Oracle® Solaris 11 システムのインストール

Part No: E25760
2012 年 3 月 E25760–03

ORACLE®

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

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際、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に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。

120412@25097

目次

はじめに 9

パート I **Oracle Solaris 11** のインストールオプション 15

1 インストールオプションの概要 17
インストールオプションの比較 17
シンプルで、事前設定されたインストール 18
サーバーの設定を必要とするインストール 19
その他のオプション 20

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 21

2 インストールの準備 23
LiveCD およびテキストインストールのシステム要件 23
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 24
システムのパーティション分割 25
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 25
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 26
適切なデバイスドライバがあることの確認 30
▼ Oracle Device Detection Tool を使用する方法 31
デバイスドライバユーティリティーの概要 31
▼ デバイスドライバユーティリティーを起動する方法 32
▼ 不足しているドライバをインストールする方法 33
▼ HCL にシステムを一覧表示させる方法 35
Oracle Configuration Manager の使用 36

3 LiveCD の使用 39
GUI インストーラを使用したインストール 39
GUI インストーラでのデフォルト設定 40
▼ GUI インストールを実行する方法 40
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 46
▼ システムがコンソールモードでブートした場合に LiveCD から Oracle Solaris をインストールする方法 46
LiveCD インストール後のソフトウェアの追加 48

4 テキストインストーラの使用 49
テキストインストーラを使用したインストール 49
テキストインストーラを使用したネットワーク構成 50
▼ テキストインストールを実行する方法 50
テキストインストール後のソフトウェアの追加 58
ネットワーク経由でのテキストインストールの実行 59
▼ 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 59

5 メディアからブートする自動インストール 61
AI メディアを使用するインストールの概要 61
AI メディアを使用したインストール 61
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 62
▼ AI メディアを使用してインストールする方法 63
カスタム AI マニフェストの作成 64
AI メディアからの SPARC システムのブート 65
AI メディアからの x86 システムのブート 66
インストールのログファイルの表示 67

6 Oracle Solaris インスタンスの構成解除または再構成 69
機能グループ 69
`sysconfig` ユーティリティーの使用 70
Oracle Solaris インスタンスの構成解除 71
システムの構成 71
▼ SCI Tool を使用して再構成する方法 72
SCI Tool を使用した構成プロファイルの作成 75

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 77

7 複数のクライアントへの自動インストール 79
自動インストールとは 79
自動インストーラの使用方法 80
自動インストーラの使用事例 81
AI を使用するための最小要件 82
インストール手順のカスタマイズ 83
システム構成手順を指定する 85
ローカルの IPS パッケージリポジトリを指定する 86
カスタムの初回ブートスクリプトを指定する 87
追加の AI インストールサービスを指定する 89

8 インストールサーバーの設定 91
AI サーバー設定の作業マップ 91
インストールサーバーの要件 92
AI サーバーのハードウェア要件 92
AI サーバーのソフトウェア要件 92
AI インストールツールをインストールする 94
インストールサーバーを構成する 95
マルチホームインストールサーバーを構成する 95
Web サーバーホストのポートを構成する 96
AI インストールサービスを作成する 97
DHCP を設定せずにインストールサービスを作成する 99
ローカル DHCP を設定してインストールサービスを作成する 102
インストールサーバーを維持する 103
インストールサービスを追加、変更、または削除する 104
クライアントをインストールサービスに関連付ける 108
クライアント固有のインストール手順をインストールサービスに関連付ける . 109
クライアント固有の構成手順をインストールサービスに関連付ける 112
AI マニフェストまたはシステム構成プロファイルをエクスポートする 114
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を変更する 115
インストールサービスの情報を表示する 116
カスタマイズしたインストールの情報を表示する 117
AI SMF サービスを管理する 118

9 インストールのカスタマイズ 121
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 121
AI マニフェストの選択 122
システム構成プロファイルの選択 123
選択条件 124
デフォルトの AI マニフェスト 127

10 クライアントシステムのプロビジョニング 131
XML AI マニフェストファイルのカスタマイズ 132
クライアントインストール時の AI マニフェストの作成 134
派生マニフェストスクリプトを作成する 135
派生マニフェストスクリプトをインストールサービスに追加する 148

11 クライアントシステムの構成 151
構成プロファイルの指定 151
システム構成プロファイルの作成 152
システム構成プロファイルの検証 152
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 153
システム構成プロファイル内での構成の指定 153
root アカウントとユーザーアカウント 155
システム識別情報 157
タイムゾーンとロケール 158
端末タイプとキー配列 159
静的なネットワーク構成 160
ネームサービス構成 162
システム構成プロファイルテンプレートの使用 163
システム構成プロファイルの例 165
システム構成プロファイルの例 165
静的なネットワーク構成の指定 167
ネームサービス構成の指定 169

12 ゾーンのインストールと構成 179
AI による非大域ゾーンのインストール方法 179
大域ゾーンの AI マニフェストでのゾーン指定 180

非大域ゾーンの構成およびインストールデータ 181
非大域ゾーンの AI マニフェスト 183
非大域ゾーンの構成プロファイル 184

13 初回ブート時のカスタムスクリプトの実行 187
初回ブート時に実行されるスクリプトの作成 187
SMF マニフェストファイルの作成 189
スクリプトとサービス用の IPS パッケージの作成 191
▼ IPS パッケージを作成し、公開する方法 191
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 192
▼ IPS パッケージのインストール方法 193

14 **AI** クライアントシステムで使用するための **Oracle Configuration Manager** の設定 195
AI クライアントでの Oracle Configuration Manager のデフォルトの動作 195
カスタムレスポンスファイルの指定 196
▼ カスタムレスポンスファイルパッケージを作成してインストールする方法 196
Oracle Configuration Manager の停止 199

15 クライアントシステムのインストール 201
クライアントのインストール方法 201
クライアントシステムの要件 202
SPARC および x86 クライアントシステムの要件 202
SPARC クライアントシステムの追加の要件 202
インストールクライアントを設定する 203
x86 クライアントの設定 203
SPARC クライアントの設定 204
サービスからのクライアントの削除 204
クライアントのインストール 205
Secure Shell を使用したリモートからのインストールの監視 205
SPARC クライアントのインストール 206
x86 クライアントのインストール 209
クライアントインストールのメッセージ 211

16 自動インストールのトラブルシューティング 213
クライアントインストールが失敗する 213
インストールログおよび手順の確認 213
DNS の確認 213
クライアントブートエラーの確認 214
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 222
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 223

はじめに

Oracle Solaris 11 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 LiveCD 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 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 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「LiveCD 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となる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「AI クライアントシステムで使用するための Oracle Configuration Manager の設定」では、Oracle Configuration Manager が AI クライアントインストールで機能するようにする方法について説明します。
- 第 15 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 16 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。

関連情報

『Oracle Solaris 11 インストールマニュアルページ』では、`aimanifest(1M)`、`distro_const(1M)`、`installadm(1M)`、`js2ai(1M)`、`ai_manifest(4)`、および `dc_manifest(4)` のマニュアルページのコピーを提供しています。

『カスタム Oracle Solaris 11 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 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 の管理: 一般的なタスク』の第 6 章「サービスの管理 (概要)」では、Oracle Solaris サービス管理機構 (SMF) 機能について説明します。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 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 パッケージリポジトリのコピーおよび作成』では、Oracle 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 システムを管理する方法の詳細は、Oracle Solaris 11 システム管理ドキュメントを参照してください。

Internet Systems Consortium (ISC) の DHCP サーバーのダウンロードおよびドキュメントについては、isc.org Web サイトの DHCP セクションを参照してください。

『Oracle Solaris 10 JumpStart から Oracle Solaris 11 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す。

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` *filename* と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	`sun%` **`grep '^#define \`** **`XV_VERSION_STRING'`**

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 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。次の Oracle Solaris インストールオプションの概要を参照してください。

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「LiveCD 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
59 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 LiveCD 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、『カスタム Oracle Solaris 11 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

LiveCD 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法です。

- 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 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。

LiveCD 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ルメディアには、汎用的なサーバーシステムにより適した、さらに小さな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には、GUI 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て済む。
- 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にブートしたあとで、`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 スライスの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できる必要があります。

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ーの容量、およびその他の特性は異なっていても構いません。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、59 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、『カスタム Oracle Solaris 11 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 11 システムの更新
: インストーラを使用して、既存の Oracle Solaris 11 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 ユーティリティー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 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。

次を参照してください。

- 第 2 章「インストールの準備」
- 第 3 章「LiveCD の使用」
- 第 4 章「テキストインストーラの使用」
- 59 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」

第 2 章

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。

- 23 ページの「LiveCD およびテキストインストールのシステム要件」
- 24 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」
- 25 ページの「システムのパーティション分割」
- 30 ページの「適切なデバイスドライバがあることの確認」
- 36 ページの「Oracle Configuration Manager の使用」

LiveCD およびテキストインストールのシステム要件

次の表に、LiveCD 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 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 Release Notes』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 LiveCD 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。ただし、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 Release Notes』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	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。x86 プラットフォーム向け Oracle Solaris の全バージョンで、GNU GRUB (GRand Unified Bootloader) が使用されます。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 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。 注 – 現在、x86 システムの Oracle Solaris オペレーティングシステムでは、協定世界時 (Universal Time Coordinated、UTC) 時間がリアルタイムクロック (RTC)、つまりハードウェアクロックに格納されます。以前、x86 システムのオペレーティングシステムでは、現地時間が RTC に格納されていました。
Linux、または Windows と Linux	Linux オペレーティングシステムのみ、または Linux および Windows の両オペレーティングシステムが x86 ベースのシステムにインストールされている場合、Oracle Solaris をインストールする前に、`menu.lst` ファイルのコピーを保存します。インストール後に `menu.lst` ファイルを編集し、以前のインストールの Linux 情報を追加する必要があります。手順については、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』の「menu.lst ファイルの編集によるブートエントリとパラメータの変更」を参照してください。 注 – Linux オペレーティングシステムもインストールされているシステムに Oracle Solaris をインストールするときは、Oracle Solaris パーティションが Linux の `swap` パーティションより前になるようにする必要があります。

表 2–1 複数のオペレーティングシステム環境 (続き)

既存のオペレーティングシステム	説明
Solaris 10 OS	LiveCD 上のインストーラは、Oracle Solaris 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)。LiveCD 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。 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。
拡張パーティション	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。LiveCD 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。

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。また、x86 および Solaris VTOC スライスにパーティション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

LiveCD の 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、x86 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では、SPARC 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こともできます。

x86 ベースのシステムでは、インストーラは 1 つ以上のドライブでの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 を使用します。Oracle Solaris をインストールするためのパーティションは、インストール前とインストール中のどちらでも作成できます。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

ンストールを行うと、ブート時に GRUB 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。

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。

Oracle Solaris の `fdisk` パーティションの作成または変更には、`fdisk` コマンドを使用できます。手順については、『Oracle Solaris の管理: デバイスとファイルシステム』の「Solaris fdisk パーティションを作成する方法」を参照してください。また、`fdisk(1M)` のマニュアルページも参照してください。

別の方法として、市販の製品やオープンソースツールを使用してハードディスクのパーティション分割を行うこともできます。

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。

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。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。

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。

- 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。
- Oracle Solaris システム上に既存の `fdisk` パーティションが存在し、ユーザーが既存のパーティションに変更を加えない場合は、インストールによって `fdisk` パーティションのみが上書きされます。ほかの既存パーティションは変更されません。
- Solaris パーティションのみが使用できます。
- 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。

- 既存の Solaris パーティションが存在する場合、そのパーティションがデフォルトとして選択されます。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。
- 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 `fdisk` パーティションがパネルに表示されます。最大 4 つのプライマリパーティションが、ディスク上の配置と同じ順番で表示されます。これらのプライマリ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。
- fdisk パーティションは OS のインストールに使用されるため、2T バイトを超えることはできません。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推奨されるサイズと最小サイズが表示されます。

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 11 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。

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)
新しい Solaris パーティションを作成する。	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できます。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。
別のプライマリパーティションに Oracle Solaris 11 オペレーティングシステムをインストールする。	別のプライマリ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。そのためには、まず既存の Solaris パーティションのタイプを「未使用」に変更する必要があります。次に、別のパーティションを選択し、そのタイプを「Solaris」に変更することができます。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。
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。	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。Solaris パーティションがすでに存在する場合、そのタイプを「未使用」に変更します。次に、新しい拡張パーティションを作成するために、パーティションタイプを「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。
既存のパーティションを削除する。	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。

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 Oracle 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 11 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す。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。

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 (続き)

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に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 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ます。`http://www.oracle.com/webfolder/technetwork/hcl/index.html` のハードウェア互換性リスト (HCL) には、Oracle Solaris が正常に動作することが確認または報告されているハードウェアについての情報が掲載されます。Solaris on x86 Platforms Device Support ツールは、どの Oracle Solaris ドライバが各種の x86 コンポーネントをサポートするかを通知します。

次のユーティリティー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**

  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が報告されます。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。このツールを使用するための手順については、31 ページの「Oracle Device Detection Tool を使用する方法」を参照してください。

- デバイスドライバユーティリティー

  デバイスドライバユーティリティーは、Oracle Device Detection Tool と同じ情報を提供します。このユーティリティーは、Oracle Solaris 11 システムのデスクトップで使用できます。また、このユーティリティーはテキストインストーラのメニューオプションを通じて使用できます。

注 – デバイスドライバユーティリティーには 1.5G バイト以上のメモリーが必要な場合があります。インストールを実行するために必要なデバイスがシステムに揃っている場合、まずインストールを完了し、次にインストールしたハードディスクをブートしてから、デバイスドライバユーティリティーを実行します。その後、ユーティリティーはインストールしたシステム上のスワップ空間を利用できます。

▼ Oracle Device Detection Tool を使用する方法

Oracle Device Detection Tool を使用すると、システム上のすべてのデバイスに対応した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。

デバイスドライバユーティリティーの概要

Oracle Device Detection Tool の代替であるデバイスドライバユーティリティーは、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。このユーティリティーは、システムに検出されたすべてのデバイスに対応するドライバが、現在ブートした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ーは、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ーを使用すると、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、次の作業について説明します。

- 32 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」
- 33 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」
- 35 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ーを起動する方法

デバイスドライバユーティリティーは、インストールイメージをブートすると自動的に実行されます。Oracle Solaris をインストールしたあとに、手動でデバイスドライバユーティリティー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ーを起動します。

- **LiveCD** のインストールイメージをブートします。

  LiveCD の ISO イメージをブートすると、デバイスドライバユーティリティーがバックグラウンドで実行されます。システムに存在しないドライバが、デフォルトの発行元が提供する IPS パッケージに見つかった場合、デバイスドライバユーティリティーはそのドライバパッケージを自動的にインストールします。不足しているドライバがほかにもある場合、デバイスドライバユーティリティーはユーザーにユーティリティーウィンドウを表示するように要求し、ユーザーがレポートを確認して不足している追加のドライバをすべてインストールできるようにします。

- **LiveCD** のデスクトップからデバイスドライバユーティリティーを手動で起動します。

  LiveCD のデスクトップからデバイスドライバユーティリティーを手動で起動するには、デスクトップ上の「デバイスドライバユーティリティー」アイコンをダブルクリックします。または、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ー」を選択します。

- **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ーを起動するには、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ー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ーを使用する前に、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ーを起動します。

