

# Oracle® Solaris 11.3 システムのインストール

ORACLE®

Part No: E62506  
2017 年 3 月



## Part No: E62506

Copyright © 2011, 2017, 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.

このソフトウェアまた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。このソフトウェアまたはハードウェアは、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。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、Oracle Corporation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はオラクル およびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

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。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除いて、Oracle Corporation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。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporation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。

### ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

### Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。



# 目次

---

このドキュメントの使用方法 .....	17
<b>I Oracle Solaris 11.3 インストールオプション .....</b>	<b>19</b>
<b>1 インストールオプションの概要 .....</b>	<b>21</b>
インストールオプションの比較 .....	21
その他のインストールオプション .....	24
Oracle Solaris 11.3 でのインストールの新機能 .....	24
<b>II インストールメディアを使用したインストール .....</b>	<b>27</b>
<b>2 インストールの準備 .....</b>	<b>29</b>
Live Media およびテキストインストールのシステム要件 .....	29
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システム の準備 .....	29
システムのパーティション分割 .....	30
適切なデバイスドライバがあることの確認 .....	35
Oracle Configuration Manager の使用 .....	35
<b>3 Live Media の使用 .....</b>	<b>39</b>
GUI インストーラを使用したインストール .....	39
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
Live Media インストール後のソフトウェアの追加 .....	48
<b>4 テキストインストーラの使用 .....</b>	<b>49</b>
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ルのタスク .....	51
<b>5 メディアからブートする自動インストール .....</b>	<b>63</b>
AI メディアを使用するインストールの概要 .....	63
AI メディアを使用したインストール .....	63
<b>6 Oracle Solaris インスタンスの構成解除または再構成 .....</b>	<b>71</b>

機能グループの概要 .....	71
Oracle Solaris インスタンスの構成解除 .....	72
システムの再構成 .....	73
SCI ツールを使用したシステム構成プロファイルの作成 .....	77
<b>III インストールサーバーを使用したインストール .....</b>	<b>79</b>
<b>7 複数システムへの自動インストール .....</b>	<b>81</b>
自動インストールとは .....	81
自動インストーラのコンポーネント .....	82
AI のセキュリティー保護 .....	85
AI およびゾーン .....	86
AI 構成プロセスの概要 .....	86
AI クライアントのブート .....	88
AI サーバーの計画 .....	89
自動インストーラの使用事例 .....	89
<b>8 AI サーバーの設定 .....</b>	<b>99</b>
AI サーバー設定タスク .....	99
AI サーバーの要件 .....	100
インストールサービスの管理者権限 .....	101
AI サーバーの構成 .....	101
インストールサービスの操作 .....	107
インストールサービスの作成 .....	107
インストールサービスへの AI クライアントの関連付け .....	114
インストール手順のカスタマイズ .....	116
AI SMF サービスを管理する .....	118
自動インストールのセキュリティーの向上 .....	118
インストールサービスの情報の表示 .....	134
インストールサービスの管理 .....	140
AI マニフェストの管理 .....	143
システム構成プロファイルの管理 .....	145
<b>9 AI クライアントへのカスタマイズの割り当て .....</b>	<b>149</b>
AI 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 .....	149
AI マニフェストの選択 .....	150
システム構成プロファイルの選択 .....	151
選択条件 .....	152
<b>10 AI クライアントのインストールパラメータの定義 .....</b>	<b>155</b>

