接続](#)』の「[IP インタフェースを構成する方法](#)」を参照してください。

デフォルトルーター

netstat -m コマンドを使用して、AI サーバーにデフォルトのルートセットがあることを確認します。AI サーバーにデフォルトのルートセットがない場合は、route add default コマンドを使用してデフォルトルートを静的に設定します。route コマンドの使用方法については、[route\(1M\)](#) のマニュアルページを参照してください。

マルチキャスト DNS

(オプション) マルチキャスト DNS を有効にして、AI サーバー上でインストールサービ

	スを通知します。106 ページの「マルチキャスト DNS の有効化」を参照してください。
ソフトウェアパッケージリポジトリ	インストールサーバーが IPS ソフトウェアパッケージリポジトリにアクセスできることを確認します。AI には <code>install/installadm</code> パッケージが必要です。104 ページの「AI インストールツールのインストール」に示すように、 <code>pkg list</code> コマンドを使用します。
DHCP	DHCP を設定します。AI サーバーを DHCP サーバーにすることもできます。または、このネットワークですでに設定されている DHCP サーバーを使用できます。クライアントアーキテクチャーごとに異なる DHCP 構成が必要です。114 ページの「ローカル DHCP 設定を含むインストールサービスの作成」に、AI を使用してインストールサーバーに DHCP を設定する例が示されています。DHCP の構成については、18 ページの「関連情報」を参照してください。

AI インストールツールのインストール

AI インストールツールパッケージには、AI インストールサービスを作成および維持できるようにする `installadm(1M)` コマンドが用意されています。

`installadm` コマンドを使用すると、次のタスクを実行できます。

- インストールサービスを作成して管理します。
- DHCP サーバーを設定および更新します。
- クライアントのインストールと構成のカスタムの手順を追加します。
- クライアントでインストールと構成のカスタムの手順を使用するための条件を設定します。

`installadm` コマンドの詳細は、117 ページの「インストールサーバーの維持」および `installadm(1M)` のマニュアルページを参照してください。

ツールパッケージをインストールするには、AI インストールサーバーが Oracle Solaris Image Packaging System (IPS) ソフトウェアパッケージリポジトリにアクセスで

きる必要があります。インターネットまたは `install/installadm` パッケージが含まれているローカル IPS パッケージサーバーに接続されていることを確認してください。

`pkg list` コマンドを使用して、`installadm` パッケージがこのシステムにすでにインストールされているかどうかを調べます。

```
$ pkg list installadm
pkg list: no packages matching 'installadm' installed
```

`-a` オプションを使用して、このイメージにインストールできる `installadm` パッケージのバージョンが IPS パッケージリポジトリに含まれているかどうかを表示します。

```
$ pkg list -a installadm
NAME (PUBLISHER)                VERSION                IFO
install/installadm             0.5.11-0.175.1.0.0.24.0 ---
```

このイメージに対して複数の発行元が定義されている場合は、`-v` オプションを使用してどの発行元が `installadm` パッケージを提供しているかを表示します。次の例は、`solaris` がこのパッケージの発行元であることを示しています。

```
$ pkg list -av installadm
FMRI                                IFO
pkg://solaris/install/installadm@0.5.11,5.11-0.175.1.0.0.24.0:20120815T024057Z ---
```

`pkg publisher` コマンドを使用して発行元の起点を表示します。この例では、`solaris` リポジトリのローカルコピーが作成されています。

```
$ pkg publisher
PUBLISHER          TYPE  STATUS  P LOCATION
solaris            origin online  F file:///export/repoSolaris11/
isv.com (non-sticky) origin online  F http://pkg.isv.com/
```

`pkg install` コマンドを使用して `installadm` パッケージをインストールします。

```
$ pfexec pkg install install/installadm
Packages to install: 1
Create boot environment: No
Create backup boot environment: No
Services to change: 2
```

```
DOWNLOAD          PKGS    FILES    XFER (MB)  SPEED
Completed          1/1     72/72     0.3/0.3    0B/s
```

```
PHASE                ITEMS
Installing new actions 138/138
Updating package state database Done
Updating image state   Done
Creating fast lookup database Done
Reading search index   Done
Updating search index  1/1
```

```
$ pkg info installadm
Name: install/installadm
Summary: installadm utility
Description: Automatic Installation Server Setup Tools
Category: System/Administration and Configuration
State: Installed
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.175.1.0.0.24.0
Packaging Date: Mon Aug 15 02:40:57 2012
Size: 1.21 MB
FMRI: pkg://solaris/install/installadm@0.5.11,5.11-0.175.1.0.0.24.0:20120815T024057Z
```

インストールサーバーの構成

このセクションでは、AIクライアントインストールの準備のためにインストールサーバーで行うことが推奨される一部の構成について説明します。

マルチキャスト DNS の有効化

インストールクライアントは、ブート時に使用するインストールサービスを検出します。インストールサービスをネットワーク上で検出できるようにするため、マルチキャスト DNS (mDNS) を有効にして、AIサーバー上でインストールサービスを通知します。

ネットワーク上に同じインストールサービス名を持つ複数のAIサーバーがある場合は、競合を避けるため、マルチキャスト DNS を無効にするようにしてください。

svcs コマンドを使用して svc:/network/dns/multicast サービスのステータスを確認し、必要な場合は svcadm コマンドを使用してこのサービスを有効にします。

```
svcs /network/dns/multicast
STATE      STIME      FMRI
disabled   10:19:28   svc:/network/dns/multicast:default
svcadm enable /network/dns/multicast
svcs /network/dns/multicast
STATE      STIME      FMRI
online     13:28:30   svc:/network/dns/multicast:default
```

マルチホームインストールサーバーの構成

マルチホームホストは、複数のインタフェースを持つシステムであり、複数の IP サブネット上でホストとして機能します。マルチホームホストの詳細は、『[Oracle Solaris 11.1 ネットワークの構成と管理](#)』の「[マルチホームホストの構成](#)」を参照してください。

デフォルトでは、AI インストールサーバーがマルチホームになっている場合は、そのサーバーが接続されているすべてのネットワーク上のインストールクライアントを処理するよう構成されます。この構成を変更するには、`svc:/system/install/server:default` SMF サービスの `all_services/networks` および `all_services/exclude_networks` プロパティを調整します。

`all_services/networks` プロパティの値は、CIDR 形式のネットワークの一覧です (192.168.56.0/24 など)。`all_services/exclude_networks` プロパティの値は、`all_services/networks` プロパティの処理方法を指定するブール型の `true` または `false` です。`exclude_networks` が `false` の場合、AI インストールサーバーは `networks` プロパティに示されたネットワークのみを処理します。`exclude_networks` が `true` の場合、AI インストールサーバーは `networks` プロパティに示されたネットワークを処理しません。

次のコマンドでは、3つのネットワークに接続されている AI インストールサーバーを、1つのネットワーク上のインストールのみを処理するように再構成します。この例では、マルチホームの AI インストールサーバーが3つのネットワーク 192.168.56.0/24、205.10.11.0/24、および 205.10.12.0/24 に接続されています。次のコマンドを実行して、192.168.56.0/24 ネットワーク上のインストールのみを処理します。

```
$ svccfg -s system/install/server:default \  
setprop all_services/networks = 192.168.56.0/24  
$ svcadm refresh system/install/server:default
```

Web サーバーホストのポートの構成

AI サーバーは、Web サーバーを使用してインストールサービスをホストします。デフォルトでは、Web サーバーはポート 5555 でホストされます。インストールサービスの Web サーバーをホストするポートをカスタマイズするには、`svc:/system/install/server:default` SMF サービスの `all_services/port` プロパティを構成します。次のコマンドでは、ポート 7000 からインストールサービスをホストするよう AI サーバーを構成します。

```
$ svccfg -s system/install/server:default setprop all_services/port = 7000  
$ svcadm refresh system/install/server:default
```

注 -port プロパティのカスタマイズは、インストールサービスの作成前に行なってください。インストールサービスの作成後に port プロパティを変更した場合は、それらの既存のインストールサービスが正しく機能しなくなるため、それらを削除して再作成する必要があります。

デフォルトのイメージパスの構成

installadm create-service コマンドで作成されたイメージのデフォルトのベースディレクトリを変更するには、all_services/default_imagepath_basedir プロパティを使用します。

イメージは、all_services/default_imagepath_basedir プロパティの値で指定された場所にある service_name ディレクトリに作成されます。このプロパティのデフォルト値は /export/auto_install です。したがって、service_name サービスのネットイメージは、デフォルトでは /export/auto_install/service_name に作成されます。

次のコマンドでは、デフォルトで /export/aimages/service_name に新しいインストールサービスを作成するように AI サーバーを構成しています。

```
$ svccfg -s system/install/server:default \  
setprop all_services/default_imagepath_basedir = /export/aimages \  
$ svcadm refresh system/install/server:default
```

特定のインストールサービスのこのデフォルトディレクトリをオーバーライドするには、installadm create-service コマンドの -d オプションを使用します。特定の既存のインストールサービスのイメージパスを変更するには、次のコマンドを使用します。

```
$ pfexec installadm set-service -o imagepath=newpath service_name
```

ISC DHCP 構成の自動更新

デフォルトでは、インストールサーバーでクライアントとサービスの構成が変更されると、ローカル ISC DHCP 構成が自動的に更新されます。ローカル ISC DHCP 構成を自動的に管理しない場合は、all_services/manage_dhcp プロパティ値を false に設定します。

114 ページの「ローカル DHCP 設定を含むインストールサービスの作成」に示すように、installadm create-service コマンドを使用して DHCP を構成するときに、all_services/manage_dhcp プロパティの値が false である場合は、コマンドが

終了します。その終了メッセージで、`all_services/manage_dhcp` プロパティの値を `true` にして `installadm create-service` コマンドを再度呼び出すように指示されます。

次のコマンドを使用して、`all_services/manage_dhcp` プロパティの値を確認します。

```
$ svcprop -p all_services/manage_dhcp svc:/system/install/server:default
true
```

AI インストールサービスの作成

インストールサーバーに複数のインストールサービスを配置することができません。クライアントのハードウェアアーキテクチャーごとに、またインストールする異なるバージョンの Oracle Solaris 11 OS ごとに個別のインストールサービスを作成します。

`installadm create-service` コマンドを使用して、AI インストールサービスを作成します。

AI インストールサービスが作成されるときに、AI SMF サービス `system/install/server` がまだ有効になっていなければ、それが有効になります。インストールサービスイメージは `/etc/netboot/svcname` にマウントされます。SPARC インストールサービスでは、`wanboot.conf` ファイルがインストールサービスイメージのルートに置かれます。x86 インストールサービスでは、GRUB メニューがインストールサービスイメージのルートに置かれます。

特定のアーキテクチャーの 1 番目のインストールサービスがインストールサーバーに作成されると、そのサービスの別名 `default-i386` または `default-sparc` が自動的に作成されます。このデフォルトのサービスは独自のマニフェストとプロファイルを備えた完全なサービスですが、このデフォルトのサービスは明示的に作成されたサービスとネットイメージを共有します。このデフォルトのサービスは、そのアーキテクチャーのクライアントのうち、`create-client` サブコマンドを使って別のインストールサービスに明示的に関連付けられなかったクライアントへのすべてのインストールで使用されます。

`default-arch` サービスが別名になっているサービスを変更するには、`set-service` サブコマンドを使用して `aliasof` プロパティを設定します。どちらかのサービスに追加されたマニフェストとプロファイルは、別名のリセット後も変わりません。サービスが使用するネットイメージが変わるだけです。`aliasof` プロパティの設定の詳細は、120 ページの「インストールサービスのプロパティの変更」を参照してください。`default-arch` サービスが別名であるサービスのネットイメージを更新するには、121 ページの「インストールサービスの更新」に示すように、`update-service` サブコマンドを使用します。

default-arch 別名が新しいインストールサービスに変更されたときに、ローカル ISC DHCP 構成が見つかった場合、all_services/manage_dhcp プロパティの値が true であれば、このデフォルトの別名のブートファイルがそのアーキテクチャーのデフォルトの DHCP サーバー全体のブートファイルとして設定されます。all_services/manage_dhcp プロパティの詳細は、108 ページの「ISC DHCP 構成の自動更新」を参照してください。

installadm create-service コマンドでは、ポート 5555 で動作する Web サーバーで ネットイメージも提供されます。たとえば、Web サーバーのアドレスは http://10.80.238.5:5555/solaris11_1-i386 のようになります。異なるポートを使用する場合は、107 ページの「Web サーバーホストのポートの構成」を参照してください。

installadm create-service コマンドには、引数やオプションは必要ありません。一般的に使用される 2 つのオプションを以下に示します。すべてのオプションについては、118 ページの「インストールサービスの作成」または installadm(1M) のマニュアルページを参照してください。

```
installadm create-service [-s source] [-y]
```

-s source source 引数には、ネットイメージのデータソースを指定します。source の値は次のいずれかです。

- IPS AI ネットイメージパッケージの FMRI 識別子 (Oracle Solaris 11.1 リリースでは install-image/solaris-auto-install)。
- AIISO イメージファイルのフルパス名。

source を指定しなかった場合は、install-image/solaris-auto-install パッケージの入手可能な最新バージョンが使用されます。パッケージは、-p オプションで指定された発行元から、またはパッケージのインスタンスが記載されているインストールサーバーの発行元設定一覧の最初の発行元から取得します。

別のバージョンのパッケージをインストールするか、または別の発行元からパッケージをインストールするには、FMRI にそのバージョンまたは発行元を指定します。たとえ

ば、pkg://publisher/install-image/solaris-auto-install または pkg://publisher/install-image/solaris-auto-install@version を指定します。特定の発行元の起点を指定するには、-p オプションを使用します。

-y -d オプションを指定しなかった場合に、自動生成されたイメージパスの使用を確認するプロンプトを抑制するには、-y オプションを使用します。

create-service コマンドは、114 ページの「ローカル DHCP 設定を含むインストールサービスの作成」に示すように、AI インストールサーバーに DHCP を設定できません。

DHCP を設定しないインストールサービスの作成

このセクションの例では、DHCP はすでに別のサーバーに設定されているか、あとで設定します。create-service コマンドによって、ISC DHCP がこのサーバーで動作していることが検出されない場合は、そのコマンドの出力に DHCP の構成手順が表示されます。これらの例では、create-service コマンドの出力は DHCP 構成に必要なブートファイルを提供します。

ISO ファイルを使用した SPARC インストールサービスの作成

この例では、ISO ファイルのネットイメージを使用して SPARC クライアント用の AI インストールサービスを作成します。

```
$ pfexec installadm create-service \
-s /var/tmp/images/sparc/sol-11_1-ai-sparc.iso -y

Creating service from: /var/tmp/images/sparc/sol-11_1-ai-sparc.iso
Setting up the image ...

Creating sparc service: solaris11_1-sparc

Image path: /export/auto_install/solaris11_1-sparc

Service discovery fallback mechanism set up
Creating SPARC configuration file
Refreshing install services

Creating default-sparc alias

Service discovery fallback mechanism set up
Creating SPARC configuration file
No local DHCP configuration found. This service is the default
alias for all SPARC clients. If not already in place, the following should
be added to the DHCP configuration:
Boot file: http://10.80.238.5:5555/cgi-bin/wanboot-cgi

Refreshing install services
$ installadm list
```

Service Name	Alias Of	Status	Arch	Image Path
default-sparc	solaris11_1-sparc	on	sparc	/export/auto_install/solaris11_1-sparc
solaris11_1-sparc	-	on	sparc	/export/auto_install/solaris11_1-sparc

この `installadm create-service` コマンドの実行結果として、次の操作が行われます。

1. インストールサービスに `solaris11_1-sparc` という名前が自動的に付けられます。
2. インストールサービスのデフォルトのネットイメージディレクトリ `/export/auto_install/solaris11_1-sparc` が作成されます。 `-y` オプションが指定されているため、このデフォルトの宛先が受け入れ可能であることを確認するプロンプトが抑制されます。
3. ISO ファイル `/var/tmp/images/sparc/sol-11_1-ai-sparc.iso` がネットイメージの場所 `/export/auto_install/solaris11_1-sparc` に展開されます。
4. このサービス用の `wanboot.conf` ファイルが `/export/auto_install/solaris11_1-sparc/wanboot.conf` に生成されます。
5. `/export/auto_install/solaris11_1-sparc` を `/etc/netboot/solaris11_1-sparc` としてマウントするように AI SMF サービス `system/install/server` がリフレッシュされます。
6. これはこのインストールサーバーで作成された最初の SPARC インストールサービスであるため、`default-sparc` サービス別名が自動的に作成されません。`solaris11_1-sparc` のイメージが別名によって使用されるため、`/export/auto_install/solaris11_1-sparc` は `/etc/netboot/default-sparc` としてもマウントされます。
7. 構成ファイル `/etc/netboot/wanboot.conf` が `/etc/netboot/default-sparc/wanboot.conf` にシンボリックリンクされます。構成ファイル `/etc/netboot/system.conf` が `/etc/netboot/default-sparc/system.conf` にシンボリックリンクされます。
8. DHCP 構成に必要なブートファイル `http://10.80.238.5:5555/cgi-bin/wanboot-cgi` が提供されます。
9. ローカル ISC DHCP サーバーがすでに構成されている場合は、新しい `default-sparc` の別名のブートファイルがすべての SPARC クライアントのデフォルトのブートファイルとして設定されます。この割り当ては、`-i` および `-c` オプションが使用されているかどうかに関係なく行われます。

IPS パッケージを使用した x86 インストールサービスの作成

この例では、IPS パッケージのネットイメージを使用して x86 クライアント用の AI インストールサービスを作成します。このコマンドは、オプションが指定されないときのデフォルトの動作も示しています。このインストールサーバーが SPARC システムである場合は、`-a i386` オプションを指定して、x86 インストールサービスを作成することを指定する必要があります。

このコマンド出力は、DHCP 構成に必要なブートファイルに加えて、DHCP 構成に必要なブートサーバー IP も提供します。


```

$ pfexec installadm create-service -y

Creating service from: pkg:/install-image/solaris-auto-install
DOWNLOAD          PKGS      FILES    XFER (MB)   SPEED
Completed          1/1       514/514  291.8/291.8 0B/s

PHASE              ITEMS
Installing new actions      661/661
Updating package state database      Done
Updating image state          Done
Creating fast lookup database      Done
Reading search index          Done
Updating search index          1/1

Creating i386 service: solaris11_1-i386

Image path: /export/auto_install/solaris11_1-i386

Refreshing install services

Creating default-i386 alias

No local DHCP configuration found. This service is the default
alias for all PXE clients. If not already in place, the following should
be added to the DHCP configuration:
Boot server IP: 192.168.1.111
Boot file(s):
  bios clients (arch 00:00): default-i386/boot/grub/pxegrub2
  uefi clients (arch 00:07): default-i386/boot/grub/grub2netx64.efi

Refreshing install services
$ installadm list

Service Name      Alias Of          Status Arch  Image Path
-----
default-i386      solaris11_1-i386 on    i386  /export/auto_install/solaris11_1-i386
default-sparc     solaris11_1-sparc on    sparc /export/auto_install/solaris11_1-sparc
solaris11_1-i386 -                on    i386  /export/auto_install/solaris11_1-i386
solaris11_1-sparc -                on    sparc /export/auto_install/solaris11_1-sparc

```

この `installadm create-service` コマンドの実行結果として、次の操作が行われます。

1. インストールサービスに `solaris11_1-i386` という名前が自動的に付けられます。
2. ネットイメージのソースのオプションが指定されていないため、最新バージョンの `install-image/solaris-auto-install` パッケージが、インストールサーバーの発行元リスト内でこのパッケージを提供する最初の発行元から取得されます。
3. インストールサービスのデフォルトのネットイメージディレクトリ `/export/auto_install/solaris11_1-i386` が作成されます。 `-y` オプションが指定されているため、このデフォルトの宛先が受け入れ可能であることを確認するプロンプトが抑制されます。
4. `install-image/solaris-auto-install` パッケージがネットイメージの場所 `/export/auto_install/solaris11_1-i386` にインストールされます。

デフォルトでは、インストールされる `install-image/solaris-auto-install` パッケージの形式は AI インストールサーバーのアーキテクチャーに適合します。この例では、インストールサーバーは x86 システムです。このサーバーで SPARC インストールサービスを作成する場合は、`-a` オプションを使用する必要があります。`-a` オプションについては、[118 ページの「インストールサービスの作成」](#)を参照してください。

5. GRUB メニューが `/etc/netboot/solaris11_1-i386/grub.cfg` にマウントされません。
6. `/export/auto_install/solaris11_1-i386` を `/etc/netboot/solaris11_1-i386` としてマウントするように AISMF サービス `system/install/server` がリフレッシュされます。
7. これはこのインストールサーバーで作成された最初の x86 インストールサービスであるため、`default-i386` サービス別名が自動的に作成されます。`solaris11_1-i386` のイメージが別名によって使用されるため、`/export/auto_install/solaris11_1-i386` は `/etc/netboot/default-i386` としてもマウントされます。
8. DHCP 構成に必要なブートサーバー IP が提供されます。DHCP 構成に必要なブートファイル `default-i386/boot/grub/pxegrub2` および `default-i386/boot/grub/grub2netx64.efi` も提供されます。
9. ローカル ISC DHCP サーバーがすでに構成されている場合は、新しい `default-i386` 別名のブートファイルがすべての x86 クライアントのデフォルトのブートファイルとして設定されます。この割り当ては、`-i` および `-c` オプションが使用されているかどうかに関係なく行われます。

ローカル DHCP 設定を含むインストールサービスの作成

`installadm create-service` コマンドを使用して、この AI インストールサーバーに DHCP サーバーを設定できます。`all_services/manage_dhcp` プロパティの値が `true` であることを確認してください。`all_services/manage_dhcp` プロパティの詳細は、[108 ページの「ISC DHCP 構成の自動更新」](#)を参照してください。

次の例では、x86 クライアント用のインストールサービスを作成します。ネットワークは 1 つのサブネットで構成されており、インストールサーバーはネットワークの DHCP サーバーとしても動作します。このインストールサービスは、`10.80.239.150` (`-i`) から始まる 20 個の IP アドレス (`-c`) を提供します。DHCP サーバーがまだ構成されていない場合は、ISC DHCP サーバーが構成されます。ISC DHCP サーバーがすでに構成されている場合は、その DHCP サーバーが更新されません。

-i と -c 引数が指定され、DHCP が構成されている場合は、作成されるインストールサービスと IP 範囲とのバインディングは存在しません。-i と -c が渡され、IP 範囲が設定されている場合は、必要に応じて新規 DHCP サーバーが作成され、その DHCP サーバーは使用するすべてのインストールサービスおよびすべてのクライアントで起動され、実行された状態のままです。DHCP サーバーに指定されたネットワーク情報には、作成されるサービスとの特定の関係はありません。

インストールサーバーが直接接続されているサブネット上にリクエストされた IP 範囲がなく、インストールサーバーがマルチホームである場合は、-B オプションを使用してブートファイルサーバーのアドレス (通常はこのシステム上の IP アドレス) を指定します。このオプションは、複数の IP アドレスがインストールサーバーに構成され、DHCP リレーが採用されている場合にのみ必要です。その他の構成では、ソフトウェアでこれを自動的に決定できます。

```
$ pfexec installadm create-service \
-s /var/tmp/images/i386/sol-11_1-ai-x86.iso \
-i 10.80.239.150 -c 20 -y

Creating service from: /var/tmp/images/i386/sol-11_1-ai-x86.iso
Setting up the image ...

Creating i386 service: solaris11_1-i386

Image path: /export/auto_install/solaris11_1-i386

Starting DHCP server...
Adding IP range to local DHCP configuration

Refreshing install services

Creating default-i386 alias.

Setting the default PXE bootfile(s) in the local DHCP configuration to:
bios clients (arch 00:00): default-i386/boot/grub/pxegrub2
uefi clients (arch 00:07): default-i386/boot/grub/grub2netx64.efi

Refreshing install services
```

この `installadm create-service` コマンドの実行結果として、次の操作が行われます。

1. インストールサービスに `solaris11_1-i386` という名前が自動的に付けられます。
2. インストールサービスのデフォルトのネットイメージディレクトリ `/export/auto_install/solaris11_1-i386` が作成されます。-y オプションが指定されているため、このデフォルトの宛先が受け入れ可能であることを確認するプロンプトが抑制されます。
3. ISO ファイル `/var/tmp/images/i386/sol-11_1-ai-x86.iso` がネットイメージの場所 `/export/auto_install/solaris11_1-i386` に展開されます。

4. GRUB メニューが `/etc/netboot/solaris11_1-i386/grub.cfg` にマウントされま
す。
5. `/export/auto_install/solaris11_1-i386` を `/etc/netboot/solaris11_1-i386` として
マウントするように AISMF サービス `system/install/server` がリフレッシュされ
ます。
6. これはこのインストールサーバーで作成された最初の x86 インストールサービス
であるため、`default-i386` サービス別名が自動的に作成されま
す。`solaris11_1-i386` のイメージが別名によって使用されるた
め、`/export/auto_install/solaris11_1-i386` は `/etc/netboot/default-i386` として
もマウントされます。
7. 必要に応じて DHCP サービスが作成され、`10.80.239.150` から `10.80.239.169` まで
の IP アドレスがプロビジョニングされます。このサーバーですでに DHCP サービ
スが設定されている場合は、`-i` および `-c` オプションによって、DHCP サーバーが
このサービスの新しい IP アドレスで更新されます。`svc:/network/dhcp/server`
サービスが `onLine` になります。
8. ブートファイル `default-i386/boot/grub/pxegrub2` および
`default-i386/boot/grub/grub2netx64.efi` が PXE クライアントのデフォルトの
ブートファイルとしてローカル DHCP 構成に追加されます。

以降のセクションでは、`installadm` によって ISC DHCP 構成用の DHCP 構成ファイル
にどのように情報が追加されるかを示します。ISC DHCP の構成の詳細は、[18 ページ](#)
の「[関連情報](#)」を参照してください。

Oracle Solaris 11.1 i386 インストールサービスの ISC DHCP 構成

次の例は、`installadm` によって、`-i` および `-c` オプションを使用して指定した IP アド
レスが、上記で作成した Oracle Solaris 11.1 i386 インストールサービスの ISC DHCP 構
成用の `/etc/inet/dhcd4.conf` ファイルにどのように追加されるかを示しています。

```
subnet 10.80.239.0 netmask 25.255.255.0 {
  range 10.80.239.150 10.80.239.169;
  option broadcast-address 10.80.239.255;
  option routers 10.80.239.1;
  next-server 10.80.239.170;
}
```

次の例は、`installadm` によって、デフォルトの PXE ブートファイルが上記で作成し
た `default-i386` Oracle Solaris 11.1 i386 インストールサービスの ISC DHCP 構成用の
`/etc/inet/dhcd4.conf` ファイルにどのように設定されるかを示しています。

```
class "PXEBoot" {
  match if (substring(option vendor-class-identifier, 0, 9) = "PXEClient");
  if option arch = 00:00 {
    filename "default-i386/boot/grub/pxegrub2";
  } else if option arch = 00:07 {
    filename "default-i386/boot/grub/grub2netx64.efi";
  }
}
```

```

    }
}

```

Oracle Solaris 11 i386 インストールサービスの ISC DHCP 構成

Oracle Solaris 11.1 サービスの代わりに Oracle Solaris 11 i386 インストールサービスを作成した場合は、次の例のような出力が表示されます。

```

If not already in place, the following should be added to the DHCP configuration:
    Boot server IP      : 10.134.125.136
    Boot file           : default-i386/boot/grub/pxegrub

```

次の例は、`installadm`によって、デフォルトの PXE ブートファイルが Oracle Solaris 11 i386 インストールサービスの ISC DHCP 構成用の `/etc/inet/dhcdpd4.conf` ファイルにどのように設定されるかを示しています。

```

class "PXEBoot" {
    match if (substring(option vendor-class-identifier, 0, 9) = "PXEclient");
    if option arch = 00:00 {
        filename "default-i386/boot/grub/pxegrub";
    }
}

```

Oracle Solaris 11.1 sparc インストールサービスの ISC DHCP 構成

i386 サービスの代わりに `sparc` インストールサービスを作成した場合は、次の例のような出力が表示されます。

```

If not already in place, the following should be added to the DHCP configuration:
Boot file: http://10.80.238.5:5555/cgi-bin/wanboot-cgi

```

次の例は、`installadm`によって、デフォルトのブートファイルが Oracle Solaris 11.1 `sparc` インストールサービスの ISC DHCP 構成用の `/etc/inet/dhcdpd4.conf` ファイルにどのように設定されるかを示しています。

```

class "SPARC" {
    match if not (substring(option vendor-class-identifier, 0, 9) = "PXEclient");
    filename "http://10.80.238.5:5555/cgi-bin/wanboot-cgi";
}

```

インストールサーバーの維持

AI インストールサーバーを設定したあとは、次のタスクのいくつかを実行することができます。[installadm\(1M\)](#) のマニュアルページも参照してください。

- [118 ページ](#)の「インストールサービスの追加、変更、または削除」
- [123 ページ](#)の「インストールサービスへのクライアントの関連付け」

- 124 ページの「インストールサービスへのクライアント固有のインストール手順の関連付け」
- 127 ページの「インストールサービスへのクライアント固有の構成手順の関連付け」
- 130 ページの「AI マニフェストまたはシステム構成プロファイルのエクспорт」
- 130 ページの「AI マニフェストまたはシステム構成プロファイルの条件の変更」
- 131 ページの「インストールサーバー上のすべてのサービスの一覧表示」
- 132 ページの「インストールサービスに関連付けられたクライアントの一覧表示」
- 133 ページの「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」

インストールサービスの追加、変更、または削除

インストールを計画しているクライアントのアーキテクチャーごとに、またクライアントシステムへのインストールを計画している Oracle Solaris 11 OS のバージョンごとに個別のインストールサービスが必要になります。

インストールサービスの作成

インストールサービスを作成するには、次のコマンドを使用します。例については、109 ページの「AI インストールサービスの作成」を参照してください。

```
installadm create-service [-n svcname] [-s source]  
    [-p publisher=origin] [-a architecture]  
    [-d imagepath] [-y] [-t aliasof]  
    [-i start] [-c count]  
    [-b property=value,...] [-B server]
```

-n *svcname*

svcname 引数の値は、英数字、アンダースコア (_)、ハイフン (-) で構成できません。 *svcname* の最初の文字をハイフンにすることはできません。 *svcname* 引数の値の長さは 63 文字を超えることはできません。インストールサービスの名前を指定しない場合は、デフォルトの名前が割り当てられます。デフォルトの名前は、アーキテクチャーと OS バージョンを示しています。

-s *source*

source 引数には、ネットイメージのデータソースを指定します。 *source* の値は次のいずれかです。

1. IPS AI ネットイメージパッケージの FMRI 識別子 (Oracle Solaris 11.1 リリースでは `install-image/solaris-auto-install`)。
2. AI ISO ネットイメージファイルのフルパス名。

source を指定しなかった場合は、`install-image/solaris-auto-install` パッケージの入手可能な最新バージョンが使用されます。パッケージは、`-p` オプションで指定された発行元から、またはパッケージのインスタンスが記載されているインストールサーバーの発行元設定一覧の最初の発行元から取得します。

別のバージョンのパッケージをインストールするか、または別の発行元からパッケージをインストールするには、FMRI にそのバージョンまたは発行元を指定します。たとえば、`pkg://publisher/install-image/solaris-auto-install` または `pkg://publisher/install-image/solaris-auto-install@version` を指定します。特定の発行元の起点を指定するには、`-p` オプションを使用します。

`-p publisher=origin`

このオプションは、ネットイメージのソースが IPS パッケージである場合にのみ使用されます。このオプションは、`install-image/solaris-auto-install` パッケージの取得元の IPS パッケージリポジトリを指定します。*publisher* 引数の値は発行元名であり、*origin* は URI (`solaris=http://pkg.oracle.com/solaris/release/` など) です。

`-p` オプションが指定されなかった場合、使用される発行元は、パッケージのインスタンスが記載されているインストールサーバーの発行元設定一覧の最初の発行元です。

`-a architecture`

このオプションは、ネットイメージのソースが IPS パッケージである場合にのみ使用されます。*architecture* 引数は、このサービスとともにインストールするクライアントのアーキテクチャーを指定します。`i386` または `sparc` のどちらかを指定できます。デフォルト値は、インストールサーバーのアーキテクチャーです。

`-d imagepath`

imagepath 引数の値は、ネットイメージを作成するパスです。`install-image/solaris-auto-install` パッケージがこの場所にインストールされるか、または指定された ISO ファイルがこの場所で展開されます。*imagepath* を指定しなかった場合、イメージは `all_services/default_imagepath_basedir` プロパティの値で定義された場所にある *svcname* ディレクトリに作成されます。このプロパティのデフォルト値やこのプロパティの値を変更する方法については、108 ページの「デフォルトのイメージパスの構成」を参照してください。*imagepath* を指定しなかった場合は、自動生成された場所を使用するかどうかを確認するプロンプトが表示されます。このプロンプトを抑制するには、`-y` オプションを指定します。

`-y`

自動生成されたイメージパスの使用を確認するプロンプトを抑制するには、`-y` オプションを指定します。

`-t aliasof`

このオプションは、新しいサービスを別名として指定します。新しいサービスは、*aliasof* サービスのネットイメージを共有しますが、独自のマニフェスト、プロファイル、およびクライアントを備えています。

-i start

このオプションは、ローカル DHCP 構成に追加される範囲の開始 IP アドレスを指定します。IP アドレスの数は、**-c** オプションで指定されます。ローカルの ISC DHCP 構成が存在せず、`all_services/manage_dhcp` プロパティの値が `true` である場合は、ISC DHCP サーバーが起動します。`all_services/manage_dhcp` プロパティの詳細は、[108 ページの「ISC DHCP 構成の自動更新」](#)を参照してください。

-c count

DHCP 構成の IP アドレスの総数を `count` の値と同じに設定します。最初の IP アドレスは、**-i** オプションで指定される `start` の値です。

-b property=value,...

x86 サービスの場合のみ。このオプションは、サービスイメージ内のサービス固有の `grub.cfg` ファイルにプロパティ値を設定します。このオプションを使用して、このサービスに固有のブートプロパティを設定します。このオプションには、コンマで区切られた複数の `property=value` ペアを指定できます。

-B server

クライアントがブートファイルを要求するブートサーバーの IP アドレスを指定するには、このオプションを使用します。このオプションは、この IP アドレスをほかの方法で決定できない場合は必須です。

インストールサービスのプロパティの変更

`svcname` インストールサービスに設定するプロパティと値を指定するには、`installadm set-service` コマンドを使用します。

```
installadm set-service -o property=value svcname
```

`property=value` ペアは、次のいずれかにする必要があります。

aliasof=aliasof

`svcname` サービスが別名になっているインストールサービスを変更します。

このプロパティを設定すると、`svcname` サービスが `aliasof` サービスの別名になるように変更されます。`svcname` サービスはすでに別名である必要があります。`default-arch` インストールサービスは別名です。`create-service` の `-t` オプションを使用して作成したサービスは別名です。`svcname` が別名であることを確認するには、[131 ページの「インストールサーバー上のすべてのサービスの一覧表示」](#)に示すように `installadm list` コマンドを使用します。

`svcname` と `aliasof` のどちらかに追加されたマニフェスト、プロファイル、およびクライアントバイndingは、別名のリセット後も変わりません。`svcname` サービスが使用するネットイメージが変わるだけです。

新しいネットイメージに関連付けられた `AIDTD` と `SMFDTD` が異なる可能性があるため、別名を設定する前に `svcname` に追加されたマニフェストとプロファイル