  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ーを起動するには、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ー」を選択します。

デバイスドライバユーティリティー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ーが検出した場合、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。33 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ーが検出した場合、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ーリストで、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ます。

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ーはこのデバイスの 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. リポジトリのメニューから「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**」と表示される場合は、次の手順に従います。

  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ー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。35 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

次のように、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ーを起動します。

   32 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。

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** に一覧表示させるには、「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。

Oracle Configuration Manager の使用

LiveCD からのインストール後またはテキストインストール後にリブートすると、Oracle Configuration Manager によって匿名のシステム構成情報が Oracle Support にアップロードされます。My Oracle Support では、インストールされたシステムの構成に関する情報は受信しますが、顧客情報は一切受信しません。

具体的には、初回リブート中に Oracle Configuration Manager サービスがはじめて実行され、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、匿名の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 インストレーションおよび管理ガイド』を参照してください。

LiveCD からのインストールまたはテキストインストールを実行するときに、次のオプションが用意されています。

- 構成情報のデフォルトの匿名登録が自動的に行われるようにします。
- インストールの終わり (システムをリブートする前) に、Oracle Configuration Manager の機能を無効にします。

  無効化の手順については、次を参照してください。

  - LiveCD インストール: 手順 13
  - テキストインストール: 手順 16

注 – インストールイメージを作成してメディアに書き込んだり、ネットワーク経由でテキストインストールを開始したりするなどの目的でインストールサーバーを使用する場合は、Oracle Configuration Manager を管理するための異なるオプションが用意されています。第 14 章「AI クライアントシステムで使用するための Oracle Configuration Manager の設定」を参照してください。

デフォルトの匿名登録の許可または取り消しのどちらを選択したとしても、あとで Oracle Configuration Manager でシステムを登録または再登録できます。

- すでに匿名で登録している場合は、今後のサポートを受けやすくするために、登録の一環としてあとで My Oracle Solaris (MOS) アカウント情報を提供することを選択することもできます。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。
- デフォルトの登録を取り消した場合は、あとで匿名で登録することも、完全な顧客登録を行うこともできます。
- ネットワークプロキシ要件のために自動登録を完了できなかった場合は、匿名で、または MOS アカウント情報を使用して登録できます。

対話型モードで `configCCR` ユーティリティー (`/usr/sbin/configCCR`) を使用することで登録できます。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
```

詳細は、次を参照してください。

- 『Oracle Solaris の管理: 一般的なタスク』の第 5 章「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』

第 3 章

LiveCD の使用

この章では、LiveCD の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 11 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。

- 23 ページの「LiveCD 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。GUI インストーラまたはテキストインストーラの使用手順を参照してください。

注 – LiveCD の ISO イメージのインストーラは、x86 プラットフォームでのみ使用できます。

あるいは、インストールの前に `fdisk` コマンドまたは他社製のパーティション分割ツールを使用すると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。25 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。

特定のオペレーティングシステムのインストール環境の準備に関する詳細は、24 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」を参照してください。

- 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 11 リリース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 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

LiveCD 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 インストールを実行する方法

1. 予備的なタスクをすべて完了します。

   a. **LiveCD** がない場合は、**LiveCD** の **ISO** イメージをダウンロードします。

   Oracle Solaris LiveCD 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 – 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、次のいずれかの手順に従います。

- 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。

- イメージをシステムに保存してから、仮想マシン内で実行します。

b. 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。

i. 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。

23 ページの「LiveCD 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。

ii. 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。

30 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。

c. **Oracle Solaris 11** リリース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。

- **Oracle Solaris 11** リリースがシステムにインストールされる唯一のオペレーティングシステムである場合は、**39** 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。

  i. **24** 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。

  ii. システムをバックアップします。

  iii. 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必要がある場合は、**25** 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。

注 – Linux オペレーティングシステムがすでにインストールされている場合は、既存の `menu.lst` ファイルのコピーを、インストール後に使用できるように USB ドライブに保存する必要があります。手順については、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』の「menu.lst ファイルの編集によるブートエントリとパラメータの変更」を参照してください。

---

2 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。

LiveCD では、GRUB 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 LiveCD 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、LiveCD 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。46 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。

- LiveCD へのログイン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。

3 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。

---

4 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。

LiveCD 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ーの使用方法に従い、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。

5 **LiveCD** のデスクトップで、「**Oracle Solaris** をインストールする」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。

6 開始パネルで、「次へ」を選択します。

注 – このパネルで最新リリースのリリースノートを確認できます。

---

7 「ディスク」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。

必要に応じて、パーティションのレイアウトを変更できます。手順については、26 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刻を調整します。

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、時刻、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 インストール仕様を確認します。

「インストールの確認」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。

11 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。

---

12 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。

13 (省略可能) インストールされたシステムの **Oracle Configuration Manager** への匿名登録を取り消す場合は、次の手順を実行して新しく作成されたブート環境をマウントし、そのブート環境に「オプトアウト」ファイルを追加してから、システムをリブートします。

注意 – デフォルトでは、インストールされたシステムのシステム構成が Oracle Configuration Manager に送信されます。これは、顧客情報が一切提供されない匿名登録です。

匿名登録は初期インストール後のリブート時に自動的に行われますが、インストール後 (インストールしたシステムのリブート前) に次の指示に従うことで登録を取り消すことができます。

インストール時にオプトアウトしない場合でも、あとでいつでもこのサービスを一時停止することができます。

詳細は、36 ページの「Oracle Configuration Manager の使用」を参照してください。

---

a. インストールされたシステムをリブートする前に、「終了」を押してインストーラを終了します。

b. 端末ウィンドウを開きます。

c. **root** の役割になります。

注 – 新しくインストールされたシステムをリブートする前は、「ユーザー」パネルでコンピュータ名を変更していたとしても、デフォルトの root パスワードは「solaris」です。

---

d. 次の例にあるように、新しく作成されたブート環境をマウントします。

```
# beadm mount solaris /a
```

リブート前のブート環境名はデフォルトで「solaris」になります。

e. エディタで、マウントされたブート環境に新しいファイルを作成し、ファイル名を **/a/etc/svc/profile/site/ocm.xml** とします。

たとえば、次のように入力します。

```
# vi /a/etc/svc/profile/site/ocm.xml
```

f. 次の内容をファイルに入力し、ファイルを保存して、ファイルを終了します。

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='profile' name='ocm'
 xmlns:xi='http://www.w3.org/2003/XInclude'>
   <service name='system/ocm' type='service' version='1'>
      <instance name='default' enabled='false'/>
   </service>
</service_bundle>
```

このファイルにより、デフォルトの SMF サービスが無効になり、プロパティーが匿名登録から「オプトアウト」に変わります。

g. このコマンド例に示すように、ブート環境をマウント解除します。

```
# beadm unmount solaris
```

h. 端末を終了します。

注 – インストールおよびリブート後に、このファイルを削除してから、次のようにサービスを有効にすることで、システムを登録できます。

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

このコマンドは、匿名登録を実行します。

システムの構成データを MOS アカウントに関連付ける場合、またはサイトでネットワークプロキシを使用する必要がある場合は、`configCCR` コマンドを使用してください。36 ページの「Oracle Configuration Manager の使用」を参照してください。

---

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に LiveCD 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 LiveCD 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、LiveCD 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。

- LiveCD の 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。
- 次の手順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 LiveCD 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、LiveCD 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 つのシステムは、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 **LiveCD** 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。

参照 次のトピックについては、『Oracle Solaris の管理: 一般的なタスク』を参照してください。

- ユーザーアカウントおよびグループの管理
- システムのブートとシャットダウン
- サービスの管理
- ハードウェア障害の管理
- システムプロセスの管理
- 次のような一般的なシステム問題のトラブルシューティング
  - リブートが失敗した場合の対処
  - root パスワードを忘れた場合の対処
  - システムがハングした場合の対処

LiveCD 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および『Oracle Solaris の管理: 一般的なタスク』の第 12 章「ソフトウェアパッケージの管理 (手順)」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。

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。

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、インストールは新しいブート環境に行うこともできます。

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。

```
# pkg install --require-new-be --be-name newBEname packagename
```

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。

第 4 章

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 11 リリース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。

- 23 ページの「LiveCD 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。

  あるいは、インストールの前に `fdisk` コマンドまたは他社製のパーティション分割ツールを使用すると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。25 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。

  特定のオペレーティングシステムのインストール環境の準備に関する詳細は、24 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」を参照してください。
- Oracle Solaris 11 の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 11 リリース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 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。
- 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。

- LiveCD 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ルメディアには、汎用的なサーバーシステムにより適した、さらに小さなソフトウェアセットが含まれています。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、58 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には、次のオプションが用意されています。

- 自動 – LiveCD 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ルを実行する方法

1. 予備的なタスクをすべて完了します。

   a. 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、次のいずれかの手順に従います。

- 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。

- イメージをシステムに保存してから、仮想マシン内で実行します。

b. 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。

i. 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。

23 ページの「LiveCD 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。

ii. 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。

30 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。

c. 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。

i. **24** 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。

ii. システムをバックアップします。

iii. インストールの前にシステムのパーティション分割を行う必要がある場合は、次のようにします。

第 2 章「インストールの準備」に記載されているガイドラインを確認します。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、25 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。

注 – Linux オペレーティングシステムがすでにインストールされている場合は、既存の `menu.lst` ファイルのコピーを、インストール後に使用できるように USB ドライブに保存する必要があります。手順については、『x86 プラットフォーム上の Oracle Solaris のブートおよびシャットダウン』の「menu.lst ファイルの編集によるブートエントリとパラメータの変更」を参照してください。

---

2 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。

---

3 (省略可能) インストールが必要なドライバについては、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、32 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ルを再起動して、インストールメニューに戻ります。

4 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 xxx 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。

---

5 開始パネルを過ぎて先に進みます。

6 「ディスク」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。

パーティション分割の詳細な手順については、26 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入力します。

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、続けて次の手順を実行します。

  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 – プロファイルがプロキシ資格レベルを指定し、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「日付と時刻」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刻を調整します。

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。ただし、root パスワードを作成する必要はあります。

- 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、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 インストール仕様を確認します。

「インストールの確認」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。

14 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。

---

15 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。

16 (省略可能) インストールされたシステムの **Oracle Configuration Manager** への匿名登録を取り消す場合は、次の手順を実行して新しく作成されたブート環境をマウントし、そのブート環境に「オプトアウト」ファイルを追加してから、システムをリブートします。

注意 – デフォルトでは、インストールされたシステムのシステム構成が Oracle Configuration Manager に送信されます。これは、顧客情報が一切提供されない匿名登録です。

匿名登録は初期インストール後のリブート時に自動的に行われますが、インストール後 (インストールしたシステムのリブート前) に次の指示に従うことで登録を取り消すことができます。

インストール時にオプトアウトしない場合でも、あとでいつでもこのサービスを一時停止することができます。

詳細は、36 ページの「Oracle Configuration Manager の使用」を参照してください。

---

a. インストールされたシステムをリブートする前に、**F9** キーを押してインストーラを終了します。

b. インストールメニューで、「**3**」を選択してシェルを開きます。

c. **root** の役割になります。

d. 次の例にあるように、新しく作成されたブート環境をマウントします。

```
# beadm mount solaris /a
```

注 – インストーラパネルでコンピュータ名を変更した場合でも、リブート前のブート環境名はデフォルトで「solaris」になります。

---

e. エディタで、マウントされたブート環境に新しいファイルを作成し、ファイル名を **/a/etc/svc/profile/site/ocm.xml** とします。

たとえば、次のように入力します。

```
# vi /a/etc/svc/profile/site/ocm.xml
```

f. 次の内容をファイルに入力し、ファイルを保存して、ファイルを終了します。

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='profile' name='ocm'
 xmlns:xi='http://www.w3.org/2003/XInclude'>
   <service name='system/ocm' type='service' version='1'>
      <instance name='default' enabled='false'/>
   </service>
</service_bundle>
```

このファイルにより、デフォルトの SMF サービスが無効になり、プロパティーが匿名登録から「オプトアウト」に変わります。

g. このコマンド例に示すように、ブート環境をマウント解除します。

```
# beadm unmount solaris
```

h. シェルを終了します。

```
# exit
```

注 – インストールおよびリブート後に、このファイルを削除してから、次のようにサービスを有効にすることで、システムを登録できます。

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

このコマンドは、匿名登録を実行します。

システムの構成データを MOS アカウントに関連付ける場合、またはサイトでネットワークプロキシを使用する必要がある場合は、`configCCR` コマンドを使用してください。36 ページの「Oracle Configuration Manager の使用」を参照してください。

---

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。

参照 次のトピックについては、『Oracle Solaris の管理: 一般的なタスク』を参照してください。

- ユーザーアカウントおよびグループの管理
- システムのブートとシャットダウン

- サービスの管理
- ハードウェア障害の管理
- システムプロセスの管理
- 次のような一般的なシステム問題のトラブルシューティング
  - リブートが失敗した場合の対処
  - root パスワードを忘れた場合の対処
  - システムがハングした場合の対処

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および『Oracle Solaris の管理: 一般的なタスク』の第 12 章「ソフトウェアパッケージの管理 (手順)」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。

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、インストールは新しいブート環境に行うこともできます。

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。

```
# pkg install packagename
```

packagename 変数は、インストールするパッケージの名前で置換します。

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。

```
# pkg install --require-new-be --be-name newBEname packagename
```

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. 次のように、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 xxx 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、50 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。自動インストールは、できるだけ多くの必要なソフトウェアを IPS リポジトリからダウンロードするように設計されています。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き、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。

```
# pkg install solaris-desktop

# pkg install solaris-large-server
```

第 5 章

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 11 OS の自動インストールは、ネットワーク経由のブートだけでなく、メディア上の AI イメージのブートによっても開始できます。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、次の作業 (省略可能) 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。

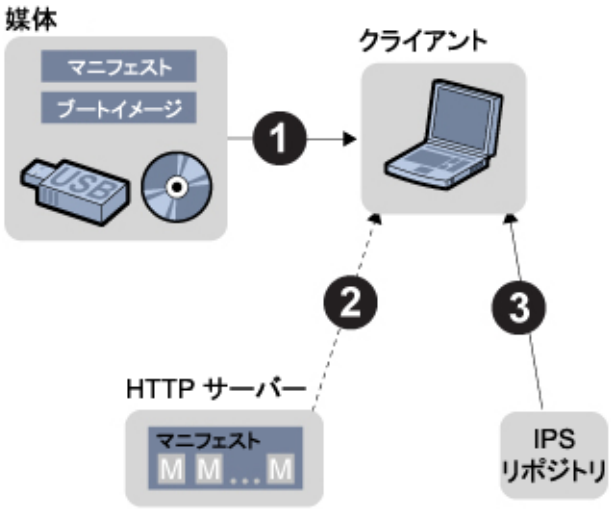
- インストールサーバーまたはインストールサービスを設定する必要がありません。
- システムをネットワーク経由でブートする必要がありません。

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。64 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 Release Notes』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 Release Notes』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	インストールするシステムは、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。 ■ ネットワーク構成情報を提供する DHCP 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 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/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、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にそのカスタムマニフェストの場所を指定できます。64 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` ユーティリティーを使用します。**

注 – 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。

---

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、65 ページの「AI メディアからの SPARC システムのブート」および66 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。

75 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 マニフェストにある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に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。