---

XML AI マニフェストファイルのカスタマイズ .....	156
クライアントインストール時の AI マニフェストの作成 .....	158
AI マニフェストウィザードを使用した AI マニフェストの作成 .....	175
installadm コマンドを使用した AI マニフェストの作成 .....	178
AI マニフェストの例 .....	182
デフォルトの AI マニフェスト .....	189
<b>11 AI クライアントのシステム構成パラメータの定義 .....</b>	<b>191</b>
構成プロファイルの指定 .....	191
システム構成プロファイル内での構成の指定 .....	194
システム構成プロファイルテンプレートの使用 .....	206
システム構成プロファイルの例 .....	208
<b>12 ゾーンのインストールと構成 .....</b>	<b>223</b>
AI による非大域ゾーンのインストール方法 .....	223
大域ゾーンの AI マニフェストでの非大域ゾーンの指定 .....	224
非大域ゾーンの構成およびインストールデータ .....	224
<b>13 初回ブート時のカスタムスクリプトの実行 .....</b>	<b>231</b>
初回ブートサービスおよびパッケージの自動作成 .....	231
初回ブートサービスパッケージの対話式での作成 .....	232
初回ブート時に実行するスクリプトの手動作成 .....	233
初回ブートサービスパッケージの手動作成 .....	236
初回ブートサービスのテスト .....	245
<b>14 AI サーバーを使用した AI クライアントのインストール .....</b>	<b>247</b>
AI クライアントのインストール方法 .....	247
SPARC および x86 AI クライアントのシステム要件 .....	248
AI クライアントの設定 .....	249
AI クライアントのインストール .....	251
<b>15 自動インストールのトラブルシューティング .....</b>	<b>259</b>
AI クライアントインストールが失敗する .....	259
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 .....	272
コマンド行から自動インストールを開始する .....	273
<b>IV 関連タスクの実行 .....</b>	<b>275</b>
<b>A Oracle Configuration Manager の操作 .....</b>	<b>277</b>
Oracle Configuration Manager の概要 .....	277
Oracle Configuration Manager の中央コレクタについて .....	278
Oracle Configuration Manager の管理 .....	279

<b>B デバイスドライバユーティリティの使用</b> .....	283
デバイスドライバユーティリティの概要 .....	283
<b>索引</b> .....	289

## 図目次

---

図 1	メディアを使用する AI インストール .....	64
図 2	AI ネットワークの例 .....	82
図 3	1つのアーキテクチャーと1つのOSをサポートするAIサーバー .....	90
図 4	2つのアーキテクチャーをサポートするAIサーバー .....	91
図 5	1つのアーキテクチャーと2つのディスクレイアウトをサポートするAIサーバー .....	92
図 6	1つのアーキテクチャーと2つのディスク構成をサポートするAIサーバー .....	93
図 7	1つのアーキテクチャーと2つのリリースをサポートするAIサーバー .....	94
図 8	1つのアーキテクチャーと一部のAIクライアント用の追加の構成をサポートするAIサーバー .....	95
図 9	多くの構成の変更をサポートするAIサーバー .....	96



## 表目次

---

表 1	インストールオプション .....	21
表 2	複数のオペレーティングシステム環境 .....	30
表 3	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 .....	33
表 4	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 .....	35
表 5	機能グループ .....	71
表 6	条件のキーワードと条件の階層 .....	152
表 7	AI クライアント属性の環境変数 .....	160
表 8	root_account プロパティグループのプロパティ .....	196
表 9	user_account プロパティグループのプロパティ .....	197
表 10	config プロパティグループのプロパティ .....	199
表 11	timezone プロパティグループのプロパティ .....	200
表 12	environment プロパティグループのプロパティ .....	200
表 13	IPv4 ネットワークインタフェースのプロパティグループのプロパ ティ .....	203
表 14	IPv6 ネットワークインタフェースのプロパティグループのプロパ ティ .....	203
表 15	svc:/system/name-service/switch プロパティグループの config プロ パティ .....	204
表 16	config プロパティグループのプロパティ .....	205
表 17	システム構成テンプレートプロファイルの変数 .....	208



# 例目次

---

例 1	システムの構成解除 .....	72
例 2	SMF データの構成解除 .....	72
例 3	特定の機能グループの構成解除 .....	72
例 4	システムの構成解除後のリブート .....	73
例 5	システム構成プロファイルを使用したシステムの再構成 .....	73
例 6	デフォルトプロファイルの使用 .....	77
例 7	プロファイルの作成と使用 .....	77
例 8	プロファイルを作成および使用して機能グループを構成する .....	78
例 9	ネットワーク上の AI サポートの無効化 .....	104
例 10	AI サーバーでサポートされるネットワークの組み込み .....	104
例 11	AI Web サーバーのポート番号の構成 .....	104
例 12	セキュア AI Web サーバーのポート番号の構成 .....	105
例 13	デフォルトのイメージパスの構成 .....	105
例 14	AI サーバーに対する AI クライアントの IP アドレスと数の定義 .....	105
例 15	AI サーバー上のローカル DHCP サービスの自動更新の無効化 .....	105
例 16	AI サーバーでの DHCP 更新の有効化 .....	106
例 17	AI サーバーの DHCP が有効な状態で ISO ファイルを使用した SPARC インストールサービスの作成 .....	108
例 18	IPS パッケージを使用した X86 インストールサービスの作成 .....	109
例 19	異なるアーキテクチャーに対するインストールサービスの作成 .....	110
例 20	x86 クライアントを自動的にインストールするサービスの作成 .....	110
例 21	インストールサービスへの SPARC クライアントの関連付け .....	114
例 22	x86: インストールサービスに対する x86 クライアントの関連付けお よびシリアルラインへの出力のリダイレクト .....	115
例 23	x86: x86 クライアントのブートプロパティの変更 .....	115
例 24	マニフェストとクライアント条件の関連付け .....	116
例 25	スクリプトとクライアント条件の関連付け .....	117
例 26	インストールサービスのデフォルトマニフェストの作成 .....	117
例 27	システム構成プロファイルとクライアント条件との関連付け .....	118
例 28	AI SMF サービスの有効化 .....	118