は、別名のリセット時に再検証されます。この検証は、後述の `create-manifest` および `create-profile` によって実行される検証と同じです。

`default-manifest=manifest`

指定されたサービスにすでに登録されている特定のマニフェストまたは派生したマニフェストスクリプトを、そのサービスのデフォルトのマニフェストまたはスクリプトに指定します。このサービスに登録されているマニフェストおよびスクリプトの一覧を表示するには、次のコマンドを使用します。

```
$ installadm list -n svcname -m
```

`imagepath=newpath`

サービスが作成されたあとでサービスのイメージを再配置します。

```
$ pfexec installadm set-service -o imagepath=/export/aiimages/solaris11_1-i386 solaris11_1-i386
```

インストールサービスの更新

IPS AI ネットイメージパッケージを使用して作成されたサービスの別名に関連付けられたイメージを更新するには、次のコマンドを使用します。

```
installadm update-service [-p|--publisher publisher=origin]  
[-s|--source FMRI] svcname
```

このコマンドは、`svcname` に関連付けられたイメージを更新します。`svcname` は、IPS AI ネットイメージパッケージを使用して作成されたサービスの別名です。更新されたイメージを使用して新規サービスが作成され、新規サービスに別名として `svcname` が設定されます。

`-p|--publisher publisher=origin`

このオプションは、`svcname` イメージを更新する IPS パッケージリポジトリを指定します。値の例は `solaris=http://pkg.oracle.com/solaris/release/` です。

`-p` オプションが指定されていない場合、`svcname` が別名であるサービスのイメージの作成に使用された発行元が使用されます。次の `pkg publisher` コマンドは、発行元 `svcname` を表示する方法を示します。

```
$ installadm list
Service Name      Alias Of          Status Arch  Image Path
-----
default-i386     solaris11_1-i386 on    i386  /export/auto_install/solaris11_1-i386
solaris11_1-i386 -                on    i386  /export/auto_install/solaris11_1-i386
$ pkg -R /export/auto_install/solaris11_1-i386 publisher
PUBLISHER      TYPE      STATUS  URI
solaris        origin   online  http://pkg.oracle.com/solaris/release/
```

`-s|--source FMRI`

このオプションは、更新用のネットイメージパッケージの FMRI を指定します。

`-s` オプションが指定されていない場合、入手可能な最新バージョンの `install-image/solaris-auto-install` パッケージが、`-p` オプションの記述子で指定された発行元から使用されます。

インストールサービスの名前変更

svcname の名前を *newsvcname* に変更するには、次のコマンドを使用します。

```
installadm rename-service svcname newsvcname
```

newsvcname 引数の値は、英数字、アンダースコア (_)、ハイフン (-) で構成できません。 *newsvcname* の最初の文字をハイフンにすることはできません。 *newsvcname* 引数の値の長さは 63 文字を超えることはできません。

インストールサービスの有効化または無効化

svcname インストールサービスを有効にするには、次のコマンドを使用します。

```
installadm enable svcname
```

svcname インストールサービスを無効にするには、次のコマンドを使用します。

```
installadm disable svcname
```

インストールサービスを無効にすると、そのインストールサービスに関連付けられたクライアントは関連付けられたままになります。 [123 ページの「インストールサービスへのクライアントの関連付け」](#) を参照してください。これは、インストールサービスを再度有効にしたときに、そのインストールサービスのクライアントを再度関連付けする必要がないことを意味します。

アーキテクチャーのデフォルトのインストールサービスまたはデフォルトサービスの別名になっているサービスを無効にすると、ほかのサービスに関連付けられていないそのアーキテクチャーのクライアントはブートしません。

インストールサービスの削除

svcname インストールサービスを削除するには、次のコマンドを使用します。

```
installadm delete-service [-r] [-y] svcname
```

このコマンドは、 *svcname* インストールサービスの AI マニフェスト、システム構成プロファイル、ネットイメージ、および Web サーバー構成を削除します。サービスがデフォルトの別名であり、ローカル ISC DHCP 構成が存在する場合は、 `all_services/manage_dhcp` プロパティの値が `true` になっていると、このサービスに関連付けられたブートファイルが ISC DHCP 構成から削除されません。 `all_services/manage_dhcp` プロパティの詳細は、 [108 ページの「ISC DHCP 構成の自動更新」](#) を参照してください。

このサービスに関連付けられたすべてのクライアントと、このサービスの別名になっているすべてのサービスを削除するには、 `-r` オプションを使用します。確認のプロンプトを抑制するには、 `-y` オプションを使用します。

アーキテクチャーのデフォルトのインストールサービスまたはデフォルトサービスの別名になっているサービスを削除すると、ほかのサービスに関連付けられていないそのアーキテクチャーのクライアントはブートしません。操作の確認を求めるプロンプトが表示されます。

インストールサービスへのクライアントの関連付け

`installadm create-client` コマンドは、クライアントを特定のインストールサービスに関連付けます。詳しい例と出力例については、[225 ページの「インストールクライアントを設定する」](#)を参照してください。

インストールサービスへのクライアントの追加

`macaddr` クライアントを `svcname` インストールサービスに関連付け、x86 クライアント用のカスタムのクライアント設定を指定するには、`installadm create-client` コマンドを使用します。

```
installadm create-client [-b property=value,...] -e macaddr -n svcname
```

クライアントが x86 システムであり、ローカル ISC DHCP 構成が存在する場合は、`all_services/manage_dhcp` プロパティの値が `true` になっていると、ISC DHCP 構成でクライアントが構成されます。`all_services/manage_dhcp` プロパティの詳細は、[108 ページの「ISC DHCP 構成の自動更新」](#)を参照してください。

システムの MAC アドレスを見つけるには、次の例に示すように `dladm` コマンドを使用します。詳細は、[dladm\(1M\)](#) のマニュアルページを参照してください。

```
$ dladm show-linkprop -p mac-address
LINK  PROPERTY      PERM VALUE      DEFAULT POSSIBLE
net0  mac-address    rw  8:0:20:0:0:1    8:0:20:0:0:1    --
net1  mac-address    rw  0:14:4f:45:c:2d 0:14:4f:45:c:2d --
```

x86 クライアントシステムの場合、`-b` オプションを使用して、`/etc/netboot` 内のクライアント固有の `grub.cfg` ファイルにブートプロパティを設定します。

次のコマンドは、MAC アドレスが `00:14:4f:a7:65:70` のクライアントを `solaris11_1-sparc` インストールサービスに追加します。

```
$ pfexec installadm create-client -e 00:14:4f:a7:65:70 -n solaris11_1-sparc
```

次の例では、x86 クライアントを追加し、インストール出力をシリアルコンソールにリダイレクトします。

```
$ pfexec installadm create-client -e c0ffeec0ffee -n solaris11_1-i386 -b console=ttya
```

別のインストールサービスへのクライアントの関連付け

クライアントは、1つのインストールサービスにのみ関連付けることができます。installadm create-client コマンドを複数回実行し、各回で同じ MAC アドレスを指定する場合、クライアントは最後に指定したインストールサービスにのみ関連付けられます。

インストールサービスからのクライアントの削除

macaddr クライアントをその関連付けられたインストールサービスから削除するには、installadm delete-client コマンドを使用します。

```
installadm delete-client macaddr
```

クライアントが x86 システムであり、ローカル ISC DHCP 構成が存在する場合は、all_services/manage_dhcp プロパティの値が true になっていると、ISC DHCP 構成でクライアントが構成解除されます。all_services/manage_dhcp プロパティの詳細は、108 ページの「ISC DHCP 構成の自動更新」を参照してください。

次のコマンドは、MAC アドレスが 00:14:4f:a7:65:70 のクライアントを削除します。クライアントを関連付けることができるインストールサービスは1つだけなので、サービス名を指定する必要はありません。

```
$ pfexec installadm delete-client 00:14:4f:a7:65:70
```

インストールサービスへのクライアント固有のインストール手順の関連付け

インストールサービスごとにインストール手順セットを複数指定できます。また、使用する手順セットをクライアントごとに指定できます。

インストールサービスへの AI マニフェストの追加

manifest カスタム AI マニフェストまたは派生マニフェストスクリプトを svcname インストールサービスに追加するには、installadm create-manifest コマンドを使用します。

```
installadm create-manifest -n svcname
  -f filename [-m manifest]
  [-c criteria=value|list|range...
  | -C criteriafile] [-d]
```

manifest 引数の値は、AI マニフェストの XML ファイルまたは派生マニフェストスクリプトです。第10章「クライアントシステムのプロビジョニング」を参照してください。create-manifest サブコマンドは、XML マニフェストファイルを検証してから、それらをインストールサービスに追加します。派生マニフェストスクリプト

ファイルを検証するには、166 ページの「インストールサービスへの派生マニフェストスクリプトの追加」に示すように `aimanifest validate` コマンドを使用します。

`manifest` 引数の値は、`installadm list` コマンドで表示される名前です。133 ページの「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」を参照してください。-m オプションを指定しなかった場合、マニフェストの名前は `ai_instance` 要素 (存在する場合) の `name` 属性の値か、または `filename` の値のベース名です。

この AI マニフェストをデフォルトの AI マニフェストにするには、-d オプションを使用します。デフォルトのマニフェストとは、このインストールサービスのほかのどのマニフェストに指定された条件にも一致しないすべてのクライアントによって使用されるマニフェストです。-d を指定した場合は、マニフェストの選択のために -c および -c オプションで指定した条件が無視されます。このサービスの以前のデフォルトの AI マニフェストにクライアント条件がない場合、それは非アクティブになります。以前のデフォルトのマニフェストに条件がある場合、それは引き続きアクティブのまま、それに関連付けられた条件が有効になります。

-d を指定しない場合は、-c または -c のどちらかを指定して、この AI マニフェストを使ってインストールを完了するクライアントを定義する必要があります。-d、-c、および -c をすべて指定しない場合、このマニフェストはサービスに追加されますが非アクティブになります。つまり、クライアントはそれを使用できません。

特定のクライアントでこの AI マニフェストを使用する場合は、まずそれらのクライアントがこの `create-manifest` コマンドで指定したインストールサービスを使用することを確認します。`create-client` コマンドで特定のインストールサービスに明示的に関連付けられていないクライアントシステムは、適切な `default-arch` インストールサービスを使用します。カスタマイズされた AI マニフェストを `default-arch` インストールサービスに追加することも、カスタマイズされた AI マニフェストを別のサービスに追加してから、`create-client` を使ってクライアントが必ずそのサービスを使用するようにすることもできます。

-c オプションは、コマンド行でクライアントの選択条件を指定します。-c オプションは、XML ファイルで条件を指定します。`criteriafile` の値は、フルパスとファイル名です。条件のキーワードとコマンド行およびファイル例の一覧については、第 9 章「インストールのカスタマイズ」を参照してください。

`installadm create-manifest` コマンドは、同じタイプの条件が重複していないことを検証します。たとえば、ある条件仕様が 10.0.0.0 から 10.255.255.255 までの IP アドレスに一致する場合、IP アドレス 10.10.10.10 に一致する条件仕様を追加しようとすると、`installadm` がエラーを伴って終了します。条件仕様の詳細については、第 9 章「インストールのカスタマイズ」を参照してください。

次のコマンドは、`manifest_t200.xml` マニフェストを `solaris11_1-sparc` インストールサービスに追加します。-c オプションは、このインストールサービスを使用

していて、Sun Fire T200 サーバーとして識別されるクライアントに `manifest_t200.xml` のインストール手順が割り当てられるように指定します。

```
$ pfexec installadm create-manifest -f ./mymanifests/manifest_t200.xml \
-m t200 -n solaris11_1-sparc -c platform="SUNW,Sun-Fire-T200"
```

次のコマンドは、`criteria_t200.xml` ファイルの内容が下に示したようになっている場合は、前のコマンドと同等です。

```
$ pfexec installadm create-manifest -f ./mymanifests/manifest_t200.xml \
-m t200 -n solaris11_1-sparc -C ./mymanifests/criteria_t200.xml
```

`criteria_t200.xml` ファイルの内容を次に示します。

```
<ai_criteria_manifest>
  <ai_criteria name="platform">
    <value>SUNW,Sun-Fire-T200</value>
  </ai_criteria>
</ai_criteria_manifest>
```

AI マニフェストの更新

`svcname` インストールサービスの *manifest* AI マニフェストまたは派生マニフェストスクリプトファイルの内容を *filename* マニフェストまたはスクリプトファイルの内容で置き換えるには、`installadm update-manifest` コマンドを使用します。更新の結果として、条件、デフォルトのステータス、および *manifest* 名は変更されません。

```
installadm update-manifest -n svcname -f filename [-m manifest]
```

`update-manifest` サブコマンドは、XML マニフェストファイルを検証してから、それらをインストールサービスに追加します。派生マニフェストスクリプトファイルを検証するには、[166 ページの「インストールサービスへの派生マニフェストスクリプトの追加」](#)に示すように `aimanifest validate` コマンドを使用します。

manifest マニフェストは `svcname` サービスにすでに存在している必要があります。確認には `installadm list` コマンドを使用します。[133 ページの「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」](#)を参照してください。

manifest を指定しなかった場合は、置き換えられるマニフェストが次のいずれかの方法で特定されます。

- *filename* マニフェスト内の `ai_instance` 要素の `name` 属性 (この属性が指定され、この属性の値がこのインストールサービスの既存のマニフェストの *manifest* 名に一致する場合)。
- *filename* の値のベース名 (この名前がこのインストールサービスの既存のマニフェストの *manifest* 名に一致する場合)。

次のコマンドは、`solaris11_1-sparc` サービスの `t200` マニフェストの内容を `./mymanifests/manifest_newt200.xml` の内容で更新します。`installadm list` でのマニフェストの名前は引き続き `t200` です。

```
$ pfexec installadm update-manifest -n solaris11_1-sparc \  
-f ./mymanifests/manifest_newt200.xml -m t200
```

AI マニフェストの削除

manifest AI マニフェストまたは派生マニフェストスクリプトを *svcname* インストールサービスから削除するには、`installadm delete-manifest` コマンドを使用します。*manifest* 引数の値は、`installadm list` コマンドによって返されるマニフェスト名です。133 ページの「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」を参照してください。

```
installadm delete-manifest -m manifest -n svcname
```

デフォルト AI マニフェストは削除できません。

次のコマンドは、`t200` AI マニフェストを `solaris11_1-sparc` インストールサービスから削除します。

```
$ pfexec installadm delete-manifest -m t200 -n solaris11_1-sparc
```

インストールサービスへのクライアント固有の構成手順の関連付け

インストールサービスごとにシステム構成手順セットを複数指定できます。複数のシステム構成プロファイルを各クライアントに関連付けることができます。

インストールサービスへのシステム構成プロファイルの追加

filename システム構成プロファイルを *svcname* インストールサービスに追加するには、`installadm create-profile` コマンドを使用します。

```
installadm create-profile -n svcname  
-f filename... [-p profile]  
[-c criteria=value|list|range... | -C criteriafile]
```

1つのクライアントで複数のシステム構成プロファイルを使用できるため、1回の `create-profile` コマンドで複数の構成プロファイルを指定できます。複数のプロファイルに対して、同じクライアント選択条件(重複する条件)を指定することも、条件を指定しないこともできます。条件を指定しない場合は、このインストールサービスを使用するすべてのクライアントによってそのプロファイルが使用されます。

`create-profile` サブコマンドは、システム構成プロファイルを検証してから、それらをインストールサービスに追加します。作成中のプロファイルを検証する場合は、後述の `validate` サブコマンドを参照してください。

filename ファイルには、インストール処理中にクライアントのインストール環境から得られた値で置換される変数を含めることができます。詳細は、185 ページの「システム構成プロファイルテンプレートの使用」を参照してください。

profile 引数の値は、プロファイルの追加後に `installadm list` コマンドで表示されるプロファイル名です。133 ページの「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」を参照してください。-p オプションを指定しなかった場合、プロファイルの名前は *filename* ファイルのベース名になります。*filename* を複数指定した場合、-p オプションは無効になります。

-c オプションは、コマンド行でクライアントの選択条件を指定します。-c オプションは、XML ファイルで条件を指定します。*criteriafile* の値は、フルパスとファイル名です。条件のキーワードとコマンド行およびファイル例の一覧については、第 9 章「インストールのカスタマイズ」を参照してください。

特定のクライアントでこのシステム構成プロファイルを使用する場合は、まずそれらのクライアントがこの `create-profile` コマンドで指定したインストールサービスを使用することを確認します。`create-client` コマンドで特定のインストールサービスに明示的に関連付けられていないクライアントシステムは、適切な `default-arch` インストールサービスを使用します。カスタマイズされたシステム構成プロファイルを `default-arch` インストールサービスに追加することも、カスタマイズされた構成プロファイルを別のサービスに追加してから、`create-client` を使ってクライアントが必ずそのサービスを使用するようにすることもできます。

次のコマンドは、`profile_t200.xml` プロファイルを `solaris11_1-sparc` インストールサービスに追加します。-c オプションは、このインストールサービスを使用していて、Sun Fire T200 サーバーとして識別されるクライアントに `profile_t200.xml` のシステム構成手順が割り当てられるように指定します。

```
$ pfexec installadm create-profile -f ./myprofiles/profile_t200.xml \  
-p t200 -n solaris11_1-sparc -c platform="SUNW,Sun-Fire-T200"
```

システム構成プロファイルの更新

svcname インストールサービスの指定したプロファイルを *filename* の内容と置き換えるには、`installadm update-profile` コマンドを使用します。プロファイルの条件は更新後もそのままです。

```
installadm update-profile -n|--service svcname  
-f|--file filename  
[-p|--profile profile]
```

更新されるプロファイルは、*svcname* インストールサービスの *profile* プロファイルです (指定した場合)。*svcname* インストールサービスの *profile* を指定しなかった場合、更新されるプロファイルの名前は *profile* のベース名を持つプロファイルです。

次のコマンドは、`solaris11_1-sparc` サービスの `t200` プロファイルの内容を `./myprofiles/profile_newt200.xml` の内容で更新します。


```
$ pfexec installadm update-profile -n solaris11_1-sparc \  
-f ./myprofiles/profile_newt200.xml -p t200
```

システム構成プロファイルの検証

システム構成プロファイルの構文の正しさを検証するには、`installadm validate` コマンドを使用します。

```
installadm validate -n svcname -P filename... | -p profile...
```

インストールサービスに追加されていないプロファイルを検証するには `-p` オプションを使用します。 `filename` 引数の値は、プロファイルファイルのフルパス名です。

`create-profile` サブコマンドを使用して `svcname` インストールサービスにすでに追加されているプロファイルを検証するには、`-p` オプションを使用します。 `profile` に指定可能な値を表示するには、133 ページの「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」に示すように、`installadm list` コマンドを使用します。 `create-profile` サブコマンドは、システム構成プロファイルを検証してから、それらをインストールサービスに追加します。 `validate -p` サブコマンドは、プロファイルが追加されて以降、壊れていないことを検証します。

`svcname` は、`filename` と `profile` の両方のプロファイルに必要です。それぞれのバージョンの OS で `service_bundle(4)` DTD が異なる可能性があるため、インストールサービスにまだ追加されていないプロファイルにはサービス名が必要です。インストールサーバーが実行しているバージョンと異なるバージョンの OS をインストールするようにインストールサービスを定義することもできます。インストールされるクライアントで使用される DTD に対してプロファイルを検証する必要があります。

検証されたプロファイルは `stdout` に出力されます。エラーは `stderr` に出力されます。

システム構成プロファイルの削除

`profile` システム構成プロファイルを `svcname` インストールサービスから削除するには、`installadm delete-profile` コマンドを使用します。 `profile` 引数の値は、`installadm list` コマンドによって返されるプロファイル名です。133 ページの「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」を参照してください。

```
installadm delete-profile -p profile... -n svcname
```

次のコマンドは、`t200` システム構成プロファイルを `solaris11_1-sparc` インストールサービスから削除します。

```
$ pfexec installadm delete-profile -p t200 -n solaris11_1-sparc
```

AI マニフェストまたはシステム構成プロファイルのエクスポート

指定された AI マニフェストまたはシステム構成プロファイルの内容を *svcname* インストールサービスから *pathname* ファイルまたはディレクトリにコピーするには、`installadm export` コマンドを使用します。

```
installadm export -n svcname
                  -m manifest... -p profile...
                  [-o pathname]
```

-o オプションを指定しなかった場合は、マニフェストとプロファイルの内容が `stdout` に出力されます。入力ファイルを1つしか指定しない場合は、*pathname* 引数の値をファイル名にできます。複数の入力ファイルを指定する場合は、*pathname* をディレクトリにする必要があります。

manifest は、XML AI マニフェストまたは派生マニフェストスクリプトの名前で、マニフェストおよび派生マニフェストスクリプトの作成については、[第10章「クライアントシステムのプロビジョニング」](#)を参照してください。

`installadm export` コマンドを使用して次のタスクを実行します。

- マニフェストおよびプロファイル内の仕様を確認します。
- 既存のマニフェストまたはプロファイルを変更します。
- 新しいマニフェストまたはプロファイルを作成するために既存のマニフェストまたはプロファイルをベースとして使用します。

AI マニフェストまたはシステム構成プロファイルの条件の変更

`create-manifest` または `create-profile` を使用して *svcname* インストールサービスにすでに追加した AI マニフェストまたはシステム構成プロファイルに関連付けられたクライアント条件を更新するには、`installadm set-criteria` コマンドを使用します。

```
installadm set-criteria -m manifest -p profile... -n svcname
                       -c criteria=value|list|range... | -C criteriafile |
                       -a criteria=value|list|range...
```

同じ `set-criteria` コマンド行には、0 または 1 つのマニフェストを、0 または任意の数のプロファイルとともに指定できます。*manifest* と *profile* の名前は、`installadm list` コマンドによって返される名前です。[133 ページの「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」](#)を参照してください。

-m オプションによってサービスのデフォルトのマニフェストを指定した場合は、条件が追加または変更されますが、この条件はクライアントのインストール時に無視されます。133 ページの「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」を参照してください。

これらの既存のマニフェストおよびプロファイルの条件を、指定された新しい条件に置き換えるには、-c または -c オプションを使用します。既存の条件を保持し、指定された条件を追加するには、-a オプションを使用します。条件の指定方法の詳細は、第 9 章「インストールのカスタマイズ」を参照してください。

次のコマンドは、最初にプラットフォームの条件仕様とともにこのサービスに追加されたマニフェストに、メモリーの条件仕様を追加します。

```
$ pfexec installadm set-criteria -m t200 -n solaris11_1-sparc -a mem="4096-unbounded"
```

条件が create-manifest で指定され、set-criteria で追加された結果、マニフェストは、このインストールサービスを使用していて、Sun Fire T200 サーバーであり、4G バイト以上のメモリーを搭載しているクライアントによって使用されます。

これと同じ結果は、次の criteria_t200.xml ファイルで -a オプションの代わりに -c オプションを使用して得ることもできます。

```
<ai_criteria_manifest>
  <ai_criteria name="platform">
    <value>SUNW,Sun-Fire-T200</value>
  </ai_criteria>
  <ai_criteria name="mem">
    <range>
      4096
      unbounded
    </range>
  </ai_criteria>
</ai_criteria_manifest>
```

インストールサービスの情報の表示

インストールサービスの情報を表示するには、installadm list コマンドを使用します。

```
installadm list [-n svcname] [-c] [-m] [-p]
```

インストールサーバー上のすべてのサービスの一覧表示

次のコマンドは、このサーバー上のすべてのインストールサービスを表示します。この例では、4つの有効なインストールサービスが見つかっています。無効なサービスの「Status」の値は off になります。

```
$ installadm list
```

Service Name	Alias Of	Status	Arch	Image Path
--------------	----------	--------	------	------------

```

-----
default-i386      solaris11_1-i386  on      i386   /export/auto_install/solaris11_1-i386
default-sparc    solaris11_1-sparc on      sparc  /export/auto_install/solaris11_1-sparc
solaris11_1-i386 -                on      i386   /export/auto_install/solaris11_1-i386
solaris11_1-sparc -                on      sparc  /export/auto_install/solaris11_1-sparc

```

default-i386 サービスは、このサーバー上で最初の i386 サービスが作成されたときに自動的に作成されました。default-i386 サービスは、create-client サブコマンドを使用して solaris11_1-i386 サービスに関連付けられていない x86 クライアントによって使用されます。default-i386 および solaris11_1-i386 サービスは、ネットイメージを共有しますが、異なる AI マニフェストとシステム構成プロファイルを持っています。

default-sparc サービスは、このサーバー上で最初の sparc サービスが作成されたときに自動的に作成されました。default-sparc サービスは、create-client サブコマンドを使用して solaris11_1-sparc サービスに関連付けられていない SPARC クライアントによって使用されます。default-sparc および solaris11_1-sparc サービスは、ネットイメージを共有しますが、異なる AI マニフェストとシステム構成プロファイルを持っています。

指定したインストールサービスの情報の表示

次のコマンドは、-n オプションで指定されたインストールサービスの情報を表示します。

```
$ installadm list -n solaris11_1-sparc
```

```

Service Name      Alias Of Status Arch  Image Path
-----
solaris11_1-sparc -          on    sparc /export/auto_install/solaris11_1-sparc

```

インストールサービスに関連付けられたクライアントの一覧表示

次のコマンドは、このインストールサーバー上のインストールサービスに関連付けられているすべてのクライアントを一覧表示します。クライアントは、installadm create-client コマンドを使用してインストールサービスに関連付けられました。[123 ページの「インストールサービスへのクライアントの追加」](#)を参照してください。

```
$ installadm list -c
```

```

Service Name      Client Address      Arch  Image Path
-----
solaris11_1-sparc 00:14:4F:A7:65:70  sparc /export/auto_install/solaris11_1-sparc
solaris11_1-i386  08:00:27:8B:BD:71  i386  /export/auto_install/solaris11_1-i386
                  01:C2:52:E6:4B:E0  i386  /export/auto_install/solaris11_1-i386

```

特定のインストールサービスに関連付けられたクライアントの一覧表示

次のコマンドは、指定したインストールサービスに追加されたすべてのクライアントを一覧表示します。次の例では、1つのクライアントが `solaris11_1-sparc` インストールサービスに関連付けられています。

```
$ installadm list -c -n solaris11_1-sparc
```

Service Name	Client Address	Arch	Image Path
solaris11_1-sparc	00:14:4f:a7:65:70	sparc	/export/auto_install/solaris11_1-sparc

カスタマイズしたインストールの情報の表示

このセクションのコマンドは、特定のインストールサービスに関連付けられている AI マニフェストとシステム構成プロファイルを表示します。また、これらのコマンドは各マニフェストおよびプロファイルに関連付けられているクライアント条件も表示します。

すべての AI マニフェストおよびシステム構成プロファイルの一覧表示

次のコマンドは、このインストールサーバー上のすべてのインストールサービスの AI マニフェスト、派生マニフェストスクリプト、およびシステム構成プロファイルを一覧表示します。「Service/Manifest Name」および「Service/Profile Name」列には、マニフェスト、スクリプト、またはプロファイルの内部名が表示されません。「Status」列では、各サービスのデフォルトのマニフェストとアクティブでないマニフェストが特定されます。マニフェストが関連付けられた条件を持たず、デフォルトでもない場合、そのマニフェストは非アクティブです。「Criteria」列には、関連付けられているクライアント条件が表示されます。

`orig_default` マニフェストは、インストールサービスが作成されたときにインストールサービスに含まれていた元のデフォルトの AI マニフェストです。`mem1` マニフェストは、メモリー条件と `-d` オプションを指定して作成され、このサービスの新しいデフォルトのマニフェストになっています。`mem1` はデフォルトのマニフェストなので、その条件は無視されます。別のマニフェストをデフォルトのマニフェストとして作成した場合は、`mem1` マニフェストを使用するクライアントの選択に `mem1` 条件が使用されます。元のデフォルトのマニフェストが非アクティブになっているのは、それを使用するクライアントを決めるための条件が関連付けられていないからです。デフォルトのマニフェストにのみ、条件を関連付けることができません。ほかのどのマニフェストを使用する条件にも一致しないクライアントは、デフォルトのマニフェストを使用します。AI マニフェストの選択の詳細は、[第9章「インストールのカスタマイズ」](#)を参照してください。

```

$ installadm list -m -p
Service/Manifest Name  Status  Criteria
-----
default-i386
  orig_default          Default  None

default-sparc
  orig_default          Default  None

solaris11_1-i386
  ipv4                  ipv4     = 10.6.68.1 - 10.6.68.200
  mem1                  Default  (Ignored: mem    = 2048 MB - 4095 MB)
  orig_default          Inactive None

solaris11_1-sparc
  t200                  mem      = 4096 MB - unbounded
                       platform = SUNWSun-Fire-T200
  mem1                  Default  (Ignored: mem    = 2048 MB - 4095 MB)
  orig_default          Inactive None

Service/Profile Name  Criteria
-----
solaris11_1-i386
  mac2                  mac      = 08:00:27:8B:BD:71
                       hostname = server2
  mac3                  mac      = 01:C2:52:E6:4B:E0
                       hostname = server3
  ipv4                  ipv4     = 10.0.2.100 - 10.0.2.199
  mem1                  mem      = 2048 MB - 4095 MB

solaris11_1-sparc
  mac1                  mac      = 01:C2:52:E6:4B:E0
                       hostname = server1
                       ipv4     = 192.168.168.251
  t200                  platform = SUNWSun-Fire-T200
                       mem      = 4096-unbounded

```

指定したインストールサービスに関連付けられているマニフェストとプロファイルの一覧表示

次の例では、インストールサービス `solaris11_1-sparc` に関連付けられているすべての AI マニフェスト、派生マニフェストスクリプト、およびシステム構成プロファイルを表示します。

```

$ installadm list -m -p -n solaris11_1-sparc
Service/Manifest Name  Status  Criteria
-----
solaris11_1-sparc
  t200                  mem      = 4096 MB - unbounded
                       platform = SUNWSun-Fire-T200
  mem1                  Default  (Ignored: mem    = 2048 MB - 4095 MB)
  orig_default          Inactive None

Service/Profile Name  Criteria
-----
solaris11_1-sparc

```

```
mac1          mac = 01:C2:52:E6:4B:E0
              hostname = server1
              ipv4 = 192.168.168.251
t200         platform = SUNWSun-Fire-T200
              mem = 4096-unbounded
```

AISMF サービスを管理する

AI サーバーの `svc:/system/install/server:default` という SMF サービスは、AI サーバーアプリケーションとすべてのインストールサービスの全体的な状態を表すサービスです。

例 8-1 AISMF サービスの有効化

AI SMF サービスは、`installadm create-service` コマンドを実行するときに有効になります。AISMF サービスは、既存のインストールサービスに影響を及ぼすほかの `installadm` コマンドを実行するときにも有効になります。手動で AI SMF サービスを有効にするには、次のコマンドを実行します。

```
$ svcadm enable svc:/system/install/server:default
```

インストールサーバー上に現在有効になっているインストールサービスがない場合、または注意を要する問題が発生した場合、AISMF サービスは保守モードになります。

例 8-2 AISMF サービスの無効化

AISMF サービスを無効にするには、次のコマンドを実行します。

```
$ svcadm disable svc:/system/install/server:default
```

いずれかの AI インストールサービスがまだ有効になっている場合は、AISMF サービスを無効にしないでください。いずれかのインストールサービスが有効になっているかどうかを確認する方法については、[131 ページの「インストールサーバー上のすべてのサービスの一覧表示」](#)を参照してください。

インストールのカスタマイズ

インストールをカスタマイズするには、インストール手順とシステム構成手順をカスタマイズします。次に、クライアント条件を指定して、その条件で識別されるクライアントに、カスタマイズされたインストール手順と構成手順を対応付けます。

AI インストールサービスには、1つ以上のインストール手順ファイル (AI マニフェスト) と 0 個以上の構成手順ファイル (SMF システム構成プロファイル) が含まれています。AI マニフェストは、各クライアントで1つのみ使用します。各クライアントは、任意の数のシステム構成プロファイルを使用できます。クライアントシステムが構成プロファイルを使用しない場合は、そのクライアントの構成を行うために、そのクライアントのインストール後の初回ブート時に対話型ツールがそのクライアント上で開きます。

クライアントとインストール手順および構成手順との対応付け

AI を使用する場合は、まずインストールサーバーを設定します。インストールサーバーには、少なくとも1つの AI ブートイメージと、そのブートイメージに関連付けられた AI インストールサービスが含まれています。クライアントは、ネットワーク経由でブートするときに、インストールサーバーのインストールサービスを使用します。

クライアントは、そのクライアントのアーキテクチャー用のデフォルトのインストールサービスか、または割り当てられたインストールサービスを使用します。インストールサービスは、この章で説明する方法を使用して、クライアントを、使用する正しいインストール手順および構成手順と照合します。

異なるブートイメージ (SPARC イメージと x86 イメージ、または別の Oracle Solaris バージョン) を使用するインストールを定義するには、イメージごとに個別のサービスを作成します。

クライアントを特定のインストールサービスに割り当てるには、そのクライアントをインストールサービスに追加します(第14章「クライアントシステムのインストール」を参照)。クライアントのMACアドレスと、このクライアントが使用するインストールサービスの名前を指定します。このMACアドレスを持つクライアントがブートすると、そのクライアントはインストールサーバーに接続され、指定されたインストールサービスを使用します。システムのMACアドレスを見つけるには、『Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化』および `dladm(1M)` のマニュアルページに記載されているように、`dladm` コマンドを使用してください。

1つのネットイメージに対して複数のタイプのインストールを定義するには、追加のAIマニフェストを作成し、システム構成プロファイルを作成します。新しいAIマニフェストと構成プロファイルをそのネットイメージのAIインストールサービスに追加します。どのクライアントがどのAIマニフェストとどのシステム構成プロファイルを使用するのかを定義する条件を指定します。124ページの「インストールサービスへのクライアント固有のインストール手順の関連付け」を参照してください。

カスタムAIマニフェストの作成方法については、第10章「クライアントシステムのプロビジョニング」を参照してください。システム構成プロファイルの作成方法については、第11章「クライアントシステムの構成」を参照してください。

AIマニフェストの選択

各クライアントでは、AIマニフェストを1つだけ使用してそのインストールを行います。次のアルゴリズムに従って、クライアントのためのAIマニフェストが選択されます。

- このインストールサービスに対してカスタムAIマニフェストが定義されていない場合は、デフォルトのAIマニフェストが使用されます。デフォルトのAIマニフェストは、どのクライアント条件にも関連付けられていません。デフォルトのAIマニフェストの例については、144ページの「デフォルトのAIマニフェスト」を参照してください。
- このインストールサービスに対してカスタムAIマニフェストが定義されているが、クライアントがどのカスタムAIマニフェストの条件にも一致しない場合、そのクライアントはデフォルトのAIマニフェストを使用します。
- クライアントが、カスタムAIマニフェストに対して指定されている条件に一致する場合、クライアントはそのカスタムマニフェストを使用します。

クライアントの特性が複数のAIマニフェストの条件に一致する場合は、インストールのためのマニフェストを選択するために、表9-1に示されている順序でクライアントの特性が評価されます。`installadm` ツールは、同じタイプの条件が重複していないことを確認します。詳細は、124ページの「インストールサービスへのAIマニフェストの追加」を参照してください。

重複しない複数の条件が、下記の表で指定されている順序で使用されます。たとえば、ある条件仕様がクライアントの MAC アドレスに一致し、別の条件仕様が同じクライアントの IP アドレスに一致した場合、選択の優先順位は macの方が ipv4 より高いため、MAC アドレスの条件仕様に関連付けられたマニフェストが使用されます。