```
# lofi_dev=$(/usr/sbin/lofiadm -a /tmp/sol-11-ai-sparc.iso)
# /usr/sbin/mount -o ro -F hsfs ${lofi_dev} /mnt
# cp /mnt/auto_install/manifest/default.xml /home/username/custom.xml
# /usr/sbin/umount /mnt
# /usr/sbin/lofiadm -d ${lofi_dev}
```

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、127 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には、`ai_manifest(4)` のマニュアルページを参照してください。

注 – Oracle Configuration Manager の設定を事前設定することもできます。第 14 章「AI クライアントシステムで使用するための Oracle Configuration Manager の設定」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にその 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に、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なったり、システムを調べたりする場合です。

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 0.97 (639K lower / 2078660K upper memory)

Oracle Solaris 11 Automated Install custom
Oracle Solaris 11 Automated Install
Oracle Solaris 11 Automated Install custom ttya
Oracle Solaris 11 Automated Install custom ttyb
Oracle Solaris 11 Automated Install ttya
Oracle Solaris 11 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 Automated Install
Oracle Solaris 11 Automated Install ttya
Oracle Solaris 11 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 Automated Install custom
Oracle Solaris 11 Automated Install custom ttya
Oracle Solaris 11 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なったり、システムを調べたりする場合です。

一般に、使用する GRUB エントリのカーネル行に `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 システムをブートするとき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 GRUB メニューエントリを確認します。その GRUB 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。

第 6 章

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	下記の組み合わせ
identity	システムのノード名	不明
kdb_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

sysconfig ユーティリティーの使用

sysconfig ユーティリティーを使用して、Oracle Solaris インスタンスに対して次の構成作業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。

  71 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。

  71 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。75 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 ユーティリティーを使用するには、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。72 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。75 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入力します。

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。
  SCI Tool 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 **SCI Tool** 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、続けて次の手順を実行します。

  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、SCI Tool 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 – プロファイルがプロキシ資格レベルを指定し、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、SCI Tool によって一連のタイムゾーンパネルおよび「日付と時刻」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。

  SCI Tool 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刻を調整します。

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。ただし、root パスワードを作成する必要はあります。

- 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、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 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。

SCI Tool を使用した構成プロファイルの作成

SCI Tool を実行して、SCI Tool 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。

SCI Tool は、SCI Tool 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。

第 7 章

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) 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。
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方法

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、DHCP を設定し、さらに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、サブネットマスク、ルーター、DNS 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。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できる必要があります。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。

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手順 SMF プロファイルファイルが含まれています。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、ターゲットデバイスやパーティション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、インスト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ことによって、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 DHCP によって 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 15 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、71 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。

また、初回ブート時に Oracle Configuration Manager はシステム構成情報を収集し、その情報を Oracle Support に送信しようとします。第 14 章「AI クライアントシステムで使用するための Oracle Configuration Manager の設定」を参照してください。

自動インストーラの使用事例

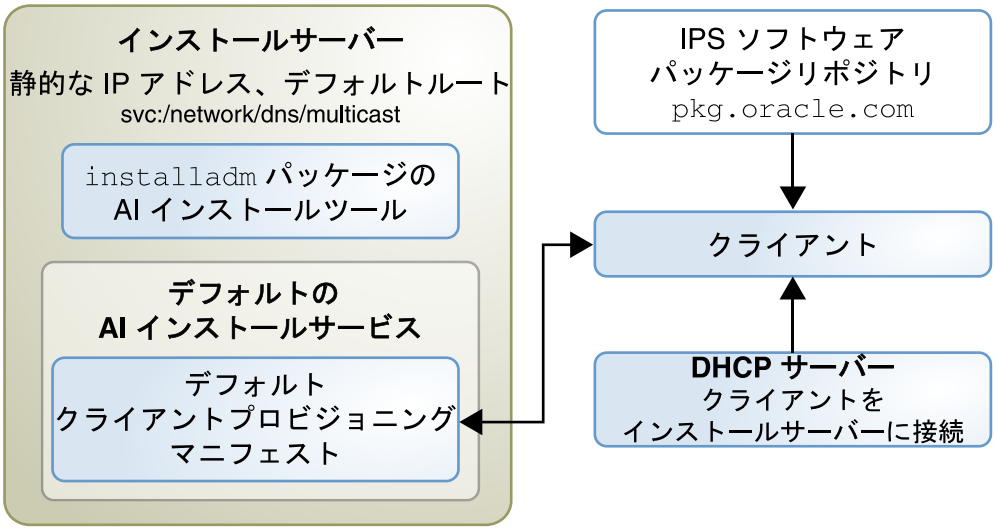
次の使用事例では、AI のもっとも明確な使用法について説明します。これらの使用事例は、相互に依存するものではありません。代わりに、各事例では AI の個別の機能について説明し、その機能に含まれない動作はすべて最小事例に含まれているものと同じになります。おそらく、これらの使用事例に記載されている機能を組み合わせて使うことになります。

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。これには、インターネット上のデフォルトの IPS パッケージリポジトリから入手できる最新バージョンの OS 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. それらのクライアントが DHCP サーバーにアクセスできることを確認します。
5. そのサービスのブートに必要な情報が DHCP 構成で使用できることを確認します。
6. 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できる必要があります。
7. クライアントをネットワーク経由でブートします。



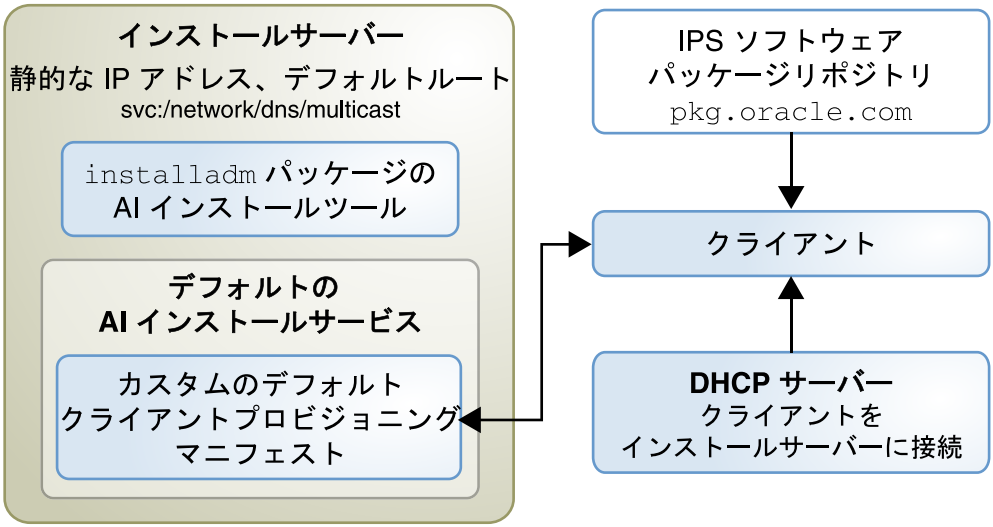
クライアントをネットワーク経由でブートすると、次のステップが実行されます。

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。
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。



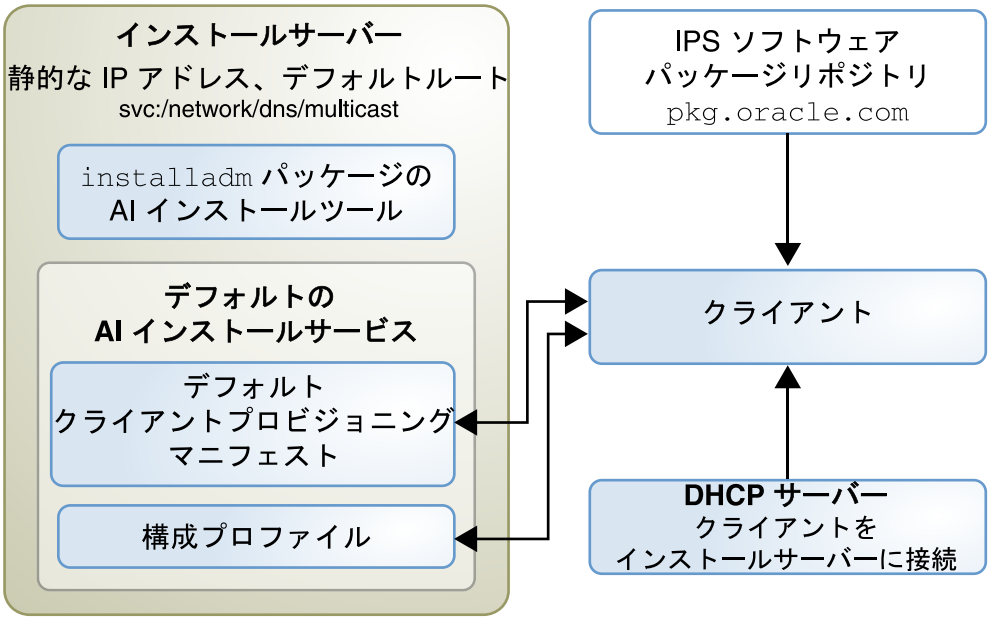
クライアントをネットワーク経由でブートすると、次のステップが実行されます。

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。
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。

システム構成手順を指定する

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。



クライアントをネットワーク経由でブートすると、次のステップが実行されます。

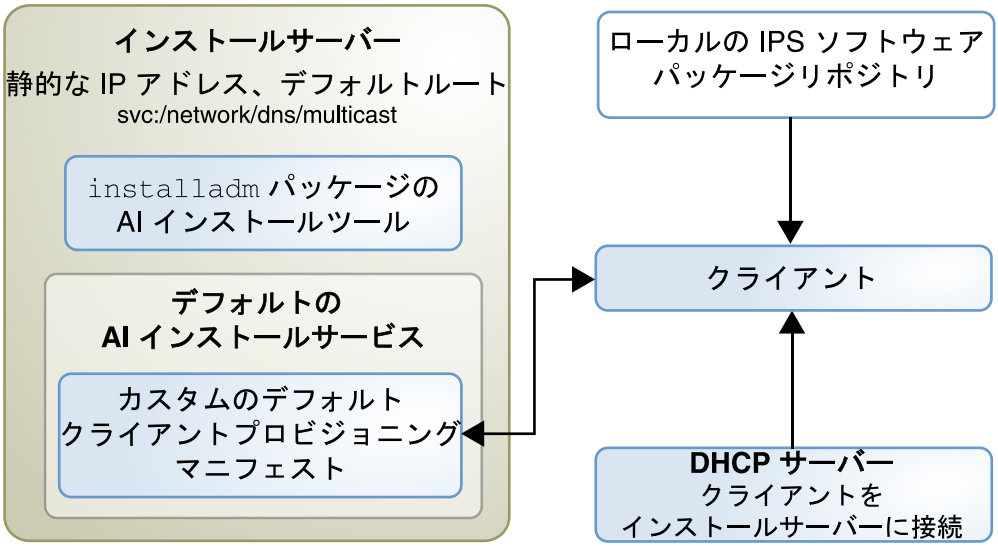
1. クライアントは DHCP サーバーからインストールサーバーアドレスを取得します。
2. アーキテクチャーが一致する場合、クライアントは `default-`*arch* インストールサービスを使用します。

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。

ローカルの IPS パッケージリポジトリを指定する

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必要な場合があります。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 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。



クライアントをネットワーク経由でブートすると、次のステップが実行されます。

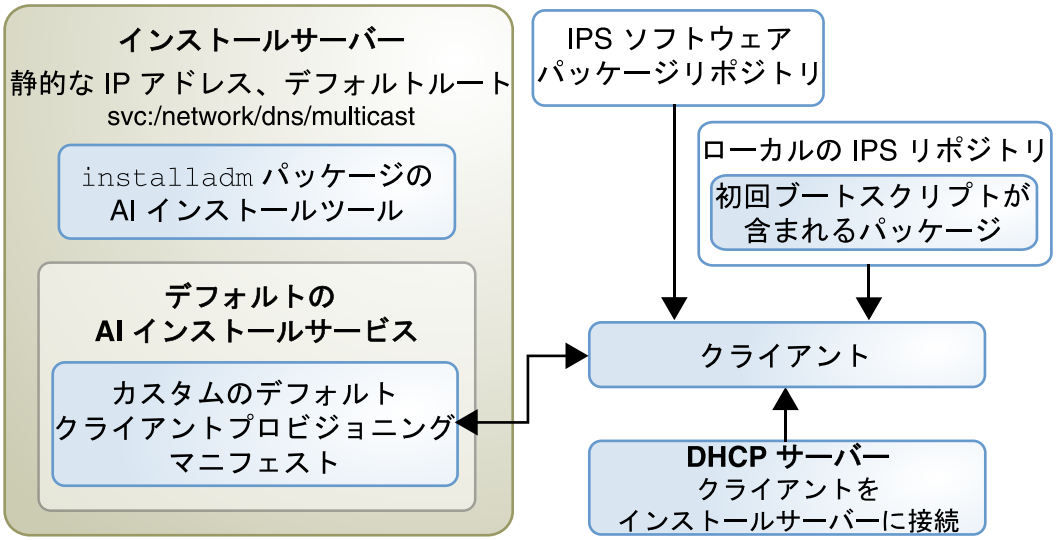
1. クライアントは DHCP サーバーからインストールサーバーアドレスを取得します。
2. アーキテクチャーが一致する場合、クライアントは `default-arch` インストールサービスを使用します。
3. カスタマイズされた AI マニフェストに従い、ローカルの IPS パッケージリポジトリを使用して、クライアントがプロビジョニングされます。
4. インストール後にクライアントがブートすると、システム構成プロファイルが指定されていないため、対話型ツールによってシステム構成情報の入力を求めるプロンプトが表示されます。

カスタムの初回ブートスクリプトを指定する

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。



クライアントをネットワーク経由でブートすると、次のステップが実行されます。

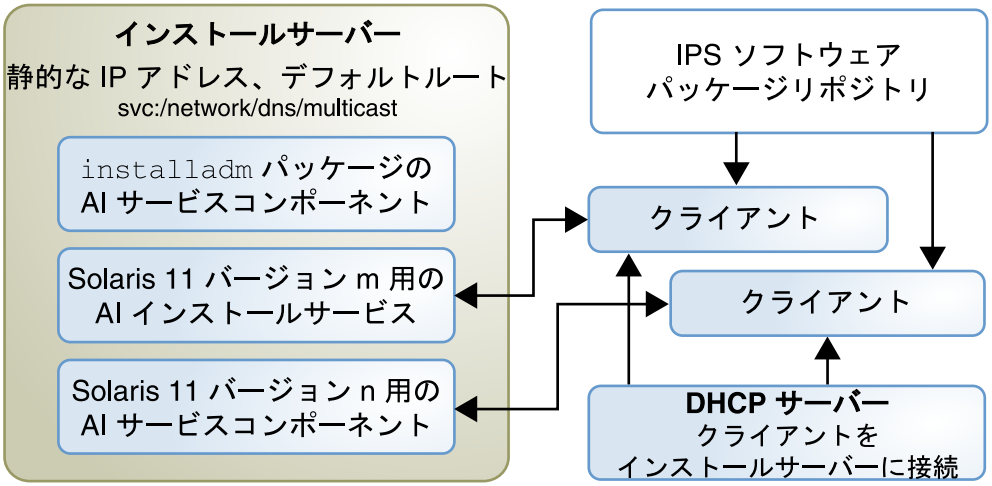
1. クライアントは DHCP サーバーからインストールサーバーアドレスを取得します。
2. アーキテクチャーが一致する場合、クライアントは `default-`*arch* インストールサービスを使用します。
3. 初回ブートスクリプトが含まれるカスタムパッケージのインストールを含む、カスタマイズされた AI マニフェストに従ってクライアントがプロビジョニングされます。
4. インストール後にクライアントがブートすると、システム構成プロファイルが指定されていないため、対話型ツールによってシステム構成情報の入力を求めるプロンプトが表示されます。
5. インストール後にクライアントがブートすると、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を指定する

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。



クライアントをネットワーク経由でブートすると、次のステップが実行されます。

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。
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。

第 8 章

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場合、AI ではインストールサーバーにするための個別のシステムが必要になります。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 Oracle Solaris 11 OS をインストールするための手順を提供します。

AI サーバー設定の作業マップ

次の作業マップに、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 作業マップ

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。	92 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、`svc:/network/dns/multicast` 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。	92 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。	94 ページの「AI インストールツールをインストールする」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。	97 ページの「AI インストールサービスを作成する」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 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 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「LiveCD 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。

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 OS を AI サーバーにインストールします。Oracle Solaris 11 を AI サーバーにインストールする場合は、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 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』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`(1M) コマンドを使用してネットワークのステータスを表示し、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、/etc/defaultrouter(4) ファイルにサーバーのネットワークの静的デフォルトルートの IP アドレスを設定することによって、静的デフォルトルートを設定できます。
マルチキャスト **DNS**	省略可能: svc:/network/dns/multicast SMF サービスを有効にします。svcs(1) コマンドを使用してこのサービスのステータスを確認してから、必要なら svcadm(1M) コマンドを使用してサービスを有効にします。例 8–1 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 install/installadm パッケージが必要です。
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。102 ページの「ローカル DHCP を設定してインストールサービスを作成する」 に、インストールサーバーで DHCP を設定する例が示されています。DHCP 構成の詳細は、『Oracle Solaris の管理: IP サービス』のパート II「DHCP」を参照してください。

例 8–1　マルチキャスト DNS の有効化

次のコマンドでは、svc:/network/dns/multicast SMF サービスのステータスを確認してから、そ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
```