例 29	AI SMF サービスの無効化 .....	118
例 30	ユーザーが提供する資格証明を使用した AI サーバー資格の生成 ....	123
例 31	インストール時の AI サーバー認証の要求 .....	125
例 32	x86: インストール時の暗号化の要求 .....	125
例 33	特定の AI クライアントのためのユーザーが提供する資格証明の使用 .....	126
例 34	特定のインストールサービスの AI クライアントの資格証明 .....	126
例 35	デフォルトの AI クライアント資格 .....	127
例 36	1 つの AI クライアントの資格の削除 .....	129
例 37	CA 証明書の削除 .....	130
例 38	AI サーバーのセキュリティー資格の削除 .....	130
例 39	Kerberos クライアントのデプロイ中の既存の鍵のダウンロード .....	132
例 40	Kerberos クライアントのデプロイ中の新しい鍵の作成 .....	133
例 41	MS AD ドメインへの Kerberos クライアントの自動参加 .....	134
例 42	AI サーバー上のすべてのインストールサービスの一覧表示 .....	135
例 43	指定したインストールサービスの情報の表示 .....	135
例 44	インストールサービスに関連付けられた AI クライアントの一覧表示 .....	135
例 45	特定のインストールサービスに関連付けられた AI クライアントの一覧表示 .....	136
例 46	すべての AI マニフェストおよびシステム構成プロファイルの一覧表示 .....	136
例 47	指定したインストールサービスに関連付けられているマニフェストとプロファイルの一覧表示 .....	137
例 48	AI サーバーのセキュリティー情報の一覧表示 .....	138
例 49	AI クライアントのセキュリティー情報の一覧表示 .....	139
例 50	インストールサービス別名の作成 .....	140
例 51	インストールサービス別名の変更 .....	140
例 52	インストールサービスを作成する際のデフォルト AI マニフェストの設定 .....	141
例 53	既存のインストールサービスを変更することによるデフォルト AI マニフェストの設定 .....	141
例 54	新しいインストールイメージのイメージパスの設定 .....	141
例 55	既存のインストールイメージのイメージパスの設定 .....	142
例 56	インストールサービスの更新 .....	142
例 57	インストールサービスを更新するときの異なるリポジトリの使用 .....	142
例 58	インストールサービスを更新するときの異なるネットイメージパッケージの使用 .....	143