例 9-1 クライアントと AI マニフェストとの照合

次の例では、2つのカスタム AI マニフェストが同じインストールサービスに追加されています。これらのマニフェストに関連付けられたクライアント条件は示されているとおりです。manifest_x86.xml の AI マニフェストは、クライアントのアーキテクチャーを指定する次の条件ファイルとともにサービスに追加されました。

```
<ai_criteria_manifest>
  <ai_criteria name="arch">
    <value>i86pc</value>
  </ai_criteria>
</ai_criteria_manifest>
```

manifest_mac1.xml の AI マニフェストは、クライアントの MAC アドレスを指定する次の条件ファイルとともにサービスに追加されました。

```
<ai_criteria_manifest>
  <ai_criteria name="mac">
    <value>00:14:4f:a7:65:70</value>
  </ai_criteria>
</ai_criteria_manifest>
```

x86 クライアントがインストールされている場合、そのクライアントには manifest_x86.xml が割り当てられます。

MAC アドレス 00:14:4f:a7:65:70 を持つ SPARC クライアントがインストールされている場合、そのクライアントには manifest_mac1.xml が割り当てられます。

ほかの何らかの MAC アドレスを持つ SPARC システムがインストールされている場合、そのシステムにはデフォルトの AI マニフェストが割り当てられます。

システム構成プロファイルの選択

クライアントのシステム構成プロファイルの選択にも、AI マニフェストの選択に使用されるのと同じ条件のキーワードが使用されます。表 9-1 を参照してください。

特定のクライアントに対して、複数のシステム構成プロファイルを選択できません。選択を 1つのプロファイルに絞り込むためのアルゴリズムは必要ありません。

クライアントの特性が複数のシステム構成プロファイルの条件に一致する場合は、一致するすべての構成プロファイルがそのシステムの構成に適用されます。た

例えば、ある条件仕様がクライアントのホスト名に一致し、別の条件仕様が同じクライアントのメモリーサイズに一致する場合は、両方の構成プロファイルがそのクライアントの構成に使用されます。

選択条件

表 9-1 は、特定の AI マニフェストまたはシステム構成プロファイルを使用するクライアントを示すために使用できる条件のキーワードを示しています。例の欄には、使用可能な値がいくつか示されています。条件のキーワードと値は、`installadm` サブコマンドの `create-manifest`、`create-profile`、および `set-criteria` とともに使用できます。

`ipv4`、`mac`、`mem`、および `network` の指定は、ハイフン (-) で区切られた範囲値で表現できます。範囲の一端に制限なしを指定するには、`unbounded` を使用します。以下の `mem` の例を参照してください。

`arch`、`cpu`、`hostname`、`platform`、および `zonename` の指定は、スペースで区切られた値の引用符付きリストとして表現されます。次の `zonename` の例を参照してください。

条件のキーワードと値は、コマンド行で `-c` オプションを使って指定します。

```
-c criteria=value|list|range  
-c mac="aa:bb:cc:dd:ee:ff"  
-c mem="2048-unbounded"  
-c zonename="zone1 zone2"
```

条件は、XML ファイルの `ai_criteria` 要素でも指定できます。このファイルの内容は、条件仕様のみにするようにしてください。コマンド行で条件ファイルを指定するには、`-c` オプションを使用します。表の中に例が示されています。

表9-1 条件のキーワードと条件の階層

条件の名前	説明	コマンド行とXML ファイルの例
mac	コロン(:)区切り文字による16進のMACアドレスまたはMACアドレスの範囲。	<p>CLI、単一のMACアドレス:</p> <pre>-c mac="0:14:4F:20:53:97"</pre> <p>CLI、MACアドレスの範囲:</p> <pre>-c mac=0:14:4F:20:53:94-0:14:4F:20:53:A0</pre> <p>XML、単一のMACアドレス:</p> <pre><ai_criteria name="mac"> <value>0:14:4F:20:53:97</value> </ai_criteria></pre> <p>XML、MACアドレスの範囲:</p> <pre><ai_criteria name="mac"> <range> 0:14:4F:20:53:94 0:14:4F:20:53:A0 </range> </ai_criteria></pre>
ipv4	IPバージョン4ネットワークアドレス、またはIPアドレスの範囲	<p>CLI、単一のIPアドレス:</p> <pre>-c ipv4="10.6.68.127"</pre> <p>CLI、IPアドレスの範囲:</p> <pre>-c ipv4="10.6.68.1-10.6.68.200"</pre> <p>XML、単一のIPアドレス:</p> <pre><ai_criteria name="ipv4"> <value>10.6.68.127</value> </ai_criteria></pre> <p>XML、IPアドレスの範囲:</p> <pre><ai_criteria name="ipv4"> <range> 10.6.68.1 10.6.68.200 </range> </ai_criteria></pre>

表 9-1 条件のキーワードと条件の階層 (続き)

条件の名前	説明	コマンド行と XML ファイルの例
platform	uname -i (x86 システム) および prtconf -b (SPARC システム) によって返されるプラットフォーム名値には次が含まれます。 i86pc SUNW,SPARC-Enterprise SUNW,Sun-Fire-T200	CLI: -c platform="SUNW,Sun-Fire-T200" XML: <ai_criteria name="platform"> <value>SUNW,Sun-Fire-T200</value> </ai_criteria>
arch	uname -m によって返されるアーキテクチャー 値: i86pc、sun4u、または sun4v	CLI: -c arch="i86pc" XML: <ai_criteria name="arch"> <value>i86pc</value> </ai_criteria>
cpu	uname -p によって返される CPU クラス 値: i386 または sparc	CLI: -c cpu="sparc" XML: <ai_criteria name="cpu"> <value>sparc</value> </ai_criteria>
network	IP バージョン 4 ネットワーク番号、またはネットワーク番号の範囲	CLI、単一の IP アドレス: -c network="10.0.0.0" CLI、IP アドレスの範囲: -c network="11.0.0.0-12.0.0.0" XML、単一の IP アドレス: <ai_criteria name="network"> <value>10.0.0.0</value> </ai_criteria> XML、IP アドレスの範囲: <ai_criteria name="network"> <range> 11.0.0.0 12.0.0.0 </range> </ai_criteria>

表 9-1 条件のキーワードと条件の階層 (続き)

条件の名前	説明	コマンド行とXMLファイルの例
mem	prtconfによって返されるMバイト単位のメモリーサイズ、またはメモリーサイズの範囲 unboundedキーワードは、範囲に上限がないことを示します。	<p>CLI、1つのメモリーサイズ: -c mem="4096"</p> <p>CLI、メモリーサイズの範囲: -c mem="2048-unbounded"</p> <p>XML、1つのメモリーサイズ: <ai_criteria name="mem"> <value>4096</value> </ai_criteria></p> <p>XML、メモリーサイズの範囲: <ai_criteria name="mem"> <range> 2048 unbounded </range> </ai_criteria></p>
zonename	zoneadm listによって示されるゾーンの名前または名前のリスト。第12章「ゾーンのインストールと構成」を参照してください。	<p>CLI、単一のゾーン名: -c zonename="myzone"</p> <p>CLI、ゾーン名のリスト: -c zonename="zoneA zoneB zoneC"</p> <p>XML、単一のゾーン名: <ai_criteria name="zonename"> <value>myzone</value> </ai_criteria></p> <p>XML、ゾーン名のリスト: <ai_criteria name="zonename"> <value>zoneA zoneB zoneC</value> </ai_criteria></p>

表 9-1 条件のキーワードと条件の階層 (続き)

条件の名前	説明	コマンド行と XML ファイルの例
hostname	クライアントホスト名またはクライアントホスト名のリスト。	<p>CLI、単一のホスト名:</p> <pre>-c hostname="host3"</pre> <p>CLI、ホスト名のリスト:</p> <pre>-c hostname="host1 host2 host6"</pre> <p>XML、単一のホスト名:</p> <pre><ai_criteria name="hostname"> <value>host3</value> </ai_criteria></pre> <p>XML、ホスト名のリスト:</p> <pre><ai_criteria name="hostname"> <value>host1 host2 host6</value> </ai_criteria></pre>

デフォルトのAI マニフェスト

新しいインストールサービスを作成する

と、`install-service-image-path/auto_install/manifest/default.xml` がそのインストールサービスの初期のデフォルト AI マニフェストになります。

このデフォルトの AI マニフェストを以下の例に示します。このデフォルトのマニフェストは、インストールイメージごとに多少異なることがあります。

デフォルトのマニフェストの `target` セクションでは、作成される ZFS ファイルシステムあるいはデータセットを定義します。デフォルトのマニフェストでは、インストールのターゲットディスクを定義しません。ターゲットディスクがマニフェストに指定されていない場合にデフォルトのターゲットインストール先を決定する方法については、[ai_manifest\(4\)](#) のマニュアルページの説明を参照してください。

`destination` セクションは、インストールするロケールの指定に使用できます。インストールされるロケールを制限するために、マニフェスト内でファセット指定を使用でき、これによって、すべてのロケールが必要ではない場合に時間とスペースを節約できます。ファセットが指定されない場合、すべてのロケールのファセットがデフォルトで `true` になります。イメージのファセットおよび属性の設定の詳細は、[ai_manifest\(4\)](#) のマニュアルページを参照してください。

ソフトウェアのインストール手順によってデフォルトのIPS パッケージリポジトリが指定され、次の2つのパッケージがインストールされます。

- `entire` パッケージは必須です。この結合パッケージによって、インストールされるシステムパッケージが互換性のあるバージョンに制約されます。適切なシステム更新と正しいパッケージ選択は、この結合の存在によって変わります。このパッケージのインストールをAI マニフェストから削除しないでください。また、インストール後にこのパッケージをアンインストールしないでください。
- `solaris-large-server` パッケージは、インストールするほとんどの環境で推奨されるツールとデバイスドライバのグループパッケージです。このパッケージにより、多数のネットワークドライバとストレージドライバ、Python ライブラリ、Perlなど、多くのものがインストールされます。`solaris-large-server` グループパッケージに含まれるパッケージの完全なリストについては、『[Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新](#)』の「[グループパッケージ内のすべてのインストール可能なパッケージの一覧表示](#)」で説明されているように、`pkg contents` コマンドを使用します。

インストールする可能性がある他のパッケージの名前を検索する方法については、『[Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新](#)』を参照してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<!--

Copyright (c) 2008, 2012, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="default">
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <!--
            Subsequent <filesystem> entries instruct an installer to create
            following ZFS datasets:

                <root_pool>/export          (mounted on /export)
                <root_pool>/export/home     (mounted on /export/home)

            Those datasets are part of standard environment and should be
            always created.

            In rare cases, if there is a need to deploy an installed system
            without these datasets, either comment out or remove <filesystem>
            entries. In such scenario, it has to be also assured that
            in case of non-interactive post-install configuration, creation
            of initial user account is disabled in related system
            configuration profile. Otherwise the installed system would fail
            to boot.
          -->
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
```

```
        <be name="solaris"/>
    </zpool>
</logical>
</target>
<software type="IPS">
    <destination>
        <image>
            <!-- Specify locales to install -->
            <facet set="false">facet.locale.*</facet>
            <facet set="true">facet.locale.de</facet>
            <facet set="true">facet.locale.de_DE</facet>
            <facet set="true">facet.locale.en</facet>
            <facet set="true">facet.locale.en_US</facet>
            <facet set="true">facet.locale.es</facet>
            <facet set="true">facet.locale.es_ES</facet>
            <facet set="true">facet.locale.fr</facet>
            <facet set="true">facet.locale.fr_FR</facet>
            <facet set="true">facet.locale.it</facet>
            <facet set="true">facet.locale.it_IT</facet>
            <facet set="true">facet.locale.ja</facet>
            <facet set="true">facet.locale.ja_*</facet>
            <facet set="true">facet.locale.ko</facet>
            <facet set="true">facet.locale.ko_*</facet>
            <facet set="true">facet.locale.pt</facet>
            <facet set="true">facet.locale.pt_BR</facet>
            <facet set="true">facet.locale.zh</facet>
            <facet set="true">facet.locale.zh_CN</facet>
            <facet set="true">facet.locale.zh_TW</facet>
        </image>
    </destination>
    <source>
        <publisher name="solaris">
            <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
    </source>
    <!--
        The version specified by the "entire" package below, is
        installed from the specified IPS repository. If another build
        is required, the build number should be appended to the
        'entire' package in the following form:

            <name>pkg:/entire@0.5.11-0.build#</name>
        -->
    <software_data action="install">
        <name>pkg:/entire@0.5.11-0.175.1</name>
        <name>pkg:/group/system/solaris-large-server</name>
    </software_data>
</software>
</ai_instance>
</auto_install>
```

クライアントシステムのプロビジョニング

AI インストールサービスを作成すると、クライアントのプロビジョニング方法を指定したデフォルトの AI マニフェストが用意されます。AI マニフェストとは、オペレーティングシステムのインストール場所とインストールするソフトウェアパッケージを指定した XML ファイルです。ストライプ化、ミラー化、パーティション分割などのディスク構成を指定することもできます。AI マニフェストの XML 要素については、[ai_manifest\(4\)](#) のマニュアルページおよび AI マニフェストの例 `install_service_image_path/auto_install/manifest/ai_manifest.xml` を参照してください。

この章では、特定のクライアントのカスタム AI マニフェストを作成する方法について説明します。

- カスタムの XML AI マニフェストファイルを作成します。この方法は、カスタム プロビジョニングを必要とするシステムがほとんどない環境にもっとも適しています。インストールされるほとんどのシステムは同一または類似したハードウェアを備えており、同じようにプロビジョニングされます。
- インストール時にクライアントごとに動的に AI マニフェストを作成するスクリプトを記述します。この方法は、インストール時に検出されるクライアント特性に基づいてクライアントごとにカスタムインストールを作成する場合に使用します。

任意のインストールサービスに XML マニフェストファイルと、マニフェストファイルを生成するためのスクリプトの両方を含めることができます。どのクライアントでも、静的な、またはスクリプトによって生成された AI マニフェストを1つしか使用しません。特定のクライアントがどの AI マニフェストを使用するかは、そのマニフェストをインストールサービスに追加するときに指定する条件によって決まります。クライアントがカスタムの AI マニフェストの使用条件に一致しない場合は、デフォルトのマニフェストが使用されます。サービス内の任意の AI マニフェストをそのサービスのデフォルトになるように指定できます。

XML AI マニフェストファイルのカスタマイズ

カスタムの XML AI マニフェストファイルを作成して適用するには、次の手順を使用します。

▼ XML AI マニフェストファイルのカスタマイズする方法

1 既存の AI マニフェストをコピーします。

AI インストールサービスを作成すると、そのインストールサービスにデフォルトの AI マニフェストが用意されます。インストールサービスの作成方法については、[第 8 章「インストールサーバーの設定」](#)を参照してください。

a. 既存のマニフェストを一覧表示します。

`installadm list` コマンドを使用して、特定のインストールサービスにすでに関連付けられている AI マニフェストを表示します。

```
$ installadm list -m -n solaris11_1-i386
Service/Manifest Name  Status  Criteria
-----
solaris11_1-i386
  orig_default          Default  None
```

b. 特定のマニフェストのコピーを取得します。

`installadm export` コマンドを使用して、このデフォルトのマニフェストまたはこのサービスに追加されているその他の AI マニフェストの内容を抽出します。

```
$ pfexec installadm export -n solaris11_1-i386 -m orig_default -o mem1.xml
これで orig_default のコピーがファイル mem1.xml に入ります。
```

2 マニフェストのコピーを変更します。

`mem1.xml` を変更し、[ai_manifest\(4\)](#) のマニュアルページの情報に従ってタグと値を追加します。

3 新しいマニフェストをインストールサービスに追加します。

新しい AI マニフェストを適切な AI インストールサービスに追加し、どのクライアントがこれらのインストール手順を使用するようにするかを定義する条件を指定します。

```
$ pfexec installadm create-manifest -n solaris11_1-i386 -f ./mem1.xml -m mem1 \
-c mem="2048-unbounded"
```

複数の `-c` オプションまたは1つの `-c` ファイルを指定できます。クライアント条件の指定方法については、第9章「インストールのカスタマイズ」および `set-criteria` サブコマンドを参照してください。

```
$ installadm list -m -n solaris11_1-i386
Service/Manifest Name  Status  Criteria
-----
solaris11_1-i386
  orig_default          Default  None
  mem1                  mem     = 2048 MB - unbounded
```

- 新しいマニフェストをデフォルトにします。

任意のマニフェストファイルまたは派生マニフェストスクリプトをサービスのデフォルトのマニフェストまたはスクリプトになるように指定できます。サービスにすでに追加したマニフェストおよびスクリプトの中でデフォルトを変更するには、`set-service` サブコマンドで `-o` オプションを使用します。

```
$ pfexec installadm set-service -o default-manifest=mem1 solaris11_1-i386
$ installadm list -m -n solaris11_1-i386
Service/Manifest Name  Status  Criteria
-----
solaris11_1-i386
  orig_default          Inactive None
  mem1                  Default  (Ignored: mem = 2048 MB - unbounded)
```

この例で元のデフォルトは、どのクライアントがそれを使用するかを指定した条件がないため、非アクティブになっています。クライアントの選択条件がなくてもアクティブな状態でいられるのは、デフォルトのマニフェストまたはスクリプトのみです。

- 新しいマニフェストをデフォルトとして追加します。

このサービス用に新しいデフォルトのマニフェストまたはスクリプトを追加する場合は、`create-manifest` で `-d` オプションを使用します。指定した条件は保存されますが、別のマニフェストがデフォルトにされるまで無視されます。

```
$ pfexec installadm create-manifest -n solaris11_1-i386 -d \
-f ./region1.xml -m region1
$ installadm list -m -n solaris11_1-i386
Service/Manifest Name  Status  Criteria
-----
solaris11_1-i386
  orig_default          Inactive None
  mem1                  mem     = 2048 MB - unbounded
  region1               Default  None
```

- 既存のマニフェストをカスタマイズします。

新しいマニフェストまたはスクリプトを追加せずに、既存のマニフェストまたはスクリプトの内容を変更するには、`installadm update-manifest` コマンドを使用します。更新の結果として、条件、デフォルトのステータス、およびマニフェスト名またはスクリプト名は変更されません。

```
$ pfexec installadm update-manifest -n solaris11_1-i386
-f ./newregion1.xml -m region1
```

4 カスタマイズしたマニフェストを検証します。

create-manifest および update-manifest サブコマンドは、XML マニフェストファイルを構文的に検証してから、それらをインストールサービスに追加します。AI では、クライアントのインストール時に AI マニフェストが意味的に検証されます。

注-クライアントに無効なマニフェストが提供された場合は、自動インストールが異常終了します。検証エラーの原因を調査するには、そのクライアント上の /system/volatile/install_log を参照してください。

installadm の list、export、create-manifest、set-criteria、update-manifest、および set-service サブコマンドの詳細は、117 ページの「インストールサーバーの維持」も参照してください。

クライアントインストール時のAIマニフェストの作成

クライアントのインストールより前にカスタムの AI マニフェストを作成する別の方法は、クライアントのインストール時にクライアントごとに動的に AI マニフェストを作成するスクリプトを記述することです。このスクリプトでは、クライアントごとに環境変数やほかのクライアント構成情報のクエリー検索を行なってカスタム AI マニフェストを作成できます。このマニフェストはインストール時に検出された各クライアントの属性に基づいているため、「派生マニフェスト」と呼ばれます。

派生マニフェストは、ほぼ同じようにインストールできるシステムが多数あるためにこれらのシステムの AI マニフェストの差異が比較的小さい場合に特に便利です。このグループのシステムに共通のインストールパラメータを指定した AI マニフェストを作成します。この共通のマニフェストをベースとして使用し、各クライアントをインストールするときに、クライアントごとに異なるパラメータを共通のマニフェストに追加する派生マニフェストスクリプトを作成します。たとえば、派生マニフェストスクリプトでは、クライアントのインストール時に各クライアントシステムに接続されているディスクの数と大きさを検出し、クライアントごとにカスタムのディスクレイアウトを指定するように AI マニフェストを変更できます。

▼ 派生マニフェストスクリプトを作成して適用する方法

1 変更するマニフェストを選択します。

変更するベースマニフェストとして使用する既存の AI マニフェストを特定します。

スクリプトを作成してテストする場合は、ローカルコピーを使って作業できます。インストール時に、この派生マニフェストスクリプトを使用する各クライアントがベースマニフェストにアクセスできる必要があります。

- 2 マニフェストを変更するスクリプトを記述します。
インストールしているクライアントの属性に基づいてインストール時にベースマニフェストを動的に変更するスクリプトを作成します。
- 3 スクリプトをインストールサービスに追加します。

派生マニフェストスクリプトを適切なAIインストールサービスに追加し、インストール時にどのクライアントがこのスクリプトを使用してインストール手順を作成するかを定義した条件を指定します。クライアントの選択条件を指定しない場合は、このスクリプトをサービスのデフォルトのAIマニフェストとして追加できます。

AIでは、クライアントのインストール時にこのスクリプトを実行してAIマニフェストのインスタンスを生成します。AIでは、結果となるマニフェストを構文的に検証します。

注- マニフェストが作成されないか、または派生マニフェストが有効でない場合、クライアントのインストールは異常終了します。検証エラーの原因を調査するには、そのクライアント上の `/system/volatile/install_log` を参照してください。

クライアントのインストールが成功した場合は、派生マニフェストがクライアント上の `/var/log/install/derived/manifest.xml` にコピーされ、マニフェストの派生に使われたスクリプトが `/var/log/install/derived/manifest_script` にコピーされます。

派生マニフェストスクリプトの作成

一般に、派生マニフェストスクリプトでは、クライアントから情報を取得し、その情報を使ってベースAIマニフェストを変更して、このクライアント専用のカスタムAIマニフェストを作成します。派生マニフェストスクリプトでは、複数の部分的なAIマニフェストを組み合わせることもできます。最終的な派生マニフェストは、完全で、検証に合格する必要があります。

派生マニフェストスクリプトは、イメージでサポートされているどの種類のスクリプトにもできます。たとえば、デフォルトでは `ksh93` と `python` がイメージに含まれています。別の種類のスクリプトを使用する場合は、必要なサポートがイメージに含まれていることを確認します。

クライアント属性の取得

派生マニフェストスクリプトは、システムの属性を読み取るコマンドを実行できます。AIでは、そのスクリプトを役割 `aiuser` として実行します。役割 `aiuser` には、非特権ユーザーが持つすべての特権と次の追加の特権があります。

```
solaris.network.autoconf.read
solaris.smf.read.*
```

役割 `aiuser` は、ほかの非特権ユーザーよりも多くの情報をシステムから読み取れることを除いては非特権です。役割 `aiuser` はシステムを変更できません。

役割、プロファイル、および特権については、『Oracle Solaris 11.1 の管理: セキュリティーサービス』のパート III 「役割、権利プロファイル、特権」を参照してください。

システムの属性を読み取るコマンドを使用するほかに、次の表に示される環境変数を通じてクライアントの属性を使用できます。

表 10-1 クライアント属性の環境変数

環境変数名	説明
<code>SI_ARCH</code>	インストールされるクライアントのアーキテクチャー。 <code>uname -p</code> の出力と同等です。
<code>SI_CPU</code>	インストールされるクライアントの ISA またはプロセッサタイプ。 <code>uname -p</code> の出力と同等です。
<code>SI_NUMDISKS</code>	クライアントのディスク数。
<code>SI_DISKNAME_#</code>	クライアントで見つかったディスクの名前を表す平坦な変数セット。 <code>SI_NUMDISKS</code> 個の <code>SI_DISKNAME_#</code> 変数が存在することになります。ここで、 <code>#</code> は 1 から <code>SI_NUMDISKS</code> までの整数に置き換えられます。この変数セットは、 <code>SI_DISKSIZE_#</code> で記述される変数セットと相関関係にあります。
<code>SI_DISKSIZE_#</code>	クライアントで見つかったディスクのディスクサイズを表す平坦な変数セット。 <code>SI_NUMDISKS</code> 個の <code>SI_DISKSIZE_#</code> 変数が存在することになります。ここで、 <code>#</code> は 1 から <code>SI_NUMDISKS</code> までの整数に置き換えられます。この変数セットは、 <code>SI_DISKNAME_#</code> で記述される変数セットと相関関係にあります。サイズは M バイトを表す整数です。
<code>SI_HOSTADDRESS</code>	インストール環境で設定されたクライアントの IP アドレス。
<code>SI_HOSTNAME</code>	インストール環境で設定されたクライアントのホスト名。
<code>SI_KARCH</code>	クライアントのカーネルアーキテクチャー。 <code>uname -m</code> の出力と同等です。
<code>SI_INSTALL_SERVICE</code>	マニフェストスクリプトの取得に使用されるインストールサービスの名前。この環境変数には、メディアからのブート用ではなく、ネットワーク経由のブート専用の値が含まれます。
<code>SI_MANIFEST_SCRIPT</code>	マニフェストスクリプトの URL。
<code>SI_MEMSIZE</code>	クライアントの物理メモリーの量。サイズは M バイトを表す整数です。
<code>SI_NATISA</code>	クライアントのネイティブの命令セットアーキテクチャー。 <code>isainfo -n</code> の出力と同等です。

表 10-1 クライアント属性の環境変数 (続き)

環境変数名	説明
SI_NETWORK	クライアントのネットワーク番号。ネットワーク番号は(IP_ADDR & netmask) です。
SI_PLATFORM (または SI_MODEL)	クライアントのプラットフォーム。uname -i (x86 システム) および prtconf -b (SPARC システム) の出力と同等です。

AI マニフェストのカスタマイズ

AI マニフェストの XML 要素を追加または変更するには、`/usr/bin/aimanifest` コマンドを使用します。

`aimanifest` によって変更するファイルには、少なくとも次の内容が含まれている必要があります。

- 作成している XML マニフェストで有効な DTD への `!DOCTYPE` 参照。
- この DTD のルート要素。

次の例は、この派生マニフェストスクリプトが追加されるインストールサービスの AI DTD ファイルの指定など、AI マニフェストのために最低限必要なベースマニフェストファイルを示しています。

```
<!DOCTYPE auto_install SYSTEM "file:///imagepath/auto_install/ai.dtd.#">
<auto_install/>
```

`#` は 1 などの整数です。`imagepath` 引数の値は、次のコマンドによって返されるパスです (`svcname` はこの派生マニフェストスクリプトが追加されるインストールサービスの名前です)。

```
$ installadm list -n svcname
```

`aimanifest` コマンドの `load` サブコマンドを使用してベースマニフェストを読み込んでから、派生マニフェストスクリプト内のほかの `aimanifest` 呼び出しを行なってください。クライアントのインストール時には、読み込んだどのファイルにもクライアントがアクセスできる必要があります。たとえば、ターゲットインストールサービスの `imagepath/auto_install/manifest/` からマニフェストを読み込むことができます。

この章の例では、ファイル `/usr/share/auto_install/manifest/default.xml` が読み込まれます。`/usr/share/auto_install/manifest/` 内のマニフェスト例は、ターゲットインストールサービスのマニフェストとは異なっている可能性があります。本稼働では、`/usr/share/auto_install/manifest/` からマニフェストを読み込まないようにしてください。

`load` サブコマンドは、部分的なマニフェストの読み込みまたは挿入にも使用できません。

新しい要素を追加するには、`add` サブコマンドを使用します。要素の属性を追加したり、要素または属性の値を変更したりするには、`set` サブコマンドを使用します。詳細は、[aimanifest\(1M\)](#) のマニュアルページを参照してください。マニュアルページと以下のスクリプト例には、`aimanifest` コマンドの使用例が含まれています。

注-`aimanifest` コマンドで指定した値に次のいずれかの文字が含まれている場合は、これらの文字がXMLのパス名の一部として解釈されないように、その値を一重引用符または二重引用符で囲む必要があります。

```
/'"@[]=
```

場合によっては、使用するシェルの規則に従って引用符の前にバックスラッシュ文字 (\) を付けてエスケープし、シェルが引用符を削除または解釈しないようにする必要があります。

次の例では、パッケージ名 `pkg:/entire` が含まれる `software_data` 要素のアクションが返されます。この例では、スラッシュ文字が特殊文字であるため、`pkg:/entire` の前後に引用符を付ける必要があります。このコマンドが `ksh93` スクリプトなどのシェルスクリプト内で呼び出される場合は、バックスラッシュ文字でこれらの引用符をエスケープする必要があります。

```
/usr/bin/aimanifest get software_data[name=\"pkg:/entire\"]@action
```

ヒント-ベストプラクティスに基づき、エラー発生時に停止するトラップを設定してください。

次のスクリプトの一部は、派生マニフェストスクリプトの良い見本です。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load baseAImanifest.xml

# Customize AI manifest. For example:
/usr/bin/aimanifest load -i manifest_fragment.xml
/usr/bin/aimanifest set origin@name file:///net/myserver/myrepo/repo.redist

exit $SCRIPT_SUCCESS
```

派生マニフェストスクリプトの例

このセクションでは、クライアントの属性を判別し、その情報を使ってAIマニフェストをカスタマイズする派生マニフェストスクリプトを作成する方法を示します。これらの例には、有効なAIマニフェストの作成に必要な情報が必ずしもすべて含まれているわけではありません。

これらの例を試すには、次の設定手順を実行します。

1. AIM_MANIFEST 環境変数を、スクリプトによってAIマニフェストが作成される場所に設定します。

\$AIM_MANIFEST ファイルは、そのファイルを変更する `aimanifest` コマンドごとに書き換えられます。`aimanifest` を `load`、`add`、または `set` サブコマンドとともに呼び出すたびに、AIM_MANIFEST ファイルが開かれ、変更され、保存されます。AIM_MANIFEST が設定されていない場合、`aimanifest` コマンドは失敗します。

2. AIM_LOGFILE 環境変数を、スクリプトによって詳細な情報とエラーメッセージが書き込まれる場所に設定します。

`aimanifest` コマンドは、サブコマンドの名前、引数の値、および各 `aimanifest` 呼び出しの戻りステータスを画面および \$AIM_MANIFEST_LOG ファイル (設定されている場合) に記録します。

3. スクリプトを実行するシステムで `aimanifest` コマンドが使用できることを確認します。`aimanifest` コマンドが使用できない場合は、`auto-install-common` パッケージをインストールします。
4. 環境変数を設定します。これらの例では、環境変数を使用してクライアントに関する情報を取得する方法を示しています。これらの例を試すには、これらの環境変数の値を設定する必要があります。

AI を使用してシステムをインストールすると、表 10-1 に示す環境変数に値が設定され、使用する派生マニフェストスクリプトで使用できるようになります。

例 10-1 ディスクサイズに基づいたディスクパーティション分割の指定

この例では、ターゲットディスクのサイズが 1T バイトを超える場合は Oracle Solaris `fdisk` パーティションにディスクの半分だけを使用するように AI マニフェストをカスタマイズします。このスクリプトの異なる実行に対して、`SI_DISKSIZE_1` を 1T バイト未満に設定してから、1T バイトを超えるサイズに設定してみます。また、スクリプトを実行する前に `SI_NUMDISKS` と `SI_DISKNAME_1` も設定します。指定されたパーティション分割は x86 クライアントのみに適用されるため、このスクリプトは x86 クライアント専用です。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
```

例10-1 ディスクサイズに基づいたディスクパーティション分割の指定 (続き)

```

    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# Check that there is only one disk on the system.
if [[ $SI_NUMDISKS -gt "1" ]]; then
    print -u2 "System has too many disks for this script."
    exit $SCRIPT_FAILURE
fi

/usr/bin/aimanifest add \
    /auto_install/ai_instance/target/disk/disk_name@name $SI_DISKNAME_1

if [[ $SI_DISKSIZE_1 -gt "1048576" ]]; then
    typeset -i PARTN_SIZE=$SI_DISKSIZE_1/2

    # Default action is to create.
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk[disk_name@name=\"$SI_DISKNAME_1\"]/partition@name 1
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk/partition[@name=1]/size@val \
        ${PARTN_SIZE}mb
else
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk[disk_name@name=\"$SI_DISKNAME_1\"]/partition@action \
        use_existing_solaris2
fi
exit $SCRIPT_SUCCESS

```

SI_DISKSIZE_1 の値が 1048576 以下であるクライアントでは、次の要素が \$AIM_MANIFEST に追加されます。

```

<target>
  <disk>
    <disk_name name="/dev/dsk/c0t0d0s0"/>
    <partition action="use_existing_solaris2"/>
  </disk>
  <!-- <logical> section -->
</target>

```

SI_DISKSIZE_1 が 1048576 より大きいクライアントでは、SI_DISKSIZE_1 の値に応じて、次のような要素が \$AIM_MANIFEST に追加されます。

```

<target>
  <disk>
    <disk_name name="/dev/dsk/c0t0d0s0"/>
    <partition name="1">
      <size val="524288mb"/>
    </partition>
  </disk>
  <!-- <logical> section -->
</target>

```

例10-1 ディスクサイズに基づいたディスクパーティション分割の指定 (続き)

パーティションを追加するためのコマンドには、そのパーティションに対して別個のディスク指定が作成されないように、`disk_name`が指定されています。この例のスクリプトは、パーティションが(ほかのディスクではなく)`$SI_DISKNAME_1`ディスク上にあることを指定しています。この例の該当する行が次の行に置き換わると、意図した結果が得られません。

```

/usr/bin/aimanifest add \
  /auto_install/ai_instance/target/disk/partition@name 1
/usr/bin/aimanifest add \
  /auto_install/ai_instance/target/disk/partition[@name=1]/size@val \
  ${PARTN_SIZE}mb
else
/usr/bin/aimanifest add \
  /auto_install/ai_instance/target/disk/partition@action \
  use_existing_solaris2

```

このスクリプトは、上記の出力ではなく、次の不正な出力を提供します。

```

<target>
  <disk>
    <disk_name name="c0t0d0s0"/>
  </disk>
  <disk>
    <partition name="1">
      <size val="524288mb"/>
    </partition>
  </disk>
</target>

```

例10-2 追加ディスクの存在に基づいたルートプールレイアウトの指定

この例では、2番目のディスクが存在する場合はルートプールのミラーを構成し、3番目のディスクが存在する場合は3方向のミラーを構成するようにAI マニフェストをカスタマイズします。スクリプトを実行する前に、`SI_NUMDISKS`と`SI_DISKNAME_1`を設定します。`SI_NUMDISKS`に設定した値に応じて、`SI_DISKNAME_2`、`SI_DISKNAME_3`などを必要に応じて設定します。AIのインストール中に、これらの環境変数が設定され、派生マニフェストスクリプトで使用できるようになります。

この例では、`aimanifest`の戻りパス(`-r`オプション)の使用方法を示しています。戻りパスの詳細は、`aimanifest(1M)`のマニュアルページを参照してください。

```

#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
  exit $SCRIPT_FAILURE
}

```

例 10-2 追加ディスクの存在に基づいたルートプールレイアウトの指定 (続き)

```

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# Use the default if there is only one disk.
if [[ $SI_NUMDISKS -ge 2 ]]; then
    typeset -i disk_num

    # Turn on mirroring. Assumes a root zpool is already set up.
    vdev=$(/usr/bin/aimanifest add -r \
        target/logical/zpool[@name=rpool]/vdev[@name mirror_vdev]
    /usr/bin/aimanifest set ${vdev}@redundancy mirror

    for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
        eval curr_disk="$SI_DISKNAME_${disk_num}"
        disk=$(/usr/bin/aimanifest add -r target/disk[@in_vdev mirror_vdev]
        /usr/bin/aimanifest set ${disk}@in_zpool rpool
        /usr/bin/aimanifest set ${disk}@whole_disk true
        disk_name=$(/usr/bin/aimanifest add -r \
            ${disk}/disk_name[@name $curr_disk]
        /usr/bin/aimanifest set ${disk_name}@name_type ctd
    done
fi
exit $SCRIPT_SUCCESS

```

c0t0d0 および c0t1d0 という名前の 2 台のディスクを含むシステムの場合、この例の出力は次の XML 要素になります。