AI インストールツールをインストールする

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、次の作業を実行できます。

- インストールサービスを作成して有効にします。
- 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、103 ページの「インストールサーバーを維持する」および『Oracle Solaris 11 インストールマニュアルページ』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 IPS パッケージリポジトリに installadm パッケージが含まれていることを確認します。

```
$ pkg list -a installadm
NAME (PUBLISHER)                                      VERSION                   IFO
install/installadm                                    0.5.11-0.175.0.0.0.0.1345 ---
```

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。

```
$ pkg list -av installadm
FMRI                                                                          IFO
pkg://solaris/install/installadm@0.5.11,5.11-0.175.0.0.0.0.1345:20110815T024057Z ---
```

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   URI
```

```
solaris                   origin  online  file:///export/Solaris11/
example.com  (non-sticky) origin  online  http://pkg.example.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。

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

DOWNLOAD                      PKGS       FILES    XFER (MB)
Completed                      1/1       66/66      0.3/0.3

PHASE                            ACTIONS
Install Phase                    119/119

PHASE                              ITEMS
Package State Update Phase           1/1
Image State Update Phase             2/2

PHASE                              ITEMS
Reading Existing Index               8/8
Indexing Packages                    1/1
Deleting content cache
```

```
$ 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: 0.175.0.0.0.0.1345
Packaging Date: Mon Aug 15 02:40:57 2011
          Size: 967.86 kB
          FMRI: pkg://solaris/install/installadm@0.5.11,5.11-0.175.0.0.0.0.1345:20110815T024057Z
```

インストールサーバーを構成する

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。

マルチホームインストールサーバーを構成する

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 プロパティーを調整します。

all_services/networks プロパティーの値は、CIDR 形式で示されたネットワークの一覧です (192.168.56.0/24 など)。all_services/exclude_networks プロパティーの値は、all_services/networks プロパティー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 プロパティー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 プロパティー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を構成する

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 プロパティーを構成します。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
# svccfg refresh system/install/server:default
```

注 – port プロパティーのカスタマイズは、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 プロパティーを変更した場合は、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。

AI インストールサービスを作成する

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す。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。意味のある名前をサービスに付け、そのサービスの作成場所となるパスを指定します。ネットワークブートイメージ (ネットイメージ) のパッケージまたは ISO ファイルのソースを指定します。

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、`menu.lst` 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。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。

別名の `default-`*`arch`* サービスが示す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` プロパティーを設定します。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` プロパティーの設定の詳細は、106 ページの「インストールサービスのプロパティーの変更」を参照してください。

`default-`*`arch`* の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は、このデフォルトの別名ブートファイルが、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。

新しい `default-`*`arch`* の別名が作成されたときにローカル ISC DHCP サーバーがすでに構成されている場合は、このアーキテクチャーのデフォルトのブートファイルが、この新しい別名のブートファイルに設定されます。

`default-`*`arch`* サービスを含む各サービスには、*`imagepath`*`/auto_install/manifest` 内にデフォルトの AI マニフェストがあります。このマニフェストは別のファイルにコ

ピーして編集できます。その後、create-manifest サブコマンドを使ってインストールサービスに追加できます。132 ページの「XML AI マニフェストファイルのカスタマイズ」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://*aiserver*:5555/export/aiserver/s11-ai-x86/s11-x86 のようになります。

すべてのオプションについては、104 ページの「インストールサービスを作成する」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 [-n *svcname*] [-s *FMRI_or_ISO*] [-d *imagepath*]

-n *svcname*
: 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。

-s *FMRI_or_ISO*
: *FMRI* は IPS AI ネットイメージパッケージの識別子であり、Oracle Solaris 11 リリースでは install-image/solaris-auto-install になります。*ISO* は AI ネットイメージ ISO ファイルのパス名です。

  FMRI_or_ISO を指定しない場合は、最新バージョンの install-image/solaris-auto-install パッケージが、pkg publisher リスト内のそのパッケージを提供する 1 番目の発行元からインストールされます。

  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。特定のパッケージリポジトリを指定するには、-p オプションを使用します。

-d *imagepath*
: *imagepath* は新しいインストールサービスの場所です。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 を指定しない場合は、サービスが /export/auto_install/*svcname* に作成され、自動生成された場所を使用すること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。

create-service コマンドは、102 ページの「ローカル DHCP を設定してインストールサービスを作成する」に示すように、AI インストールサーバーに DHCP を設定できます。別の DHCP サーバーを設定するか、または既存の DHCP サーバーを AI で使用するように設定する場合は、『Oracle Solaris の管理: IP サービス』の

パート II「DHCP」を参照してください。DHCP サーバーは、インストールされるシステムに DNS 情報を提供できる必要があります。

DHCP を設定せずにインストールサービスを作成する

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 ファイルを使用して SPARC インストールサービスを作成する

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。

```
# installadm create-service -n s11-sparc \
-s /var/tmp/images/sparc/sol-11-dev-170-ai-sparc.iso \
-d /install/images/s11-sparc

Creating service: s11-sparc

Setting up the target image at /install/images/s11-sparc ...
Service discovery fallback mechanism set up
Creating SPARC configuration file
Refreshing install services

Creating default-sparc alias.

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

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

上記の 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 s11-sparc という名前が付けられます。
2. インストールサービスのターゲットディレクトリ /install/images/s11-sparc が作成されます。
3. ISO ファイル /var/tmp/images/sparc/sol-11-dev-170-ai-sparc.iso がネットイメージの場所 /install/images/s11-sparc に展開されます。

4. このサービス用の wanboot.conf ファイルが /install/images/s11-sparc/wanboot.conf に生成されます。
5. /install/images/s11-sparc を /etc/netboot/s11-sparc としてマウントするように AI SMF サービス system/install/server が更新されます。
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 というサービスの別名が自動的に作成されます。s11-sparc のイメージが別名によって使用されるため、/install/images/s11-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。これは、-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当てはまります。

IPS パッケージを使用して x86 インストールサービスを作成する

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。このコマンドでは、インストールサービス名のオプションしか指定されていないため、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。このコマンドでは、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。

```
# installadm create-service -n s11-i386 -y

Creating service from: pkg:/install-image/solaris-auto-install
Download: install-image/solaris-auto-install ...  Done
Install Phase ...  Done
Package State Update Phase ...  Done
Image State Update Phase ...  Done
Reading Existing Index ...  Done
Indexing Packages ...  Done

Creating service: s11-i386

Image path: /export/auto_install/s11-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       : 10.134.125.136
        Boot file            : default-i386/boot/grub/pxegrub

Refreshing install services
```

上記の 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 s11-i386 という名前が付けられます。
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発行元リスト内のこのパッケージを提供する 1 番目の発行元から取得されます。
3. -d オプションでネットイメージの宛先が指定されていないため、イメージがデフォルトのディレクトリ /export/auto_install/s11-i386 に作成されます。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/s11-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、104 ページの「インストールサービスを作成する」を参照してください。
5. pxegrub メニューが /export/auto_install/s11-i386/menu.lst に作成されます。
6. /export/auto_install/s11-i386 を /etc/netboot/s11-i386 としてマウントするように AI SMF サービス system/install/server が更新されます。
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 というサービスの別名が自動的に作成されます。s11-i386 のイメージが別名によって使用されるため、/export/auto_install/s11-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/pxegrub 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-i386 の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。これは、-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当てはまります。

ローカル DHCP を設定してインストールサービスを作成する

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。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います。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、DNS を使用してホスト名を解決します。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す。

-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、要求された 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、-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。このアドレスは、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。その他のすべての構成では、これはソフトウェアで自動的に決定できます。

```
# installadm create-service -n s11-x86 \
-s /var/tmp/images/i386/sol-11-dev-171-ai-x86.iso \
-d /install/images/s11-x86 \
-i 10.80.239.150 -c 20

Creating service from: /var/tmp/images/i386/sol-11-dev-171-ai-x86.iso
Setting up the image ...

Creating service: s11-x86

Image path: /install/images/s11-x86

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

Refreshing install services

Creating default-i386 alias.

Setting the default PXE bootfile in the local DHCP configuration to
'default-i386/boot/grub/pxegrub'

Refreshing install services
```

上記の 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 s11-x86 という名前が付けられます。
2. インストールサービスのターゲットディレクトリ /install/images/s11-x86 が作成されます。
3. ISO ファイル /var/tmp/images/i386/sol-11-dev-171-ai-x86.iso がネットイメージの場所 /install/images/s11-x86 に展開されます。
4. pxegrub メニューが /install/images/s11-x86/menu.lst に作成されます。
5. /install/images/s11-x86 を /etc/netboot/s11-x86 としてマウントするように AI SMF サービス system/install/server が更新されます。
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 というサービスの別名が自動的に作成されます。s11-x86 のイメージが別名によって使用されるため、/install/images/s11-x86 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。
8. default-i386/boot/grub/pxegrub ブートファイルが 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。

インストールサーバーを維持する

AI インストールサーバーを設定したあとは、次の作業のいくつかを実行することができます。詳細は、installadm(1M) のマニュアルページを参照してください。

- 104 ページの「インストールサービスを追加、変更、または削除する」
- 108 ページの「クライアントをインストールサービスに関連付ける」
- 109 ページの「クライアント固有のインストール手順をインストールサービスに関連付ける」
- 116 ページの「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を一覧表示する」
- 116 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を一覧表示する」
- 117 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を一覧表示する」

インストールサービスを追加、変更、または削除する

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を作成する

インストールサービスを作成にするには、次のコマンドを使用します。例については、97 ページの「AI インストールサービスを作成する」を参照してください。

```
installadm create-service [-n svcname] [-s FMRI_or_ISO]
    [-p prefix=origin] [-a architecture]
    [-d imagepath] [-y] [-t existing_service]
    [-i dhcp_ip_start] [-c count_of_ipaddr]
    [-b boot_property=value,...] [-B server_ipaddr]
```

`-n` *svcname*	*svcname* は、英数字、アンダースコア (_)、ハイフン (-) で構成できます。*svcname* の最初の文字をハイフンにす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。
`-s` *FMRI_or_ISO*	このオプションは、ネットブートイメージのソースを指定します。*FMRI* は IPS AI ネットイメージパッケージの識別子であり、Oracle Solaris 11 リリースでは `install-image/solaris-auto-install` になります。AI ネットイメージ ISO ファイルを使用する場合は、ネットイメージ ISO ファイルのパス名を指定します。 *FMRI_or_ISO* を指定しない場合は、最新バージョンの `install-image/solaris-auto-install` パッケージが、`pkg publisher` リスト内のそのパッケージを提供する 1 番目の発行元からインストールされます。 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。特定のパッケージリポジトリを指定するには、`-p` オプションを使用します。

-p *prefix= origin*	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。*prefix* は発行元名であり、*origin* は URI (solaris=http://pkg.oracle.com/solaris/release/ など) です。 -s と -p が指定されない場合は、最新バージョンの install-image/solaris-auto-install パッケージが、pkg publisher リスト内のそのパッケージを提供する 1 番目の発行元からインストールされます。
-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* は、インストールされ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。 IPS パッケージからサービスを作成する場合、デフォルトでインストールされるパッケージの形式は、そのサービスが作成されるシステムのアーキテクチャーに適合する形式になります。たとえば、AI インストールサーバーが x86 の場合、create-service がデフォルトでインストールする solaris-auto-install パッケージの形式は i386 形式になります。SPARC クライアントをインストールするためのサービスを作成する場合は、-a sparc と指定して、sparc 形式の solaris-auto-install パッケージをインストールサービスにインストールします。
-d *imagepath*	*imagepath* は新しいインストールサービスの場所です。*imagepath* を指定しない場合は、サービスが /export/auto_install/*svcname* に作成され、自動生成された場所を使用すること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 *imagepath* の使用を確認するプロンプトを抑制するために、-y オプションを指定します。
-t *existing_service*	新しいサービスを別名として指定します。これは *existing_service* 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 *dhcp_ip_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 ISC DHCP 構成が存在しない場合は、ISC DHCP サーバー が起動されます。

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

-b *boot_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 menu.lst ファイルにプロパティー値を設定します。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ーを設定します。このオプションでは、コンマで区切った *boot_property= value* のペアを複数指定できます。

-B *server_ipaddr*
: 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ーの変更

svcname インストールサービスに設定するプロパティー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 prop= value svcname
```

prop=value ペアは、次のいずれかにする必要があります。

aliasof=*another_svcname*
: *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。

このプロパティーを設定すると、*svcname* サービスが *another_svcname* 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、116 ページの「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を一覧表示する」に示すように installadm list コマンドを使用します。

svcname または *another_svcname* 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ンディングは、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 AI と SMF DTD 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_or_script_name*
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スクリプトを、そのサービスのデフォルト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
```

インストールサービスの名前を変更する

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 は、英数字、アンダースコア (_)、ハイフン (-) で構成できます。*newsvcname* の最初の文字をハイフンにすることはできません。

インストールサービスを有効にしたり無効にしたりする

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削除する

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、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す。

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。

クライアントをインストールサービスに関連付ける

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、203 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。

クライアントをインストールサービスに追加する

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。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の説明に従って dladm 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、ISC DHCP 構成でクライアントが構成されます。

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 menu.lst ファイルにブートプロパティーを設定します。

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 s11-sparc インストールサービスに追加します。