例 59	AI クライアントと AI マニフェストとの照合 .....	151
例 60	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 .....	164
例 61	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 .....	165
例 62	特定のサイズのディスクが2つ以上ある場合のミラー化構成の指 定 .....	166
例 63	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 .....	169
例 64	特定のサイズ以上のディスクのみをターゲットディスクにする指 定 .....	169
例 65	システム構成プロファイルの追加 .....	170
例 66	マニフェスト仕様が正しくないスクリプト .....	171
例 67	AI マニフェストウィザードの無効化 .....	175
例 68	マニフェストファイルを AI サーバーに保存できるようにする .....	176
例 69	walk モードでの installadm によるマニフェストエントリの作 成 .....	180
例 70	installadm によるオブジェクトの順序の変更 .....	181
例 71	パスワードが必ず期限切れになる root アカウントの構成 .....	196
例 72	パスワードを使用しないアカウントの構成 .....	198
例 73	SSH 鍵の構成 .....	198
例 74	ホスト名の構成 .....	199
例 75	ノード名マッピングの無効化 .....	199
例 76	タイムゾーンの構成 .....	200
例 77	ロケールの構成 .....	201
例 78	端末タイプの構成 .....	201
例 79	キーボード配列の構成 .....	202
例 80	NIS の構成および DNS の無効化 .....	215
例 81	指定されたドメインでの NIS と DNS の有効化 .....	216
例 82	検索リストを使用した DNS の構成 .....	217
例 83	LDAP および LDAP 検索ベースの構成 .....	218
例 84	セキュアな LDAP サーバーを使用した LDAP の構成 .....	219
例 85	ゾーンのデフォルトの AI マニフェスト .....	227
例 86	コマンド行モードでの初回ブートサービスおよびパッケージの作 成 .....	231
例 87	コマンド行モードでの初回ブートサービスおよびパッケージの作成 (カスタマイズあり) .....	232
例 88	初回ブートサービスで使用するシェルスクリプト .....	232
例 89	テンプレート初回ブートスクリプト .....	234
例 90	複数の IP インタフェースを構成する初回ブートスクリプト .....	235
例 91	生成された SMF サービスマニフェスト .....	237

例 92	カスタマイズしたサービスマニフェスト: スクリプトの実行が可能な時間の増加 .....	239
例 93	カスタマイズしたサービスマニフェスト: 非大域ゾーンがインストールされたあとでのスクリプトの実行 .....	241

## このドキュメントの使用方法

---

- **概要** – Oracle Solaris 11.3 リリースをインストールして、自動インストールサーバーを構成する方法について説明します。
- **対象読者** – 技術者、システム管理者、および認定サービスプロバイダ
- **前提知識** – ハードウェアのトラブルシューティングや交換に関する豊富な経験

## 製品ドキュメントライブラリ

この製品および関連製品のドキュメントとリソースは <http://www.oracle.com/pls/topic/lookup?ctx=E62101-01> で入手可能です。

## フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。



## パート I

### Oracle Solaris 11.3 インストールオプション

このセクションでは、インストールオプションの一般的な概要について説明します。  
この章の内容は次のとおりです。

- [21 ページの「インストールオプションの比較」](#)
- [24 ページの「その他のインストールオプション」](#)



# ◆◆◆ 第 1 章

## インストールオプションの概要

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

### インストールオプションの比較

次の表では、各種インストールオプションの機能を比較しています。

表 1 インストールオプション

インストールオプション	詳細の参照先	最低限の準備	AI サーバーが必要	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: GUI インストーラ	『Oracle Solaris 11.3 システムのインストール』の第 3 章, 「Live Media の使用」	はい	いいえ、メディアからインストールします	いいえ
テキストインストーラ	『Oracle Solaris 11.3 システムのインストール』の第 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	いいえ
サーバーがブートを補助するテキストインストール	『Oracle Solaris 11.3 システムのインストール』の「ネットワーク経由でテキストインストールを開始する方法」	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	いいえ
メディアを使用した自動インストーラ	『Oracle Solaris 11.3 システムのインストール』の第 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	はい
複数システム用の自動インストーラ	『Oracle Solaris 11.3 システムのインストール』	いいえ	はい、サーバーが必要です	はい

インストールオプション	詳細の参照先	最低限の準備	AI サーバーが必要	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
	<a href="#">『Oracle Solaris 11.3 カスタムインストールイメージの作成』の第7章, 「複数システムへの自動インストール」</a>			

インストールイメージ (カスタムの Live Media イメージ、テキストインストーライメージ、および自動インストールイメージを含む) を作成することもできます。『[Oracle Solaris 11.3 カスタムインストールイメージの作成](#)』を参照してください。

## 簡易インストール

Live Media 上の GUI インストーラやテキストインストーラは、シンプルなインストール方式です。

- どちらの方法を使用しても x86 プラットフォームに Oracle Solaris をインストールできます。テキストインストーラを使用して、SPARC プラットフォームに Oracle Solaris をインストールすることもできます。
- どちらのインストーラも最低限のメモリーで機能します。メモリー要件については、『[Oracle Solaris 11.3 ご使用にあたって](#)』を参照してください。
- どちらのインストーラを使用しても、インストール中にディスクのパーティションを選択、作成、または変更できます。