```

<target>
  <disk in_vdev="mirror_vdev" in_zpool="rpool" whole_disk="true">
    <disk_name name="c0t0d0" name_type="ctd"/>
  </disk>
  <disk in_vdev="mirror_vdev" in_zpool="rpool" whole_disk="true">
    <disk_name name="c0t1d0" name_type="ctd"/>
  </disk>
  <logical>
    <zpool name="rpool" is_root="true">
      <vdev name="mirror_vdev" redundancy="mirror"/>
      <filesystem name="export" mountpoint="/export"/>
      <filesystem name="export/home"/>
      <be name="solaris"/>
    </zpool>
  </logical>
</target>

```

例 10-3 特定のサイズのディスクが2つ以上ある場合のミラー化構成の指定

この例では、システムに 200G バイトのディスクが 2 つ以上ある場合にミラー化構成を指定するように AI マニフェストをカスタマイズします。少なくとも 200G バイトであることがわかった最初の 2 つのディスクを使用します。スクリプトを実行する前に、テスト環境で SI_NUMDISKS、SI_DISKNAME_1、および SI_DISKSIZE_1 を設定します。また、SI_NUMDISKS に設定した値に応じて、SI_DISKNAME_2、SI_DISKSIZE_2 など

例 10-3 特定のサイズのディスクが2つ以上ある場合のミラー化構成の指定 (続き)

を必要に応じて設定します。AIのインストール中に、これらの環境変数が設定され、派生マニフェストスクリプトで使用できるようになります。

この例では、同じパスを持つノードが複数存在している場合にノードを変更する方法を示しています。シェルの実装では、`aimanifest` の戻りパス (`-r`) オプションを使用して特定のノードへのパスを返し、そのパスを使用して同じノードに追加の変更を行います。Python の実装では、サブパス (ノードパスの内側で `[]` を使用) を使って同じノードに追加の変更を行う方法を示しています。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

# Find the disks first.
typeset found_1
typeset found_2
typeset -i disk_num

for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
    eval curr_disk="$SI_DISKNAME_${disk_num}"
    eval curr_disk_size="$SI_DISKSIZE_${disk_num}"
    if [[ $curr_disk_size -ge "204800" ]] ; then
        if [ -z $found_1 ] ; then
            found_1=$curr_disk
        else
            found_2=$curr_disk
            break
        fi
    fi
done

# Now, install them into the manifest.
# Let the installer take the default action if two large disks are not found.

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

if [[ -n $found_2 ]] ; then
    # Turn on mirroring.
    vdev=$(/usr/bin/aimanifest add -r \
        /auto_install/ai_instance/target/logical/zpool/vdev@redundancy mirror)
    /usr/bin/aimanifest set ${vdev}@name mirror_vdev

    disk=$(/usr/bin/aimanifest add -r \
        /auto_install/ai_instance/target/disk@in_vdev mirror_vdev)
    disk_name=$(/usr/bin/aimanifest add -r ${disk}/disk_name@name $found_1)
```

例 10-3 特定のサイズのディスクが2つ以上ある場合のミラー化構成の指定 (続き)

```
/usr/bin/aimanifest set ${disk_name}@name_type ctd

disk=$(/usr/bin/aimanifest add -r \
    /auto_install/ai_instance/target/disk@in_vdev mirror_vdev)
disk_name=$(/usr/bin/aimanifest add -r ${disk}/disk_name@name $found_2)
/usr/bin/aimanifest set ${disk_name}@name_type ctd
fi

exit $SCRIPT_SUCCESS
```

次のスクリプトは、前の Korn shell 版を Python 版にしたものです。

```
#!/usr/bin/python2.6

import os
import sys

from subprocess import check_call, CalledProcessError

SCRIPT_SUCCESS = 0
SCRIPT_FAILURE = 1

def main():

    # Find the disks first.
    found_1 = ""
    found_2 = ""

    si_numdisks = int(os.environ["SI_NUMDISKS"])
    for disk_num in range(1, si_numdisks + 1):
        curr_disk_var = "SI_DISKNAME_" + str(disk_num)
        curr_disk = os.environ[curr_disk_var]
        curr_disk_size_var = "SI_DISKSIZE_" + str(disk_num)
        curr_disk_size = os.environ[curr_disk_size_var]
        if curr_disk_size >= "204800":
            if not len(found_1):
                found_1 = curr_disk
            else:
                found_2 = curr_disk
            break

    # Now, write the disk specifications into the manifest.
    # Let the installer take the default action if two large disks are not found.

    try:
        check_call(["/usr/bin/aimanifest", "load",
            "/usr/share/auto_install/manifest/default.xml"])
    except CalledProcessError as err:
        sys.exit(err.returncode)

    if len(found_2):
        try:
            check_call(["/usr/bin/aimanifest", "add",
                "target/logical/zpool[@name=rpool]/vdev@redundancy", "mirror"])
            check_call(["/usr/bin/aimanifest", "set",
```


例 10-3 特定のサイズのディスクが2つ以上ある場合のミラー化構成の指定 (続き)

```

"target/logical/zpool/vdev[@redundancy='mirror']@name", "mirror_vdev"])

check_call(["usr/bin/aimanifest", "add",
           "target/disk/disk_name@name", found_1])
check_call(["usr/bin/aimanifest", "set",
           "target/disk/disk_name[@name='" + found_1 + "'" + "@name_type", "ctd"])
check_call(["usr/bin/aimanifest", "set",
           "target/disk[disk_name@name='" + found_1 + "'" + "@in_vdev", "mirror_vdev"])

check_call(["usr/bin/aimanifest", "add",
           "target/disk/disk_name@name", found_2])
check_call(["usr/bin/aimanifest", "set",
           "target/disk/disk_name[@name='" + found_2 + "'" + "@name_type", "ctd"])
check_call(["usr/bin/aimanifest", "set",
           "target/disk[disk_name@name='" + found_2 + "'" + "@in_vdev", "mirror_vdev"])
except CalledProcessError as err:
    sys.exit(err.returncode)

sys.exit(SCRIPT_SUCCESS)

if __name__ == "__main__":
    main()

```

例 10-4 IPアドレスに基づくインストールするパッケージの指定

この例では、クライアントのIPアドレスが指定の範囲にある場合はあるパッケージをインストールし、クライアントのIPアドレスが別の範囲にある場合は別のパッケージをインストールするようにAIマニフェストをカスタマイズします。スクリプトを実行する前に、テスト環境でSI_HOSTADDRESSを設定します。AIのインストール中に、この環境変数が設定され、派生マニフェストスクリプトで使用できるようになります。

```

#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# First determine which range the host IP address of the client is in.
echo $SI_HOSTADDRESS | sed 's/\./ /g' | read a b c d

# Assume all systems are on the same class A and B subnets.

# If the system is on class C subnet = 100, then install the /pkg100 package.
# If the system is on class C subnet = 101, then install the /pkg101 package.

```

例10-4 IPアドレスに基づくインストールするパッケージの指定 (続き)

```
# Otherwise, do not install any other additional package.

if ((c == 100)) ; then
    /usr/bin/aimanifest add \
        software/software_data[@action='install']/name pkg:/pkg100
fi
if ((c == 101)) ; then
    /usr/bin/aimanifest add \
        software/software_data[@action='install']/name pkg:/pkg101
fi

exit $SCRIPT_SUCCESS
```

例10-5 特定のサイズ以上のディスクのみをターゲットディスクにする指定

この例では、50Gバイト以上のディスクにのみインストールするようにAIマニフェストをカスタマイズします。それよりも小さいディスクは無視します。スクリプトを実行する前に、テスト環境でSI_NUMDISKS、SI_DISKNAME_1、およびSI_DISKSIZE_1を設定します。また、SI_NUMDISKSに設定した値に応じて、SI_DISKNAME_2、SI_DISKSIZE_2などを必要に応じて設定します。AIのインストール中に、これらの環境変数が設定され、派生マニフェストスクリプトで使用できるようになります。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

typeset found
typeset -i disk_num
for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
    eval curr_disk="$SI_DISKNAME_${disk_num}"
    eval curr_disk_size="$SI_DISKSIZE_${disk_num}"
    if [[ $curr_disk_size -ge "512000" ]] ; then
        found=$curr_disk
        /usr/bin/aimanifest add \
            /auto_install/ai_instance/target/disk/disk_name@name $found
        break
    fi
done

if [[ -z $found ]] ; then
    exit $SCRIPT_FAILURE
fi
```

例10-5 特定のサイズ以上のディスクのみをターゲットディスクにする指定 (続き)

```
exit $SCRIPT_SUCCESS
```

例10-6 マニフェスト仕様が正しくないスクリプト

この例のスクリプトにはエラーが含まれています。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

/usr/bin/aimanifest set \
    software[@type="IPS"]/software_data/name pkg:/driver/pcmcia
/usr/bin/aimanifest set \
    software/software_data[@name=pkg:/driver/pcmcia]@action uninstall

return $SCRIPT_SUCCESS
```

この例には、\$AIM_MANIFEST への書き込みに関する問題が3つあります。

1. `aimanifest` の `set` サブコマンドでは、既存の要素または属性の値を変更したり、新しい属性を作成したりできます。 `set` サブコマンドで新しい要素を作成することはできません。最初の `set` サブコマンドでは、新しいパッケージ名を作成するのではなく、マニフェスト内の既存のパッケージ名を変更しようとし、マニフェスト内に複数のパッケージ名が存在する場合は、変更するパッケージを特定できないため、あいまいエラーが発生します。この例の最初の `set` サブコマンドは `add` サブコマンドにするべきでした。
2. この例の2番目の `set` サブコマンドでは、値 `pkg:/driver/pcmcia` を持つ要素 `name` が `@` 記号を前に付けて指定されています。属性値は `@` 記号を前に付けて指定しますが、要素値はそのようにしません。
3. 値 `pkg:/driver/pcmcia` は引用符で囲むようにしてください。スラッシュなどの特殊文字が含まれる値には引用符を付ける必要があります。

この例の2つの `set` 行は、次の行で置き換えるようにしてください。

```
/usr/bin/aimanifest add \
    software[@type="IPS"]/software_data@action uninstall
/usr/bin/aimanifest add \
    software/software_data[@action=uninstall]/name pkg:/driver/pcmcia
```

例 10-6 マニフェスト仕様が正しくないスクリプト (続き)

これらの2つの `add` サブコマンドによって、作成中のマニフェストの `software` セクションの最後に次の行が追加されます。

```
<software_data action="uninstall">
  <name>pkg:/driver/pcmcia</name>
</software_data>
```

派生マニフェストスクリプトのテスト

派生マニフェストスクリプトをテストするには、AIインストール環境に似ている環境でスクリプトを実行します。

1. 変更するスクリプト用のベース AI マニフェストを設定します。

- a. スクリプト内の最初の `aimanifest` コマンドが `aimanifest load` コマンドであることを確認します。読み込まれているファイルに、ターゲットインストールサービスの AI マニフェストの検証に使用する適切な DTD を指定した `<!DOCTYPE>` 定義が含まれていることを確認します。次の例は、この派生マニフェストスクリプトが追加されるインストールサービスの AI DTD ファイルの指定など、AI マニフェストのために最低限必要なベースマニフェストファイルを示しています。

```
<!DOCTYPE auto_install SYSTEM "file:///imagepath/auto_install/ai.dtd.#">
<auto_install/>
```

`#` は 1 などの整数です。 `imagepath` 引数の値は、次のコマンドによって返されるパスです (`svcname` はこの派生マニフェストスクリプトが追加されるインストールサービスの名前です)。

```
$ installadm list -n svcname
```

- b. `AIM_MANIFEST` を、スクリプトによって AI マニフェストが作成される場所に設定します。この場所は、非特権ユーザー `aiuser` によって書き込み可能である必要があります。

注-AI がインストールを行なっているときは、`AIM_MANIFEST` を設定する必要はありません。AI によってデフォルト値が設定されます。

2. `AIM_LOGFILE` を、スクリプトによって詳細な情報とエラーメッセージが書き込まれる場所に設定します。この場所は、非特権ユーザー `aiuser` によって書き込み可能である必要があります。

注-AIがインストールを行なっているときは、AIM_LOGFILEを設定する必要はありません。このログ情報は、さらに大きなインストールログ /system/volatile/install_log の一部です。

3. スクリプトをテストするシステムで `aimanifest` コマンドが使用できることを確認します。`aimanifest` コマンドが使用できない場合は、`auto-install-common` パッケージをインストールします。
4. `root` の役割になれることを確認します。`root` の役割から、パスワードを指定しないで役割 `aiuser` になることができます。

```
$ su
Password:
# su aiuser -c ./script
#
```

AIは、役割 `aiuser` として派生マニフェストスクリプトを実行します。AIインストール環境に近づけるために、役割 `aiuser` でスクリプトを実行します。役割 `aiuser` に含まれるのと異なる特権を持つユーザーとしてスクリプトを実行した場合、スクリプト内のいくつかの操作で結果が異なる可能性があります。

5. テスト環境の環境変数を、この派生マニフェストスクリプトを使ってインストールされるクライアントシステムを表す値で設定します。サンプルファイル `/usr/share/auto_install/derived_manifest_test_env.sh` をテンプレートとして使用できます。必要に応じて値を変更します。

AIがインストールを行なっているときに、表 10-1 に示す環境変数に値が設定され、使用する派生マニフェストスクリプトで使用できるようになります。

意図されたクライアントシステムが、派生マニフェストスクリプトをテストするインストールサーバーやその他のシステムと大きく異なる場合があります。スクリプト内で呼び出すコマンドが使用できない場合や、動作の異なる別のバージョンである場合があります。システムのアーキテクチャーが異なる場合や、ディスクの数とサイズが異なる場合があります。説明したようにテスト環境で環境変数を設定することによって、これらの差異のいくつかに対処できます。

▼ 派生マニフェストスクリプトをテストする方法

この手順では、意図されたクライアントシステムのいずれかで派生マニフェストスクリプトをテストする方法について説明します。

- 1 そのクライアントシステムでAIイメージをブートします。
そのクライアントシステムで、AIイメージを「Text Installer and command line」モードでブートします。
- 2 インストーラの初期メニューから「シェル」を選択します。

- 3 AI インストールサーバーからスクリプトをコピーします。
AIインストールサーバーからスクリプトをコピーするには、`wget` または `sftp` を使用します。
- 4 スクリプトをデバッグします。
次のいずれかの方法を使用して、スクリプトをデバッグします。
 - スクリプトを手動で実行します。
 - AI をテストモードで実行します。
次のコマンドを実行して、AI をテストモードで実行します。

```
$ auto-install -m script -i
```

AI ログファイル `/system/volatile/install_log` を調べます。ログファイルに、スクリプトが有効であることを示す次の行が含まれているはずですが、

```
Derived Manifest Module: XML validation completed successfully
```
- 5 スクリプトをコピーしてインストールサーバーに戻します。
変更が行われた場合は、スクリプトをコピーしてインストールサーバーに戻します。

インストールサービスへの派生マニフェストスクリプトの追加

XML マニフェストをインストールサービスに追加するのと同じ方法で、スクリプトを AI インストールサービスに追加します。同じオプションを使用して、インストールの際にこのスクリプトを使ってマニフェストを作成するクライアントの選択条件を指定します。XML マニフェストを更新するのとまったく同じようにスクリプトを更新できます。スクリプトをサービスのデフォルトのマニフェストになるように設定できます。1つのサービスに関連付けられたマニフェストを一覧表示するときに、スクリプトが表示されます。XML マニフェストをエクスポートするのとまったく同じようにスクリプトの内容をエクスポートできます。

XML マニフェストをインストールサービスに追加するときに、そのマニフェストが検証されます。スクリプトをインストールサービスに追加するときに、そのスクリプトは検証されません。

スクリプトをインストールサービスに追加する前に、派生した AI マニフェストを検証してください。

1. 意図されたクライアントシステムに似ている環境でスクリプトを実行します。
2. 結果となるマニフェストに対して `validate` サブコマンドを使用します。

```
$ /usr/bin/aimanifest validate
```

検証に失敗した場合にのみメッセージが表示されます。

そのスクリプトを適切な AI インストールサービスに追加し、どのクライアントがこれらのインストール手順を使用するかを定義する条件を指定します。クライアントの選択条件を指定しない場合は、`-d` オプションを使用して、このスクリプトをサービスのデフォルトの AI マニフェストとして追加できます。

```
$ pfexec installadm create-manifest -n solaris11_1-i386 -f ./mac1.ksh -m mac1 \
-c mac=BB:AA:AA:AA:AA:AA
```

複数の `-c` オプションまたは 1 つの `-c` ファイルを指定できます。set-criteria サブコマンドも参照してください。クライアント条件の指定方法については、[第 9 章「インストールのカスタマイズ」](#) を参照してください。

installadm の list、export、create-manifest、set-criteria、update-manifest、および set-service サブコマンドについては、[117 ページの「インストールサーバーの維持」](#) を参照してください。

AI マニフェストの例

このセクションの例は、完成した AI マニフェストが規定された結果を実現するために持っている必要がある XML 要素を示しています。これらのマニフェストを作成するには、XML を直接編集するか、または派生マニフェストスクリプトを使用します。

このセクションに示すマニフェストはすべて、default.xml マニフェストを基にして、必要な変更を加えたものです。説明を簡単にするため、software 要素内の destination 要素は省略しています。

iSCSI ターゲットデバイスの指定

この例では、インストールのターゲットは iSCSI デバイスです。target 要素内の disk 要素内の iscsi 要素を使用します。disk 要素の whole_disk 属性は、iSCSI ディスクで一般的な true に設定されています。target_name、target_lun、および target_ip 属性の説明については、[ai_manifest\(4\)](#) のマニュアルページを参照してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Copyright (c) 2008, 2012, Oracle and/or its affiliates. All rights reserved.
-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="default">
    <target>
```

```

<disk whole_disk="true">
  <iscsi target_name="iqn.1986-03.com.sun:02:1234567890abcdef" \
    target_lun="1" target_ip="129.158.144.200"/>
</disk>
<logical>
  <zpool name="rpool" is_root="true">
    <filesystem name="export" mountpoint="/export"/>
    <filesystem name="export/home"/>
    <be name="solaris"/>
  </zpool>
</logical>
</target>
<software type="IPS">
  <source>
    <publisher name="solaris">
      <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
  </source>
  <software_data action="install">
    <name>pkg:/entire@0.5.11-0.175.1</name>
    <name>pkg:/group/system/solaris-large-server</name>
  </software_data>
</software>
</ai_instance>
</auto_install>

```

RAID 構成の指定

この例では、2 台のディスク `c0t0d0` および `c0t1d0` を使用した RAID 構成を指定しています。このマニフェストは、例 10-3 に示したミラー化構成のマニフェストと似ています。2 つのマニフェストの違いの 1 つは、`redundancy` 属性の値が `mirror` ではなく `raidz` であることです。冗長性タイプについては、`zpool(1M)` のマニュアルページを参照してください。もう 1 つの違いは、ZFS プールの名前が `rpool` ではないことです (`rpool` はルートプールを意味するためです)。デフォルトでは、`zpool` 要素の `is_root` 属性の値は `false` であるため、この例では割り当てを省略できます。ルートプールが指定されていないため、このインストールの初期ユーザーを構成しないでください。

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
  Copyright (c) 2008, 2012, Oracle and/or its affiliates. All rights reserved.
-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="default">
    <target>
      <disk in_vdev="raid_vdev" in_zpool="raidpool" whole_disk="true">
        <disk_name name="c0t0d0" name_type="ctd"/>
      </disk>
      <disk in_vdev="raid_vdev" in_zpool="raidpool" whole_disk="true">
        <disk_name name="c0t1d0" name_type="ctd"/>
      </disk>
    </logical>
  </target>
</ai_instance>
</auto_install>

```



```

    <zpool name="raidpool" is_root="false">
      <vdev name="raid_vdev" redundancy="raidz"/>
    </zpool>
  </logical>
</target>
<software type="IPS">
  <source>
    <publisher name="solaris">
      <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
  </source>
  <software_data action="install">
    <name>pkg:/entire@0.5.11-0.175.1</name>
    <name>pkg:/group/system/solaris-large-server</name>
  </software_data>
</software>
</ai_instance>
</auto_install>

```

SVR4 パッケージのインストール

この例は、SVR4 パッケージをインストールする方法を示しています。SVR4 パッケージの名前は、type が SVR4 である software 要素内で指定する必要があります。発行元の起点の name 属性の値は、SVR4 パッケージのサブディレクトリを含むディレクトリまたは SVR4 パッケージのデータストリームファイルです。SVR4 パッケージサブディレクトリのこの起点名は、ファイルのフルディレクトリパスまたはファイルの URI にすることができます。SVR4 パッケージデータストリームファイルのこの起点名は、ファイルのフルディレクトリパス、ファイルの URI、または HTTP URI にすることができます。

ヒント-インストールの一部として、ユーザー入力が必要なパッケージをインストールしないでください。

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
  Copyright (c) 2008, 2012, Oracle and/or its affiliates. All rights reserved.
-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="default">
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">

```

```
<source>
  <publisher name="solaris">
    <origin name="http://pkg.oracle.com/solaris/release"/>
  </publisher>
</source>
<software_data action="install">
  <name>pkg:/entire@0.5.11-0.175.1</name>
  <name>pkg:/group/system/solaris-large-server</name>
</software_data>
</software>
<software type="SVR4">
  <source>
    <publisher>
      <origin name="/net/host2/usr/dist"/>
    </publisher>
  </source>
  <software_data>
    <name>SUNWpackage</name>
  </software_data>
</software>
</ai_instance>
</auto_install>
```

クライアントシステムの構成

この章では、インストール後にクライアントシステムを構成するために必要な情報を指定する方法について説明します。構成可能な項目の構成は、`smf(5)` プロパティを使用して指定できます。

構成プロファイルの指定

システム構成プロファイルは、クライアントシステムの構成を一連の構成パラメータとしてサービス管理機能 (SMF) プロファイルの形式で指定します。システム構成プロファイルは、適切な SMF サービスに対して SMF プロパティを設定します。

システム構成プロファイルは、AI インストール後のシステムの初回ブート時に適用されます。特定の構成領域を担当する SMF サービスは、SMF プロパティを処理し、それに応じてシステムを構成します。

各クライアントは、任意の数のシステム構成プロファイルを使用できます。たとえば、1つのクライアントに、そのクライアントのホスト名と IP アドレスのみを提供するプロファイルを割り当てることができます。同じクライアントとほかの多くのクライアントに、より広範囲に適用できるプロパティ値を設定するほかのプロファイルを割り当てることができます。

特定のクライアントにシステム構成プロファイルが提供されない場合は、対話型構成ツールがそのクライアントで開きます。対話型で構成ツールを使用する方法については、[79 ページ](#)の「[システムの構成](#)」を参照してください。

システム構成プロファイルの作成

システム構成プロファイルの作成には、次のいずれかの方法を使用します。

- 対話型構成ツールを実行し、出力をファイルに保存します。次のコマンドでは、対話型で入力した応答から有効なプロファイルを `sc.xml` に作成します。

```
# sysconfig create-profile -o sc.xml
```

構成ツールを使用してプロファイルファイルを作成する方法については、[84 ページの「SCI Tool を使用した構成プロファイルの作成」](#)を参照してください。

- [174 ページの「システム構成プロファイル内での構成の指定」](#) および [187 ページの「システム構成プロファイルの例」](#) に示されるプロパティー仕様を使って、システム構成プロファイルを手動で作成します。

すべてのシステム構成プロファイルに次の行を含めます。

```
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- service, property_group, property, and propval specifications -->
</service_bundle>
```

該当しないサービスまたはプロパティーを指定した場合、その指定は無視されません。

特定のプロパティーを何度も指定しないでください。

システム構成プロファイルでは、プロパティーと属性の値を2つの方法で表現できます。1つのプロファイルで両方の方法を使用できます。

- この章に示されるプロパティー仕様を使って、プロファイルがインストールサービスに追加される前に値を明示的に入力できます。
- システム構成プロファイルには、そのプロファイルを使用してクライアントシステムをインストールするときに有効な値に置き換えられる変数を含めることができます。[185 ページの「システム構成プロファイルテンプレートの使用」](#)を参照してください。

システム構成プロファイルの検証

作成中のシステム構成プロファイルの構文の正しさを検証するには、`installadm validate` コマンドを使用します。このプロファイルを追加する予定のインストールサービスはすでに存在している必要があります。`validate` サブコマンドの詳細は、[129 ページの「システム構成プロファイルの検証」](#)を参照してください。

インストールサービスへのシステム構成プロファイルの追加

システム構成プロファイルをインストールサービスに追加するには、`installadm create-profile` コマンドを使用します。`create-profile` サブコマンドは、プロファイルを検証してから、それらをインストールサービスに追加します。

適切なクライアントがそのシステム構成プロファイルを選択するように条件を指定します。条件を指定しない場合は、すべてのクライアントがこのプロファイルを使用します。

単一のクライアントが複数のシステム構成プロファイルに一致し、複数のシステム構成プロファイルを使用する場合があります。特定のプロパティが何度も指定されるような一連のプロファイルを使用するクライアントがないようにしてください。クライアントが特定のプロパティの複数の指定を受け取る場合は、そのプロパティの値が各指定で同じであっても、構成している SMF サービスの動作は未定義です。

クライアントがインストールサービスのシステム構成プロファイル用に指定されたどの条件にも一致しない場合は、対話型構成ツールがそのクライアントで開きません。

指定のインストールサービスに追加されているプロファイルを一覧表示し、プロファイルごとに指定されている条件を一覧表示するには、`installadm list` コマンドを使用します。

プロファイル用に指定されたクライアント選択条件を変更するか、またはそれに追加するには、`installadm set-criteria` コマンドを使用します。

インストールサービスに追加されているプロファイルの内容のコピーを取得するには、`installadm export` コマンドを使用します。そのコピーを変更して別のプロファイルを作成できます。

インストールサービスにすでに追加されているプロファイルの内容を置き換えるには、`installadm update-profile` コマンドを使用します。

`create-profile`、`update-profile`、`list`、`set-criteria`、および `export` サブコマンドの詳細は、[117 ページの「インストールサーバーの維持」](#) および `installadm(1M)` のマニュアルページを参照してください。

システム構成プロファイル内での構成の指定

構成可能なシステム特性の構成は、smf(5) プロパティを使用して指定できます。たとえば、システム構成プロファイルは、root アカウント、初期ユーザー、キーボード配列、端末タイプ、IPv4 ネットワークインタフェース (静的または DHCP) とデフォルトルート、IPv6 ネットワークインタフェース (静的または addrconf) とデフォルトルート、およびネームサービス (ネームサーバーリスト、検索リスト、ドメイン) を構成できます。該当しないサービスまたはプロパティを指定した場合、その指定は無視されます。特定のプロパティを何度も指定しないでください。

どの SMF プロパティを指定する必要があるか確信を持っていない場合は、svccfg コマンドの describe サブコマンドを使用して、サービスのプロパティグループとプロパティの説明 (使用可能な設定を含む) を表示できます。svccfg(1M) のマニュアルページの「プロパティの検査および変更のサブコマンド」を参照してください。

```
svccfg -s FMRI describe [-v] [-t] [propertygroup/property]
```

プロパティグループまたは特定のプロパティのクエリー検索を行うには、プロパティグループ名、またはプロパティグループ名とプロパティ名をスラッシュ (/) で区切ったものを引数として指定します。

-v オプションを指定すると、現在の設定、制約、使用可能なその他の設定の選択肢の説明など、使用可能なすべての情報が提供されます。

-t オプションを指定すると、選択内容のテンプレートデータのみが表示され (smf_template(5) のマニュアルページを参照)、プロパティグループやプロパティの現在の設定は表示されません。

```
$ svccfg -s name-service/switch describe config
config                application
  Name service switch configuration data as described in nsswitch.conf(4).
config/value_authorization  astring                solaris.smf.value.name-service.switch
config/default            astring                files
  Default configuration database entry.
config/host                astring                "files dns mdns"
  Override configuration for host database lookups. (both IPv4 and IPv6 hosts)
config/printer            astring                "user files"
  Override configuration for printer database lookups.
$ svccfg -s name-service/switch describe -v config
config                application
  name: config
  type: application
  required: true
  target: this
  description: Name service switch configuration data as described in nsswitch.conf(4).
config/value_authorization  astring                solaris.smf.value.name-service.switch
config/default            astring                files
  type: astring
  required: true
  Default configuration database entry.
```

```

visibility: readwrite
minimum number of values: 1
maximum number of values: 1
value: files
...
$ svccfg -s name-service/switch describe -t config
name: config
type: application
  Name service switch configuration data as described in nsswitch.conf(4).
name: default
type: astring
  Default configuration database entry.
name: host
type: astring
  Override configuration for host database lookups. (both IPv4 and IPv6 hosts)
name: password
type: astring
  Override configuration for passwd database lookups. Also used with the shadow and user_attr databases.
name: group
type: astring
  Override configuration for group database lookups.
name: network
type: astring
  Override configuration for network database lookups.
...
$ svccfg -s system/config-user describe root_account
root_account          application
root_account/expire  astring
root_account/password astring
root_account/read_authorization astring          solaris.smf.read.system-config
root_account/stability astring          Evolving
root_account/type     astring

```

root アカウントとユーザーアカウントの構成

root ユーザーと初期ユーザーを構成する有効なプロファイルを生成するには、次の `sysconfig create-profile` コマンドを `users` グループとともに入力します。

```
# sysconfig create-profile -g users -o sc_users.xml
```

`svc:/system/config-user` SMF サービスは、ユーザーアカウントと root アカウントを構成します。このサービスは、2つのプロパティグループを認識します。

- `root_account` プロパティグループには、root アカウントを構成する SMF プロパティが含まれています。
- `user_account` プロパティグループには、ユーザーアカウントを構成する SMF プロパティが含まれています。

ヒント - Oracle Solaris OS の暗号化されたパスワードを生成する 1 つの方法は、意図された名前とパスワードのユーザーを作成し、ユーザーのレコードの 1 番目のコロンと 2 番目のコロンの間にある `/etc/shadow` ファイルからパスワードをコピーし、その情報をマニフェストの `password` の値に追加します。

root アカウントの構成

`root_account` プロパティグループには、次の表に示すプロパティが含まれています。

表 11-1 `root_account` プロパティグループのプロパティ

プロパティ	タイプ	必要	説明
<code>password</code>	<code>astring</code>	必須	暗号化された <code>root</code> パスワードを指定します。 <code>root</code> パスワードを指定しない場合、 <code>root</code> パスワードは空になります。
<code>type</code>	<code>astring</code>	オプション	アカウントタイプ: <code>normal</code> または <code>role</code> 。デフォルトは <code>normal</code> です。
<code>expire</code>	<code>string</code>	オプション	ログインの有効期限。0 (ゼロ) に設定した場合、ユーザーは次のログイン時に <code>root</code> パスワードを変更するように要求されます。

例 11-1 パスワードが必ず期限切れになる `root` アカウントの構成

```
<service name="system/config-user" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="root_account" type="application">
      <propval name="password" value="encrypted_password"/>
      <propval name="type" value="normal"/>
      <propval name="expire" value="0"/>
    </property_group>
  </instance>
</service>
```

ユーザーアカウントの構成

このセクションでは、次の内容について説明します。

- 176 ページの「オートマウントに依存しないユーザーアカウントの作成」
- 177 ページの「ユーザーアカウントプロパティ」
- 178 ページの「複数の初期ユーザーの構成」

オートマウントに依存しないユーザーアカウントの作成

デフォルトでは、初期ユーザーアカウントが作成されると、ホームディレクトリはオートマウントによって管理され、`/home/login` ディレクトリに基づいてアクセスさ

れます。オートマウンタに依存せずに初期ユーザーアカウントを作成するには、構成プロファイル内の `user_account/autohome` プロパティを空の文字列 ("") に設定します。

`user_account/autohome` プロパティを空の文字列に設定すると、次の効果があります。

- `/etc/passwd` ファイル内のホームディレクトリエントリが `/home/login` ではなくホーム ZFS データセットのマウントポイントに設定されます。ホーム ZFS データセットのデフォルトのマウントポイントは `/export/home/login` です。
- `/etc/auto_home` ファイルにマッピングエントリは追加されません。

ユーザーアカウントプロパティ

`user_account` プロパティグループには、次の表に示すプロパティが含まれています。

表 11-2 `user_account` プロパティグループのプロパティ

プロパティ	タイプ	必要	説明
<code>login</code>	<code>astring</code>	<code>required</code>	ユーザーのログイン。
<code>password</code>	<code>astring</code>	<code>required</code>	暗号化されたユーザーパスワード。
<code>description</code>	<code>astring</code>	<code>optional</code>	通常はユーザーのフルネーム。
<code>shell</code>	<code>astring</code>	<code>optional</code>	ユーザーのログインシェルとして使用するプログラムのフルパス名。
<code>uid</code>	<code>count</code>	<code>optional</code>	新しいユーザーの UID。デフォルトの UID は 101 です。
<code>gid</code>	<code>count</code>	<code>optional</code>	ユーザーのプライマリグループのメンバーシップ。デフォルトの GID は 10 です。
<code>type</code>	<code>astring</code>	<code>optional</code>	アカウントタイプ: <code>normal</code> または <code>role</code> 。デフォルトは <code>normal</code> です。
<code>profiles</code>	<code>astring</code>	<code>optional</code>	<code>prof_attr(4)</code> のマニュアルページで定義された、コマンドで区切られた 1 つ以上の実行プロファイル。
<code>roles</code>	<code>astring</code>	<code>optional</code>	<code>user_attr(4)</code> のマニュアルページで定義された、コマンドで区切られた 1 つ以上の役割。
<code>sudoers</code>	<code>astring</code>	<code>optional</code>	<code>sudoers</code> ファイルに <code>login</code> とともに追加されるエントリ。
<code>expire</code>	<code>astring</code>	<code>optional</code>	ログインの有効期限。0 (ゼロ) に設定した場合、ユーザーは次のログイン時にパスワードを変更するように要求されます。

表 11-2 user_account プロパティグループのプロパティ (続き)

プロパティ	タイプ	必要	説明
home_zfs_dataset	aststring	optional	ユーザーのホームディレクトリの ZFS データセット。デフォルトは <code>root_pool/export/home/login</code> です。
home_mountpoint	aststring	optional	ユーザーのホームディレクトリのマウントポイント。デフォルトは <code>/export/home/login</code> です。
autohome	aststring	optional	ユーザーの自動ホームディレクトリのマウントポイント。この値は、構成されるユーザーの <code>/etc/auto_home</code> ファイルに入力されます。デフォルト値は <code>localhost:/export/home/login</code> です。autohome プロパティが空の文字列 ("") に設定されると、オートマウントに依存せずにユーザーアカウントが作成されます。

複数の初期ユーザーの構成

新規にインストールされたシステムに複数のユーザーを構成するには、スクリプト内で `useradd(1M)` コマンドを使用してユーザーを指定します。次に、1 度だけ実行する SMF サービスを使用して初回ブート時にスクリプトを実行します。手順については、第 13 章「初回ブート時のカスタムスクリプトの実行」を参照してください。

システム識別情報の設定

システムノード名を構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `identity` グループとともに使用します。