```
# installadm create-client -e 00:14:4f:a7:65:70 -n s11-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。

```
# installadm create-client -e c0ffeec0ffee -n s11-x86 -b 'console=ttya'
```

クライアントを別のインストールサービスに関連付ける

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クライアントを削除する

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、ISC DHCP 構成でクライアントが構成解除されます。

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。

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

クライアント固有のインストール手順をインストールサービスに関連付ける

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。

AI マニフェストを追加する

manifest_or_script_filename カスタム AI マニフェストを *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 manifest_or_script_filename [-m manifest_or_script_name]
    [-c criteria=value|list|range...
    | -C criteriafile] [-d]
```

manifest_or_script_filename は AI マニフェストの XML ファイルにすることも、派生マニフェストスクリプトにすることもできます。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。`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、148 ページの「派生マニフェストスクリプトをインストールサービスに追加する」に示すように `aimanifest validate` コマンドを使用します。

manifest_or_script_name は、`installadm list` コマンドで表示される名前です。117 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を一覧表示する」を参照してください。*manifest_or_script_name* を指定しない場合、*manifest_or_script_name* は `ai_instance` 要素の `name` 属性の値 (存在する場合)、または *manifest_or_script_filename* のベース名になります。

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` と `-C` 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で、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 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 s11-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。

```
# installadm create-manifest -f ./mymanifests/manifest_t200.xml \
-m t200 -n s11-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。

```
# installadm create-manifest -f ./mymanifests/manifest_t200.xml \
-m t200 -n s11-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 マニフェストを更新する

svcname インストールサービスの *manifest_or_script_name* AI マニフェストの内容を *manifest_or_script_filename* AI マニフェストに置き換えるには、`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_or_script_name* は変更されません。

```
installadm update-manifest -n svcname -f manifest_or_script_filename [-m manifest_or_script_name ]
```

`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、148 ページの「派生マニフェストスクリプトをインストールサービスに追加する」に示すように `aimanifest validate` コマンドを使用します。

manifest_or_script_name 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。117 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を一覧表示する」を参照してください。

manifest_or_script_name を指定しない場合は、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。

- *manifest_or_script_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_or_script_name* に一致する場合)。
- *manifest_or_script_filename* マニフェストのベース名 (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_or_script_name* に一致する場合)。

次のコマンドは、`s11-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。

```
# installadm update-manifest -n s11-sparc \
-f ./mymanifests/manifest_newt200.xml -m t200
```

AI マニフェストを削除する

manifest_or_script_name AI マニフェストを *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_or_script_name* は、`installadm list` コマンドで返されるマニフェスト名です。117 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を一覧表示する」を参照してください。

```
installadm delete-manifest -m manifest_or_script_name -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 s11-sparc インストールサービスから削除します。

```
# installadm delete-manifest -m t200 -n s11-sparc
```

クライアント固有の構成手順をインストールサービスに関連付ける

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。

システム構成プロファイルを追加する

profile_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 profile_filename... [-p profile_name]
    [-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。

profile_filename には、create-profile コマンドで指定された条件から、または環境変数から値を取得する置換タグを含めることができます。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。

profile_name は、installadm list コマンドで表示される名前です。117 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を一覧表示する」を参照してください。*profile_name* を指定しない場合、*profile_name* は *profile_filename* のベース名になります。-p オプションは、*profile_filename* を複数指定した場合は有効ではありません。

-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 s11-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。

```
# installadm create-profile -f ./mymanifests/profile_t200.xml \
-p t200 -n s11-sparc -c platform="SUNW,Sun-Fire-T200"
```

システム構成プロファイルを検証する

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 profile_filename... | -p profile_name...
```

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。*profile_filename* はそのファイルへのフルパス名です。

117 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を一覧表示する」に示すように create-profile 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。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、*profile_filename* と *profile_name* 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を削除する

profile_name 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_name* は、`installadm list` コマンドで返されるプロファイル名です。117 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を一覧表示する」を参照してください。

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

次のコマンドは、`t200` システム構成プロファイルを `s11-sparc` インストールサービスから削除します。

```
# installadm delete-profile -p t200 -n s11-sparc
```

AI マニフェストまたはシステム構成プロファイルをエクスポートする

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_or_script_name... -p profile_name...
    [-o pathname]
```

pathname を指定しない場合は、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* 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_or_script_name は、XML AI マニフェストにも派生マニフェストスクリプトにもできます。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を行います。

- 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を変更する

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てある 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_or_script_name -p profile_name... -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_or_script_name* と *profile_name* の名前は、installadm list コマンドで返される名前です。117 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を一覧表示する」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。

```
# installadm set-criteria -m t200 -n s11-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を表示する

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インストールサービスを一覧表示する

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、2 つの有効なインストールサービスが見つかります。無効なサービスの「Status」の値は off になります。指定のアーキテクチャー用に作成された最初のサービスは、そのアーキテクチャーのクライアントのデフォルトサービスです。104 ページの「インストールサービスを追加、変更、または削除する」を参照してください。

```
$ installadm list

Service Name  Alias Of  Status Arch  Image Path
------------  --------  ------ ----  ----------
default-i386  s11-x86   on     x86   /install/images/s11_x86
default-sparc s11-sparc on     Sparc /install/images/s11_sparc
s11-sparc               on     Sparc /install/images/s11_sparc
s11-x86                 on     x86   /install/images/s11_x86
```

指定したインストールサービスの情報を表示する

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 s11-sparc

Service Name  Alias Of  Status Arch  Image Path
------------  --------  ------ ----  ----------
s11-sparc               on     Sparc /install/images/s11_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を一覧表示する

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。108 ページの「クライアントをインストールサービスに追加する」を参照してください。

```
$ installadm list -c

Service Name Client Address    Arch  Image Path
```

```
------------ --------------    ----  ----------
s11-sparc    00:14:4F:A7:65:70 Sparc /install/images/s11_sparc
s11-x86      08:00:27:8B:BD:71 x86   /install/images/s11_x86
             01:C2:52:E6:4B:E0 x86   /install/images/s11_x86
```

特定のインストールサービスに関連付けられているクライアントを一覧表示する

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 `s11-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 s11-sparc

Service Name Client Address    Arch  Image Path
------------ --------------    ----  ----------
s11-sparc    00:14:4f:a7:65:70 Sparc /install/images/s11_sparc
```

カスタマイズしたインストールの情報を表示する

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を一覧表示する

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すべて一覧表示します。「Manifest」または「Profile」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す。

```
# installadm list -m -p

Service Name  Manifest
------------  --------
s11-sparc     t200
s11-x86       ipv4
              mem1

Service Name  Profile
------------  --------
s11-sparc     mac1
              t200
s11-x86       mac2
              mac3
              ipv4
              mem1
```

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を一覧表示する

次の例では、インストールサービス s11-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。「Manifest」または「Profile」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、この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 -n s11-sparc

Manifest      Status    Criteria
--------      ------    --------
orig_default  Inactive  None
mem1          Default   (Ignored: mem = 2048 - 4095)
t200                    platform = SUNW,Sun-Fire-T200
                        mem = 4096-unbounded

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

AI SMF 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 8–2　AI SMF 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。AI SMF 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

例 8–2　AI SMF サービスの有効化　　*(続き)*

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、AI SMF サービスは保守モードになります。

例 8–3　AI SMF サービスの無効化

AI SMF 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、AI SMF 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、116 ページの「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を一覧表示する」を参照してください。

第 9 章

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 DHCP サーバーと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。クライアントがブートすると、そのクライアントは DHCP によってインストールサーバーに接続されます。

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を使用するか、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 15 章「クライアントシステムのインストール」を参照してください。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 DHCP によって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の説明に従って `dladm` コマンドを使用します。

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。109 ページの「クライアント固有のインストール手順をインストールサービスに関連付ける」を参照してください。

カスタムの AI マニフェストを作成する場合は、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を作成する場合は、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、127 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。109 ページの「AI マニフェストを追加する」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。hostname という条件のキーワードは構成プロファイルにのみ使用でき、AI マニフェストには使用できません。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す。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。たとえば、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

次の表には、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が記載されています。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。

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 zonename="zone1 zone2"
-c mem="2048-unbounded"
```

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。	**CLI**、単一の **MAC** アドレス:
		`-c mac="0:14:4F:20:53:97"`
		CLI、**MAC** アドレスの範囲:
		`-c mac=0:14:4F:20:53:94-0:14:4F:20:53:A0`
		XML、単一の **MAC** アドレス:
		`<ai_criteria name="mac">`
		`  <value>0:14:4F:20:53:97</value>`
		`</ai_criteria>`
		XML、**MAC** アドレスの範囲:
		`<ai_criteria name="mac">`
		`  <range>`
		`  0:14:4F:20:53:94`
		`  0:14:4F:20:53:A0`
		`  </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

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	**CLI、単一の IP アドレス:**
		`-c ipv4="10.6.68.127"`
		CLI、IP アドレスの範囲:
		`-c ipv4="10.6.68.1-10.6.68.200"`
		XML、単一の IP アドレス:
		`<ai_criteria name="ipv4">`
		`  <value>10.6.68.127</value>`
		`</ai_criteria>`
		XML、IP アドレスの範囲:
		`<ai_criteria name="ipv4">`
		`  <range>`
		`  10.6.68.1`
		`  10.6.68.200`
		`  </range>`
		`</ai_criteria>`

platform	uname -i 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>`

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

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>`

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。
		CLI、1 つのメモリーサイズ:
		`-c mem="4096"`
		CLI、メモリーサイズの範囲:
		`-c mem="2048-unbounded"`
		XML、1 つのメモリーサイズ:
		`<ai_criteria name="mem">`
		`  <value>4096</value>`
		`</ai_criteria>`
		XML、メモリーサイズの範囲:
		`<ai_criteria name="mem">`
		`  <range>`
		`  2048`
		`  unbounded`
		`  </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

zonename	zoneadm list によって示される zones(5) 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。	**CLI**、単一のゾーン名:
		`-c zonename="myzone"`
		CLI、ゾーン名のリスト:
		`-c zonename="zoneA zoneB zoneC"`
		XML、単一のゾーン名:
		`<ai_criteria name="zonename">`
		`<value>myzone</value>`
		`</ai_criteria>`
		XML、ゾーン名のリスト:
		`<ai_criteria name="zonename">`
		`<value>zoneA zoneB zoneC</value>`
		`</ai_criteria>`

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。
	この条件のキーワードはシステム構成プロファイルにのみ使用でき、AI マニフェストには使用できません。
		CLI、単一のホスト名:
		`-c hostname="host3"`
		CLI、ホスト名のリスト:
		`-c hostname="host1 host2 host6"`
		XML、単一のホスト名:
		`<ai_criteria name="hostname">`
		`<value>host3</value>`
		`</ai_criteria>`
		XML、ホスト名のリスト:
		`<ai_criteria name="hostname">`
		`<value>host1 host2 host6</value>`
		`</ai_criteria>`

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の AI マニフェストになります。

このデフォルトの AI マニフェストを下に示します。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。インストールするべき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ます。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 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」に記載されているように、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 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, 2011, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.@DTD_VERSION_AI@">
<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">
      <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>
      <!--
        By default the latest build available, in the specified IPS
        repository, is installed.  If another build is required, the
        build number has to 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@latest</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

第 10 章

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、カスタムの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マニフェストまたはスクリプトを使用するかは、そのマニフェストまたはスクリプ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マニフェストまたはスクリプ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マニフェストまたはスクリプ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、次の手順に従います。

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、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。

   list サブ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 s11-x86

   Manifest          Status   Criteria
   -----------       -------  --------
   orig_default      Default  None
   ```

   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 マニフェストをコピーします。

   ```
   # installadm export -n s11-x86 -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. 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 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。

   ```
   # installadm create-manifest -n s11-x86 -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 -n s11-x86 -m

   Manifest          Status   Criteria
   -----------       -------  --------
   orig_default      Default  None
   mem1                       mem  = 2048 MB - unbounded
   ```

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。

```
# installadm set-service -o default-manifest=mem1 s11-x86
# installadm list -n s11-x86 -m
```

```
Manifest          Status    Criteria
-----------       -------   --------
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。

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、クライアント条件は指定しません。

```
# installadm create-manifest -n s11-x86 -d \
-f ./region1.xml -m region1
# installadm list -n s11-x86 -m

Manifest          Status    Criteria
-----------       -------   --------
orig_default      Inactive  None
mem1                        mem  = 2048 MB - unbounded
region1           Default   None
```

インストールサービスにすでに追加されている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 *manifest_or_script_name* は変更されません。

```
# installadm update-manifest -n s11-x86
    -f ./newregion1.xml -m region1
```

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 マニフェストを構文的に検証します。

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。

---

list、export、create-manifest、set-criteria、update-manifest、および set-service サブコマンドの詳細は、103 ページの「インストールサーバーを維持する」を参照してください。

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には、次の手順に従います。

1. 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. 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. 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。

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/sadm/system/logs/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/sadm/system/logs/derived/manifest_script` にコピーされます。

派生マニフェストスクリプトを作成する

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。

クライアントの属性を取得する

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 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も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
SI_ARCH	インストールされるクライアントのアーキテクチャー。uname -p の出力と同等です。
SI_CPU	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。uname -p の出力と同等です。
SI_NUMDISKS	クライアントのディスク数。
SI_DISKNAME_#	クライアントで見つかったディスクの ctds 名を表す平坦な変数セット。SI_NUMDISKS 個の SI_DISKNAME_# 変数が存在することになります。ここで、# は 1 から SI_NUMDISKS までの整数に置き換えられます。この変数セットは、SI_DISKSIZE_# で記述される変数セットと相関関係にあります。

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_DISKSIZE_#	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。SI_NUMDISKS 個の SI_DISKSIZE_# 変数が存在することになります。ここで、# は 1 から SI_NUMDISKS までの整数に置き換えられます。この変数セットは、SI_DISKNAME_# 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。
SI_HOSTADDRESS	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。
SI_HOSTNAME	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。
SI_KARCH	クライアントのカーネルアーキテクチャー。uname -m の出力と同等です。
SI_INSTALL_SERVICE	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。
SI_MANIFEST_SCRIPT	マニフェストスクリプトの URL。
SI_MEMSIZE	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。
SI_MODEL	クライアントのモデル名。uname -i の出力と同等です。
SI_NATISA	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。isainfo -n の出力と同等です。
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	クライアントのプラットフォーム。uname -i の出力と同等です。

AI マニフェストをカスタマイズする

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 が変更できるファイルには、少なくとも次の両方の要素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:///image_path/auto_install/ai.dtd.#">
<auto_install/>
```

は 1 などの整数です。*image_path* は次のコマンドで返されるパスです。ここで、*service_name* 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 service_name
```

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 *image_path*/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す。

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。aimanifest コマンドの使用例については、aimanifest(1M) のマニュアルページと下のスクリプト例を参照してください。

aimanifest コマンドで指定した値にスラッシュ (/)、一重引用符、二重引用符などの特殊文字が含まれている場合は、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よい見本です。

```
#!/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、ディスクのサイズが 1T バイトを超える場合に Solaris fdisk パーティションにターゲットディスクの半分だけ使用するように 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。