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

テキストインストーラには、GUI インストーラにはない次の利点があります。

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

---

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

---

- テキストインストーラでは、パーティションを変更する機能に加え、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。

単純なインストールを実行する方法の詳細については、『[Oracle Solaris 11.3 システムのインストール](#)』の第2章、「[インストールの準備](#)」を参照してください。

## AI 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。

以降で説明している AI メディアからのブートのみを使用する場合を除き、AI を使用する場合は、最初にネットワーク上に AI サーバーをセットアップする必要があります。AI クライアントがブートしたときに、システムは AI 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、ソフトウェアがシステムにインストールされます。

---

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

---

AI は、x86 ベースおよび SPARC ベースの両方のシステムで「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。AI 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、[81 ページ](#)の「[自動インストールとは](#)」を参照してください。

「手入力なし」のネットワークインストールに加えて、ネットワーク経由で対話型のテキストインストールを実行できます。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テキストインストールを開始する方法](#)」を参照してください。

AI イメージにアクセスできる場合は、AI サーバーが構成されていなくても、イメージをダウンロードしてネットワーク上またはローカルに格納できます。次に、イメー

ジをリムーバブルメディアにコピーできます。すると、AI メディアまたは ISO イメージを各システム上で直接ブートできるようになります。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。

## その他のインストールオプション

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。詳細については、『[Oracle Solaris 11.3 カスタムインストールイメージの作成](#)』を参照してください。

インストール済みの Oracle Solaris システムの更新

インストーラを使用して、既存の Oracle Solaris がインストールされているシステムを更新することはできません。代わりに、pkg ユーティリティを使用してパッケージリポジトリにアクセスし、システム用の新規または更新されたソフトウェアパッケージをダウンロードする必要があります。詳細については、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』を参照してください。

## Oracle Solaris 11.3 でのインストールの新機能

このセクションでは、このリリースでのインストールに関連する重要な新しい機能に関する既存のお客様のための情報を紹介します。

- テキストインストーラが拡張され、ブートプール用の新しい構成パネルが含まれています。[53 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。
- `installadm` コマンドは拡張されています。
  - システム構成プロファイルのコマンドの変更点については、[193 ページの「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」](#)を参照してください。

- 対話型エディタを使用したマニフェストの管理については、[178 ページの「installadm コマンドを使用した AI マニフェストの作成」](#)を参照してください。
- AI マニフェストは次のアクションをサポートするように構成できるようになりました。
  - [183 ページの「AI マニフェストでのルートプールとブートプールの指定」](#)
  - [187 ページの「SSL クライアント認証を使用した統合アーカイブへのアクセス」](#)
  - [187 ページの「HTTP 認証トークンを使用した統合アーカイブへのアクセス」](#)
  - [188 ページの「セキュアな IPS リポジトリへのアクセス」](#)
- システム構成プロファイルを使用して Infiniband (IB) リンクを構成できるようになりました。[213 ページの「システム構成プロファイル内での IB リンクの指定」](#)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

このセクションでは、メディアを使用する場合のインストールオプションの概要について説明します。この章の内容は次のとおりです。

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



# ◆◆◆ 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、インストールのシステム要件、システムのパーティション分割の推奨事項、および Oracle Configuration Manager のヘルプを含むこの章の情報を確認してください。

### Live Media およびテキストインストールのシステム要件

Live Media インストールイメージまたはテキストインストールイメージを使用して Oracle Solaris 11.3 リリースをインストールするための最小メモリーおよびディスク容量の要件を確認するには、『[Oracle Solaris 11.3 ご使用にあたって](#)』を参照してください。

テキストインストーラに必要なメモリーは Live Media インストーラよりも少なくなります。正確な最小要件は、システムの仕様によって変わります。GUI インストーラを実行するための十分なメモリーがシステムに搭載されていない場合は、代わりにテキストインストーラを使用してください。

---

**注記** - Intel® Virtualization Technology for Directed I/O (VT-d) を備えた Oracle x86 以外のシステムに Oracle Solaris をインストールする場合は、その前に、それらのシステムの Intel VT-d パラメータを Enabled に設定しておく必要があります。このパラメータを設定する手順については、それぞれのドキュメント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システムの準備

Oracle Solaris を複数のブート環境システム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 複数のオペレーティングシステム環境