```
# sysconfig create-profile -g identity -o sc_identity.xml
```

`svc:/system/identity:node` SMF サービスは、システムのホスト名を設定します。node は `svc:/system/identity` のインスタンスです。

`identity` プロパティグループには、次の表に示すプロパティが含まれています。

表 11-3 config プロパティグループのプロパティ

プロパティ	タイプ	必要	説明
nodename	aststring	オプション	システムのホスト名。デフォルトは <code>solaris</code> です。
enable_mapping	boolean	オプション	ノード名マッピングを無効にするために使用する値。デフォルトは <code>true</code> です。
loopback	aststring	オプション	<code>loopback</code> にマップされるホスト名。デフォルトは <code>solaris</code> です。

例11-2 ホスト名の構成

この例では、システムのホスト名を `solaris` に設定しています。

```
<service name="system/identity" version="1" type="service">
  <instance name="node" enabled="true">
    <property_group name="config" type="application">
      <propval name="nodename" value="solaris"/>
    </property_group>
  </instance>
</service>
```

例11-3 ノード名マッピングの無効化

Oracle Solaris 11 OS または Oracle Solaris 11 の更新リリースをインストールすると、デフォルトではシステムノード名が `loopback` またはインストールの一部として構成されるインタフェースの IP アドレスにマップされます。このデフォルトのマッピングを無効にするには、次の例に示すように、`enable_mapping` プロパティを `false` に設定します。

```
<service name="system/identity" version="1" type="service">
  <instance name="node" enabled="true">
    <property_group name="config" type="application">
      <propval name="nodename" value="solaris"/>
      <propval name="enable_mapping" value="false"/>
    </property_group>
  </instance>
</service>
```

タイムゾーンとロケールの設定

タイムゾーンとロケールを構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `location` グループとともに使用します。

```
# sysconfig create-profile -g location -o sc_location.xml
```

`svc:/system/timezone` SMF サービスは、システムのタイムゾーンを設定します。

`timezone` プロパティグループには、次の表に示すプロパティが含まれていません。

表 11-4 `timezone` プロパティグループのプロパティ

プロパティ	タイプ	必要	説明
<code>localtime</code>	<code>astring</code>	オプ ション	システムのタイムゾーン。デフォルトは UTC です。

例11-4 タイムゾーンの構成

この例では、タイムゾーンを中央ヨーロッパ標準時/プラハ、チェコに設定しています。

```
<service name='system/timezone' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='timezone'>
      <propval name='localtime' value='Europe/Prague' />
    </property_group>
  </instance>
</service>
```

svc:/system/environment:init SMF サービスは、システムのロケールを設定します。

environment プロパティグループは、次の環境変数を定義できます。環境変数については、[environ\(5\)](#)のマニュアルページを参照してください。

表11-5 environment プロパティグループのプロパティ

環境変数	タイプ	必要	デフォルト値
LC_CTYPE	astring	オプション	C
LC_NUMERIC	astring	オプション	C
LC_TIME	astring	オプション	C
LC_COLLATE	astring	オプション	C
LC_MONETARY	astring	オプション	C
LC_MESSAGES	astring	オプション	C
LC_ALL	astring	オプション	C
LANG	astring	オプション	C

例11-5 ロケールの構成

この例では、ロケールをチェコ語 (cs) およびチェコ共和国 (cz) に設定しています。

```
<service name='system/environment' version='1'>
  <instance name='init' enabled='true'>
    <property_group name='environment'>
```

例 11-5 ロケールの構成 (続き)

```
<propval name='LC_ALL' value='cs_CZ.UTF-8' />
</property_group>
</instance>
</service>
```

端末タイプとキーボード配列の設定

例 11-6 端末タイプの構成

svc:/system/console-login SMF サービスは、端末タイプを構成します。関連する SMF プロパティの定義については、ttymon(1M) のマニュアルページを参照してください。

この例では、端末タイプを vt100 に設定しています。

```
<service name="system/console-login" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="ttymon" type="application">
      <propval name="terminal_type" value="vt100"/>
    </property_group>
  </instance>
</service>
```

例 11-7 キーボード配列の構成

キーボード配列を構成する有効なプロファイルを生成するには、sysconfig create-profile コマンドを kbd_layout グループとともに使用します。

```
# sysconfig create-profile -g kbd_layout -o sc_kdb.xml
```

svc:/system/keymap SMF サービスは、キーボード配列を構成します。関連する SMF プロパティの定義については、kbd(1) のマニュアルページを参照してください。

この例では、キーボード配列をチェコ語に設定しています。

```
<service name='system/keymap' version='1' type='service'>
  <instance name='default' enabled='true'>
    <property_group name='keymap' type='system'>
      <propval name='layout' value='Czech' />
    </property_group>
  </instance>
</service>
```

静的なネットワークの構成

ネットワークを構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `network` グループとともに使用します。

```
# sysconfig create-profile -g network -o sc_network.xml
```

`svc:/network/install` SMF サービスは、初期の物理ネットワークインタフェースを構成します。このサービスは、システム構成が行われないプロパティ値で最初は無効になっています。

注-インストールのターゲットが iSCSI デバイスである場合は、そのインストールのシステム構成プロファイルでそのネットワークインタフェースを構成しないでください。iSCSI ブートの場合は、iSCSI デバイスのネットワークインタフェースがクライアントのブートプロセスの早い段階で構成されます。その同じインタフェースを再度構成すると、インタフェースの `network/install` サービスが保守状態になります。

複数のネットワークインタフェースを構成するには、スクリプトにその構成を指定し、1度だけ実行する SMF サービスを使用して初回ブート時にそのスクリプトを実行します。手順とスクリプトの例については、[第 13 章「初回ブート時のカスタムスクリプトの実行」](#)を参照してください。

`svc:/network/install` サービスは、1つの IPv4 インタフェースと1つの IPv6 インタフェースの構成をサポートしています。また、任意で、これらのインタフェースが到達できるデフォルトルートの構成をサポートしています。このサービスは、2つのプロパティグループを定義します。IPv4 のインタフェースに1つのプロパティグループ、IPv6 のインタフェースに1つのプロパティグループです。このサービスは、そのプロパティと `ipadm(1M)` を使用してネットワークインタフェースを構成します。同様に、このサービスは、そのプロパティと `route(1M)` を使用してデフォルトルートを定義します。

[189 ページの「静的なネットワーク構成の指定」](#)の例を参照してください。

`install_ipv4_interface` プロパティグループには、次の表に示すプロパティが含まれています。

表 11-6 `install_ipv4_interface` プロパティグループのプロパティ

プロパティ	タイプ	必要	説明
<code>name</code>	<code>astring</code>	<code>required</code>	ネットワークインタフェースの名前。
<code>address_type</code>	<code>astring</code>	<code>required</code>	<code>ipadm(1M) create-addr</code> サブコマンドの <code>-T</code> オプションの構築に使用する値。有効な値は <code>static</code> または <code>dhcp</code> です。

表 11-6 install_ipv4_interface プロパティグループのプロパティ (続き)

プロパティ	タイプ	必要	説明
static_address	net_address_v4	optional	static の address_type でのみ必要です。ipadm(1M) create-addr サブコマンドのローカルアドレスの構築に使用されます。
dhcp_wait	astiring	optional	dhcp の address_type でのみ適用されます。このプロパティを定義した場合は、ipadm(1M) create-addr サブコマンドの -w seconds (または forever) の部分の構築に使用されます。
default_route	net_address_v4	optional	route(1M) を使用したデフォルトルート の定義に使用されます。 # /usr/sbin/route \ -p add default default-route \ -ifp ifname ifname の値は、name プロパティのインタフェース名の部分です。

install_ipv6_interface プロパティグループには、次の表に示すプロパティが含まれています。

表 11-7 install_ipv6_interface プロパティグループのプロパティ

プロパティ	タイプ	必要	説明
name	astiring	required	ネットワークインタフェースの名前。
address_type	astiring	required	ipadm(1M) create-addr サブコマンドの -r オプションの構築に使用する値。有効な値は static または addrconf です。
static_address	net_address_v6	optional	static の address_type でのみ必要です。ipadm(1M) create-addr サブコマンドのローカルアドレスの構築に使用されます。
interface_id	net_address_v6	optional	addrconf の address_type でのみ適用されます。ipadm(1M) create-addr サブコマンドの -i interface_id の部分の構築に使用されます。
stateless	astiring	optional	addrconf の address_type でのみ適用されます。ipadm(1M) create-addr サブコマンドの -p stateless=yes no の部分の構築に使用されません。
stateful	astiring	optional	addrconf の address_type でのみ適用されます。ipadm(1M) create-addr サブコマンドの -p stateful=yes no の部分の構築に使用されません。

表 11-7 install_ipv6_interface プロパティグループのプロパティ (続き)

プロパティ	タイプ	必要	説明
default_route	net_address_v6	optional	route(1M) を使用したデフォルトルートの定義に使用されます。 # /usr/sbin/route \ -p add default default-route \ -ifp ifname ifname の値は、name プロパティのインタフェース名の部分です。

svc:/network/dns/client サービスは、DNS クライアントの構成をサポートしていません。このサービスは、1つのプロパティグループ (config) を定義します。このサービスは、そのプロパティを使用して DNS の [resolv.conf\(4\)](#) ファイルを構築します。

config プロパティグループには、次の表に示すプロパティが含まれています。

表 11-8 config プロパティグループのプロパティ

プロパティ	タイプ	必要	説明
domain	astring	optional	ローカルドメイン名。resolv.conf(4) の domain ディレクティブの構築に使用されます。
nameserver	net_address_list	required	IPv4 および IPv6 アドレスのリスト。resolv.conf(4) の nameserver ディレクティブの構築に使用されます。
search	astring_list	optional	ホスト名検索用の検索リストのためのドメイン値のリスト。resolv.conf(4) の search ディレクティブの構築に使用されます。

ネームサービスの構成

DNS、NIS、および LDAP クライアントとネームサービススイッチを構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `naming_services` グループとともに使用します。

```
# sysconfig create-profile -g naming_services -o sc_ns.xml
```

svc:/network/dns/client SMF サービスは、初期の DNS クライアント構成を構成します。このサービスは、システム構成が行われないプロパティ値で最初は無効になっています。191 ページの「[ネームサービス構成の指定](#)」の例を参照してください。

Oracle Configuration Manager と Oracle Auto Service Request の設定

Oracle Configuration Manager では、システム構成を My Oracle Support に登録でき、Oracle Auto Service Request では、特定のハードウェア障害に対するサービスリクエストを自動的に生成できます。

Oracle Configuration Manager と Oracle Auto Service Request を構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `support` グループとともに使用します。

```
# sysconfig create-profile -g support -o sc_support.xml
```

出力のプロファイルは登録の最初のフェーズを設定しますが、これは次の条件に一致するすべてのクライアントで同じになります。

- システムは、同じ My Oracle Support 資格を使用して登録します。このプロファイルを使用するすべてのクライアントシステムは、同じ方法で My Oracle Support に登録します。これらすべてのクライアントのデータは、同じ My Oracle Support アカウントに関連付けられます。
- システムは、同じネットワーク構成を介して My Oracle Support にアクセスします。このプロファイルを使用するすべてのクライアントシステムは、たとえば同じプロキシサーバーとアグリゲーションハブを介して My Oracle Support にアクセスします。

AI クライアントシステムの別のグループのために追加のプロファイルを作成する必要がある場合は、既存のプロファイルをコピーして編集せずに、`sysconfig create-profile` コマンドを再実行するようにしてください。プロキシサーバーにユーザー名とパスワードがある場合は、パスワードが暗号化されているため、`sysconfig create-profile` を再実行する必要があります。

システム構成プロファイルテンプレートの使用

プロファイルには、インストール処理中にクライアントのインストール環境から得られた値で置換される変数を含めることができます。このように、単一のプロファイルファイルで、さまざまなクライアント上にさまざまな構成パラメータを設定できます。使用できる変数の一覧については、[表 11-9](#) を参照してください。

次の `hostIPnet.xml` という名前のプロファイルの例では、`AI_HOSTNAME` はクライアントシステムのホスト名のプレースホルダであり、`AI_IPV4` はクライアントシステムの IP アドレスのプレースホルダです。

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
```

```

<service name="system/identity" version="1" type="service">
  <instance name="node" enabled="true">
    <property_group name="config" type="application">
      <propval name="nodename" value="{AI_HOSTNAME}"/>
    </property_group>
    <property_group name="install_ipv4_interface" type="application">
      <propval name="name" value="net0/v4"/>
      <propval name="address_type" value="static"/>
      <propval name="static_address" type="net_address_v4" value="{AI_IPV4}}/8"/>
      <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
    </property_group>
  </instance>
</service>
</service_bundle>

```

次のコマンドでは、入力の `hostandIP.xml` ファイルを変更せずに、インストールクライアントごとにカスタマイズされるシステム構成プロファイルをインストールサービス内に作成しています。

```
$ pfexec installadm create-profile -n solaris11_1-i386 -f /export/hostIPnet.xml
```

`hostandIP.xml` ファイルは変更されませんが、クライアントに適用されるプロファイルがカスタマイズされます。たとえば、`server1` というホスト名を持つクライアントがインストールされると、`hostandIP.xml` プロファイルの内容は次のようになる可能性があります。

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service name="system/identity" version="1" type="service">
    <instance name="node" enabled="true">
      <property_group name="config" type="application">
        <propval name="nodename" value="server1"/>
      </property_group>
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" value="net0/v4"/>
        <propval name="address_type" value="static"/>
        <propval name="static_address" type="net_address_v4" value="10.0.0.2/8"/>
        <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
      </property_group>
    </instance>
  </service>
</service_bundle>

```

次の表に、テンプレートプロファイル内でプレースホルダとして使用できる変数を示します。

表 11-9 システム構成テンプレートプロファイルの変数

変数名	説明
AI_ARCH	<code>uname -m</code> からのカーネルアーキテクチャー
AI_CPU	<code>uname -p</code> からのプロセッサタイプ

表 11-9 システム構成テンプレートプロファイルの変数 (続き)

変数名	説明
AI_HOSTNAME	クライアントの DNS 名
AI_IPV4	IP バージョン 4 のネットワークアドレス
AI_MAC	コロン (:) で区切られた 16 進数の MAC アドレス
AI_MEM	prtconf によって返される M バイト単位のメモリーサイズ
AI_NETWORK	IP バージョン 4 のネットワーク識別子
AI_SERVICE	インストールサービス名
AI_ZONE_NAME	zoneadm list で示される zones(5) ゾーンの名前

システム構成プロファイルの例

このセクションの例は、`installadm create-profile` コマンドを使用してインストールサービスに追加できる完全なシステム構成プロファイルです。

システム構成プロファイルの例

このセクションでは、変更するベースとして使用できるシステム構成プロファイル例を示します。この例は、`/usr/share/auto_install/sc_profiles/sc_sample.xml` で利用できます。インストールサービスの作成が完了したら、`image-path/auto_install/sc_profiles/sc_sample.xml` でこの構成プロファイル例を利用できます。

```
<?xml version="1.0"?>
<!--
Copyright (c) 2011, 2012, Oracle and/or its affiliates. All rights reserved.
-->

<!--
Sample system configuration profile for use with Automated Installer

Configures the following:
* User account name 'jack', password 'jack', GID 10, UID 101, root role, bash shell
* 'root' role with password 'solaris'
* Keyboard mappings set to US-English
* Time zone set to UTC
* Network configuration is automated with Network Auto-magic
* DNS name service client is enabled

See installadm(1M) for usage of 'create-profile' subcommand.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
```

```
<service_bundle type="profile" name="system configuration">
  <service name="system/config-user" version="1">
    <instance name="default" enabled="true">
      <property_group name="user_account">
        <propval name="login" value="jack"/>
        <propval name="password" value="9Nd/cwBcNWFZg"/>
        <propval name="description" value="default user"/>
        <propval name="shell" value="/usr/bin/bash"/>
        <propval name="gid" value="10"/>
        <propval name="uid" value="101"/>
        <propval name="type" value="normal"/>
        <propval name="roles" value="root"/>
        <propval name="profiles" value="System Administrator"/>
      </property_group>
      <property_group name="root_account">
        <propval name="password" value="encrypted_password"/>
        <propval name="type" value="role"/>
      </property_group>
    </instance>
  </service>

  <service version="1" name="system/identity">
    <instance enabled="true" name="node">
      <property_group name="config">
        <propval name="nodename" value="solaris"/>
      </property_group>
    </instance>
  </service>

  <service name="system/console-login" version="1">
    <instance name="default" enabled="true">
      <property_group name="ttymon">
        <propval name="terminal_type" value="sun"/>
      </property_group>
    </instance>
  </service>

  <service name="system/keymap" version="1">
    <instance name="default" enabled="true">
      <property_group name="keymap">
        <propval name="layout" value="US-English"/>
      </property_group>
    </instance>
  </service>

  <service name="system/timezone" version="1">
    <instance name="default" enabled="true">
      <property_group name="timezone">
        <propval name="localtime" value="UTC"/>
      </property_group>
    </instance>
  </service>

  <service name="system/environment" version="1">
    <instance name="init" enabled="true">
      <property_group name="environment">
        <propval name="LANG" value="en_US.UTF-8"/>
      </property_group>
    </instance>
  </service>
```

```

</service>
<service name="network/physical" version="1">
  <instance name="default" enabled="true">
    <property_group name="netcfg" type="application">
      <propval name="active_ncp" type="astring" value="Automatic"/>
    </property_group>
  </instance>
</service>
</service_bundle>

```

静的なネットワーク構成の指定

このプロファイル例のバージョン

は、`/usr/share/auto_install/sc_profiles/static_network.xml` で利用できます。このプロファイルの以下に示すバージョンは、次のパラメータを構成するために変更されています。

- IPv4 の静的アドレスが `10.0.0.10` でネットマスクが `255.0.0.0` の `bge0`
- `10.0.0.1` の IPv4 デフォルトルート
- IPv6 のアドレスタイプが `addrconf` の `bge1`
- DNS `8.8.8.8` ネームサーバー
- `example1.com` と `example2.com` (ホスト名検索の DNS 検索リストとして)

ネットワークは、`IPaddress/netmask` という表記で指定されます。ここで、`netmask` はネットマスクの上位ビットの数を示す数値です。

<code>netmask</code> の値	ネットマスクの例
8	255.0.0.0
16	255.255.0.0
24	255.255.255.0

```

<?xml version="1.0"?>
<!--
Copyright (c) 2011, 2012, Oracle and/or its affiliates. All rights reserved.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="system configuration">
  <service name="system/config-user" version="1">
    <instance name="default" enabled="true">
      <property_group name="user_account">
        <propval name="login" value="jack"/>
        <propval name="password" value="9Nd/cwBcNWFZg"/>
        <propval name="description" value="default user"/>
        <propval name="shell" value="/usr/bin/bash"/>
        <propval name="gid" value="10"/>
        <propval name="type" value="normal"/>
      </property_group>
    </instance>
  </service>
</service_bundle>

```

```
<propval name="roles" value="root"/>
<propval name="profiles" value="System Administrator"/>
</property_group>
<property_group name="root_account">
  <propval name="password" value="$5$dnRfcZse$Hx4aBQ161Uvn9ZxJFKMdRiy8tCf4gMT2s2rtkFba2y4"/>
  <propval name="type" value="role"/>
</property_group>
</instance>
</service>

<service version="1" name="system/identity">
  <instance enabled="true" name="node">
    <property_group name="config">
      <propval name="nodename" value="solaris"/>
    </property_group>
  </instance>
</service>

<service name="system/console-login" version="1">
  <instance name="default" enabled="true">
    <property_group name="ttymon">
      <propval name="terminal_type" value="sun"/>
    </property_group>
  </instance>
</service>

<service name="system/keymap" version="1">
  <instance name="default" enabled="true">
    <property_group name="keymap">
      <propval name="layout" value="US-English"/>
    </property_group>
  </instance>
</service>

<service name="system/timezone" version="1">
  <instance name="default" enabled="true">
    <property_group name="timezone">
      <propval name="localtime" value="UTC"/>
    </property_group>
  </instance>
</service>

<service name="system/environment" version="1">
  <instance name="init" enabled="true">
    <property_group name="environment">
      <propval name="LANG" value="en_US.UTF-8"/>
    </property_group>
  </instance>
</service>

<service name="network/physical" version="1">
  <instance name="default" enabled="true">
    <property_group name="netcfg" type="application">
      <propval name="active_ncp" type="astring" value="DefaultFixed"/>
    </property_group>
  </instance>
</service>

<service name="network/install" version="1" type="service">
```

```

<instance name="default" enabled="true">
  <property_group name="install_ipv4_interface" type="application">
    <propval name="name" type="astring" value="bge0/v4"/>
    <propval name="address_type" type="astring" value="static"/>
    <propval name="static_address" type="net_address_v4" value="10.0.0.10/8"/>
    <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
  </property_group>

  <property_group name="install_ipv6_interface" type="application">
    <propval name="name" type="astring" value="bge1/v6"/>
    <propval name="address_type" type="astring" value="addrconf"/>
    <propval name="stateless" type="astring" value="yes"/>
    <propval name="stateful" type="astring" value="yes"/>
  </property_group>
</instance>
</service>

<service name="network/dns/client" version="1">
  <property_group name="config">
    <property name="nameserver">
      <net_address_list>
        <value_node value="8.8.8.8"/>
      </net_address_list>
    </property>
    <property name="search">
      <astring_list>
        <value_node value="example1.com example2.com"/>
      </astring_list>
    </property>
  </property_group>
  <instance name="default" enabled="true"/>
</service>

<service version="1" name="system/name-service/switch">
  <property_group name="config">
    <propval name="default" value="files"/>
    <propval name="host" value="files dns mdns"/>
    <propval name="printer" value="user files"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>

<service version="1" name="system/name-service/cache">
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

ネームサービス構成の指定

このセクションのプロファイル例をテンプレートとして使用して独自のプロファイルを作成できます。または、`sysconfig` ツールを `namingservices` グループとともに使用して、プロンプトへの応答に基づいてプロファイルを作成できます。`sysconfig` を使用したシステム構成プロファイルの作成の詳細は、[84 ページの「SCI Tool を使用した構成プロファイルの作成」](#) および [sysconfig\(1M\) のマニュアルページ](#) を参照してください。

ネームサービス NIS の構成

例 11-8 指定されたドメインでの NIS の有効化

このプロファイル例では、次の構成を行います。

- my.domain.com に対して NIS を有効にします
- ブロードキャストを使用して NIS サーバーを検出します。これは同じサブネット上に置かれている必要があります
- ネームサービスキャッシュサービスを有効にします。これは必須です

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
  Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
  -->
<service_bundle type='profile' name='default'>
  <service name='network/nis/domain' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='domainname' type='hostname' value='my.domain.com' />
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='network/nis/client' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='use_broadcast' type='boolean' value='true' />
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='system/name-service/switch' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='default' type='astring' value='files nis' />
      <propval name='printer' type='astring' value='user files nis' />
      <propval name='netgroup' type='astring' value='nis' />
    </property_group>
    <instance name='default' enabled='true' />
  </service>
  <service name='system/name-service/cache' type='service' version='1'>
    <instance name='default' enabled='true' />
  </service>
</service_bundle>
```

例 11-9 NIS の構成および DNS の無効化

このプロファイル例では、次の構成を行います。

- NIS サーバーに対して自動ブロードキャストでネームサービス NIS を構成します。NIS サーバーは同じサブネット上に置かれている必要があります
- NIS ドメイン my.domain.com を構成します
- ネームサービスキャッシュサービスを有効にします。これは必須です
- DNS ネームサービスを無効にします

例 11-9 NIS の構成および DNS の無効化 (続き)

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- service name-service/switch below for NIS only - (see nsswitch.conf(4)) -->
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files nis"/>
      <propval type="astring" name="printer" value="user files nis"/>
      <propval type="astring" name="netgroup" value="nis"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <!-- service name-service/cache must be present along with name-service/switch -->
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <!-- if no DNS, must be explicitly disabled to avoid error msgs -->
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>
  <service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">
      <propval type="hostname" name="domainname" value="my.domain.com"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <!-- configure the NIS client service to broadcast the subnet for a NIS server -->
  <service version="1" type="service" name="network/nis/client">
    <property_group type="application" name="config">
      <propval type="boolean" name="use_broadcast" value="true"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
</service_bundle>

```

例 11-10 NIS の構成

次のプロファイルは、サーバーの IP アドレス `10.0.0.10` とドメイン `mydomain.com` を使用してネームサービス NIS を構成します。サーバーの IP アドレスを明示的に指定する場合は、NIS サーバーは同じサブネット上に置かれている必要はありません。

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- name-service/switch below for NIS only - (see nsswitch.conf(4)) -->
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files nis"/>
      <propval type="astring" name="printer" value="user files nis"/>
      <propval type="astring" name="netgroup" value="nis"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <!-- name-service/cache must be present along with name-service/switch -->

```

例 11-10 NIS の構成 (続き)

```

<service version="1" type="service" name="system/name-service/cache">
  <instance enabled="true" name="default"/>
</service>
<!-- if no DNS, must be explicitly disabled to avoid error msgs -->
<service version="1" type="service" name="network/dns/client">
  <instance enabled="false" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="mydomain.com"/>
    <!-- Note: use property with net_address_list and value_node as below -->
    <property type="net_address" name="ypservers">
      <net_address_list>
        <value_node value="10.0.0.10"/>
      </net_address_list>
    </property>
  </property_group>
  <!-- configure default instance separate from property_group -->
  <instance enabled="true" name="default"/>
</service>
<!-- enable the NIS client service -->
<service version="1" type="service" name="network/nis/client">
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

例 11-11 指定されたドメインでの NIS と DNS の有効化

この例では、DNS と NIS の両方のネームサービスを構成します。

- 複数の DNS ネームサーバーを指定します
- DNS ドメイン検索リストを指定します
- NIS ドメインを指定します
- ブロードキャストを指定して NIS サーバーを検出します

```

<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<!--
  Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
-->
<service_bundle type='profile' name='default'>
  <service name='network/dns/client' type='service' version='1'>
    <property_group name='config' type='application'>
      <propval name='domain' type='astring' value='us.example.com' />
      <property name='nameserver' type='net_address'>
        <net_address_list>
          <value_node value='130.35.249.52' />
          <value_node value='130.35.249.41' />
          <value_node value='130.35.202.15' />
        </net_address_list>
      </property>
      <property name='search' type='astring'>
        <astring_list>
          <value_node value='us.example.com example.com example.com' />

```

例 11-11 指定されたドメインでの NIS と DNS の有効化 (続き)

```

        </astring_list>
    </property>
</property_group>
</service>
</service>
<service name='network/nis/domain' type='service' version='1'>
<property_group name='config' type='application'>
    <propval name='domainname' type='hostname' value='mydomain.com' />
</property_group>
<instance name='default' enabled='true' />
</service>
</service>
<service name='network/nis/client' type='service' version='1'>
<property_group name='config' type='application'>
    <propval name='use_broadcast' type='boolean' value='true' />
</property_group>
<instance name='default' enabled='true' />
</service>
</service>
<service name='system/name-service/switch' type='service' version='1'>
<property_group name='config' type='application'>
    <propval name='default' type='astring' value='files nis' />
    <propval name='host' type='astring' value='files dns' />
    <propval name='printer' type='astring' value='user files nis' />
    <propval name='netgroup' type='astring' value='nis' />
</property_group>
<instance name='default' enabled='true' />
</service>
</service>
<service name='system/name-service/cache' type='service' version='1'>
<instance name='default' enabled='true' />
</service>
</service_bundle>

```

ネームサービス DNS の構成

例 11-12 検索リストを使用した DNS の構成

次のプロファイル例では、次のパラメータを構成します。

- ネームサービス DNS
- サーバーの IP アドレス 1.1.1.1 および 2.2.2.2
- ドメイン dom.ain.com

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <!-- name-service/switch below for DNS only - (see nsswitch.conf(4)) -->
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files"/>
      <propval type="astring" name="host" value="files dns"/>
      <propval type="astring" name="printer" value="user files"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
</service_bundle>

```

例 11-12 検索リストを使用した DNS の構成 (続き)

```

<!-- name-service/cache must be present along with name-service/switch -->
<service version="1" type="service" name="system/name-service/cache">
  <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/dns/client">
  <property_group type="application" name="config">
    <!-- Note: use property with net_address_list and value_node as below -->
    <property type="net_address" name="nameserver">
      <net_address_list>
        <value_node value="1.1.1.1"/>
        <value_node value="2.2.2.2"/>
      </net_address_list>
    </property>
    <!-- Note: use property with astring_list and value_node,
      concatenating search names, as below -->
    <property type="astring" name="search">
      <astring_list>
        <value_node value="dom.ain.com ain.com"/>
      </astring_list>
    </property>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

ネームサービス LDAP の構成

例 11-13 LDAP および LDAP 検索ベースの構成

このプロファイル例では、次のパラメータを構成します。

- サーバーの IP アドレスが **10.0.0.10** のネームサービス LDAP
- サービス `system/nis/domain` に指定されたドメイン `my.domain.com`
- LDAP 検索ベース (必須)、`dc=my,dc=domain,dc=com`

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files ldap"/>
      <propval type="astring" name="printer" value="user files ldap"/>
      <propval type="astring" name="netgroup" value="ldap"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>
  <service version="1" type="service" name="network/ldap/client">

```

例 11-13 LDAP および LDAP 検索ベースの構成 (続き)

```

<property_group type="application" name="config">
  <propval type="astring" name="profile" value="default"/>
  <property type="host" name="server_list">
    <host_list>
      <value_node value="10.0.0.10"/>
    </host_list>
  </property>
  <propval type="astring" name="search_base" value="dc=my,dc=domain,dc=com"/>
</property_group>
<instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="my.domain.com"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

例 11-14 セキュアな LDAP サーバーを使用した LDAP の構成

このプロファイル例では、次のパラメータを構成します。

- サーバーの IP アドレスが 10.0.0.10 のネームサービス LDAP
- サービス system/nis/domain に指定されたドメイン my.domain.com
- LDAP 検索ベース (必須)、dc=my,dc=domain,dc=com
- LDAP プロキシバインド識別名
cn=proxyagent,ou=profile,dc=my,dc=domain,dc=com
- LDAP プロキシバインドパスワード。セキュリティ対策として暗号化されています。暗号化された値を見つけるには、次のいずれかの方法を使用します。
 - sysconfig create-profile から bind_passwd プロパティー値を取得します。
 - LDAP サーバーの SMF 構成から値を取得します。

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files ldap"/>
      <propval type="astring" name="printer" value="user files ldap"/>
      <propval type="astring" name="netgroup" value="ldap"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>

```

例 11-14 セキュアな LDAP サーバーを使用した LDAP の構成 (続き)

```

<service version="1" type="service" name="network/ldap/client">
  <property_group type="application" name="config">
    <propval type="astring" name="profile" value="default"/>
    <property type="host" name="server_list">
      <host_list>
        <value_node value="10.0.0.10"/>
      </host_list>
    </property>
    <propval type="astring" name="search_base" value="dc=my,dc=domain,dc=com"/>
  </property_group>
  <property_group type="application" name="cred">
    <propval type="astring" name="bind_dn" value="cn=proxyagent,ou=profile,dc=my,dc=domain,dc=com"/>
    <!-- note that the password below is encrypted -->
    <propval type="astring" name="bind_passwd" value="{NS1}c2ab873ae7c5ceefa4b9"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="my.domain.com"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

DNS を LDAP とともに使用する

DNS ネームサービスを LDAP ネームサービスとともに使用できます。一般的な使用方法では、DNS でノード名 (LDAP サーバー名を含む) を解決し、LDAP でほかのすべての名前を解決します。この例の最初の service 要素に示したように、サービス system/name-service/switch は、DNS をノード名の検索に指定し、LDAP をほかの名前の解決に指定するために使用します。

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files ldap"/>
      <propval type="astring" name="host" value="files dns"/>
      <propval type="astring" name="printer" value="user files ldap"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <property_group type="application" name="config">
      <property type="net_address" name="nameserver">
        <net_address_list>
          <value_node value="10.0.0.10"/>
        </net_address_list>

```

```

</property>
<propval type="astring" name="domain" value="my.domain.com"/>
<property type="astring" name="search">
  <astring_list>
    <value_node value="my.domain.com"/>
  </astring_list>
</property>
</property_group>
<instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/ldap/client">
  <property_group type="application" name="config">
    <propval type="astring" name="profile" value="default"/>
    <property type="host" name="server_list">
      <host_list>
        <!-- here, DNS is expected to resolve the LDAP server by name -->
        <value_node value="ldapservers.my.domain.com"/>
      </host_list>
    </property>
    <propval type="astring" name="search_base" value="dc=my,dc=domain,dc=com"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="my.domain.com"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

NIS を DNS とともに使用する

同様に、NIS を DNS とともに使用できます。

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files nis"/>
      <propval type="astring" name="host" value="files dns"/>
      <propval type="astring" name="printer" value="user files nis"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <property_group type="application" name="config">
      <property type="net_address" name="nameserver">
        <net_address_list>
          <value_node value="10.0.0.10"/>
        </net_address_list>
      </property>
      <propval type="astring" name="domain" value="my.domain.com"/>
    </property_group>
  </service>

```

```
<property type="astring" name="search">
  <astring_list>
    <value_node value="my.domain.com"/>
  </astring_list>
</property>
</property_group>
<instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="my.domain.com"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/client">
  <property_group type="application" name="config">
    <propval type="boolean" name="use_broadcast" value="true"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
</service_bundle>
```


ゾーンのインストールと構成

この章では、AIクライアントインストールの一環として非大域ゾーンのインストールと構成を指定する方法について説明します。

AIによる非大域ゾーンのインストール方法

非大域ゾーンは、大域ゾーンのインストール後の初回リブート時にインストールされ、構成されます。

1. AIを使ってシステムをインストールする場合は、AIマニフェストの `configuration` 要素を使用して非大域ゾーンをそのシステムにインストールできます。`configuration` 要素については、[202 ページの「大域ゾーンのAIマニフェストでの非大域ゾーンの指定」](#)を参照してください。
2. 大域ゾーンのインストール後にシステムが最初にブートしたときに、ゾーンの自己アセンブリ SMF サービス (`svc:/system/zones-install:default`) によって、大域ゾーンの AI マニフェストに定義されている各非大域ゾーンが構成され、インストールされます。非大域ゾーンのインストールに使用されるデータについては、[203 ページの「非大域ゾーンの構成およびインストールデータ」](#)を参照してください。
3. `autoboot=true` を指定してゾーンを構成した場合は、ゾーンのインストール後に `system/zones-install` サービスによってそのゾーンがブートされます。

`system/zones-install` サービスを使用して、ラベル付きゾーンを作成してインストールすることができます。ラベル付きゾーンが自動的にブートされるのは、そのゾーンが `autoboot=true` を指定して構成され、さらに大域ゾーンにもラベルが付いている場合のみです。AIによって大域ゾーンのインストールが完了し、`system/zones-install` サービスによってラベル付きの非大域ゾーンの作成とインストールが完了したら、必要な変更を行なって大域ゾーンをラベル付きにすることができます。システムのリブート時に、`svc:/system/zones:default` サービスによって、`autoboot=true` が設定されているラベル付きゾーンがすべてブートされます。

system/zones-install サービスはオンラインのままですが、再起動されるまで新しい構成情報は処理しません。system/zones-install サービスを無効にしたり有効にしたりしないようにしてください。このサービスの再起動のみを行うようにしてください。

非大域ゾーンのインストールを監視するには、system/zones-install サービス、または zoneadm list -cv の出力を監視します。

次のいずれかのエラーが発生した場合、ゾーンはインストールされません。

- ゾーンの config ファイルが構文的に正しくありません
- インストールされるゾーンセット内のゾーン名、ゾーンパス、または委任された ZFS データセットの間に競合が存在します。
- 必要なデータセットが大域ゾーンに構成されていません

大域ゾーンの AI マニフェストでの非大域ゾーンの指定

非大域ゾーンを指定するには、クライアントシステムの AI マニフェストに含まれる configuration 要素を使用します。ゾーンの名前を指定するには、configuration 要素の name 属性を使用します。ゾーンの config ファイルの場所を指定するには、source 属性を使用します。source の場所は、インストール中にクライアントがアクセスできる任意の http:// または file:// の場所にできます。

次の AI マニフェスト例では、2つの非大域ゾーンを指定します。