```
#!/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

# Assume there is only one disk on the system for this example.
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/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
fi
exit $SCRIPT_SUCCESS
```

一部のクライアントでは、次の要素が $AIM_MANIFEST に追加されます。

```
<target>
  <disk>
    <partition action="use_existing_solaris2"/>
  </disk>
</target>
```

ほかのクライアントでは、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>
    <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
}

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 2 つのディスクを備えたシステムでは、この例の出力は次の XML 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">
```

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)

```
    <disk_name name="c0t1d0" name_type="ctd"/>
  </disk>
  <logical>
    <zpool name="rpool">
      <vdev name="mirror_vdev" redundancy="mirror"/>
    </zpool>
  </logical>
</target>
```

例 10–3 指定された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 2 つの 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
```

例 10–3 指定された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)

```
        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)
    /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 Kornshell 版を Python 版にしたものです。

```
import os
from subprocess import call, check_call, CalledProcessError

SCRIPT_SUCCESS = 0
SCRIPT_FAILURE = 1

def main():

    # Find the disks first.
    found_1 = ""
    found_2 = ""

    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.
```

例 10–3 指定された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)

```
    try:
        subprocess.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:            subprocess.check_call(["/usr/bin/aimanifest", "set",
                "target/logical/zpool/vdev@redundancy", "mirror"])
            subprocess.check_call(["/usr/bin/aimanifest", "set",
               "target/logical/zpool/vdev[@redundancy='mirror']@name", "mirror_vdev"])

            subprocess.check_call(["/usr/bin/aimanifest", "add",
                "target/disk/disk_name@name", "found_1"])
            subprocess.check_call(["/usr/bin/aimanifest", "set",
                "target/disk/disk_name[@name='" + found_1 + "']" + "@name_type", "ctd"])
            subprocess.check_call(["/usr/bin/aimanifest", "set",
                "target/disk[disk_name@name='" + found_1 + "']" + "@in_vdev", "mirror_vdev"])

            subprocess.check_call(["/usr/bin/aimanifest", "add",
                "target/disk/disk_name@name", "found_2"])
            subprocess.check_call(["/usr/bin/aimanifest", "set",
                "target/disk/disk_name[@name='" + found_2 + "']" + "@name_type", "ctd"])
            subprocess.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
}
```

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)

```
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.
# 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 ターゲットディスクが必ず特定のサイズになるように指定

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}
```

例 10–5 ターゲットディスクが必ず特定のサイズになるように指定 (続き)

```
    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

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ます。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、どのパッケージを変更してよいか判断できないため、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。要素値はそのように指定しません。

例 10–6 マニフェスト仕様が正しくないスクリプト (続き)

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
```

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:///image_path/auto_install/ai.dtd.#">
      <auto_install/>
      ```

      # は 1 などの整数です。*image_path* は次のコマンドで返されるパスです。ここで、*service_name* 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 service_name
      ```

   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。上記の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。

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には、次の方法を使用します。

1. 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. `wget` または `sftp` を使用して、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。
4. 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。
   - 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。

   ```
   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。

派生マニフェストスクリプトをインストールサービスに追加する

スクリプトを AI インストールサービスに追加する方法は、XML マニフェストをインストールサービスに追加する場合と同じです。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。

```
# installadm create-manifest -n s11-x86 -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。

103 ページの「インストールサーバーを維持する」、export、create-manifest、set-criteria、update-manifest、および set-service サブコマンドについては、Maintain an Install Server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

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) プロパティー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 プロパティーを設定します。

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 プロパティーを処理し、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ー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、71 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、75 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。

- 153 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165 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す。

  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。
- プロファイルがインストールサービスに追加されるときに有効な値に置き換えられる変数をシステム構成プロファイルに含めることができます。163 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、113 ページの「システム構成プロファイルを検証する」を参照してください。

163 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」の説明に従ってプロファイルテンプレートを検証するには、環境変数に変数値を指定します。

```
# export AI_HOSTNAME=server1;export AI_IPV4=10.0.0.1;export AI_NETWORK=10.0.0.0
# installadm validate -n svc1 -P /export/hostIPnet.xml
```

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もあります。特定のプロパティー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ーの複数の指定を受け取る場合は、そのプロパティー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す。

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。

create-profile、list、set-criteria、export サブコマンドの詳細は、103 ページの「インストールサーバーを維持する」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

構成可能な項目の構成は、smf(5) プロパティー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 DNS (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ーを指定した場合、その指定は無視されます。特定のプロパティーを何度も指定しないでください。

どの SMF プロパティーを指定する必要があるか確信を持てない場合は、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ーグループやプロパティーの説明 (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「プロパティー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*]

プロパティーグループや特定のプロパティーのクエリー検索を行うには、プロパティーグループ名か、プロパティーグループ名とプロパティー名をスラッシュ (/) 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 オプションを指定した場合は、選択内容のテンプレートデータのみが表示され (smf_template(5) を参照)、プロパティーグループやプロパティー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

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、sysconfig create-profile コマンドを users グループとともに使用します。

```
# 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 つのプロパティーグループを認識します。

- root_account プロパティーグループには、root アカウントを構成する SMF プロパティーが含まれています。
- user_account プロパティーグループには、ユーザーアカウントを構成する SMF プロパティーが含まれています。

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 プロパティーグループには、次のプロパティーが含まれています。

表 11–1　root_account プロパティーグループのプロパティー

プロパティー	タイプ	必要	説明
password	astring	はい	暗号化された root パスワードを指定します。root パスワードを指定しない場合、root パスワードは空になります。
type	astring	いいえ	アカウントタイプ: normal または role。デフォルトは normal です。
expire	string	いいえ	ログインの有効期限。0 (ゼロ) に設定した場合、ユーザーは次回のログイン時に root 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

user_account プロパティーグループには、次のプロパティーが含まれています。

表 11–2　user_account プロパティーグループのプロパティー

プロパティー	タイプ	必要	説明
login	astring	yes	ユーザーのログイン。
password	astring	yes	暗号化されたユーザーパスワード。
description	astring	no	通常はユーザーのフルネーム。
shell	astring	no	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。
uid	count	no	新しいユーザーの UID。デフォルトの UID は 101 です。
gid	count	no	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。
type	astring	no	アカウントタイプ: normal または role。デフォルトは normal です。

表 11–2 user_account プロパティーグループのプロパティー (続き)

プロパティー	タイプ	必要	説明
profiles	astring	no	prof_attr (4) で定義された、コンマで区切られた 1 つ以上の実行プロファイル。
roles	astring	no	user_attr(4) で定義された、コンマで区切られた 1 つ以上の役割。
sudoers	astring	no	sudoers(4) ファイルに login とともに追加されるエントリ。
expire	astring	no	ログインの有効期限。0 (ゼロ) に設定した場合、ユーザーは次回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。
home_zfs_dataset	astring	no	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 *root_pool* /export/home/*login* です。
home_mountpoint	astring	no	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 /export/home/ *login* です。

システム識別情報

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 プロパティーグループには、次のプロパティーが含まれています。

表 11–3 identity プロパティーグループのプロパティー

プロパティー	タイプ	必要	説明
nodename	astring	いいえ	システムのホスト名。デフォルトは unknown 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"/>
```

例 11–2 ホスト名の構成 (続き)

```
        </property_group>
    </instance>
</service>
```

タイムゾーンとロケール

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 プロパティーグループには、次のプロパティーが含まれています。

表 11–4 timezone プロパティーグループのプロパティー

プロパティー	タイプ	必要	説明
localtime	astring	いいえ	システムのタイムゾーン。デフォルトは UTC です。

例 11–3 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 プロパティーグループは、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 プロパティーグループのプロパティー

環境変数	タイプ	必要	デフォルト値
LC_CTYPE	astring	いいえ	C
LC_NUMERIC	astring	いいえ	C

表 11–5 environment プロパティーグループのプロパティー (続き)

環境変数	タイプ	必要	デフォルト値
LC_TIME	astring	いいえ	C
LC_COLLATE	astring	いいえ	C
LC_MONETARY	astring	いいえ	C
LC_MESSAGES	astring	いいえ	C
LC_ALL	astring	いいえ	C
LANG	astring	いいえ	C

例 11–4 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'>
      <propval name='LC_ALL' value='cs_CZ.UTF-8'/>
    </property_group>
  </instance>
</service>
```

端末タイプとキー配列

例 11–5 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 プロパティーの定義については、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 11–6 キー配列の構成

キー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 kdb_layout グループとともに使用します。

```
# sysconfig create-profile -g kdb_layout -o sc_kdb.xml
```

例 11–6 キー配列の構成 (続き)

svc:/system/keymap SMF サービスは、キー配列を構成します。関連する SMF プロパティーの定義については、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ー値で最初は無効になっています。

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 つのプロパティーグループを定義します。IPv4 のインタフェースに 1 つのプロパティーグループ、IPv6 のインタフェースに 1 つのプロパティーグループです。このサービスは、そのプロパティーと 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、そのプロパティーと route(1M) を使用してデフォルトルートを定義します。

167 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 プロパティーグループには、次のプロパティーが含まれています。

表 11–6 install_ipv4_interface プロパティーグループのプロパティー

プロパティー	タイプ	必要	説明
name	astring	yes	ネットワークインタフェースの名前。

表 11–6 install_ipv4_interface プロパティーグループのプロパティー (続き)

プロパティー	タイプ	必要	説明
address_type	astring	yes	ipadm (1M) create-addr サブコマンドの -T オプションの構築に使用する値。有効な値は static または dhcp です。
static_address	net_address_v4	no	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	astring	no	dhcp の address_type でのみ適用されます。このプロパティーを定義した場合は、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	no	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 プロパティーのインタフェース名の部分です。

install_ipv6_interface プロパティーグループには、次のプロパティーが含まれています。

表 11–7 install_ipv6_interface プロパティーグループのプロパティー

プロパティー	タイプ	必要	説明
name	astring	yes	ネットワークインタフェースの名前。
address_type	astring	yes	ipadm (1M) create-addr サブコマンドの -T 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	no	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	no	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	astring	no	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す。

表 11–7 install_ipv6_interface プロパティーグループのプロパティー (続き)

プロパティー	タイプ	必要	説明
stateful	astring	no	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す。
default_route	net_address_v6	no	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 プロパティー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す。このサービスは、1 つのプロパティーグループ (config) を定義します。このサービスは、そのプロパティーを使用して DNS の resolv.conf(4) ファイルを構築します。

config プロパティーグループには、次のプロパティーが含まれています。

表 11–8 config プロパティーグループのプロパティー

プロパティー	タイプ	必要	説明
domain	astring	no	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	yes	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	no	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ー値で最初は無効になっています。169 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。

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、インストールサーバー環境から、または create-profile サブコマンドで指定された条件から構成パラメータを取得す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 6–1 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、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、そして AI_NETWORK はクライアントシステムの IP バージョン 4 ネットワーク番号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="{{AI_NETWORK}}"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

次のコマンドでは、入力の hostandIP.xml ファイルを変更しないで、カスタマイズされたシステム構成プロファイルをインストールサービスに作成しています。

```
# installadm create-profile -n svc1 -f /export/hostIPnet.xml \
-p server1 -c hostname="server1" -c ipv4="10.0.0.1" -c network="10.0.0.0"
# installadm create-profile -n svc1 -f /export/hostIPnet.xml \
-p server2 -c hostname="server2" -c ipv4="10.0.0.2" -c network="10.0.0.0"
$ installadm list -n svc1 -p
Profile  Criteria
-------  --------
server1  hostname = server1
         ipv4 = 10.0.0.1
         network = 10.0.0.0
server2  hostname = server2
         ipv4 = 10.0.0.2
         network = 10.0.0.0
```

hostandIP.xml ファイルは変わらないまま、svc1 インストールサービスの内部にある server1 および server2 プロファイルがカスタマイズされます。たとえば、server1 プロファイルの内容は次のとおりです。

```
# installadm export -n svc1 -p server1
<?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="10.0.0.1/8"/>
        <propval name="default_route" type="net_address_v4" value="10.0.0.0"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

環境変数を設定することで、これと同じ結果を得ることができます。MAC アドレスやホスト名など、このクライアントを一意に識別する -c オプションを 1 つ指定して、これらの構成値がそのクライアントにのみ適用されるようにする必要があります。

```
# export AI_HOSTNAME=server1;export AI_IPV4=10.0.0.1;export AI_NETWORK=10.0.0.0
# installadm create-profile -n svc1 -f /export/hostIPnet.xml \
-p server1 -c mac="aa:bb:cc:dd:ee:ff"
```

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。これらの変数は、環境変数としても指定できます。

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	条件の名前	説明
AI_ARCH	arch	uname -m からのカーネルアーキテクチャー
AI_CPU	cpu	uname -p からのプロセッサタイプ
AI_HOSTNAME	hostname	クライアントの DNS 名
AI_IPV4	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
AI_MAC	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。
AI_MEM	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

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	条件の名前	説明
AI_NETWORK	network	IP バージョン 4 ネットワーク識別子、またはネットワーク識別子の範囲
AI_SERVICE		インストールサービス名
AI_ZONENAME	zonename	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, 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
* Timezone 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="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='default' enabled='true'>
    <property_group name='environment'>
      <propval name='LC_ALL' 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='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。

このプロファイル例では、次のパラメータを構成します。

- 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 (ローカルの DNS ドメイン名として)
- example2.com と example3.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。

netmask 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

```
<!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"/>
          <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='default' enabled='true'>
      <property_group name='environment'>
        <propval name='LC_ALL' 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='net0/v4'/>
              <propval name='address_type' type='astring' value='static'/>
              <propval name='static_address' type='net_address_v4' value='x.x.x.x/n'/>
              <propval name='default_route' type='net_address_v4' value='x.x.x.x'/>
          </property_group>

          <property_group name='install_ipv6_interface' type='application'>
              <propval name='name' type='astring' value='net0/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='x.x.x.x'/>
                </net_address_list>
            </property>
            <property name='search'>
                <astring_list>
                    <value_node value='example.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 naming_services 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、75 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 11–7　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 11–8　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 11–8 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 11–9 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、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 11–9 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 11–10 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.oracle.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.oracle.com oracle.com oraclecorp.com' />
```

例 11–10 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>
    <instance name='default' enabled='true' />
    </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 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='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 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 11–11 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>
```

例 11–11　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 11–12　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。これは必須で、`dc=my,dc=domain,dc=com` 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 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 11–12 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 11–13 セキュリティー保護された 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。これは必須で、`dc=my,dc=domain,dc=com` です
- 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。セキュリティー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 11–13 セキュリティー保護された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="ldapserver.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 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>
```

第 12 章

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、180 ページの「大域ゾーンの AI マニフェストでのゾーン指定」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、181 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 マニフェストでのゾーン指定

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>
```

```
    <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、183 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、184 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、71 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 s11-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。

```
# installadm create-manifest -n s11-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 s11-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。

```
# installadm create-profile -n s11-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 s11-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。

```
# installadm create-profile -n s11-sparc -f /tmp/z2profile.xml \
-p z2profile -c zonename=zone2
```

次の例は、s11-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 s11-sparc -m -p

Manifest         Status   Criteria
--------         ------   --------
orig_default     Default  None
line1-netra2000           mac       = 00:14:4F:2D:7A:DC
zmanifest                 zonename  = zone1 zone2

Profile    Criteria
-------    --------
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。

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

 Copyright (c) 2011, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.@DTD_VERSION_AI@">