既存のオペレーティングシステム	説明
Microsoft Windows	<p>Oracle Solaris リリースをインストールするための十分なディスク容量を設定します。このリリースでは、x86 プラットフォーム用の Oracle Solaris は GRand Unified Bootloader の新しいバージョン (GRUB 2) を使用します。Oracle Solaris は Windows を認識して、インストール中に Windows パーティションを変更しないようにします。インストールが完了すると、システムがリブートし、GRUB 2 メニューに Windows と Oracle Solaris の両方のブートエントリが表示されます。</p> <p>GRUB 2 の詳細については、『<a href="#">Oracle Solaris 11.3 システムのブートとシャットダウン</a>』の「<a href="#">GRUB 2 の概要</a>」を参照してください。</p>
Solaris 10 OS	<p>Live Media インストーラを使用して Oracle Solaris オペレーティングシステムの複数のインスタンスをインストールすることはできません。一方で、テキストインストーラは同じパーティションにある Oracle Solaris オペレーティングシステムの複数のインスタンスをサポートします (それらのインスタンスが異なるスライス上にある場合に限る)。Live Media インストーラとテキストインストーラを使用して、Oracle Solaris の複数のインスタンスがインストールされている既存のシステム上の Solaris 10 1/06 以降のリリースを置き換えることは可能です。</p> <p>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)。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。  <b>注記</b> - 現在のオペレーティングシステムで特定の Solaris VTOC (Volume Table of Contents) スライスを保持する必要がある場合は、テキストインストーラを使用します。</p>
拡張パーティション	<p>拡張パーティション上に別のオペレーティングシステムが存在する場合、インストール中に既存の拡張パーティションを変更する必要はありません。Live Media インストーラ、テキストインストーラ、自動インストーラのいずれかを使用して Oracle Solaris をインストールするときに、拡張パーティションを作成、サイズ変更、または削除できます。拡張パーティション内部の論理パーティションに Oracle Solaris をインストールすることも選択できます。</p>

## システムのパーティション分割

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

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

---

**注記** - サポートされる SPARC ベースのシステムで GPT 対応ファームウェアを適用する方法については、[SPARC: GPT Labeled Disk Support](#)を参照してください。

---

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

## インストール前にシステムをパーティション分割するためのガイドライン



**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。

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

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

GRUB 2 の詳細については、『[Oracle Solaris 11.3 システムのブートとシャットダウン](#)』の「[GRUB 2 の概要](#)」を参照してください。

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

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

- パーティション分割の次の仕様に注意してください。
  - ディスクに既存の DOS パーティションが含まれている場合は、最大 4 つの DOS プライマリパーティションが表示されます。DOS 拡張パーティションが存在する場合は、その論理パーティションも、拡張パーティション内のディスクの配置順に表示されます。Solaris パーティションは 1 つしか許可されないため、その Solaris パーティションをインストールに使用する必要があります。Solaris パーティションは、拡張パーティション内の論理パーティションである場合があります。
  - ディスクに既存の GPT パーティションが含まれている場合は、それらの GPT パーティションが表示されます。最大 7 つの GPT パーティションがサポートされます。インストール中に 1 つまたは複数の Solaris パーティションを作成できますが、インストール先として 1 つの Solaris パーティションを選択する必要があります。既存の Solaris GPT パーティションが複数存在する場合、デフォルトでは、最初の適切な Solaris GPT パーティションがインストール先として選択されます。
- 次のいずれかに当てはまる場合、Oracle Solaris のインストールによってディスクレイアウト全体が上書きされます
  - ディスクテーブルが読み込めない。
  - ディスクが事前にパーティション分割されていない。
  - インストール用にディスク全体を選択する。
- 既存の Oracle Solaris パーティションが存在し、ユーザーがほかの既存のパーティションのいずれにも変更を加えない場合、インストールのデフォルトでは Oracle Solaris パーティションのみが上書きされます。そのパーティションは、既存の拡張パーティション内にある論理パーティションである可能性があります。ほかの既存パーティションは変更されません。
- インストールには Solaris パーティションを使用する必要があります。
- ディスクのパーティション分割またはスライスの変更は、インストーラパネルで選択を完了し、インストールが開始するまでは有効になりません。インストール前であればいつでも、変更を取り消して元の設定に戻すことができます。
- 既存のパーティションテーブルを読み取れない場合は、提案されたパーティション分割の情報が表示されます。