```
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance>
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@latest</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

```
<configuration type="zone" name="zone1" source="http://server/zone1/config"/>
<configuration type="zone" name="zone2" source="file:///net/server/zone2/config"/>

</ai_instance>
</auto_install>
```

非大域ゾーンの構成およびインストールデータ

次のファイルは、非大域ゾーンの構成とインストールに使用します。

config ファイル 必須。config ファイルは、`zonecfg export` コマンドの出力から得られるファイル形式のゾーンの構成です。

config ファイルの場所は、AI マニフェストの `configuration` 要素の `source` 属性で指定します。AI では、インストールされたクライアントシステムにこの config ファイルをコピーしてゾーンの構成に使用します。

AI マニフェスト 省略可能。ゾーンのインストール用のこの AI マニフェストには、ゾーンにインストールされるパッケージ、発行元情報、および必要に応じて証明書と鍵のファイルが指定されます。ゾーンのカスタム AI マニフェストの作成については、[205 ページの「非大域ゾーンの AI マニフェスト」](#)を参照してください。

カスタム AI マニフェストをゾーンに指定するには、大域ゾーンをインストールしているインストールサービスにそのマニフェストを追加します。`create-manifest` コマンドで、この AI マニフェストを使用するすべてのゾーンの名前とともに `zonename` 条件キーワードを指定します。

カスタム AI マニフェストを非大域ゾーンに指定しない場合は、[例 12-1](#)に示すようにゾーンのデフォルトの AI マニフェストが使用されます。

構成プロファイル 省略可能。非大域ゾーンに 0 個以上の構成ファイルを指定できます。これらの構成プロファイルは、大域ゾーンを構成するためのシステム構成プロファイルに似ています。システム構成プロファイルファイルについては、[第 11 章「クライアントシステムの構成」](#)を参照してください。構成プロファイルファイルを指定して、ユーザーや、ゾーン管理者の `root` パスワードなどのゾーン構成を指定することができます。非大域ゾーンの構成プロファイル例については、[207 ページの「非大域ゾーンの構成プロファイル」](#)を参照してください。

構成プロファイルファイルをゾーンに指定するには、大域ゾーンをインストールしているインストールサービスにその構

成プロファイルを追加します。create-profile コマンドで、この構成プロファイルを使用するすべてのゾーンの名前とともに zonename 条件キーワードを指定します。

構成プロファイルファイルを指定しない場合は、ゾーンの初回ブート時にシステム構成対話型ツールが実行され、必要なデータがクエリー検索されます。対話型構成ツールの使用法については、[79 ページ](#)の「システムの構成」を参照してください。

次の例では、/tmp/zmanifest.xml AI マニフェストを solaris11_1-sparc インストールサービスに追加し、zone1 と zone2 がこのマニフェストを使用することを指定します。

```
$ pfexec installadm create-manifest -n solaris11_1-sparc -f /tmp/zmanifest.xml \
-m zmanifest -c zonename="zone1 zone2"
```

次の例では、/tmp/z1profile.xml 構成プロファイルを solaris11_1-sparc インストールサービスに追加し、zone1 と zone2 がこのプロファイルを使用することを指定します。

```
$ pfexec installadm create-profile -n solaris11_1-sparc -f /tmp/z1profile.xml \
-p z1profile -c zonename="zone1 zone2"
```

次の例では、/tmp/z2profile.xml 構成プロファイルを solaris11_1-sparc インストールサービスに追加し、zone2 がこのプロファイルを使用することを指定します。

```
$ pfexec installadm create-profile -n solaris11_1-sparc -f /tmp/z2profile.xml \
-p z2profile -c zonename=zone2
```

次の例は、solaris11_1-sparc インストールサービスに追加されている AI マニフェストと構成プロファイルを示しています。

```
$ installadm list -n solaris11_1-sparc -m -p
Service/Manifest Name  Status  Criteria
-----
solaris11_1-sparc
  orig_default          Default  None
  line1-netra2000       mac      = 00:14:4F:2D:7A:DC
  zmanifest             zonename = zone1 zone2

Service/Profile Name  Criteria
-----
solaris11_1-sparc
  z1profile            zonename = zone1 zone2
  z2profile            zonename = zone2
```

非大域ゾーンのAIマニフェスト

非大域ゾーンのインストール用のこのAIマニフェストは、大域ゾーンのインストール用のAIマニフェストに似ています。AIマニフェストの要素と属性については、[ai_manifest\(4\)](#)のマニュアルページを参照してください。

非大域ゾーンのAIマニフェスト内で次の要素または属性は使用しないでください。

- `ai_instance` 要素の `auto_reboot` 属性
- `ai_instance` 要素の `http_proxy` 属性
- `target` 要素の `disk` 子要素
- `logical` 要素の `noswap` 属性
- `logical` 要素の `nodump` 属性
- `configuration` 要素

非大域ゾーンのAIマニフェストでは、`target` 要素の `logical` 子要素のみを使用できます。`logical` セクションでは、追加のファイルシステムまたはデータセットを定義します。

非大域ゾーンのAIマニフェストでは、`logical` 要素の `zpool` 要素で、`filesystem` および `be` 子要素のみを使用できます。

`software` 要素の `type` 属性でサポートされている値はIPSのみで、これはデフォルト値です。

例12-1 ゾーンのデフォルトのAIマニフェスト

次のファイルは、非大域ゾーンのデフォルトのAIマニフェストを示しています。このマニフェストは、カスタムAIマニフェストをゾーンに指定しない場合に使用されます。このマニフェストは、`/usr/share/auto_install/manifest/zone_default.xml` で入手できます。

`target` セクションは、ゾーンのZFSファイルシステムを定義します。`destination` セクションは、インストールするロケールを指定します。`software_data` セクションは、`solaris-small-server` パッケージのインストールを指定します。`solaris-small-server` パッケージは、インストールするほとんどの非大域ゾーンで推奨されるツールとデバイスドライバのグループパッケージです。`solaris-small-server` グループパッケージに含まれるパッケージの完全なリストについては、『Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新』の「グループパッケージ内のすべてのインストール可能なパッケージの一覧表示」で説明されているように、`pkg contents` コマンドを使用します。

パッケージソースが指定されていないことに注意してください。システムリポジトリについては、[pkg.sysrepo\(1M\)](#) を参照してください。

例 12-1 ゾーンのデフォルトの AI マニフェスト (続き)

```

<?xml version="1.0" encoding="UTF-8"?>
<!--

Copyright (c) 2011, 2012, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="zone_default">
    <target>
      <logical>
        <zpool name="rpool">
          <!--
Subsequent <filesystem> entries instruct an installer
to create following ZFS datasets:

          <root_pool>/export          (mounted on /export)
          <root_pool>/export/home     (mounted on /export/home)

Those datasets are part of standard environment
and should be always created.

In rare cases, if there is a need to deploy a zone
without these datasets, either comment out or remove
<filesystem> entries. In such scenario, it has to be also
assured that in case of non-interactive post-install
configuration, creation of initial user account is
disabled in related system configuration profile.
Otherwise the installed zone would fail to boot.
          -->
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris">
            <options>
              <option name="compression" value="on"/>
            </options>
          </be>
        </zpool>
      </logical>
    </target>

    <software type="IPS">
      <destination>
        <image>
          <!-- Specify locales to install -->
          <facet set="false">facet.locale.*</facet>
          <facet set="true">facet.locale.de</facet>
          <facet set="true">facet.locale.de_DE</facet>
          <facet set="true">facet.locale.en</facet>
          <facet set="true">facet.locale.en_US</facet>
          <facet set="true">facet.locale.es</facet>
          <facet set="true">facet.locale.es_ES</facet>
          <facet set="true">facet.locale.fr</facet>
          <facet set="true">facet.locale.fr_FR</facet>

```

例 12-1 ゾーンのデフォルトの AI マニフェスト (続き)

```

    <facet set="true">facet.locale.it</facet>
    <facet set="true">facet.locale.it_IT</facet>
    <facet set="true">facet.locale.ja</facet>
    <facet set="true">facet.locale.ja_*/</facet>
    <facet set="true">facet.locale.ko</facet>
    <facet set="true">facet.locale.ko_*/</facet>
    <facet set="true">facet.locale.pt</facet>
    <facet set="true">facet.locale.pt_BR</facet>
    <facet set="true">facet.locale.zh</facet>
    <facet set="true">facet.locale.zh_CN</facet>
    <facet set="true">facet.locale.zh_TW</facet>
  </image>
</destination>
<software_data action="install">
  <name>pkg:/group/system/solaris-small-server</name>
</software_data>
</software>
</ai_instance>
</auto_install>

```

非大域ゾーンの構成プロファイル

構成プロファイルをゾーンに指定して、言語、ロケール、タイムゾーン、端末、ユーザー、ゾーン管理者の root パスワードなどのゾーンパラメータを構成できます。タイムゾーンは構成できますが、時間を設定することはできません。ネームサービスを構成できます。

ゾーンで許可されていない構成を指定した場合、それらのプロパティ設定は無視されます。

次のファイルは、非大域ゾーンの構成プロファイルファイル例を示しています。

```

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/config-user">
    <instance enabled="true" name="default">
      <property_group type="application" name="root_account">
        <propval type="astring" name="login" value="root"/>
        <propval type="astring" name="password" value="encrypted_password"/>
        <propval type="astring" name="type" value="normal"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/timezone">
    <instance enabled="true" name="default">
      <property_group type="application" name="timezone">
        <propval type="astring" name="localtime" value="UTC"/>
      </property_group>
    </instance>
  </service>

```

```
<service version="1" type="service" name="system/environment">
  <instance enabled="true" name="init">
    <property_group type="application" name="environment">
      <propval type="astring" name="LC_ALL" value="C"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="system/identity">
  <instance enabled="true" name="node">
    <property_group type="application" name="config">
      <propval type="astring" name="nodename" value="z2-test"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="system/keymap">
  <instance enabled="true" name="default">
    <property_group type="system" name="keymap">
      <propval type="astring" name="layout" value="US-English"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="system/console-login">
  <instance enabled="true" name="default">
    <property_group type="application" name="ttymon">
      <propval type="astring" name="terminal_type" value="vt100"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="network/physical">
  <instance enabled="true" name="default">
    <property_group type="application" name="netcfg"/>
  </instance>
</service>
</service_bundle>
```


初回ブート時のカスタムスクリプトの実行

AI マニフェストまたはシステム構成プロファイルで実行できない追加のインストールまたは構成を行う場合は、1 度だけ実行する SMF サービスによって初回ブート時に実行されるスクリプトを作成できます。

1. 初回ブートスクリプトを作成します。
2. 初回ブート時に 1 度だけ実行され、そのスクリプトを実行する SMF サービスのマニフェストを作成します。
3. そのサービスマニフェストとスクリプトが含まれる IPS パッケージを作成します。
4. そのパッケージを IPS パッケージリポジトリに追加します。
5. そのパッケージを AI マニフェストに指定することにより、AI インストール中にそのパッケージをインストールします。

AI インストール後の初回リブート時に、サービスが実行され、スクリプトが実行されます。

初回ブート時に 1 度だけ実行する制御の実装

次の手順は、スクリプトが新規にインストールされたシステムの初回ブート時のみ実行され、かつスクリプトが 1 度だけ実行されるようにする方法を示しています。

▼ 初回ブート時に1度だけ実行されるようにする方法

- 1 スクリプトを実行するサービスを作成します。
この単純なサービスを作成するには、213 ページの「マニフェスト作成ツールの使用」に示すように、`svcbundle` コマンドを使用するのがもっとも簡単な方法です。
- 2 スクリプトを実行する前に、スクリプト完了フラグを設定します。
サービスマニフェストにブール型の完了プロパティを定義し、その値を `false` に設定します。例 13-3 のマニフェストの `completed` プロパティを参照してください。
- 3 スクリプトの最後でスクリプト完了フラグを設定します。
スクリプトの最後で、`svccfg` コマンドを使用して `completed` プロパティを `true` に設定します。`svcadm` コマンドを使用して、新しいプロパティ値でサービスをリフレッシュします。例 13-1 のサンプルスクリプトの最後を参照してください。
- 4 スクリプトが完了した場合は、サービスを無効にします。
サービスマニフェストで、デフォルトのサービスインスタンスが作成され、有効化されます。このサービスはスクリプトで無効化されます。初回ブートスクリプトを終了するときに、`SMF_EXIT_TEMP_DISABLE` 終了コードを使用してサービスの `start` メソッドを終了し、サービスを一時的に無効にします。サービスが無効になり、サービスの `stop` メソッドは実行されません。

サービスをより簡単に再有効化できるため、サービスを永続的に無効にするより、サービスを一時的に無効にすることをお勧めします。状況によっては、実行された構成作業（ゾーンのクローニングや移行）を更新するためにスクリプト（したがってサービス）を再実行する必要があります。サービスが永続的に無効になっている場合は、`svcadm enable` コマンドを実行してサービスを再有効化する必要があります。

サービスをオンラインのままにせずに、やはりサービスを一時的に無効にすることをお勧めします。オンラインのサービスは、リブートのたびに作業を実行しているように見える場合があります。この例では、サービスの名前は `site/first-boot-script-svc` です。クライアントがブートされたあとで、サービスが `disabled` 状態になっていることがわかります。

```
$ svcs first-boot-script-svc
STATE          STIME      FMRI
disabled       8:24:16   svc:/site/first-boot-script-svc:default
```

初回ブート時に実行するスクリプトの作成

スクリプトにどのソースを使用できるかを知るためには、初回ブート時にどのツールがクライアントシステムにインストールされているかを知る必要があります。デフォルトでは、`solaris-large-server` パッケージがインストールされています。そのグループパッケージをインストールした場合は、Python、bash、ksh、およびその他のツールが初回ブート時に使用可能です。`solaris-large-server` グループパッケージに含まれるパッケージの完全なリストについては、『Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新』の「グループパッケージ内のすべてのインストール可能なパッケージの一覧表示」で説明されているように、`pkg contents` コマンドを使用します。`solaris-large-server` パッケージに用意されていないソースをスクリプトに使用する場合は、必要なパッケージを特定し、それを AI マニフェストに指定します。インストールする可能性がある他のパッケージの名前を検索する方法については、『Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新』を参照してください。

ヒント-

- 異なるスクリプト内のそれぞれのコマンドが互いに競合しないようにするため、使用する初回ブートスクリプトは1つのみにしてください。
- 初回ブートスクリプト内ではリブートしないでください。

例 13-1 テンプレート初回ブートスクリプト

この例は、初回ブートスクリプトで実行すべき操作を示しています。

- 初回ブートスクリプトでは、SMF メソッドの終了コードなどの定義を使用するために `/lib/svc/share/smf_include.sh` を読み込む必要があります。
- スクリプトでは、前のブートですでに実行されたかどうかを検査するようにしてください。`completed` プロパティがすでに `true` に設定されている場合は、`start` メソッドを終了し、サービスを一時的に無効にします。

スクリプトの次の行は、`site/first-boot-script-svc:default` サービスインスタンスの `config` プロパティグループ内の `completed` プロパティの値を取得し、その値をローカルの `completed` 変数に割り当てます。

```
completed=$(svcprop -p config/completed site/first-boot-script-svc:default)
```

スクリプトの次の行は、`SMF_EXIT_TEMP_DISABLE` 終了コードとともに、終了の簡単な理由として `method_completed`、および終了の理由を示す長い説明として "Configuration completed" をサービスの `start` メソッドに送ります。

```
smf_method_exit $SMF_EXIT_TEMP_DISABLE script_completed "Configuration completed"
```

- 初回ブートスクリプトでは、AI インストールによって作成されたばかりのブート環境 (BE) のコピーを保存するようにしてください。初回ブートスクリプトで変更される前に BE のコピーを保存しておく、スクリプトによって何らかの問題が発生した場合に、保存された BE にブートするだけで簡単に回復できます。

例 13-1 テンプレート初回ブートスクリプト (続き)

- スクリプトでは、その作業が完了したときに、`completed` プロパティの値を `true` に設定し、新しいプロパティ値でサービスをリフレッシュし、`start` メソッドを終了してサービスを一時的に無効にする必要があります。 `svccfg` コマンドを使用して `completed` プロパティを `true` に設定し、`svcadm` コマンドを使用してサービスをリフレッシュします。

デフォルトでは `sh` が `ksh93` であることを覚えておいてください。

```
#!/bin/sh

# Load SMF shell support definitions
. /lib/svc/share/smf_include.sh

# If nothing to do, exit with temporary disable
completed='svccprop -p config/completed site/first-boot-script-svc:default'
[ "${completed}" = "true" ] && \
    smf_method_exit $SMF_EXIT_TEMP_DISABLE completed "Configuration completed"

# Obtain the active BE name from beadm: The active BE on reboot has an R in
# the third column of 'beadm list' output. Its name is in column one.
bename='beadm list -Hd|nawk -F ' '; ' '$3 ~ /R/ {print $1}'
beadm create ${bename}.orig
echo "Original boot environment saved as ${bename}.orig"

# Place your one-time configuration tasks here

# Record that this script's work is done
svccfg -s site/first-boot-script-svc:default setprop config/completed = true
svcadm refresh site/first-boot-script-svc:default

smf_method_exit $SMF_EXIT_TEMP_DISABLE method_completed "Configuration completed"
```

例 13-2 複数の IP インタフェースを構成する初回ブートスクリプト

この例は、2つの IP インタフェースのアドレスを構成し、デフォルトルートを追加する `first-boot-script.sh` という名前の初回ブートスクリプトを示しています。

```
#!/bin/sh

# Load SMF shell support definitions
. /lib/svc/share/smf_include.sh

# If nothing to do, exit with temporary disable
completed='svccprop -p config/completed site/first-boot-script-svc:default'
[ "${completed}" = "true" ] && \
    smf_method_exit $SMF_EXIT_TEMP_DISABLE completed "Configuration completed"

# Obtain the active BE name from beadm: The active BE on reboot has an R in
# the third column of 'beadm list' output. Its name is in column one.
bename='beadm list -Hd|nawk -F ' '; ' '$3 ~ /R/ {print $1}'
beadm create ${bename}.orig
echo "Original boot environment saved as ${bename}.orig"
```

例 13-2 複数の IP インタフェースを構成する初回ブートスクリプト (続き)

```
# Create and configure addresses on two IP interfaces
/usr/sbin/ipadm create-ip net0
/usr/sbin/ipadm create-ip net1
/usr/sbin/ipadm create-addr -a 10.153.125.222/24 net0
/usr/sbin/ipadm create-addr -a 169.254.182.77/24 net1

# Add a default route with net0 as the gateway
/usr/sbin/route add default 10.153.125.1 -ifp net0

# Record that this script's work is done
svccfg -s site/first-boot-script-svc:default setprop config/completed = true
svcadm refresh site/first-boot-script-svc:default

smf_method_exit $SMF_EXIT_TEMP_DISABLE method_completed "Configuration completed"
```

SMF マニフェストファイルの作成

スクリプトを実行するサービスを定義した SMF マニフェストファイルを作成します。

- サービスの `start` メソッドによって初回ブートスクリプトが実行されます。
- この例では、`multi-user` 依存性を指定して、初回ブート後の起動シーケンスの終わりに初回ブートスクリプトが実行されるようにしています。初回ブートスクリプトの実行内容によっては、このような依存性は必要ない場合もあります。このような依存性を指定しなかった場合は、システムが適切に構成される前にスクリプトが実行される可能性があります。

ヒント-スクリプトの依存性を評価し、その依存性が満たされたあとでスクリプトを実行するようにサービスを作成してください。

- `completed` プロパティは `false` の値で定義されます。

マニフェスト作成ツールの使用

`svcbundle` コマンドを使用して、有効なサービスマニフェストを生成できます。次の例では、デフォルトで、`svcbundle` コマンドによって生成されたマニフェストが一時的なサービスを指定し、`multi-user` 依存性を指定しています。

例 13-3 生成された SMF サービスマニフェスト

次のコマンドでは、211 ページの「初回ブート時に実行するスクリプトの作成」に示したスクリプトの名前が `start-method` の値として指定されています。217 ページ

例 13-3 生成された SMF サービスマニフェスト (続き)

の「スクリプトとサービス用の IPS パッケージの作成」で作成されたパッケージは /opt/site/first-boot-script.sh に first-boot-script.sh スクリプトをインストールするため、スクリプトの名前は /opt/site/first-boot-script.sh として指定されています。

次のコマンドでは、completed プロパティーが、プロパティーグループ名、プロパティー名、プロパティーのタイプ、およびプロパティーの初期値をコロンで区切ったリストで指定されています。

```
$ svcbundle -s service-name=site/first-boot-script-svc \
-s start-method=/opt/site/first-boot-script.sh \
-s instance-property=config:completed:boolean:false \
> first-boot-script-svc-manifest.xml
```

以下に示す生成されたサービスマニフェストでは、初回ブートスクリプト (/opt/site/first-boot-script.sh) は start メソッドの exec 属性の値です。completed プロパティーは、このサービスのデフォルトインスタンス (first-boot-script-svc:default) を定義する instance 要素に指定されています。

```
<?xml version="1.0" ?>
<!DOCTYPE service_bundle
  SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<!--
  Manifest created by svcbundle (2012-Jul-13 16:39:30-0700)
-->
<service_bundle type="manifest" name="site/first-boot-script-svc">
  <service version="1" type="service" name="site/first-boot-script-svc">
    <!--
      The following dependency keeps us from starting until the
      multi-user milestone is reached.
    -->
    <dependency restart_on="none" type="service"
      name="multi_user_dependency" grouping="require_all">
      <service_fmri value="svc:/milestone/multi-user"/>
    </dependency>
    <exec_method timeout_seconds="60" type="method" name="start"
      exec="/opt/site/first-boot-script.sh"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute to stop the service. See smf_method(5) for more
      details.
    -->
    <exec_method timeout_seconds="60" type="method" name="stop"
      exec=":true"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute when the service is refreshed. Services are
      typically refreshed when their properties are changed in the
      SMF repository. See smf_method(5) for more details. It is
      common to retain the value of :true which means that SMF will
      take no action when the service is refreshed. Alternatively,
      you may wish to provide a method to reread the SMF repository
```

例 13-3 生成された SMF サービスマニフェスト (続き)

```

        and act on any configuration changes.
-->
<exec_method timeout_seconds="60" type="method" name="refresh"
  exec=":true"/>
<property_group type="framework" name="startd">
  <propval type="astring" name="duration" value="transient"/>
</property_group>
<instance enabled="true" name="default">
  <property_group type="application" name="config">
    <propval type="boolean" name="completed" value="false"/>
  </property_group>
</instance>
<template>
  <common_name>
    <loctext xml:lang="C">
      <!--
        Replace this comment with a short name for the
        service.
      -->
    </loctext>
  </common_name>
  <description>
    <loctext xml:lang="C">
      <!--
        Replace this comment with a brief description of
        the service
      -->
    </loctext>
  </description>
</template>
</service>
</service_bundle>

```

生成されたマニフェストのカスタマイズ

svcbundle コマンドで生成されたサービスマニフェストは、何も変更せずに要件を満たしている可能性があります。次の例は、サービスマニフェストの変更を示しています。

サービスマニフェストを変更した場合は、svccfg validate コマンドを使用して、マニフェストが引き続き有効であることを確認します。

例 13-4 カスタマイズされた SMF サービスマニフェスト

生成されたサービスマニフェストの次のコピーでは、exec_method のデフォルトのタイムアウト (60 秒) を start メソッドのために増やしています。start メソッドが初回ブートスクリプトを実行するための十分な時間を確保します。

```

<?xml version="1.0" ?>
<!DOCTYPE service_bundle
  SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>

```

例 13-4 カスタマイズされた SMF サービスマニフェスト (続き)

```

<!--
Manifest created by svcbundle (2012-Jul-13 16:39:30-0700)
-->
<service_bundle type="manifest" name="site/first-boot-script-svc">
  <service version="1" type="service" name="site/first-boot-script-svc">
    <!--
      The following dependency keeps us from starting until the
      multi-user milestone is reached.
    -->
    <dependency restart_on="none" type="service"
      name="multi_user_dependency" grouping="require_all">
      <service_fmri value="svc:/milestone/multi-user"/>
    </dependency>
    <!--
      Make sure the start method has adequate time to run the script.
    -->
    <exec_method timeout_seconds="360" type="method" name="start"
      exec="/opt/site/first-boot-script.sh"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute to stop the service. See smf_method(5) for more
      details.
    -->
    <exec_method timeout_seconds="60" type="method" name="stop"
      exec=":true"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute when the service is refreshed. Services are
      typically refreshed when their properties are changed in the
      SMF repository. See smf_method(5) for more details. It is
      common to retain the value of :true which means that SMF will
      take no action when the service is refreshed. Alternatively,
      you may wish to provide a method to reread the SMF repository
      and act on any configuration changes.
    -->
    <exec_method timeout_seconds="60" type="method" name="refresh"
      exec=":true"/>
    <property_group type="framework" name="startd">
      <propval type="astring" name="duration" value="transient"/>
    </property_group>
    <instance enabled="true" name="default">
      <property_group type="application" name="config">
        <propval type="boolean" name="completed" value="false"/>
      </property_group>
    </instance>
    <template>
      <common_name>
        <loctext xml:lang="C">
          <!--
            Replace this comment with a short name for the
            service.
          -->
        </loctext>
      </common_name>
      <description>
        <loctext xml:lang="C">

```


例 13-4 カスタマイズされた SMF サービスマニフェスト (続き)

```

        <!--
            Replace this comment with a brief description of
            the service
        -->
    </loctext>
</description>
</template>
</service>
</service_bundle>

$ svccfg validate first-boot-script-svc-manifest.xml

```

スクリプトとサービス用の IPS パッケージの作成

次の内容の IPS パッケージを作成します。

- 213 ページの「SMF マニフェストファイルの作成」で得られるサービスマニフェストファイル。
- 211 ページの「初回ブート時に実行するスクリプトの作成」で得られる初回ブートスクリプト。
- インストールサーバーなどの別の場所から提供できない、スクリプトで必要なすべてのファイル。

▼ IPS パッケージを作成して公開する方法

- 1 ディレクトリ階層を作成します。

この例では、サービスマニフェストは `/lib/svc/manifest/site` にインストールされ、初回ブートスクリプトは `/opt/site` にインストールされます。

```

$ mkdir -p proto/lib/svc/manifest/site
$ mkdir -p proto/opt/site
$ cp first-boot-script-svc-manifest.xml proto/lib/svc/manifest/site
$ cp first-boot-script.sh proto/opt/site

```

- 2 パッケージマニフェストを作成します。

`first-boot-script.p5m` という名前の次のファイルを作成します。

```

set name=pkg.fmri value=first-boot-script@1.0,5.11-0
set name=pkg.summary value="AI first-boot script"
set name=pkg.description value="Script that runs at first boot after AI installation"
set name=info.classification value=\
    "org.opensolaris.category.2008:System/Administration and Configuration"
file lib/svc/manifest/site/first-boot-script-svc-manifest.xml \
    path=lib/svc/manifest/site/first-boot-script-svc-manifest.xml owner=root \
    group=sys mode=0444

```

```
dir path=opt/site owner=root group=sys mode=0755
file opt/site/first-boot-script.sh path=opt/site/first-boot-script.sh \
owner=root group=sys mode=0555
```

初回ブートスクリプトの実行内容によっては、依存性を指定することが必要な場合もあります。このマニフェストを変更した場合は、新しいマニフェストが正しいことを検証します。警告は無視できます。パッケージの作成方法 (pkgdepend、pkgmogrify、および pkglint コマンドに関する情報を含む) については、『Oracle Solaris 11.1 での Image Packaging System を使用したソフトウェアのパッケージ化および配布』の第 2 章「IPS を使用したソフトウェアのパッケージ化」を参照してください。

3 パッケージのリポジトリを作成します。

この例では、発行元として firstboot を指定して、ローカルディレクトリにリポジトリを作成します。

注-インストール時に AI クライアントがアクセスできるディレクトリにリポジトリを作成してください。

```
$ pkgrepo create firstbootrepo
$ pkgrepo -s firstbootrepo add-publisher firstboot
```

4 パッケージを発行します。

```
$ pkgsend publish -d ./proto -s ./firstbootrepo first-boot-script.p5m
pkg://firstboot/first-boot-script@1.0,5.11-0:20120716T022508Z
PUBLISHED
```

クライアントは、firstbootrepo リポジトリからパッケージをインストールできます。firstbootrepo 起点を持つ firstboot 発行元は、次のセクションに示すように AI マニフェストに定義されます。

5 パッケージが使用可能であることを確認します。

パッケージを一覧表示して、パッケージが使用可能であることを検証します。

```
$ pkg list -g ./firstbootrepo first-boot-script
NAME (PUBLISHER)          VERSION  IFO
first-boot-script (firstboot)  1.0-0   ---
```

6 (オプション)パッケージのインストールをテストします。

-n オプションは、パッケージをインストールしないことを示します。

```
$ pfxec pkg set-publisher -g ./firstbootrepo firstboot
$ pkg publisher
PUBLISHER  TYPE    STATUS P LOCATION
solaris   origin  online F http://http://pkg.oracle.com/solaris/release/
firstboot origin  online F file:///home/user1/firstboot/firstbootrepo/
$ pkg list -af first-boot-script
NAME (PUBLISHER)          VERSION  IFO
first-boot-script (firstboot)  1.0-0   ---
```

```

$ pfexec pkg install -nv first-boot-script
  Packages to install:      1
    Estimated space available: 50.68 GB
Estimated space to be consumed: 64.66 MB
    Create boot environment:      No
Create backup boot environment:  No
    Rebuild boot archive:        No

Changed packages:
firstboot
  first-boot-script
    None -> 1.0,5.11-0:20120716T022508Z
Planning linked: 0/2 done; 1 working: zone:z2
Linked image 'zone:z2' output:
|   Estimated space available: 50.68 GB
|   Estimated space to be consumed: 62.07 MB
|   Rebuild boot archive:      No
|
Planning linked: 1/2 done; 1 working: zone:z1
Linked image 'zone:z1' output:
|   Estimated space available: 50.67 GB
|   Estimated space to be consumed: 62.07 MB
|   Rebuild boot archive:      No

```

次の手順 クライアントシステムが NFS 共有または HTTP を介して新しいリポジトリにアクセスできるようにする手順については、『[Oracle Solaris 11.1 パッケージリポジトリのコピーおよび作成](#)』を参照してください。

AIクライアントへの初回ブートパッケージのインストール

カスタム AI マニフェストファイルを作成して、新しいパッケージ、発行元、およびリポジトリ情報を追加します。

▼ IPS パッケージのインストール方法

- 1 パッケージを AI マニフェストに追加します。

パッケージを AI マニフェストのソフトウェアインストールセクションに追加します。AI マニフェストの XML ファイルをカスタマイズするか、派生マニフェストスクリプトを作成して、これらの要素を追加します。AI マニフェストのカスタマイズについては、[第 10 章「クライアントシステムのプロビジョニング」](#)を参照してください。

1 つ以上の既存の AI マニフェストの内容を取得するには、`installadm export` コマンドを使用します。次の例では、追加する必要がある XML 要素を示しています。

```

<software type="IPS">
  <source>
    <publisher name="solaris">

```

```

    <origin name="http://pkg.oracle.com/solaris/release"/>
  </publisher>
  <publisher name="firstboot">
    <origin name="file:///net/host1/export/firstbootrepo"/>
  </publisher>
</source>
<software_data action="install">
  <name>pkg:/first-boot-script</name>
</software_data>
</software>

```

起点が、AI インストール中にクライアントがアクセスできる URIであることを確認します。クライアントがローカルリポジトリにアクセスできるように、`zfs set sharenfs` を使用してリポジトリをエクスポートします。

- 2 AI インストールサービス内の変更された AI マニフェストを更新します。

`installadm update-manifest` コマンドを使用して、AI マニフェストの内容を、初回ブートスクリプトパッケージが含まれる内容に置き換えます。更新後もマニフェストまたはスクリプトの条件やデフォルトステータスはそのままです。

- 3 クライアントをネットワーク経由でブートします。

クライアントをネットワーク経由でブートし、AI を使用して Oracle Solaris 11 OS とカスタムの `first-boot-script` パッケージをインストールします。インストール後にクライアントがブートされると、そのサービスが実行され、初回ブートスクリプトが実行されます。

初回ブートサービスのテスト

AI インストールをテストする前にサービスをテストするには、テストシステムにパッケージをインストールし、そのテストシステムをリポートするだけです。

```
$ pfexec pkg install first-boot-script
```

```

  Packages to install: 1
  Create boot environment: No
  Create backup boot environment: No

```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
Completed	1/1	2/2	0.0/0.0	0B/s

PHASE	ITEMS
Installing new actions	7/7
Updating package state database	Done
Updating image state	Done
Creating fast lookup database	Done
Reading search index	Done

```
$ pkg list first-boot-script
```

NAME (PUBLISHER)	VERSION	IFO
first-boot-script (firstboot)	1.0-0	i--

```
$ pkg info first-boot-script
```

```
Name: first-boot-script
```

```

Summary: AI first-boot script
Description: Script that runs at first boot after AI installation
Category: System/Administration and Configuration
State: Installed
Publisher: firstboot
Version: 1.0
Build Release: 5.11
Branch: 0
Packaging Date: July 23, 2012 02:50:31 PM
Size: 3.89 kB
FMRI: pkg://firstboot/first-boot-script@1.0,5.11-0:20120723T145031Z

```

テストシステムをリブートします。スクリプトが上に示したように新しいブート環境を作成したら、必ずその新しいブート環境からブートしてください。

スクリプトが /opt/site ディレクトリ内にあり、スクリプトの効果が正しいことを確認します。

サービスの状態を確認します。スクリプトが完了して正常終了した場合は、サービスが無効な状態になっているはずです。

```

$ svcs first-boot-script-svc
STATE          STIME          FMRI
disabled       8:24:16      svc:/site/first-boot-script-svc:default

```

次のいずれかのコマンドを使用して、completed プロパティの値を確認します。

```

$ svcprop first-boot-script-svc:default
config/completed boolean true
$ svcprop -p config/completed first-boot-script-svc:default
true

```

サービスのログファイルを確認する場合は、次のコマンドを使用してログファイルの場所を見つけます。

```

$ svcs -x first-boot-script-svc
svc:/site/first-boot-script-svc:default (?)
State: disabled since July 23, 2012 08:24:16 AM PDT
Reason: Temporarily disabled by service method: "Configuration completed."
See: http://support.oracle.com/msg/SMF-8000-15
See: /var/svc/log/site-first-boot-script-svc:default.log
Impact: This service is not running.