<auto_install>
    <ai_instance name="zone_default">
        <target>
            <logical>
                <zpool name="rpool">
                    <filesystem name="export" mountpoint="/export"/>
                    <filesystem name="export/home"/>
                    <be name="solaris">
                        <options>
                            <option name="compression" value="on"/>
```

例 12–1 ゾーンのデフォルトの AI マニフェスト (続き)

```
                        </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>
                    <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。DNS は構成できます。

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ー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>
```

第 13 章

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。

初回ブート時に実行されるスクリプトの作成

189 ページの「SMF マニフェストファイルの作成」に示す SMF サービスマニフェストの初めの方で、サービスが次の行によって有効になります。

```
<create_default_instance enabled='true' />
```

初回ブートスクリプトの終わりで、初回ブートスクリプトが 1 度しか実行されないようにサービスを無効にし、パッケージをアンインストールします。

```
#!/bin/sh

svcadm disable svc:/site/first-boot-script-svc:default
pkg uninstall pkg:/first-boot-script
```

```
exit $SMF_EXIT_OK
```

この例では、first-boot-script-svc は189 ページの「SMF マニフェストファイルの作成」で作成された SMF サービス、および first-boot-script は191 ページの「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 IPS パッケージです。

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リブートする必要がある場合は、そのリブートをスクリプトの最後のアクションにしてください。

例 13–1　初回ブートスクリプトの例

この例は、/opt/site/first-boot-script.sh という名前の初回ブートスクリプトの例を示しています。このスクリプトでは最初に、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) のコピーを保存します。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と、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。

```
#!/bin/sh

# Load SMF shell support definitions
. /lib/svc/share/smf_include.sh

echo "Save original boot environment first."
# 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

# Add support for faster serial console
echo "Setting up support for faster serial console"
! grep console115200 >/dev/null /etc/ttydefs && \
    echo "console115200:115200 hupcl opost onlcr:115200::console115200" \
    >>/etc/ttydefs

echo "Configure ssh server for root login and X11 forwarding"
ed - << EOF
r /etc/ssh/sshd_config
/PermitRootLogin/
c
PermitRootLogin yes
.
/X11Forwarding/
c
X11Forwarding yes
.
w
q
```

例 13–1 初回ブートスクリプトの例 (続き)

```
EOF
svcadm refresh ssh

# Set up coreadm
echo "Setting core file configuration"
coreadm -G default -g /var/cores/%f.%u.%p.%t.core
coreadm -e global
coreadm -e process
coreadm -e proc-setid
coreadm -e log

# Disable service and uninstall package
svcadm disable svc:/site/first-boot-script-svc:default
pkg uninstall pkg:/first-boot-script

echo "Site first-boot script done. Rebooting in 5 seconds." > /dev/console

sleep 5 && reboot -p || reboot &

exit $SMF_EXIT_OK
```

SMF マニフェストファイルの作成

スクリプトを実行する一時的な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。

- 下のサービスマニフェスト例の終わり近くにある duration プロパティーの値は transient になっています。一時的なサービスでは、start メソッドが 1 度実行され、そのメソッドが $SMF_EXIT_OK で終了した場合は再度実行されません。svc.startd デーモンは、スクリプトの初回実行後にスクリプトの再起動を試みません。
- 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。
- この例のサービスの名前は site/first-boot-script-svc です。クライアントのブート後に、次のコマンドの出力でそのサービスを確認できます。

  ```
  $ svcs -a|grep first-boot-script
  STATE       STIME    FMRI
  disabled    13:51:42 svc:/site/first-boot-script-svc:default
  ```

- この例では、multi-user 依存性を指定して、初回ブート後のブート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い場合は、スクリプトで想定されるようにシステムが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。

例 13–2　SMF サービスマニフェストの例

次のファイルは、first-boot-script-svc サービス用の SMF マニフェストファイル first-boot-script-svc-manifest.xml です。このサービスはデフォルトで有効になっており、再起動しません。この例でサービスが実行するスクリプトは /opt/site/first-boot-script.sh です。

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type='manifest' name='first-boot-script:site-first-boot-script-svc'>

<service
    name='site/first-boot-script-svc'
    type='service'
    version='1'>
    <create_default_instance enabled='true' />
    <single_instance />

<!-- Run the script late in the startup sequence after first boot. -->
<dependency name='multi-user' grouping='require_all' restart_on='none' type='service'>
    <service_fmri value='svc:/milestone/multi-user:default' />
</dependency>

<exec_method
    type='method'
    name='start'
    exec='/opt/site/first-boot-script.sh'
    timeout_seconds='360'>
    <method_context>
        <method_credential user='root'/>
    </method_context>
</exec_method>

<exec_method
    type='method'
    name='stop'
    exec=':true'
    timeout_seconds='60'
/>

<property_group name='startd' type='framework'>
    <propval name='duration' type='astring' value='transient' />
</property_group>

</service>
</service_bundle>
```

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。

- 189 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。
- 187 ページの「初回ブート時に実行され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、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
set name=pkg.summary value="AI first boot script"
set name=pkg.description value="Script that runs at first boot after AI installation"
file path=lib/svc/manifest/site/first-boot-script-svc-manifest.xml mode=0444 \
    owner=root group=sys
dir path=opt/site mode=0755 owner=root group=sys
file path=opt/site/first-boot-script.sh mode=0555 owner=root group=sys
```

   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する場合は、pkglint を使用して、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。

```
# pkglint first-boot-script.p5m
```

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@1.0,5.11:20111101T024901Z
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。

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        ---
```

パッケージのテストインストールを試みることをお勧めします。-n オプションは、実際にはパッケージをインストールしないことを意味します。

```
# pkg set-publisher -g ./firstbootrepo firstboot
# pkg install -nv first-boot-script
           Packages to install:         1
     Estimated space avaialble: 111.26 GB
Estimated space to be consumed:  72.42 MB
       Create boot environment:        No
Create backup boot environment:        No
          Rebuild boot archive:        No

Changed packages:
firstboot
  first-boot-script
    None -> 1.0,5.11:20111101T024901Z
```

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 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。

第 14 章

AI クライアントシステムで使用するための Oracle Configuration Manager の設定

Oracle Configuration Manager を使用すると、My Oracle Support にシステム構成を記録できます。これを行うと、プロアクティブで的が絞られたサポートを Oracle から受けることができます。

この章では、Oracle Configuration Manager が AI クライアントインストールで確実に機能する方法について説明します。

AI クライアントでの Oracle Configuration Manager のデフォルトの動作

クライアントの初回ブートで network マイルストーンに達すると、system/ocm SMF サービスが開始します。

system/ocm サービスは、opt_out プロパティーが指定されているかどうかを確認します。デフォルトでは opt_out プロパティーは指定されていません。

- opt_out プロパティーが指定されている場合は、サービス自体が無効になり、終了します。opt_out プロパティーを指定する場合は、199 ページの「Oracle Configuration Manager の停止」を参照してください。
- opt_out プロパティーが指定されていない場合、system/ocm サービスは response_file_pkg_name プロパティーが指定されているかどうかを確認します。デフォルトでは response_file_pkg_name プロパティーは指定されていません。
  - response_file_pkg_name プロパティーが指定されていない場合、Oracle Configuration Manager はデフォルトのレスポンスファイルを使用します。

    Oracle Configuration Manager は、匿名登録を実行し、システムデータの収集を開始して、そのデータをインターネット上の場所 https:ccr.oracle.com/ に送信します。

- response_file_pkg_name プロパティーが指定されている場合、Oracle Configuration Manager はそのパッケージで提供されるカスタムレスポンスファイルを使用します。このプロパティーには、そのファイル名ではなくパッケージ名が指定されています。response_file_pkg_name プロパティーを指定する場合は、196 ページの「カスタムレスポンスファイルの指定」を参照してください。

  このカスタムレスポンスファイルは、サポート識別子をこのシステム情報に関連付けたり、プロキシを指定したり、またはインターネットにアクセスできないシステム用に Oracle サポートハブを指定したりするために使用できます。

カスタムレスポンスファイルの指定

次のいずれかの情報を指定する場合は、カスタムレスポンスファイルを用意する必要があります。

- このデータを企業に関連付けるための顧客サポート識別子。
- プロキシ、またはインターネットにアクセスできないシステムでのデータ収集や送信を可能にする Oracle サポートハブ。

▼ カスタムレスポンスファイルパッケージを作成してインストールする方法

カスタム IPS パッケージにカスタムレスポンスファイルを指定するには、次の手順を実行します。

1. **カスタムレスポンスファイルを作成します。**

   既存の Oracle Configuration Manager インストールを使用してレスポンスファイルを作成します。カスタムレスポンスファイルの作成については、『Oracle Configuration Manager インストレーションおよび管理ガイド』の「レスポンスファイルの作成 (http://download.oracle.com/docs/cd/E23562_01/doc.1035/e22050/admin.htm#BAJICIFE)」を参照してください。レスポンスファイルの名前は ocm.rsp にする必要があります。

2. **カスタム IPS パッケージを作成します。**

   カスタムレスポンスファイルが含まれている IPS パッケージを作成します。それぞれのシステム用にカスタムレスポンスファイルを作成する場合は、各レスポンスファイルを個別のパッケージに入れます。それぞれのカスタムレスポンスファイルのパスとファイル名は同じになります。パッケージ名は違っていても構いませ

ん。AI マニフェストに含めることができる Oracle Configuration Manager のカスタムレスポンスファイルパッケージは 1 つのみです。

a. ディレクトリ階層を作成します。

カスタムレスポンスファイルは AI クライアントシステム上の /var/ocm/.rsp/ocm.rsp としてインストールする必要があります。

```
$ mkdir -p proto/var/ocm/.rsp
$ cp ocm.rsp proto/var/ocm/.rsp
```

b. パッケージマニフェストを作成します。

ocsresp.p5m という名前の次のファイルを作成します。

```
set name=pkg.fmri value=ocsresp@1.0,5.11
set name=pkg.summary value="Oracle Configuration Manager custom response file"
set name=pkg.description value="Oracle Configuration Manager custom response file"
depend fmri=pkg:/system/ocm type=require
file var/ocm/.rsp/ocm.rsp group=root mode=0444 owner=root path=var/ocm/.rsp/ocm.rsp
```

depend および file 行のみが必須です。summary および description 行は、ユーザーがリポジトリでこのパッケージの検索または表示を試みる場合に役立ちます。

c. 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 ocm 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 customocm
# pkgrepo -s customocm add-publisher ocm
```

d. パッケージを発行します。

```
$ pkgsend publish -d ./proto -s ./customocm ocsresp.p5m
pkg://ocm/ocsresp@1.0,5.11:20110708T174359Z
PUBLISHED
```

クライアントは、customocm リポジトリからパッケージをインストールできます。customocm 起点を持つ ocm 発行元は、次の手順に示すように AI マニフェストに定義されます。

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 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。

3 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。AI マニフェストに含めることができる Oracle Configuration Manager のカスタムレスポンスファイルパッケージは 1 つのみです。

```
<software type="IPS">
  <source>
    <publisher name="solaris">
      <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
    <publisher name="ocm">
      <origin name="file:///net/host/export/customocm"/>
    </publisher>
  </source>
  <software_data action="install">
    <name>pkg://ocm/ocsresp</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。

4 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、カスタムの Oracle Configuration Manager レスポンスファイル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。

5 カスタムレスポンスファイルパッケージのプロパティーをシステム構成プロファイルに追加します。

クライアントでは任意の数のシステム構成プロファイルを使用できるため、このプロパティー設定を個別のプロファイルに入れて、任意の、またはすべてのクライアントでこの設定を簡単に追加することができます。

```
<service name='system/ocm' type='service' version='1' >
  <instance name='default' enabled='true' >
    <property_group name='reg' type='framework'>
      <propval name='response_file_pkg_name' type='astring' value='pkg:/ocsresp' />
    </property_group>
  </instance>
</service>
```

6 変更されたシステム構成プロファイルを **AI** インストールサービスに追加します。

installadm コマンドの create-profile サブコマンドを使用して、新規の、またはカスタマイズされたシステム構成プロファイルを AI インストールサービスに追加します。

次の例では、filename は構成プロファイルのフルパス名です。ファイル criteria.xml は、このプロファイルを使用するクライアントを指定します。このプロファイルにレスポンスファイルパッケージ名の設定が 1 つしか含まれていない場合で、さらにすべてのクライアントで同じカスタムレスポンスファイルを使用する場合は、-C または -c オプションを省略します。

```
# installadm create-profile -n svcname -f filename -C criteria.xml
```

Oracle Configuration Manager の停止

一部のクライアントで Oracle Configuration Manager の使用を停止する場合は、次の要素をそれらのクライアントのシステム構成プロファイルに追加します。

```
<service name='system/ocm' type='service' version='1' >
  <instance name='default' enabled='false' >
    <property_group name='reg' type='framework'>
      <propval name='opt_out' type='astring' value='true' />
    </property_group>
  </instance>
</service>
```

新規の、または変更された構成プロファイルを、それらのクライアントを特定する条件とともに AI インストールサービスに追加します。

第 15 章

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 IP アドレス、ブートファイル、および必要に応じてインストールサーバーの IP アドレスを取得します。
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、122 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。
   - 123 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します。
   - 対話型システム構成ツールでの管理者の応答を使用します。

AI クライアントインストールが終了すると、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。

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。

SPARC および x86 クライアントシステムの要件

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。 ■ 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の追加の要件

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。

ネットワーク経由でブートするには、AI 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 eeprom 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、ブートプロパティーを任意で指定できます。

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 s11-x86 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 s11-x86 インストールサービスをブートできません。

```
# installadm create-client -n s11-x86 -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           : 0100E0815DBFE0
```

また、installadm create-client コマンドの結果は、/etc/netboot ディレクトリでも表示できます。このリストには、このクライアントのインストールサービスファイルのみが表示されます。menu.lst ファイルの行は読みやすくするために分割されています。

```
# cd /etc/netboot
# ls -l
lrwxrwxrwx  Aug 26 08:27 0100E0815DBFE0 -> ./s11-x86/boot/grub/pxegrub
-rw-r--r--  Aug 26 08:27 menu.lst.0100E0815DBFE0
drwxr-xr-x  Aug 26 08:26 s11-x86

# cat menu.lst.0100E0815DBFE0
default=0
timeout=30
min_mem64=0
title Oracle Solaris 11 11/11 Text Installer and command line
        kernel$ /s11-x86/platform/i86pc/kernel/$ISADIR/unix -B
install_media=http://$serverIP:5555//export/auto_install/s11-x86,install_service=s11-x86,
install_svc_address=$serverIP:5555
        module$ /s11-x86/platform/i86pc/$ISADIR/boot_archive

title Oracle Solaris 11 11/11 Automated Install
        kernel$ /s11-x86/platform/i86pc/kernel/$ISADIR/unix -B install=true,
install_media=http://$serverIP:5555//export/auto_install/s11-x86,install_service=s11-x86,
install_svc_address=$serverIP:5555,livemode=text
        module$ /s11-x86/platform/i86pc/$ISADIR/boot_archive
```

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 s11-sparc インストールサービスに関連付けています。

```
# installadm create-client -n s11-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。97 ページの「AI インストールサービスを作成する」を参照してください。

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。

```
# 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を正確に示します。また、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すると、進行中のインストールをリモートから監視できます。

リモートアクセスは、インストール設定ファイルでオプション livessh を enable に設定することで有効にします。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。

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、menu.lst 構成ファイルが次の方法で作成されます。

- installadm create-client コマンドを使用した場合、ファイル名は /etc/netboot/menu.lst.01*MAC_address* です。ここで、*MAC_address* は installadm create-client コマンドで指定された MAC アドレスです。
- installadm create-client コマンドを使用しなかった場合、ファイル名は /etc/netboot/*service_name*/menu.lst です。ここで、*service_name* は installadm create-service コマンドで作成されたインストールサービスの名前です。

このファイルでは、オプションはカーネルのパラメータとして指定されます。次の例では、livessh オプションと install_debug オプションは enable に設定されています。

```
kernel$ ... -B install_media=...,livessh=enable,install_debug=enable
```

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、/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できます。

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/svc1/system.conf
...
livessh=enable
...
```

SPARC クライアントのインストール

OBP プロンプトから SPARC クライアントをブートするには、次のコマンドを使用します。

```
ok boot net:dhcp - install
```

SPARC クライアントのネットワークブート処理

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、DHCP サーバーから IP アドレスと `wanboot-cgi` ファイルの場所を取得します。
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。

SPARC のネットワークブートの出力例

ブート処理によって次の出力が表示されます。

```
{0} ok boot net:dhcp - install

SPARC Enterprise T5120, No Keyboard
Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.29.1, 16256 MB memory available, Serial #81036844.
Ethernet address 0:14:4f:d4:86:2c, Host ID: 84d4862c.