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

---

- インストール中、「ディスクをパーティション分割する」オプションを選択した場合は、選択されたディスクの既存のパーティションがディスク上の配置と同じ順序でパネルに表示されます。これらのパーティションの未使用のディスク領域が表示されます。各パーティションのパーティションタイプ、現在のサイズ、および利

用可能なディスク容量の最大値も表示されます。拡張パーティションが存在する場合、その論理パーティションも、拡張パーティション内のディスクと同じ順番で表示されます。

- インストールを成功させるために十分な容量のないディスクやパーティションには、そのことを示すラベルが付けられます。

## x86: 対話型インストール中のパーティションの設定

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

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

表 3 対話型インストール中にディスクをパーティション分割するためのオプション

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

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)
	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。 <b>注記</b> - システムに既存の DOS パーティションがある場合、Solaris パーティションは1つしか許可されません。最初に既存の Solaris パーティションのタイプを「未使用」に変更してから、拡張パーティション内に Solaris パーティションを作成する必要があります。
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、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。

---

**注記** - `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` という名前のプール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、隣接するスライスのタイプを「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 4 テキストインストール中の VTOC スライス変更のオプション

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

## 適切なデバイスドライバがあることの確認

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

## Oracle Configuration Manager の使用

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

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

---

**注記** - すべてのデータはセキュアモードで送信されます。

---

対話型インストールを実行するときは、次のオプションが用意されています

- 顧客情報を指定せずに匿名のシステム構成を My Oracle Support に送信する場合は、デフォルトの「サポート登録」インストーラパネルの匿名の登録アドレスを使用するか、別の電子メールアドレスをパスワードなしで使用します。

- My Oracle Support で顧客情報を参照し、セキュリティ更新を受け取る場合は、「サポートの構成」パネルの匿名の電子メールアドレスを My Oracle Support のログイン ID に置き換えて、My Oracle Support のパスワードを追加します。このオプションを使用すると、Oracle Auto Service Request も起動されます。  
顧客の構成データが定期的にアップロードされると、顧客サポート担当者はこのデータを分析して、よりよいサービスを提供できます。たとえば、サービスリクエストを登録すると、サポート担当者は構成データを直接そのサービスリクエストに関連付けることができます。そして、顧客サポート担当者はシステムの一覧を表示し、それによって問題を解決できます。
- My Oracle Support にデータを自動的に送信しない場合は、「サポートの構成」パネルの匿名の電子メールアドレスを削除して、そのフィールドを空白のままにします。Oracle Configuration Manager が切断モードで起動されます。このモードでも、データを送信するために Oracle Configuration Manager を手動でアクティブ化できます。たとえば、技術サポート担当者にシステム上のデータを提供するように求められた場合は、Oracle Configuration Manager を手動で使用して、そのデータを提供できます。

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

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

次のような状況で登録または再登録を選択できます。

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

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

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

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

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

登録完了後は、次のようにサービスを有効にできます。

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

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

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

- [付録A Oracle Configuration Manager の操作](#)
- [configCCR\(8\) のマニュアルページ](#)
- 『*Oracle Configuration Manager* インストレーションおよび管理ガイド』
- <http://www.oracle.com/us/support/policies/index.html>
- <http://www.oracle.com/us/support/auto-service-request/index.html> にある Oracle Auto Service Request のドキュメント



## Live Media の使用

---

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

### GUI インストーラを使用したインストール

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

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

次の事項に注意してください。

- ディスク全体またはフォーマットされていないディスクにインストールする場合、インストーラは GPT フォーマットを使用します。ただし、既存の GPT パーティションまたは DOS パーティションがデフォルトで保持され、インストーラに表示されるため、既存のパーティションを保持してそこにインストールできます。詳細は、[31 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。
- このリリースでは、x86 プラットフォーム用の Oracle Solaris は GRand Unified Bootloader の新しいバージョン (GRUB 2) をインストールします。GRUB 2 については、『[Oracle Solaris 11.3 システムのブートとシャットダウン](#)』の「[GRUB 2 の概要](#)」を参照してください。

必要な場合は、インストールの前にサードパーティーまたはオープンソースのパーティション分割ツールを使用して、新しいパーティションを作成したり、既存のパーティションを調整したりできます。[31 ページの「インストール前にシステムをパーティション分割するためのガイドライン」](#)を参照してください。