```

ログファイルには次の情報が含まれています。

```

[ Jul 23 08:22:57 Enabled. ]
[ Jul 23 08:24:14 Executing start method ("/opt/site/first-boot-script.sh"). ]
[ Jul 23 08:24:16 Method "start" exited with status 101. ]
[ Jul 23 08:24:16 "start" method requested temporary disable: "Configuration completed"
]
[ Jul 23 08:24:16 Rereading configuration. ]

```

▼ スクリプトまたはサービスを更新する方法

スクリプトまたはサービスマニフェストを変更した場合は、この手順を使用して更新をインストールします。

- 1 更新されたファイルをプロトタイプディレクトリにコピーします。

```
$ cp first-boot-script-svc-manifest.xml proto/lib/svc/manifest/site
$ cp first-boot-script.sh proto/opt/site
```

- 2 パッケージのバージョンを増分します。

パッケージマニフェストで、`pkg.fmri` 属性の値をたとえば次のように変更します。

```
first-boot-script@1.0,5.11-0.1
```

- 3 新しいバージョンを公開します。

パッケージの新しいバージョンをリポジトリに公開します。

```
$ pkgsend publish -d ./proto -s ./firstbootrepo first-boot-script.p5m
pkg://firstboot/first-boot-script@1.0,5.11-0.1:20120723T231948Z
PUBLISHED
```

- 4 パッケージを更新します。

`pkg list -af` コマンドを使用して、新しいバージョンにアクセスできるようにします。場合によっては、`pkg refresh firstboot` コマンドを使用してパッケージリストをリフレッシュする必要があります。 `pkg update` コマンドを使用してパッケージを更新します。

- 5 テストシステムをリブートします。

クライアントシステムのインストール

この章では、AIクライアントのシステム要件を示すとともに、各クライアントを正しいAIインストールサービスに関連付ける方法について説明します。

クライアントのインストール方法

インストールサーバーを設定したときに、クライアントのアーキテクチャーと、インストールを計画している Oracle Solaris OS のバージョンごとに少なくとも1つのインストールサービスを作成しました。各インストールサービスを作成したときに、必要に応じて各種クライアントのカスタマイズされたインストール手順とシステム構成手順を作成しました。自動インストールを開始するには、単にクライアントをブートするだけです。

クライアントをネットワーク経由でブートしたあと、インストールサービスによって提供されるネットイメージ、インストール仕様、およびシステム構成仕様を使ってクライアントのインストールと構成が行われます。

1. 管理者がクライアントをネットワーク経由でブートします。
2. クライアントシステムがDHCPサーバーに接続し、クライアントのネットワーク構成とインストールサーバーの場所を取得します。SPARCクライアントは、必要に応じて、OBPに設定された `network-boot-arguments` 変数を使用してこの情報を取得することもできます。
3. クライアントシステムが次のいずれかのソースからネットイメージを読み込みます。
 - `installadm create-client` コマンドを使ってこのクライアントに割り当てられたインストールサービス
 - このアーキテクチャー用のデフォルトのインストールサービス
4. クライアントシステムが、138 ページの「AI マニフェストの選択」の説明に従って決められた AI マニフェストを使用してそのインストールを完了します。

5. AI マニフェストに `auto_reboot` が設定されている場合、クライアントシステムがリブートします。または、システム管理者がクライアントをリブートします。
6. リブート中に、クライアントシステムが次のいずれかの方法で構成されます。
 - 139 ページの「システム構成プロファイルの選択」の説明に従って決められたシステム構成プロファイルを使用する
 - 対話型システム構成ツールでの管理者の応答を使用する

AI クライアントインストールが完了すると、自動インストールの成功メッセージが画面に表示され、完了メッセージが `/system/volatile/install_log` ログファイルに表示され、そのクライアント上の `svc:/application/auto-installer SMF` サービスが `onLine` 状態になります。

SPARC および x86 クライアントシステムの要件

自動インストールを実行するクライアントシステムは、次の要件を満たしている必要があります。ノートパソコン、デスクトップパソコン、仮想マシン、エンタープライズサーバーなど、次に示す要件を満たすシステムを自動インストールクライアントとして使用できます。

ネットワーク経由で AI インストールを実行する SPARC および x86 クライアントは、次の要件を満たしている必要があります。

メモリー	最小 1G バイト
ディスク容量	最小 13G バイト
ネットワークアクセス	クライアントシステムは、インストール中に次のリソースにアクセスできる必要があります。 <ul style="list-style-type: none"> ■ ネットワーク構成情報を提供する DHCP サーバー ■ AI インストールサーバー ■ クライアントシステムにインストールするパッケージが含まれる IPS リポジトリ

SPARC クライアントシステムは、ネットワーク経由の AI インストールに関する次の追加要件を満たす必要があります。

ファームウェア	SPARC クライアント上のファームウェアは、最新の WAN ブートサポートを含む最新バージョンの OBP (Open Boot PROM) を含むように更新されている必要があります。
WAN ブート	ネットワーク経由で AI インストールを実行する SPARC クライアントは、WAN ブートをサポートしている必要があります。

AI では、ネットワーク経由でブートするために、SPARC クライアントに対する WAN ブートサポートが必要です。 `network-boot-arguments` が `eeeprom` で設定できる有

効な変数であるかどうかを確認することによって、クライアントのOBP (Open Boot PROM) がWAN ブートをサポートしているかどうかを確認できます。

変数 `network-boot-arguments` が表示される場合、またはこのコマンドによって出力 `network-boot-arguments: data not available` が返される場合は、OBP によって WAN ブートがサポートされているため、ネットワーク経由でクライアントをインストールできます。

```
# eeprom | grep network-boot-arguments
network-boot-arguments: data not available
```

このコマンドによって出力が表示されない場合は、WAN ブートがサポートされていないため、ネットワーク経由でクライアントをインストールすることはできません。[第5章「メディアからブートする自動インストール」](#)を参照してください。

```
# eeprom | grep network-boot-arguments
```

インストールクライアントを設定する

インストールサーバー上で、特定のクライアントを特定のインストールサービスに関連付けるには `installadm create-client` コマンドを使用します。

`installadm create-client` コマンドには、次の情報が必要です。

- クライアントのMACアドレス
- クライアントがインストールに使用するインストールサービスの名前

x86 クライアントの場合は、必要に応じて、`installadm create-client` コマンドに `-b` オプションを使用してブートプロパティを指定できます。SPARC クライアントの場合は、OBP の `network-boot-arguments` 変数を使用してブートプロパティを設定できます。

SPARC クライアントの設定

次の例では、MAC アドレス `00:14:4f:a7:65:70` を持つ SPARC クライアントを `solaris11_1-sparc` インストールサービスに関連付けています。

```
$ pfexec installadm create-client -n solaris11_1-sparc -e 00:14:4f:a7:65:70
```

SPARC の `wanboot-cgi` ブートファイルが `create-service` によってすでに構成されているため、DHCP サーバーでの構成は必要ありません。詳細は、[109 ページの「AI インストールサービスの作成」](#)を参照してください。

この `installadm create-client` コマンドの次の結果が `/etc/netboot` ディレクトリに表示されます。

```
lrwxrwxrwx 1 root staff 33 2012-05-09 08:53 0100144FA76570 -> /etc/netboot/solaris11_1-sparc
```

x86 クライアントの設定

次の例では、MACアドレス `0:e0:81:5d:bf:e0` を持つ x86 クライアントを `solaris11_1-i386` インストールサービスに関連付けています。このコマンドによって出力される DHCP 構成を DHCP サーバーに追加する必要があります。この DHCP 構成が行われない場合、クライアントは `solaris11_1-i386` インストールサービスをブートできません。

```
$ pfexec installadm create-client -n solaris11_1-i386 -e 0:e0:81:5d:bf:e0
No local DHCP configuration found. If not already configured, the
following should be added to the DHCP configuration:
  Boot server IP       : 10.80.239.5
  Boot file(s)        :
    bios clients (arch 00:00): 0100E0815DBFE0.bios
    uefi clients (arch 00:07): 0100E0815DBFE0.uefi
```

次の例は、`installadm` によって、Oracle Solaris 11.1 i386 インストールサービスの ISC DHCP 構成に対応する `/etc/inet/dhcpd4.conf` ファイルに、このクライアントのデフォルトの PXE ブートファイルを設定する方法を示しています。

```
host 00E0815DBFE0 {
  hardware ethernet 00:E0:81:5D:BF:E0;
  if option arch = 00:00 {
    filename "0100E0815DBFE0.bios";
  } else if option arch = 00:07 {
    filename "0100E0815DBFE0.uefi";
  }
}
```

この `installadm create-client` コマンドの次の結果が `/etc/netboot` ディレクトリに表示されます。

```
lrwxrwxrwx 1 root staff 47 2012-05-08 17:49 0100E0815DBFE0.uefi -> ./solaris11_1-i386/boot/grub/grub2netx64.efi
lrwxrwxrwx 1 root staff 21 2012-05-08 17:49 0100E0815DBFE0 -> ./0100E0815DBFE0.bios
lrwxrwxrwx 1 root staff 40 2012-05-08 17:49 0100E0815DBFE0.bios -> ./solaris11_1-i386/boot/grub/pxegrub2
-rw-r--r-- 1 root root 1744 2012-05-08 17:49 grub.cfg.0100E0815DBFE0
-rw-r--r-- 1 root root 1212 2012-05-08 17:49 menu.conf.0100E0815DBFE0
```

サービスからのクライアントの削除

インストールサービスからクライアントを削除するには、`installadm delete-client` コマンドを使用します。

```
$ pfexec installadm delete-client macaddr
```

クライアントを関連付けることができるインストールサービスは1つだけなので、サービス名を指定する必要はありません。

クライアントのインストール

インストールを開始するには、クライアントをブートします。このセクションでは、SPARCまたはx86クライアントをブートする方法について説明します。また、このセクションではインストールの進捗状況をリモートから監視する方法についても説明します。

Secure Shell を使用したリモートからのインストールの監視

sshを使用することで、自動インストールクライアントへのネットワークアクセスを有効にできます。このアクセスを使用して、`/system/volatile/install_log` インストールログファイルの進捗状況を監視することにより、進行中のインストールをリモートから監視できます。

特定のインストールサービスのすべてのクライアントでリモートアクセスを有効にするには、インストール構成ファイルでオプション `livessh` を `enable` に設定します。このアクセスが有効になっている場合は、ユーザー名 `jack` とパスワード `jack` を使用してAIクライアントにログインできます。

個々のクライアントで、ブートコマンド行にこのオプションを設定することもできます。

x86クライアントのインストールの監視

x86システムの場合は、次の例に示すように、`create-service` サブコマンドで `-b` オプションを使用して、そのサービスを使用するすべてのクライアントのブートプロパティを設定します。

```
$ pfexec installadm create-service -a i386 -b livessh=enable
```

次の抜粋は、このプロパティが `/etc/netboot/svcname/grub.cfg` ファイルにどのように表示されるかを示しています。

```
$multiboot $kern /platform/i86pc/kernel/amd64/unix -B livessh=enable,...
```

1つのx86クライアントのsshを有効にするには、ブートコマンド行に `livessh` を指定します。手順については、『Oracle Solaris 11.1 システムのブートおよびシャットダウン』の「ブート時に GRUB メニューを編集してカーネル引数を追加する」を参照してください。

SPARC クライアントのインストールの監視

SPARC システムの場合は、`/etc/netboot` ディレクトリ下にマウントされたサービスのネットイメージディレクトリ `/etc/netboot/svcname/system.conf` を介して `system.conf` ファイルにアクセスします。

`system.conf` ファイルでは、オプションは名前と値のペアとして定義されます。次の例では、`livessh` オプションは `enable` に設定されています。

```
$ cat /etc/netboot/solaris11_1-sparc/system.conf
...install_service=solaris11_1-sparc
install_svc_address=$serverIP:5555
livessh=enable
...
```

1つのSPARCクライアントのsshを有効にするには、ブートコマンド行に `livessh` を指定します。次の例は、この引数を指定する2種類の方法を示しています。

```
ok boot net:dhcp - livessh
ok boot net:dhcp - livessh=enable
```

ブートコマンド行での `livessh` の指定は、サービスの `system.conf` ファイルに指定されたすべての設定をオーバーライドします。たとえば、`system.conf` ファイルに `livessh=enable` が指定されている場合に、特定のクライアントで `livessh` を無効にするには、ブートコマンド行で `livessh=disable` を指定します。

```
ok boot net:dhcp - livessh=disable
```

SPARC クライアントのインストール

OBP プロンプトから SPARC クライアントをネットワーク経由でブートします。

- DHCP を使用している場合は、次のネットワークブートコマンドを使用します。

```
ok boot net:dhcp - install
```

- DHCP を使用していない場合は、次のコマンドを使用して OBP の `network-boot-arguments` 変数を設定します。この変数は、OBP に持続的に設定されます。

```
ok setenv network-boot-arguments host-ip=client-ip,
router-ip=router-ip,subnet-mask=subnet-mask,hostname=hostname,
file=wanboot-cgi-file
```

次に、次のコマンドを使用してクライアントをネットワーク経由でブートします。

```
ok boot net - install
```

注 - `network-boot-arguments` 変数を使用するときは、SPARCクライアントにDNSの構成情報がありません。このクライアントで使用されているAIマニフェストに、IPSパッケージリポジトリの場所やマニフェスト内のその他のURIとして、ホスト名ではなくIPアドレスが指定されていることを確認してください。

SPARCクライアントのAIブート中に、次のイベントが発生します。

1. クライアントがブートし、そのネットワーク構成と `wanboot-cgi` ファイルの場所をDHCPサーバーまたはそのOBPに設定された `network-boot-arguments` 変数から取得します。
2. `wanboot-cgi` プログラムが `wanboot.conf` を読み取り、WANブートバイナリの場所をクライアントに送信します。
3. WANブートバイナリがHTTPを使用してダウンロードされ、クライアントがWANブートプログラムをブートします。
4. WANブートが `boot_archive` ファイルを取得し、Oracle Solaris OSがブートされます。
5. イメージのアーカイブである `solaris.zlib` と `solarismisc.zlib` が、HTTPを使用してダウンロードされます。
6. AIマニフェストとシステム構成プロファイルが、mDNS検索または `system.conf` ファイルのどちらかで指定されたAIインストールサービスからダウンロードされます。
7. AIインストールプログラムがAIマニフェストを使用して呼び出され、クライアントへのOracle Solaris OSのインストールを実行します。

x86クライアントのインストール

次のいずれかの方法を使用してネットワークからブートすることによって、x86クライアントのインストールを開始します。

- 適切なファンクションキーを押します。たとえば、一部のシステムでは、ネットワークからブートするためにF12キーを使用します。
- BIOS内のブート順序を変更します。

クライアントがブートしたら、ブートするネットワークデバイスを選択します。

x86 クライアントの AI ブート中に、次のイベントが発生します。

1. クライアントがブートして IP アドレスを取得し、ブートファイルが DHCP サーバーによって提供された場所からダウンロードされます。
2. ブートファイルが読み込まれ、GRUB メニューファイルを読み取ります。
3. ユーザーが GRUB メニューから 2 番目のオプションである「Oracle Solaris 11.1 Automated Install」を選択します。
4. ブートファイルがブートアーカイブファイルを取得し、Oracle Solaris OS が TFTP を使用してブートされます。
5. GRUB メニューによって提供される HTTP を使用して、ネットイメージのアーカイブ `solaris.zlib` および `solarismisc.zlib` がダウンロードされます。
6. AI マニフェストとシステム構成プロファイルが、mDNS 検索またはブートされた GRUB メニューエントリで指定された AI インストールサービスからダウンロードされます。
7. AI マニフェストで AI インストールプログラムが呼び出され、インストールが実行されます。

システムの PXE ブートが正常に行われると、次のメッセージが少しの間表示されてから、GRUB メニューが表示されます。

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation
```

```
CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49
GATEWAY: 10.6.68.1
```

GRUB メニューに 2 つのメニューエントリが表示されます。自動インストールを開始するには、2 番目のエントリを選択します。

```
Oracle Solaris 11.1 Text Installer and command line
Oracle Solaris 11.1 Automated Install
```

デフォルトの GRUB メニューエントリ「テキストインストーラおよびコマンド行」では、イメージはブートされますが、手入力なしの自動インストールは開始されません。自動インストールを開始するには、GRUB メニュー内の 2 番目のエントリ「自動インストール」を選択します。1 番目のメニューエントリを選択した場合は、クライアントがブートされると、[243 ページの「インストールを開始しないでブートしたあとにインストールを開始する」](#)に示すように、メニューが表示されます。このメニューを使用して、システムを確認するかインストールします。

クライアントインストールのメッセージ

次のメッセージは、SPARC と x86 の両方のインストールに共通しています。

自動インストールの開始メッセージ

クライアントがブートに成功し、インストールファイルをダウンロードできる場合は、次のメッセージが表示されます。

```
Automated Installation started
The progress of the Automated Installation will be output to the console
Detailed logging is in the logfile at /system/volatile/install_log
Press RETURN to get a login prompt at any time.
```

パスワード `solaris` を使用して `root` としてログインし、`/system/volatile/install_log` 内のインストールメッセージを監視できます。

自動インストールの成功メッセージ

次のメッセージが表示された場合は、インストールが成功しました。

```
Automated Installation finished successfully
The system can be rebooted now
Please refer to the /system/volatile/install_log file for details
After reboot it will be located at /var/log/install/install_log
```

AI マニフェストで自動リブートを設定している場合は、この時点でシステムがリブートします。インストールが正常に完了したあとの自動リブートを指定するには、`<ai_instance>` タグの `auto_reboot` 属性を `true` に設定します。デフォルト値は `false` です。インストールが正常に完了したあと、クライアントは自動的にリブートしません。

◆◆◆ 15

第 15 章

自動インストールのトラブルシューティング

この章では、考えられるいくつかの障害と回復の方法について説明します。

クライアントインストールが失敗する

このセクションでは、クライアントインストールに失敗した場合に実行するアクションに関する推奨事項を示します。

インストールログおよび手順の確認

クライアントシステムへのインストールに失敗した場合は、ログを `/system/volatile/install_log` で見つけることができます。

このクライアントに使用された AI マニフェストは、`/system/volatile/ai.xml` 内にあります。このクライアントに使用されたシステム構成プロファイルは、`/system/volatile/profile/*` 内にあります。

DNS の確認

空でない `/etc/resolv.conf` ファイルが存在することを検証することで、クライアントで DNS が構成されているかどうかを確認します。

`/etc/resolv.conf` が存在しないか、空である場合は、DHCP サーバーが DNS サーバー情報をクライアントに提供していることを確認します。

```
# /sbin/dhcpinfo DNSserv
```

このコマンドが何も返さない場合、DHCP サーバーは DNS サーバー情報をクライアントに提供するように設定されていません。DHCP 管理者に問い合わせるこの問題を修正してください。

/etc/resolv.conf ファイルが存在していて、正しく構成されている場合は、考えられる次の問題がないか確認して、システム管理者に解決方法を問い合わせてください。

- DNS サーバーが IPS リポジトリのサーバー名を解決していない可能性があります。
- DNS サーバーに到達するデフォルトルートが存在しません。

クライアントブートエラーの確認

クライアントシステムのブート時に発生するエラーに関する次の詳細情報を確認してください。

- 234 ページの「SPARC のネットワークブートエラーと考えられるエラー原因」
- 237 ページの「x86 のネットワークブートエラーと考えられるエラー原因」
- 239 ページの「SPARC および x86 のエラーメッセージ」

SPARC のネットワークブートエラーと考えられるエラー原因

このセクションでは、ネットワーク経由で SPARC クライアントをブートしたときに起こり得るエラーまたは問題と、考えられる原因について説明します。

- 234 ページの「BOOTP/DHCP 応答を待機してタイムアウトになった」
- 235 ページの「ブートの読み込みに失敗」
- 235 ページの「内部サーバーのエラーまたは WAN ブートの警告」
- 236 ページの「エラーメッセージ 403: Forbidden または 404 Not Found」
- 236 ページの「自動インストーラが無効になっている」

BOOTP/DHCP 応答を待機してタイムアウトになった

DHCP サーバーが SPARC クライアントの要求に応答しない場合、次のメッセージが表示されます。

```
...
OpenBoot 4.23.4, 8184 MB memory available, Serial #69329298.
Ethernet address 0:14:4f:21:e1:92, Host ID: 8421e192.
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
Timed out waiting for BOOTP/DHCP reply
Timed out waiting for BOOTP/DHCP reply
Timed out waiting for BOOTP/DHCP reply
Timed out waiting for BOOTP/DHCP reply
```

タイムアウトのメッセージは、クライアントが DHCP リクエストを送信し、そのリクエストに対する応答がないことを示しています。このエラーは、DHCP 構成問題によって発生している可能性があります。DHCP サーバーでクライアントが正しく構成されているかどうかを確認してください。

ブートの読み込みに失敗

AIクライアントが `boot_archive` のダウンロードを開始したあと、「Boot load failed」エラーが表示されてダウンロードに失敗した場合、クライアント DHCP 情報が正しく構成されていないことを示しています。

```
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
HTTP: Bad Response: 500 Internal Server Error
Evaluating:

Boot load failed
```

このエラーは、別の DHCP サーバーがクライアントに応答している場合に発生する可能性があります。このクライアントの DHCP 構成を確認してください。構成が正しいと思われる場合は、サブネットに別の DHCP サーバーがあるかどうかを確認してください。

内部サーバーのエラーまたは WAN ブートの警告

AIクライアントが、ブートアーカイブのダウンロードを開始するための IP アドレスと始値を取得したあと、クライアントが `boot_archive` を検出またはダウンロードできない場合があります。

- クライアントが `boot_archive` を検出できない場合は、次のエラーが表示されません。

```
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 366 of 366 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Tue Aug 5 20:46:43 wanboot alert: miniinfo: Request returned code 500
Tue Aug 5 20:46:44 wanboot alert: Internal Server Error \
(root filesystem image missing)
```

- AIクライアントが `boot_archive` ファイルを検出しても、そのファイルにアクセスできない場合は、次のエラーが表示されます。

```
Rebooting with command: boot net:dhcp - install
Boot device: /pci@7c0/pci@0/network@4:dhcp File and args:
1000 Mbps FDX Link up
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 366 of 366 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Tue Aug 5 20:53:02 wanboot alert: miniroot: Request returned code 403
Tue Aug 5 20:53:03 wanboot alert: Forbidden
```

これらの両方の問題を解決するには、このクライアント用に構成された boot_archive ファイルを修正します。boot_archive のパス名とアクセス権を、\$IMAGE/boot/boot_archive で確認します。

エラーメッセージ 403:Forbidden または 404 Not Found

「ERROR 403:Forbidden」および「ERROR 404:Not Found」メッセージは、AIクライアントが boot_archive を正常にダウンロードして Oracle Solaris カーネルをブートしたあと、イメージアーカイブの1つを取得できなかった場合に表示されます。エラーメッセージには、どのファイルが問題を引き起こしているのかが表示されます。たとえば、次の SPARC クライアントでの出力では、solaris.zlib ファイルが存在しないか、指定した場所でアクセスできません。

```
<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 368 of 368 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Thu Jul  5 18:57:36 wanboot progress: miniroot: Read 235737 of 235737 kB (100%)
Thu Jul  5 18:57:36 wanboot info: miniroot: Download complete
SunOS Release 5.11 Version 11.1 64-bit
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2012-07-05 18:52:30-- http://10.134.125.136:5555/export/auto_install/11_1_sparc/solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2012-07-05 18:52:30 ERROR 404: Not Found.

Could not obtain http://10.134.125.136:5555/export/auto_install/11_1_sparc/solaris.zlib from install server
Please verify that the install server is correctly configured and reachable from the client
```

この問題は、次のいずれかの状態によって引き起こされる場合があります。

- WAN ブートで構成されたイメージパスが正しくない。
- イメージパスが存在しないか不完全である。
- アクセス権の問題により、アクセスが拒否される。

DHCP の構成、または installadm create-service を実行したときに指定したネットイメージの内容を確認してください。WAN ブート構成を確認してください。

自動インストーラが無効になっている

Oracle Solaris OS をクライアントシステムにインストールする場合は、インストールを開始するためにブートするときに install 引数を含める必要があります。

```
ok boot net:dhcp - install
```

install ブート引数なしでブートすると、SPARC クライアントで自動インストーラのブートイメージがブートされますが、インストールは開始されません。この時点か

ら自動インストールを開始する方法の手順については、243 ページの「インストールを開始しないでブートしたあとにインストールを開始する」を参照してください。

x86 のネットワークブートエラーと考えられるエラー原因

このセクションでは、ネットワーク経由で x86 クライアントをブートしたときに起こり得るエラーまたは問題と、考えられる原因について説明します。

- 237 ページの「DHCP またはプロキシ DHCP の応答が受信されなかった」
- 237 ページの「TFTP エラー、または GATEWAY メッセージ表示後のシステムのハングアップ」
- 238 ページの「GRUB メニューのエントリが選択されたあとのシステムのハングアップ」
- 238 ページの「送信した HTTP 要求の結果が 403 Forbidden または 404 Not Found になる」
- 239 ページの「自動インストーラが無効になっている」

DHCP またはプロキシ DHCP の応答が受信されなかった

DHCP サーバーが x86 クライアントの要求に対して応答しない場合、次のメッセージが表示されます。

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation
```

```
CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
DHCP..... No DHCP or ProxyDHCP offers were received
PXE-MOF: Exiting Intel Boot Agent
```

タイムアウトメッセージは、クライアントが DHCP 要求を送信し、応答を受信していないことを示しています。この問題は、DHCP 構成のエラーが原因である場合があります。DHCP サーバーでクライアントが正しく構成されているかどうかを確認してください。

TFTP エラー、または GATEWAY メッセージ表示後のシステムのハングアップ

DHCP サーバーは、最初のブートプログラムの IP アドレスと場所を DHCP 応答の一部として提供します。

- ブートプログラムが存在しない場合は、AI クライアントのブートを続行できません。次のメッセージが表示されます。

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation
```

```
CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49
GATEWAY: 10.6.68.1
```

```
TFTP.  
PXE-T02: Access Violation  
PXE-E3C: TFTP Error - Access violation  
PXE-MOF: Exiting Intel Boot Agent
```

- ブートプログラムが存在してもプログラムが正しくない場合、AIクライアントはこのメッセージを表示したあとにハングアップします。

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)  
Copyright(C) 1997-2007, Intel Corporation
```

```
CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400  
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49  
GATEWAY: 10.6.68.1
```

GRUB メニューのエントリが選択されたあとのシステムのハングアップ

クライアントが最初のブートを実行できてもカーネルをブートできない場合、システムは、GRUB メニューからエントリを選択したあとにハングアップします。

インストールサーバーで、このクライアントの `grub.cfg` ファイルまたは `menu.lst` ファイルが有効なブートアーカイブを指しているかどうかを確認してください。サーバー上のイメージのブートディレクトリは、`installadm list` で表示されたイメージパスに対する `df -k` のこの抜粋例に示すように、`/etc/netboot` ディレクトリの下にループバックマウントされます。

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/export/auto_install/solaris11_1-i386					
	92052473	36629085	55423388	40%	/etc/netboot/default-i386
/export/auto_install/solaris11_1-i386					
	92052473	36629085	55423388	40%	/etc/netboot/solaris11_1-i386

送信した HTTP 要求の結果が 403 Forbidden または 404 Not Found になる

インストールサーバー上で、`/etc/netboot` 下の `grub.cfg` ファイルまたは `menu.lst` ファイルで指定された場所にあるインストールプログラムのいずれかがアクセス不可能であるか、またはその場所に存在しない場合、クライアントはブートできますが、そのファイルをダウンロードできません。エラーメッセージには、どのファイルが問題を引き起こしているのかが表示されます。たとえば、次の x86 クライアント上の出力では、`solaris.zlib` ファイルが指定の場所に存在しません。

```
SunOS Release 5.11 Version 11.1 64-bit  
Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.  
Remounting root read/write  
Probing for device nodes ...  
Preparing network image for use  
Downloading solaris.zlib  
--2012-07-18 20:02:26-- http://10.134.125.136:5555/export/auto_install/solaris11_1-i386/solaris.zlib  
Connecting to 10.134.125.136:5555... connected.  
HTTP request sent, awaiting response... 404 Not Found  
2012-07-18 20:02:26 ERROR 404: Not Found.
```

```
Could not obtain http://10.134.125.136:5555/export/auto_install/solaris11_1-i386/solaris.zlib from install server
Please verify that the install server is correctly configured and reachable from the client
```

```
Requesting System Maintenance Mode
(See /lib/svc/share/README for more information.)
Console login service(s) cannot run
```

installadm create-service コマンドを実行したときに指定したターゲットディレクトリの内容を確認してください。

自動インストーラが無効になっている

Oracle Solaris OS を x86 クライアントシステムにインストールするときに、ネットワーク経由でブートするインストールでは、GRUB ブートメニューの 2 番目のエントリを選択して自動インストールを開始する必要があります。通常、メニューのエントリは次のように表示されます。

```
Oracle Solaris 11.1 Text Installer and command line
Oracle Solaris 11.1 Automated Install
```

最初の GRUB メニューエントリを選択した場合、またはプロンプトがタイムアウトした場合は、システムで自動インストールのブートイメージがブートされますが、インストールは開始されません。この時点から自動インストールを開始する方法の手順については、[243 ページの「インストールを開始しないでブートしたあとにインストールを開始する」](#)を参照してください。

SPARC および x86 のエラーメッセージ

次のエラーは、SPARC と x86 の両方のインストールに共通しています。

- [239 ページの「自動インストール失敗メッセージ」](#)
- [239 ページの「有効なパッケージサーバーに接続不能」](#)
- [241 ページの「パッケージが見つからない」](#)

自動インストール失敗メッセージ

インストール中に障害が発生した場合、次のメッセージが表示されます。

```
21:43:34 Automated Installation Failed. See install log at /system/volatile/install_log
Automated Installation failed
Please refer to the /system/volatile/install_log file for details
Jul 6 21:43:34 solaris svc.startd[9]: application/auto-installer:default failed fatally:
transitioned to maintenance (see 'svcs -xv' for details)
```

有効なパッケージサーバーに接続不能

インストールクライアントは、Oracle Solaris OS をインストールするために、AI マニフェストで定義された IPS パッケージリポジトリに到達する必要があります。クライ

アントがそのパッケージリポジトリにアクセスできない場合は、インストールに失敗し、`application/auto-installer` サービスが保守に移行します。次の出力は、コンソールに表示される内容の例です。

```
15:54:46 Creating IPS image
15:54:46 Error occurred during execution of 'generated-transfer-1341-1' checkpoint.
15:54:47 Failed Checkpoints:
15:54:47     generated-transfer-1341-1
15:54:47 Checkpoint execution error:
15:54:47     Framework error: code: 6 reason: Couldn't resolve host 'pkg.example.com'
15:54:47     URL: 'http://pkg.example.com/solaris/release/versions/0/'.
15:54:47 Automated Installation Failed. See install log at /system/volatile/install_log
Automated Installation failed
Please refer to the /system/volatile/install_log file for details
Aug 31 15:54:47 line2-v445 svc.startd[8]: application/auto-installer:default failed fatally:
transitioned to maintenance (see 'svcs -xv' for details)
...
SUNW-MSG-ID: SMF-8000-YX, TYPE: defect, VER: 1, SEVERITY: major
EVENT-TIME: Wed Aug 31 15:54:47 UTC 2011
PLATFORM: SUNW,Sun-Fire-V445, CSN: -, HOSTNAME: line2-v445
SOURCE: software-diagnosis, REV: 0.1
EVENT-ID: c8a5b809-ece4-4399-9646-d8c64d78aac7
DESC: A service failed - a start, stop or refresh method failed.
AUTO-RESPONSE: The service has been placed into the maintenance state.
IMPACT: svc:/application/auto-installer:default is unavailable.
REC-ACTION: Run 'svcs -xv svc:/application/auto-installer:default' to determine the generic reason
why the service failed, the location of any logfiles, and a list of other services impacted. Please
refer to the associated reference document at http://support.oracle.com/msg/SMF-8000-YX for the latest service
procedures and policies regarding this diagnosis.
```

`/system/volatile/install_log` ファイルで次のようなメッセージを確認してください。

```
TransportFailures: Framework error: code: 6 reason: Couldn't resolve host
'pkg.example.com'
URL: 'http://pkg.example.com/solaris/versions/0/'
```

```
TransportFailures: Framework error: code: 7 reason: Failed connect to
pkg.example.com:80; Connection refused
URL: 'http://pkg.example.com/solaris/versions/0/'
```

```
TransportFailures: http protocol error: code: 404 reason: Not Found
URL: 'http://pkg.oracle.com/mysolaris/versions/0/'
```

表示されるメッセージに応じて、次の考えられる対処法を試してください。

- ping を使用するなどして、失敗したクライアントシステムからパッケージサーバーへの接続を試みます。
- DNS を使用している場合は、DNS が AI クライアント上で正しく構成されているかどうかを確認します。233 ページの「DNS の確認」を参照してください。

- ローカルリポジトリを使用している場合は、すべてのクライアントがそのリポジトリにアクセスできるようにしてあるかどうかを確認します。『Oracle Solaris 11.1 パッケージリポジトリのコピーおよび作成』の第3章「リポジトリへのアクセスの提供」を参照してください。
- AI マニフェスト内の URI に誤字がないことを確認します。
- 次のようなコマンドを使用して、パッケージリポジトリが有効かどうかを確認します。

```
$ pkg list -g http://pkg.example.com/solaris/ entire
```

カタログをリフレッシュしたり、索引を再作成したりすることが必要な場合もあります。

パッケージが見つからない

AI マニフェストで指定されたパッケージのいずれかが、IPS リポジトリで見つからない場合、ディスクにパッケージをインストールする前にインストーラが失敗します。次の例では、インストーラがパッケージ `mypkg` を IPS リポジトリで見つけることができませんでした。次の出力は、コンソールに表示される内容の例です。

```
14:04:02 Failed Checkpoints:
14:04:02
14:04:02     generated-transfer-1230-1
14:04:02
14:04:02 Checkpoint execution error:
14:04:02
14:04:02     The following pattern(s) did not match any allowable packages. Try
14:04:02     using a different matching pattern, or refreshing publisher information:
14:04:02
14:04:02         pkg:/mypkg
14:04:02
14:04:02 Automated Installation Failed. See install log at /system/volatile/install_log
```

次の出力は、`/system/volatile/install_log` ログファイルの一部分の例です。

```
PlanCreationException: The following pattern(s) did not match any allowable packages.
Try using a different matching pattern, or refreshing publisher information:

pkg:/mypkg
```

問題のパッケージが有効なパッケージかどうかを確認してください。別の IPS リポジトリからこのパッケージを入手できる場合は、`source` 要素に `publisher` 要素をもう1つ追加して、その IPS リポジトリを AI マニフェストに追加します。

インストールを開始しないでインストール環境をブートする

自動インストールを開始しないでインストール環境をブートするには、次のいずれかの方法を使用します。クライアントがブートされると、243 ページの「インストールを開始しないでブートしたあとにインストールを開始する」に示すように、メニューが表示されます。このメニューを使用して、システムを確認するかインストールします。

ネットワーク経由での SPARC クライアントのブート

自動インストールを開始しないで SPARC クライアントをネットワーク経由でブートするには、次のコマンドを使用します。

```
ok boot net:dhcp
```

install フラグをブート引数として指定しないでください。

メディアからの SPARC クライアントのブート

インストールを開始しないで SPARC クライアントをメディアからブートするには、次のコマンドを使用します。

```
ok boot cdrom
```

install フラグをブート引数として指定しないでください。

ネットワーク経由での x86 クライアントのブート

ネットワーク経由でブートする x86 インストールの場合、次の GRUB メニューが表示されます。

```
Oracle Solaris 11.1 Text Installer and command line
Oracle Solaris 11.1 Automated Install
```