Boot device: /pci@0/pci@0/pci@1/pci@0/pci@2/network@0:dhcp  File and args: - install
1000 Mbps full duplex  Link up
```

```
Timed out waiting for BOOTP/DHCP reply
<time unavailable> wanboot info: WAN boot messages->console
<time unavailable> wanboot info: configuring /pci@0/pci@0/pci@1/pci@0/pci@2/network@0:dhcp

1000 Mbps full duplex  Link up
<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
Fri Aug 26 14:34:53 wanboot progress: miniroot: Read 221327 of 221327 kB (100%)
Fri Aug 26 14:34:53 wanboot info: miniroot: Download complete
SunOS Release 5.11 Version snv_175 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2011-08-26 13:47:31--  http://host1:5555/install/images/sparc_snv175//solaris.zlib
idn_decode failed (9): ‘System iconv failed’
Resolving host1... 10.80.238.5
idn_decode failed (9): ‘System iconv failed’
Connecting to host1|10.80.238.5|:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 135977984 (130M) [text/plain]
Saving to: ‘/tmp/solaris.zlib’

100%[=====================================>] 135,977,984 22.3M/s   in 6.0s

2011-08-26 13:47:37 (21.7 MB/s) - ‘/tmp/solaris.zlib’ saved [135977984/135977984]

Downloading solarismisc.zlib
--2011-08-26 13:47:37--  http://host1:5555/install/images/sparc_snv175//solarismisc.zlib
idn_decode failed (9): ‘System iconv failed’
Resolving host1... 10.80.238.5
idn_decode failed (9): ‘System iconv failed’
Connecting to host1|10.80.238.5|:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18826752 (18M) [text/plain]
Saving to: ‘/tmp/solarismisc.zlib’

100%[=====================================>] 18,826,752  21.7M/s   in 0.8s

2011-08-26 13:47:37 (21.7 MB/s) - ‘/tmp/solarismisc.zlib’ saved [18826752/18826752]

Downloading .image_info
--2011-08-26 13:47:37--  http://host1:5555/install/images/sparc_snv175//.image_info
idn_decode failed (9): ‘System iconv failed’
Resolving host1... 10.80.238.5
idn_decode failed (9): ‘System iconv failed’
Connecting to host1|10.80.238.5|:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 82 [text/plain]
Saving to: ‘/tmp/.image_info’

100%[=====================================>] 82          --.-K/s   in 0s

2011-08-26 13:47:37 (2.22 MB/s) - ‘/tmp/.image_info’ saved [82/82]
```

```
Done mounting image
Configuring devices.
Hostname: solaris
Service discovery phase initiated
Service name to look up: default-sparc
Service discovery finished successfully
Process of obtaining install manifest initiated
Using the install manifest obtained via service discovery

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.

solaris console login: 13:48:35    Install Log: /system/volatile/install_log
13:48:35    Using XML Manifest: /system/volatile/ai.xml
13:48:35    Using profile specification: /system/volatile/profile
13:48:35    Using service list file: /var/run/service_list
13:48:35    Starting installation.
13:48:35    0% Preparing for Installation
13:48:36    100% manifest-parser completed.
13:48:36    0% Preparing for Installation
13:48:36    1% Preparing for Installation
13:48:37    2% Preparing for Installation
13:48:37    3% Preparing for Installation
13:48:37    4% Preparing for Installation
13:48:47    7% target-discovery completed.
13:48:47    === Executing Target Selection Checkpoint ==
13:48:48    Selected Disk(s) : c3t0d0
13:48:48    13% target-selection completed.
13:48:48    17% ai-configuration completed.
13:49:01    19% target-instantiation completed.
13:49:02    19% Beginning IPS transfer
13:49:02    Creating IPS image
13:50:58    Installing packages from:
13:50:58        solaris
13:50:58            origin:  http://pkg.example.com/solaris/
14:48:40    21% generated-transfer-1491-1 completed.
14:48:41    23% initialize-smf completed.
14:48:43    Installing SPARC bootblk to root pool devices: ['/dev/rdsk/c3t0d0s0']
14:48:43    Setting openprom boot-device
14:48:44    33% boot-configuration completed.
14:48:44    35% update-dump-adm completed.
14:48:45    37% setup-swap completed.
14:48:45    40% set-flush-ips-content-cache completed.
14:48:47    42% device-config completed.
14:48:49    44% apply-sysconfig completed.
14:48:49    46% transfer-zpool-cache completed.
14:49:03    87% boot-archive completed.
14:49:04    89% transfer-ai-files completed.
14:49:04    99% create-snapshot completed.
14:49:05    Automated Installation succeeded.
14:49:05    You may wish to reboot the system at this time.
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/sadm/system/logs/install_log
```

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 クライアントのネットワークブート処理

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、IP アドレスを取得します。ブートファイル `pxegrub` 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. `pxegrub` ブートファイルが読み込まれ、`menu.lst` ファイルを読み取ります。
3. `pxegrub` ブートファイルが `boot_archive` ファイルを取得し、TFTP を使用して Oracle Solaris OS がブートされます。
4. 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。
5. 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。
6. 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。

x86 のネットワークブートの出力例

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 11/11 Text Installer and command line
Oracle Solaris 11 11/11 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、223 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。

イメージが選択されると、次のメッセージが表示されます。

```
Remounting root read/write snv_175 64-bit
Probing for device nodes ...acle and/or its affiliates. All rights reserved.
Preparing network image for use
Downloading solaris.zlib
--2011-08-26 07:35:13--  http://10.80.238.5:5555//install/images/i386_snv175/solaris.zlib
Connecting to 10.80.238.5:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 130032128 (124M) [text/plain]
Saving to: ‘/tmp/solaris.zlib’

100%[=====================================>] 130,032,128 16.4M/s   in 8.3s

2011-08-26 07:35:22 (14.9 MB/s) - ‘/tmp/solaris.zlib’ saved [130032128/130032128]

Downloading solarismisc.zlib
--2011-08-26 07:35:22--  http://10.80.238.5:5555//install/images/i386_snv175/solarismisc.zlib
Connecting to 10.80.238.5:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18758144 (18M) [text/plain]
Saving to: ‘/tmp/solarismisc.zlib’

100%[=====================================>] 18,758,144  21.7M/s   in 0.8s

2011-08-26 07:35:23 (21.7 MB/s) - ‘/tmp/solarismisc.zlib’ saved [18758144/18758144]

Downloading .image_info
--2011-08-26 07:35:23--  http://10.80.238.5:5555//install/images/i386_snv175/.image_info
Connecting to 10.80.238.5:5555... connected.
HTTP request sent, awaiting response... 200 OK
Length: 241 [text/plain]
Saving to: ‘/tmp/.image_info’

100%[=====================================>] 241         --.-K/s   in 0s

2011-08-26 07:35:23 (29.3 MB/s) - ‘/tmp/.image_info’ saved [241/241]

Done mounting image
Configuring devices.
Hostname: solaris
Service discovery phase initiated
Service name to look up: default-i386
Service discovery finished successfully
Process of obtaining install manifest initiated
Using the install manifest obtained via service discovery
```

```
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.

solaris console login: 07:35:35    Install Log: /system/volatile/install_log
07:35:35    Using XML Manifest: /system/volatile/ai.xml
07:35:35    Using profile specification: /system/volatile/profile
07:35:35    Using service list file: /var/run/service_list
07:35:36    Starting installation.
07:35:36    0% Preparing for Installation
07:35:36    100% manifest-parser completed.
07:35:36    0% Preparing for Installation
07:35:36    1% Preparing for Installation
07:35:36    2% Preparing for Installation
07:35:36    3% Preparing for Installation
07:35:37    4% Preparing for Installation
07:35:51    7% target-discovery completed.
07:35:51    === Executing Target Selection Checkpoint ==
07:35:51    Selected Disk(s) : c7t0d0
07:35:51    13% target-selection completed.
07:35:51    17% ai-configuration completed.
07:36:23    19% target-instantiation completed.
07:36:23    19% Beginning IPS transfer
07:36:23    Creating IPS image
07:36:41    Installing packages from:
07:36:41        solaris
07:36:41            origin:  http://pkg.example.com/solaris/
07:53:29    21% generated-transfer-1006-1 completed.
07:53:29    23% initialize-smf completed.
07:53:30    Setting console boot device property to ttya
07:53:30    Disabling graphical console in boot loader
07:53:30    Creating Legacy GRUB config directory:
       /rpool/boot/grub
07:53:30    Installing boot loader to devices: ['/dev/rdsk/c7t0d0s0']
07:53:31    33% boot-configuration completed.
07:53:31    35% update-dump-adm completed.
07:53:31    37% setup-swap completed.
07:53:31    40% set-flush-ips-content-cache completed.
07:53:32    42% device-config completed.
07:53:46    44% apply-sysconfig completed.
07:53:46    46% transfer-zpool-cache completed.
07:54:08    87% boot-archive completed.
07:54:08    89% transfer-ai-files completed.
07:54:09    99% create-snapshot completed.
07:54:09    Automated Installation succeeded.
07:54:09    You may wish to reboot the system at this time.
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/sadm/system/logs/install_log
```

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。IPS からのパッケージのインストールが開始されると、このログファイルへの更新が長時間表示されないことがあります。

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/sadm/system/logs/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はリブートしません。

第 16 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て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。

- 214 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」
- 217 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」
- 220 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、ネットワーク経由による SPARC クライアントのブート時に起こりう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。

- 214 ページの「BOOTP/DHCP 応答を待機してタイムアウトになった」
- 215 ページの「ブートの読み込みに失敗」
- 215 ページの「内部サーバーのエラーまたは WAN ブートの警告」
- 216 ページの「エラーメッセージ 403: Forbidden または 404 Not Found」
- 217 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 要求を送信し、その要求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 20:46:43 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

これらの「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、イメージのアーカイブの 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
Fri Aug 26 16:26:52 wanboot progress: miniroot: Read 221327 of 221327 kB (100%)
Fri Aug 26 16:26:53 wanboot info: miniroot: Download complete
Warning: 'SUNW,UltraSPARC-IIIi' support will be removed in a future release of Solaris.
SunOS Release 5.11 Version snv_175 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
WARNING: i2c_0 failed to add interrupt.
WARNING: i2c_0 operating in POLL MODE only

Hardware watchdog enabled
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2011-08-26 23:19:57--  http://10.134.125.136:5555/export/auto_install/175s//solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2011-08-26 23:19:57 ERROR 404: Not Found.

Could not obtain http://10.134.125.136:5555/export/auto_install/175s//solaris.zlib from install server
Please verify that the install server is correctly configured and reachable from the client
Requesting System Maintenance Mode
```

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ターゲットディレクトリ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、次に示すように、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 ブート引数を含めずにブートした場合、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。次のメッセージが表示されます。

```
Auto-installer disabled. Enable the auto-installer service
by running the following command:
svcadm enable svc:/application/auto-installer:default
```

自動インストールを開始する場合は、ログインして、メッセージに示されているようにインストールサービスを有効にするか、上に示したコマンドで install 引数を使用してシステムをリブートできます。

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、ネットワーク経由による x86 クライアントのブート時に起こりう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。

- 217 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」
- 218 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」
- 218 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」
- 218 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」
- 219 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 `menu.lst` ファイルが有効なブートアーカイブを提示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、この抜粋例に示すように、`df -k` から `/etc/netboot` ディレクトリ下にループバックマウントされます。

```
/install/images/x86_snv175 \
28046887     8432439    19614448    31%    /etc/netboot/x86_snv175
```

`installadm create-service` コマンドで使用したターゲットディレクトリ名がわかる場合は、その情報を利用して、そのターゲットディレクトリがマウント済みであるかどうかを判別できます。

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で、`/etc/netboot` 下の `menu.lst` ファイルで指定された場所でインストールプログラムの 1 つにアクセスできないか、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 クライアントでの出力では、solaris.zlib ファイルが指定の場所に存在しません。

```
SunOS Release 5.11 Version snv_175 64-bit
Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2011-08-18 20:02:26--  http://10.134.125.136:5555//export/auto_install/s11-x86/solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2011-08-18 20:02:26 ERROR 404: Not Found.

Could not obtain http://10.134.125.136:5555//export/auto_install/s11-x86/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 11/11 Text Installer and command line
Oracle Solaris 11 11/11 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。次のメッセージが表示されます。

```
Auto-installer disabled. Enable the auto-installer service
by running the following command:
svcadm enable svc:/application/auto-installer:default
```

自動インストールを開始する場合は、ログインして、メッセージに示されているようにインストールサービスを有効にするか、システムをリブートして 2 番目のメニューエントリを選択できます。

SPARC および x86 のエラーメッセージ

次のエラーは、SPARC と x86 両方のインストールに共通しています。

- 220 ページの「自動インストール失敗メッセージ」
- 220 ページの「有効なパッケージサーバーに接続不能」
- 221 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。

```
Automated Installation failed. Please refer to /system/volatile/install_log file
   for details
   Apr  9 14:28:09 solaris svc.startd[7]: 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
15:54:47        generated-transfer-1341-1
15:54:47
15:54:47    Checkpoint execution error:
15:54:47
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
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://sun.com/msg/SMF-8000-YX for the latest service
procedures and policies regarding this diagnosis.
```

次のメッセージと同じようなメッセージを /system/volatile/install_log ファイルで確認してください。

```
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(1M) 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。213 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 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
  ```

  カタログを更新したり、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 entirity 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        ips
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:/entirity
```

```
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:/entirity
```

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、223 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 11/11 Text Installer and command line
  Oracle Solaris 11 11/11 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 ブートプロパティー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` ブートプロパティーを削除します。

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` ブートプロパティーを指定した場合は、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` ブートプロパティーが指定されていないことを確認します。このプロパティーが指定されている場合は、ブートエントリのカーネル行を編集してこのプロパティーを削除します。

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
```

このメニューにはデフォルトの選択はありません。

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
```