デフォルトのエントリ「テキストインストーラおよびコマンド行」では、イメージはブートされますが、手入力なしの自動インストールは開始されません。

そのエントリのカーネル行に `install=true` ブートプロパティが指定されていないことを確認してください。

メディアからの x86 クライアントのブート

メディアから x86 システムをブートしたあと、インストールを開始しない場合は、GRUB メニューを編集して、ブートするエントリのカーネル行から `install=true` ブートプロパティを削除します。

一般に、x86 インストールでは、ブート元の GRUB エントリのカーネル行に `install=true` ブートプロパティを指定した場合は、インストールが自動的に開始されます。自動インストールを開始せずに x86 ベースのシステムをブートする場合は、GRUB ブートエントリに `install=true` ブートプロパティが指定されていないことを確認します。このプロパティが指定されている場合は、『Oracle Solaris 11.1

システムのブートおよびシャットダウン』の「ブート時に GRUB メニューを編集してカーネル引数を追加する」の説明に従ってブートエントリを編集し、このプロパティを削除します。

インストールを開始しないでブートしたあとにインストールを開始する

インストールを開始しないブートオプションを選択した場合は、次のメニューが表示されます。

```
1 Install Oracle Solaris
2 Install Additional Drivers
3 Shell
4 Terminal type (currently sun)
5 Reboot
```

Please enter a number [1]:

オプション 3 を選択してシェルを開きます。

次のコマンドを使用して、自動インストールを開始します。

```
$ svcadm enable manifest-locator:default
$ svcadm enable svc:/application/auto-installer:default
```


パート IV

関連タスクの実行



Oracle Configuration Manager の操作

この章では、Oracle Configuration Manager の概要に加えて、Oracle Solaris リリースを実行しているシステム上でこのサービスを使用する手順について説明します。この章の内容は次のとおりです。

- 247 ページの「[Oracle Configuration Manager の概要](#)」
- 249 ページの「[Oracle Configuration Manager の中央コレクタについて](#)」
- 250 ページの「[Oracle Configuration Manager の管理\(タスク\)](#)」

Oracle Configuration Manager の概要

Oracle Configuration Manager は、システムの構成情報を収集し、それを Oracle リポジトリにアップロードするために使用します。この情報のコレクタは、サーバー上のすべての製品に関する情報を収集する中央コレクタとして構成することも、個々の収集サイト内の情報を収集するように構成することもできます。詳細は、[249 ページの「Oracle Configuration Manager の中央コレクタについて」](#)を参照してください。

顧客サポート担当者はこの情報を利用して、より適切なサービスを提供できません。Oracle Configuration Manager を使用する利点のいくつかを次に示します。

- サポート問題の解決時間を短縮します
- 予防的な問題回避を実現します
- ベストプラクティスや Oracle ナレッジベースへのアクセスが向上します
- 顧客のビジネスニーズへの理解を高め、一貫した対応とサービスを提供します

Oracle Configuration Manager は、接続または切断のいずれかのモードで実行できます。切断モードは、システムがインターネットに接続しておらず、Oracle Support Hub を構成できない場合にのみ必要です。このモードでは、構成情報を手動で収集し、サービスリクエストのために Oracle に情報をアップロードできます。

接続モードでは、次のような複数のネットワーク構成で Oracle Configuration Manager を実行できます。

- システムをインターネットに直接接続できます。
- プロキシサーバー経由でシステムをインターネットに接続できます。
- システムはインターネットに直接アクセスできませんが、Oracle Support Hub 経由でインターネットに接続するイントラネットプロキシサーバーにアクセスできます。
- システムはインターネットに直接アクセスできませんが、プロキシサーバー経由でインターネットに接続する Oracle Support Hub にアクセスできます。

Oracle Configuration Manager の設定および構成の詳細は、『[Oracle Configuration Manager インストレーションおよび管理ガイド](#)』を参照してください。このドキュメントの残りでは、Oracle Configuration Manager に関連する Oracle Solaris タスクに焦点を当てます。

注 - プロキシまたは Oracle Support Hub を使用するように Oracle Configuration Manager を構成するには、configCCR コマンドを対話型モードで実行する必要があります。詳細は、『[Oracle Support Hub](#)』を参照してください。

Oracle Solaris 11 のインストール中、ソフトウェアは Oracle リポジトリへの匿名接続を設定しようとします。成功した場合は、情報の提供を一切求められることなく、この接続によりインストールプロセスを進めることができます。可能であれば、登録またはネットワーク構成の変更は、システムが完全にインストールされたあとで行うようにしてください。匿名で読み込まれたデータはどの組織にも結び付けられません。ソフトウェアが Oracle リポジトリに接続できなかった場合は、システムを手動で登録してから、Oracle Configuration Manager サービスを有効化できます。

Oracle Configuration Manager の中央コレクタについて

Oracle Solaris オペレーティングシステムの一部としてインストールされた Oracle Configuration Manager コレクタは、中央コレクタとして構成され、指定されません。Oracle Configuration Manager コレクタの利点 (パーソナライズされたサポートエクスペリエンス、サポート問題の迅速な解決、予防的な問題回避など) を得るには、個々の Oracle インストールの構成データを収集してアップロードする必要があります。これは、通常、Oracle ホームにインストールされたコレクタのタスクです。しかし、Oracle ホームのコレクタが構成されていない場合や、切断されたままになっている場合もあります。中央コレクタの目的は、これらの Oracle ホームを収集し、それらを独自の My Oracle Support (MOS) 資格でアップロードすることです。中央コレクタには次のような特徴があります。

- 中央コレクタは次を収集します。
 - 自身が配置されている Oracle ホーム
 - 構成済みのコレクタが存在しないホスト上の Oracle ホーム
 - コレクタが切断モードになっている Oracle ホーム
 - コレクタが登録を認証している Oracle ホーム

Oracle ホームのコレクタが `ORACLE_CONFIG_HOME` の指定を使用して構成されている場合、中央コレクタはそのホームを収集しません。

- コレクタインストールを中央コレクタとして指定するには、`root` の役割を使用して、`setupCCR` および `configCCR` コマンドに `-c` オプションを指定します。その後、`-c` オプションを指定せずに `configCCR` コマンドを実行すると、コレクタから中央コレクタの指定が破棄されます。`-c` オプションを指定して `setupCCR` および `configCCR` コマンドを実行すると、コレクタが中央コレクタとして指定されます。Oracle Solaris オペレーティングシステムの一部としてインストールされたコレクタは、`root` アクセス権を使用してインストールされるため、ホストの中央コレクタとして動作します。
- Oracle Universal Installer の中央インベントリは、中央コレクタが収集する Oracle ホームの候補のセットを取得するソースです。中央インベントリは、ドキュメントで説明されているように、インストーラによって検索されます。Oracle Solaris オペレーティングシステムのインストーラ中央インベントリポイントのデフォルトの場所は、`/var/opt/oracle/oraInst.loc` です。Oracle インストールのインベントリを別の場所に配置するように選択すると、中央インベントリはそれを検索および収集できません。
- このリリースでは、Oracle Solaris OS の構成情報のほかに、Oracle WebLogic を使用する Oracle Database および Oracle Fusion Middleware ベースの製品のみが中央コレクタによって収集されます。
- 中央コレクタが Oracle ホームから収集した構成データはすべて、中央コレクタの My Oracle Support 資格を使用してアップロードされます。

Oracle Configuration Manager の管理 (タスク)

次のタスクマップには、Oracle Solaris システム上での Oracle Configuration Manager の使用に関連する複数の手順が含まれています。

タスク	説明	手順
Oracle Configuration Manager サービスを有効にします。	構成変更を行なったあとで、Oracle Configuration Manager サービスを有効にします。	250 ページの「Oracle Configuration Manager サービスを有効にする方法」
Oracle Configuration Manager サービスを無効にします。	重大な構成変更を行う前に、Oracle Configuration Manager サービスを無効にします。	250 ページの「Oracle Configuration Manager サービスを無効にする方法」
システムを Oracle リポジトリに手動で登録します。	登録資格を変更します。	251 ページの「システムを Oracle リポジトリに手動で登録する方法」
データ収集時間を変更します。	データ収集の頻度と時間を再設定します。	251 ページの「Oracle Configuration Manager のデータ収集の時間と頻度を変更する方法」

▼ Oracle Configuration Manager サービスを有効にする方法

- 1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 Oracle Configuration Manager サービスを有効にします。

```
# svcadm enable system/ocm
```

▼ Oracle Configuration Manager サービスを無効にする方法

- 1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 Oracle Configuration Manager サービスを無効にします。

```
# svcadm disable system/ocm
```



注意 - Oracle Solaris システム上で `emCCR stop` コマンドを実行しないでください。サービスに対するすべての変更は、Oracle Solaris のサービス管理機能 (SMF) を使用して行う必要があります。

▼ システムを **Oracle** リポジトリに手動で登録する方法

- 1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

- 2 ユーザー登録を変更します。

```
# configCCR
```

Oracle Configuration Manager ソフトウェアにより、電子メールアドレスとパスワードの指定を求められます。お使いの My Oracle サポート ID に関連付けられた電子メールアドレスを使用することをお勧めします。

可能な場合、システムは登録サーバーと直接通信します。そうでない場合は、Oracle Support Hub の URL の指定を求められます。サイトで URL が使用可能な場合は、それをここに指定します。Oracle Support Hub の URL を指定しなかった場合、または登録サーバーとまだ通信できない場合は、ネットワークプロキシの指定を求められます。

登録が完了すると、データ収集が開始されます。

参照 configCCR コマンドの詳細は、configCCR(1M) のマニュアルページまたは『Oracle Configuration Manager インストールおよび管理ガイド』を参照してください。configCCR コマンドを使用した対話型セッションの完全な例については、「configCCR」のページを参照してください。

▼ Oracle Configuration Manager のデータ収集の時間と頻度を変更する方法

- 1 管理者になります。

詳細は、『Oracle Solaris 11.1 の管理: セキュリティーサービス』の「割り当てられている管理権限を使用する方法」を参照してください。

2 データ収集の頻度を再設定します。

この例では、データ収集時間を毎週月曜日の午前 6:00 に再設定します。

```
# emCCR set collection_interval=FREQ=WEEKLY\; BYDAY=MON\; BYHOUR=6
```

参照 emCCR コマンドの詳細は、emCCR(1M)のマニュアルページまたは『[Oracle Configuration Manager インストールおよび管理ガイド](#)』を参照してください。

デバイスドライバユーティリティの使用

Oracle デバイスドライバユーティリティ (DDU) は、インストール済みのシステムで検出されたデバイスが最新のリリースによってサポートされているかどうかを報告します。

デバイスドライバユーティリティの概要

デバイスドライバユーティリティは、インストール済みのシステム上のデバイスとそれらのデバイスを管理するドライバに関する情報を提供します。DDU は、システムで検出されたすべてのデバイスに対するドライバが現在ブートしているオペレーティングシステムに存在するかどうかを報告します。デバイスにドライバが関連付けられていない場合、デバイスドライバユーティリティは、インストールするドライバパッケージを推奨します。

デバイスドライバユーティリティを使用すると、使用しているシステムの情報を HCL (<http://www.oracle.com/webfolder/technetwork/hcl/index.html>) に送信することもできます。情報を送信すると、使用しているシステムとそのコンポーネントが「Reported to Work」として HCL に掲載されます。

このセクションでは、次のタスクについて説明します。

- 253 ページの「デバイスドライバユーティリティを起動する方法」
- 254 ページの「不足しているドライバをインストールする方法」
- 256 ページの「HCL にシステムを一覧表示させる方法」

▼ デバイスドライバユーティリティを起動する方法

デバイスドライバユーティリティは、インストール済みのシステムをブートすると自動的に実行されます。Oracle Solaris OS をインストールしたあとで、手動でデバイスドライバユーティリティを起動することもできます。

- 次のいずれかの方法でデバイスドライバユーティリティを起動します。
 - **Oracle Solaris** テキストインストーライメージをブートします。
 テキストインストーラからデバイスドライバユーティリティを起動するには、初期メニューから「Install Additional Drivers」を選択します。

注-テキストインストーラがブートすると、デフォルトで自動ネットワーク接続が設定されます。DHCPを使用している場合、デバイスドライバユーティリティを使用するための追加のネットワーク設定は不要です。DHCPを使用していない場合は、初期メニューで「シェル」オプションを選択し、デバイスドライバユーティリティを使用する前に、適切なコマンドを使ってネットワーク設定を手動で構成します。

- インストール先システムでデバイスドライバユーティリティを起動します。
 インストール先システムのデスクトップからデバイスドライバユーティリティを起動するには、メインメニューから「アプリケーション」→「システムツール」→「デバイスドライバユーティリティ」を選択します。

デバイスドライバユーティリティがシステムをスキャンし、検出されたデバイスのリストを表示します。このリストでは、デバイスを現在管理しているドライバの製造元、モデル、名前などの情報が、検出されたデバイスごとに表示されます。

次の手順 ドライバが関連付けられていないデバイスをユーティリティが検出した場合、デバイスリストでそのデバイスが選択されます。そのデバイスの詳細情報を表示したり、不足しているドライバをインストールしたりできます。[254 ページの「不足しているドライバをインストールする方法」](#)を参照してください。

▼ 不足しているドライバをインストールする方法

ドライバが関連付けられていないデバイスをユーティリティが検出した場合、デバイスリストでそのデバイスが選択されます。そのデバイスの詳細情報を表示したり、不足しているドライバをインストールしたりできます。

- 1 デバイスドライバユーティリティのリストで、デバイス名を右クリックし、次にポップアップメニューから「詳細を表示」を選択します。
 「Device and Driver Details」ウィンドウが表示されます。デバイス名、ベンダー名、ノード名、ドライバ名など、そのデバイスに関する詳細情報が表示されず。
- 2 不足しているドライバについて詳細を表示するには、選択したデバイスの「情報」リンクをクリックします。

現在デバイスを管理しているドライバが存在しない場合、デバイスリストの「ドライバ」列に、そのデバイスのドライバのステータスが表示されます。不足しているドライバは、次のいずれかのカテゴリに分類されて表示されます。

- IPS – 構成済みの IPS パッケージリポジトリの 1 つ。
- SVR4 – System V Revision 4 (SVR4) パッケージ。
- DU – DU パッケージ。
- UNK – デバイスドライバユーティリティはこのデバイスの Oracle Solaris ドライバを見つけられません。

ヒント – 追加情報を表示するには、「ヘルプ」ボタンをクリックします。

3 不足しているドライバをインストールします。

■ IPS ドライバの場合:

- a. 表の対応する行の「情報」リンクをクリックして、そのデバイスのドライバが含まれる IPS パッケージについての情報を表示します。

「パッケージ」ラジオボタンのテキストフィールドに、関連するパッケージ情報が入力されます。正しい発行元が指定されます。

- b. パッケージをインストールするには「インストール」ボタンをクリックします。

- 「情報」リンクによって、未構成の発行元が提供する IPS パッケージが表示された場合は、次の手順に従います。

- i. 「リポジトリ」メニューから「リポジトリを追加」を選択します。

「リポジトリマネージャー」ウィンドウが表示されます。

- ii. 新しいリポジトリの名前と URI を追加し、「追加」をクリックします。

- 「パッケージ」フィールドに情報が入力されない場合、「情報」リンクから IPS パッケージの名前を入力して、「インストール」をクリックします。

■ SVR4 または DU ドライバの場合:

- パッケージの URL がわかっている場合、「ファイル/URL」フィールドに URL を入力して「インストール」をクリックします。

- パッケージのコピーがシステム上にある場合、「参照」ボタンをクリックしてパッケージを選択し、「インストール」をクリックします。

- ドライバステータスが「UNK」と表示される場合:
 - a. このドライバで管理するデバイスの名前を選択します。
 - b. 関連するパッケージ情報を「パッケージ」フィールドまたは「ファイル/URL」フィールドに入力し、「インストール」をクリックします。
 - c. (省略可能) デバイスに対応するドライバについての情報を共有するには、「送信」ボタンをクリックします。

次の手順 デバイスドライバユーティリティで作業している場合は、特定のデバイスに対して機能することがわかっているドライバに関する情報をほかのユーザーと共有できます。256 ページの「[HCL にシステムを一覧表示させる方法](#)」を参照してください。

▼ HCL にシステムを一覧表示させる方法

特定のデバイスに対して機能することがわかっているドライバに関する情報をほかのユーザーと共有できます。

- 1 デバイスドライバユーティリティを起動します。
253 ページの「[デバイスドライバユーティリティを起動する方法](#)」を参照してください。
- 2 使用中のシステムとそのコンポーネントを「Reported to Work」として HCL に一覧表示するには、「送信」ボタンをクリックします。
「ハードウェア互換リストに情報を送信」ウィンドウが開きます。このウィンドウには、システムについて収集されたすべての情報が表示されます。
 - a. 「システムタイプ」を選択します。
 - b. 情報が自動入力されなかったフィールドに、適切な情報を入力します。
 - 製造元名 - システムのメーカーの名称 (たとえば、Toshiba、Hewlett-Packard、Dell など)。
 - 完全なモデル番号。
「BIOS/ファームウェアのメーカー」は、通常、システムのブート時に表示される BIOS 設定画面の情報です。
 - CPU タイプ - CPU のメーカーの名称。
 - c. 氏名と電子メールアドレスを入力します。

- d. 追加のコメントがあれば「一般的なメモ」フィールドに入力し、「保存」をクリックします。保存されたファイルを **device-detect-feedback_ww@oracle.com** に送信します。

索引

A

AI, 「自動インストーラ (AI)」を参照

`aimanifest` コマンド

`add` サブコマンド, 154

`load` サブコマンド, 153

`set` サブコマンド, 154

`validate` サブコマンド, 166-167

AI インストールサーバー, 101-135

DHCP の設定, 104

維持, 117-135

構成, 106-109

Web サーバーホストのポート, 107-108

マルチキャスト DNS, 106

マルチホーム, 107

要件, 101-102

オペレーティングシステム, 103

管理者権限, 102-103

静的 IP アドレス, 103

ソフトウェアパッケージリポジトリ, 104

デフォルトルーター, 103

ハードウェア, 102

AI インストールサービス

`default-arch` サービス別名の変更, 109

DHCP 構成, 108-109, 114-117

`grub.cfg` ファイル, 114, 116

GRUB メニュー, 114, 116

`system.conf` ファイル, 112

`wanboot.cgi` ファイル, 112

`wanboot.conf` ファイル, 112

Web サーバーホストのポート, 107-108

インストールサービスからのクライアントの削除, 124, 226-227

AI インストールサービス (続き)

インストールサービスに関する情報の表示, 131-133

インストールサービスの更新, 121

インストールサービスの削除, 122-123

インストールサービスの作成, 109-117, 118-120

インストールサービスの名前変更, 122

インストールサービスの無効化, 122

インストールサービスの有効化, 122

インストール手順

「AI マニフェスト」を参照

クライアントのインストール手順

「AI マニフェスト」を参照

クライアントの関連付け, 123, 225-227

クライアントの構成手順

「AI システム構成プロファイル」を参照

ネットイメージ

IPS パッケージ, 112-114

ISO ファイル, 111-112

デフォルトソース, 113

デフォルトの宛先, 112

デフォルトのベースディレクトリ, 108

ブートファイル

SPARC の `wanboot-cgi` ファイル, 112

x86 の `grub2netx64.efi` ファイル, 114, 116

x86 の `pxegrub2` ファイル, 114, 116

プロパティ, 変更

`aliasof` プロパティ, 120

`default-manifest` プロパティ, 121

`imagepath` プロパティ, 121

- AI インストールサービスのネットイメージ
 - IPS パッケージ, 112-114
 - ISO ファイル, 111-112
 - デフォルトソース, 113
 - デフォルトの宛先, 112
 - デフォルトのベースディレクトリ, 108
- AI クライアントインストール
 - SPARC インストールのトラブルシューティング
 - DHCP サーバーが応答しない, 234
 - Wan ブートの警告, 235-236
 - インストールが開始しない, 236-237
 - ファイルが見つからない, 236
 - ブートの読み込みに失敗, 235
 - ssh コマンドを使用した監視, 227-228
 - /system/volatile/install_log ログファイル, 224, 233
 - x86 インストールのトラブルシューティング
 - DHCP サーバーが応答しない, 237
 - TFTP エラー, 237-238
 - インストールが開始しない, 239
 - システムのハングアップ, 238
 - ファイルが見つからない, 238-239
 - インストールのメッセージ
 - インストールの開始, 231
 - インストールの成功, 231
 - インストールを開始しないでブートする, 242-243
 - 概要, 223-224
 - クライアントの要件, 224-225
 - コマンド行でのインストールの開始, 243
 - トラブルシューティング
 - DNS の構成, 233-234
 - インストールに失敗, 239
 - パッケージが見つからない, 241
 - パッケージリポジトリが見つからない, 239
 - ブートエラー, 234-241
 - ネットワークブート
 - network-boot-arguments OBP 変数, 228
 - SPARC クライアント, 228-229
 - SPARC の WAN ブートサポート, 224
 - x86 クライアント, 229-230
- AI システム構成プロファイル
 - 「システム構成」も参照
 - /usr/share/auto_install/sc_profiles プロファイル, 187-200
 - インストールサービスからの削除, 129
 - インストールサービスへの追加, 127-128, 173
 - プロファイルの検証, 129, 172
 - プロファイルの更新, 128-129
 - プロファイルのコピー, 130
 - プロファイルの作成, 171-173
 - プロファイルを選択するための条件, 128, 130-131, 140-144
 - 例, 187-200
- AI マニフェスト
 - インストール開始前の作成, 148-150
 - インストールサービスからの削除, 127
 - インストールサービスへの追加, 124-126
 - カスタム IPS パッケージのインストール, 219-220
 - 既存のマニフェストの変更, 148-150
 - クライアントインストール時の作成
 - 「派生マニフェスト」を参照
 - 選択アルゴリズム, 138-139
 - ゾーンの configuration 要素, 201-208
 - デフォルトの AI マニフェスト, 144-146
 - マニフェストの更新, 126-127
 - マニフェストのコピー, 130
 - マニフェストを選択するための条件, 125, 130-131, 140-144
 - 例, 167-170
 - iSCSI ターゲット, 167-168
 - RAID 構成, 168-169
 - SVR4 パッケージのインストール, 169-170
- all_services プロパティグループ
 - default_imagepath_basedir プロパティ, 108
 - exclude_networks プロパティ, 107
 - manage_dhcp プロパティ, 108-109, 114-117
 - networks プロパティ, 107
 - port プロパティ, 107-108
- ASR, 「Oracle Auto Service Request」を参照
- AI クライアント構成, 「システム構成」を参照

C

configCCR コマンド
 -c オプション, 249
 手動登録と, 251

D**DHCP**

AIクライアントインストールのトラブル
 シューティング, 234, 237
 dhcpinfo コマンド, 233-234
 /etc/inet/dhcdpd4.conf 構成ファイル, 116-117,
 117
 Live Media または GUI インストール中の構
 成, 46-50
 svc:/network/dhcp/server SMF サービス, 116
 構成ファイル, 116, 226
 自動構成, 108-109, 114-117
 設定, 104
 dhcpinfo コマンド, 233-234
 DNS, AIクライアントインストールのトラブル
 シューティング, 233-234

E

emCCR コマンド, データ収集の変更, 251-252
 /etc/auto_home ファイル, 176-177
 /etc/passwd ファイル, 176-177
 /etc/resolv.conf 構成ファイル, 233-234

G

grub.cfg ファイル, 114, 116
 grub2netx64.efi ファイル, 114, 116
 GRUB メニュー, 114, 116
 GUI インストーラ
 DHCP, インストール中の構成, 46-50
 iSCSI, インストール中の構成, 46-50
 インストール, 準備, 45-46
 インストール後のパッケージの追加, 52-53

GUI インストーラ (続き)

インストール時にグラフィックスカードが未サ
 ポートまたは存在しない場合のインス
 トール, 50-52
 インストールに使用されるデフォルトのネット
 ワーク設定とセキュリティ設定, 44
 サポートされているプラットフォーム, 43-50
 パーティション分割のガイドライン, 43-50

I

Image Packaging System (IPS) パッケージリポジト
 リ, 95-96
 install/installadm ソフトウェアパッケージ, 104
 installadm コマンド, 104-106
 create-client サブコマンド, 123, 225-227
 create-manifest サブコマンド, 124-126
 create-profile サブコマンド, 127-128
 create-service サブコマンド, 109-117, 118-120
 delete-client サブコマンド, 124, 226-227
 delete-manifest サブコマンド, 127
 delete-profile サブコマンド, 129
 delete-service サブコマンド, 122-123
 disable サブコマンド, 122
 enable サブコマンド, 122
 export サブコマンド, 130
 list サブコマンド, 131-133
 rename-service サブコマンド, 122
 set-criteria サブコマンド, 130-131
 set-service サブコマンド, 120-121
 update-manifest サブコマンド, 126-127
 update-profile サブコマンド, 128-129
 update-service サブコマンド, 121
 validate サブコマンド, 129, 172
 IPS ソフトウェアパッケージリポジトリ, 95-96
 iSCSI
 Live Media または GUI インストーラ, インス
 トール中の構成, 46-50
 テキストインストーラ, インストール中の構
 成, 56

L

Live Media

- iSCSI, インストール中の構成, 46-50
- インストール, 準備, 45-46
- インストール後のパッケージの追加, 52-53
- インストール時にグラフィックスカードが未サポートまたは存在しない場合のインストール, 50-52
- インストールに使用されるデフォルトのネットワーク設定とセキュリティー設定, 44
- サポートされているプラットフォーム, 43-50
- パーティション分割のガイドライン, 43-50

M

MOS, 「My Oracle Support」を参照

My Oracle Support

- AI インストール, 185
- 資格
- Oracle Configuration Manager, 249

N

network-boot-arguments OBP 変数, 228

O

OCM, 「Oracle Configuration Manager」を参照

Oracle Auto Service Request

- AI インストール用の構成, 185
- 使用, 39-41

Oracle Configuration Manager

- AI インストール用の構成, 185
- Oracle Universal Installer と, 249
- 概要, 247-248
- 手動登録, 251
- 使用, 39-41
- 中央コレクタ, 249
- データ収集, 251-252
- 無効化, 250-251
- 有効化, 250

Oracle Device Detection Tool, 使用, 38-39

Oracle Universal Installer, Oracle Configuration Manager と, 249

Oracle Solaris インストール, システム要件, 31
OUI, 「Oracle Universal Installer」を参照

P

pkg コマンド

- Live Media または GUI インストール後の適用, 52-53
- 既存のインストールを更新するために使用, 28
- テキストインストール後のソフトウェアの追加, 65-66
- pxegrub2 ファイル, 114, 116

S

setupCCR コマンド, -c オプション, 249

SMF サービス

- svc:/application/auto-installer, 224
 - svc:/network/dhcp/server, 116
 - svc:/network/dns/client, 184
 - svc:/network/dns/multicast, 106
 - svc:/network/install, 182-184
 - svc:/system/config-user, 175-178
 - svc:/system/console-login, 181
 - svc:/system/environment:init, 179-181
 - svc:/system/identity, 178-179
 - svc:/system/install/server, 106-109, 135
 - svc:/system/keymap, 181
 - svc:/system/name-service/switch, 198-199
 - svc:/system/timezone, 179-181
 - svc:/system/zones-install, 201-202
 - 初回ブート時に1度だけ実行する, 209
- SMF サービスマニフェスト
- svcbundle コマンド, 213-215
 - 作成, 213-217
 - 初回ブート時に1度だけ実行するサービスの例, 215-217
 - マニフェスト作成ツール
 - 「svcbundle コマンド」を参照
- SMF プロパティー
- AI クライアント構成, 171-200

SMF プロパティ (続き)

- all_services プロパティグループ, 106-109
- config プロパティグループ, 178-179, 179, 184
- enable_mapping プロパティ, 179
- environment プロパティグループ, 179-181
- install_ipv4_interface プロパティグループ, 182-184
- install_ipv6_interface プロパティグループ, 183
- root_account プロパティグループ, 176
- timezone プロパティグループ, 179-181
- user_account プロパティグループ, 177-178
表示, 174-185
- ssh コマンド, AI クライアントインストールの監視, 227-228
- svcbundle コマンド, 213-215
- svccfg コマンド, プロパティ情報の表示, 174-185
- sysconfig create-profile コマンド, 172
- system.conf ファイル, 112
- /system/volatile/install_log ログファイル, 224, 233
自動インストール, 75

U

- USB イメージ, テキストインストーラ用の取得と構成, 57-58
- /usr/sbin/configCCR コマンド, 手動登録と, 251
- /usr/sbin/emCCR コマンド, データ収集の変更, 251-252

V

- /var/sadm/system/logs/install_log ログファイル, 自動インストール, 75
- VTOC スライス
 - rpool と ZFS ルートプール, 37
 - インストール中の選択と変更, 37-38

W

- wanboot.conf ファイル, 112
- WAN ブートサポート, 224

あ

- 暗号化されたパスワード, etc/shadow ファイルからのコピー, 175-178

い

- インストール
 - インストール前のデフォルトの root パスワード, 52
 - 自動インストーラ (AI) の使用
「AI クライアントインストール」を参照
追加オプション
 - pkg コマンド, 既存のインストールを更新するため使用, 28
 - カスタム ISO の作成 (ディストリビューションコンストラクタ), 28
 - テキストインストーラと USB イメージの使用, 57-58
 - ネットワーク経由でのテキストインストーラの使用, 66-67
 - インストールサーバー, 「AI インストールサーバー」を参照
 - インストールサービスの Web サーバーホストのポート, 107-108
 - インストールの要件, 31

け

- 権限
 - sudo コマンド, 103
 - 権利プロファイル, 102
 - 役割, 103

こ

- 構成, 「システム構成」を参照

さ

サービス管理機能 (SMF) プロファイル, AI クライアント構成, 171-200

し

システム構成, 171-200

Oracle Auto Service Request, 185

Oracle Configuration Manager, 185

sysconfig create-profile コマンド, 172

インストールサービスへのプロファイルの追加, 173

カスタム IPS パッケージ, 217-219

キーボード配列, 181

クライアントのインストール時, 185-187

構成プロファイルの検証, 172

構成プロファイルの作成, 171-173

構成プロファイルの例, 187-200

LDAP プロファイル, 196-198

LDAP を DNS プロファイルとともに使用, 198-199

NIS サービスのプロファイル, 192-195

NIS と DNS の両方のプロファイル, 194-195

NIS を DNS プロファイルとともに使用, 199-200

検索リストを使用した DNS のプロファイル, 195-196

静的なネットワークのプロファイル, 189-191

セキュアな LDAP プロファイル, 197-198

ネームサービスのプロファイル, 191-200

システム識別情報, 178-179

システムのロケール, 179-181

初回ブートスクリプト, 209

作成, 211-213

テンプレート, 211-212

複数の IP インタフェースの構成, 212-213

静的なネットワーク, 182-184

ゾーン構成プロファイル, 207-208

ゾーンのデフォルトの AI マニフェスト, 205

タイムゾーン, 179-181

端末タイプ, 181

ネームサービス, 184

ノード名マッピング, 179

システム構成 (続き)

ホスト名, 179

ユーザー

autohome プロパティ, 176-177

/etc/auto_home ファイル, 176-177

/etc/passwd ファイル, 176-177

root ユーザーアカウント, 175-178

暗号化されたパスワード, 175-178

オートマウントへの依存, 176-177

初期ユーザーアカウント, 175-178

複数のユーザーアカウント, 178

システム構成プロファイルテンプレート, 185-187

変数, 186-187

システムのパーティション分割

DOS, 34-38

GPT, 34-38, 43-50

GRUB 2, 33-34, 43-50

GUI インストーラまたは Live Media ISO イメージ, 使用, 43-50

Linux スワップ, 33-34

Oracle Solaris, 34-38

VTOC スライス, インストール中の選択と変更, 34-38

インストール前, 33-34

対話型インストール, 35-37

システム要件, 31

自動インストール, 70-71

自動インストーラ (AI)

インストール手順, 92-93

インストールのカスタマイズ, 137-146

概要, 87

構成手順, 93-94

異なる OS バージョンのインストール, 98-99

最小要件, 90-91

初回ブート構成スクリプト, 96-97

ローカル IPS パッケージリポジトリ, 95-96

自動インストール

インストールの概要, 71-72

概要, 69

カスタムマニフェスト, 使用, 72

システム要件, 70-71

手動登録, Oracle Configuration Manager, 251

そ

ゾーン

- AI クライアントでのインストール, 201-208
- AI マニフェスト, 202-203, 205-207
 - configuration 要素, 201-208
 - デフォルト, 205
- config ファイル, 203
- インストールサービスへのプロファイルの追加, 204
- インストールサービスへのマニフェストの追加, 204
- 構成プロファイル, 207-208
- ソフトウェアパッケージリポジトリ, 95-96

た

- 対話型インストール, システムのパーティション分割, 35-37

ち

- 中央コレクタ, Oracle Configuration Manager, 249

つ

- ツールを使用したデバイスドライバの特定, 253-257

て

- ディストリビューションコンストラクタ, 追加情報の検索, 28
- データ収集, Oracle Configuration Manager, 251-252
- テキストインストーラ
 - GUI インストーラに優る利点, 26
 - iSCSI, インストール中の構成, 56
 - NCP, 構成, 57
 - USB イメージ, インストール用の取得, 57-58
 - インストール後のソフトウェアの追加, 65-66
 - インストールの準備, 57-58

テキストインストーラ (続き)

- システムのパーティション分割
 - GPT 形式の使用法, および, 55-66
 - GRUB 2, 使用, 55-66
- デフォルトのパッケージセット, 26
- ネットワーク, 構成, 57
- ネットワーク経由でのインストールの実行, 66-67
- デバイスドライバ
 - Oracle Device Detection Tool, 使用, 38-39
 - 関連情報を見つける, 38-39
 - 情報の特定, 253-257
 - デバイスドライバユーティリティの使用, 253-257
- デバイスドライバを見つけるツール, 使用, 38-39
- デフォルトのパスワード, Live Media または GUI インストール, 46-50

と

ドライバ

- 特定, 253-257
- 見つける, 38-39
- トラブルシューティング
 - AI インストール, 233-243
 - Live Media または GUI インストール
 - コンソールモードでのシステムブート, 50-52

は

派生マニフェスト

- AIM_LOGFILE 環境変数, 155-164
- AIM_MANIFEST 環境変数, 155-164
- aimanifest コマンド, 153-154
- aiuser の役割, 151-153
- インストールサービスへの追加, 166-167
- クライアント属性の環境変数, 151-153
- 作成と適用, 150-167
- スクリプトの検証, 166-167
- スクリプトのテスト, 164-166
- スクリプトの例, 155-164
- 変更する初期のマニフェスト, 153-154

パッケージマネージャー

Live Media または GUI インストール後の適
用, 52-53

テキストインストール後のソフトウェアの追
加, 65-66

パッケージリポジトリ, 95-96

ひ

非大域ゾーン, 「ゾーン」を参照

ふ

ブート環境, 複数のオペレーティングシステムを
インストールするための準備, 32

ブートファイル

SPARC の wanboot-cgi ファイル, 112, 225

x86 クライアントの .bios ファイル, 226

x86 クライアントの .uefi ファイル, 226

x86 の grub2netx64.efi ファイル, 114, 116

x86 の pxegrub2 ファイル, 114, 116

ま

マルチキャスト DNS (mDNS), 106

マルチホームインストールサーバー, 107

む

無効化, Oracle Configuration Manager, 250-251

ゆ

有効化, Oracle Configuration Manager, 250