- このリリースでは、iSCSI ターゲットがブートディスクとして機能でき、iSCSI ブートに必要なサポートがシステムに存在する場合は、GUI インストーラを使用

して iSCSI ターゲットに Oracle Solaris オペレーティングシステムをインストールできます。

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

詳細は、この章の [42 ページ](#) の「[GUI インストールを実行する方法](#)」のインストール手順を参照してください。 [iscsiadm\(1M\)](#) のマニュアルページも参照してください。

- GUI インストーラはオペレーティングシステムをアップグレードできません。ただし、Oracle Solaris オペレーティングシステムのインストール後に Image Packaging System を使用すると、システム上のパッケージのうち、利用可能な更新があるすべてのパッケージを更新できます。『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』を参照してください。
- GUI インストーラは、ディスク全体またはディスク上の Oracle Solaris x86 パーティションに対して初期インストールを実行できます。



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

---

## GUI インストーラでのデフォルト設定

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

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

## ▼ GUI インストールを準備する方法

GUI インストールを実行する前に、この手順のアクションを完了します。

1. **Live Media がない場合は、Live Media ISO イメージをダウンロードします。**

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

---

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

---

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

---

**注記** - USB イメージの場合は、イメージを USB フラッシュドライブにコピーするために `usbcopy` ユーティリティが必要です。このユーティリティをシステムに追加するには、`pkg:/install/distribution-constructor` パッケージをインストールします。

---

2. 使用しているシステムでインストーラを実行するための要件と制限事項を確認します。
  - a. すべての必要なシステム要件が満たされていることを確認します。  
[29 ページの「Live Media およびテキストインストールのシステム要件」](#) を参照してください。
  - b. 必要なデバイスドライバがすべて揃っていることを確認します。  
[35 ページの「適切なデバイスドライバがあることの確認」](#) を参照してください。
3. 複数のオペレーティングシステムをインストールする場合は、必要な環境を設定します。
  - a. [29 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システムの準備」](#) に記載されている仕様を確認します。
  - b. システムをバックアップします。
  - c. インストールの前にシステムをパーティション分割する場合は、[30 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。

次の手順 [42 ページの「GUI インストールを実行する方法」](#) を参照してください。

## ▼ GUI インストールを実行する方法

1. **インストールメディアを挿入して、システムをブートします。**  
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。

---

**注記** - システムのグラフィックカードが Live Media のインストール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、メディア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。

---

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

---

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

---

3. **インストールに必要な、不足しているドライバをすべてインストールします。**  
Live Media からブートした場合、不足しているドライバが存在すると、プロンプトが表示されます。デバイスドライバユーティリティの使用方法に従い、インストールに必要なドライバをすべて見つけて、インストールします。
4. **Live Media のデスクトップで、「Oracle Solaris のインストール」アイコンをダブルクリックして GUI インストーラを起動します。**
5. **開始パネルで、「次へ」を選択します。**
6. **「ディスク検出」パネルで、インストーラが検出するディスクのタイプを選択します。**
  - 「ローカルディスク」 - これは、コンピュータに接続されているディスク (内蔵および外付けハードディスクを含む) のデフォルトのオプションです。
  - 「iSCSI」 - iSCSI 標準を使用してネットワーク経由でアクセスできるリモートディスクをインストーラで検索する場合は、このオプションを選択します。追加のフィールドが次のように表示されます。
    - 「DHCP 自動検出を使用」 - システムで iSCSI ディスクの自動検出がサポートされている場合は、このオプションが有効になります。このオプションを選択

すると、条件フィールドに自動検出から返された値が取り込まれます。「検索条件の指定」オプションを選択すると、これらの値をさらに細かく調整できます。

- 「検索条件の指定」 – このオプションを選択して、iSCSI 検索の値を手動で設定できます。

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

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

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

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

次の事項に注意してください。

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

パーティションのレイアウトを変更することもできます。手順については、[31 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て、現在の現地時間に一致するように日付と時間を調整します。**

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

**9. ユーザー設定を完了します。**

- ユーザー名とパスワードを入力します。  
ユーザーのアカウントの設定を完了するには、ログイン名とパスワードを指定する必要があります。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。

---

**注記** - 作成したユーザーアカウントには管理者権限が割り当てられます。

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

---

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

**10. 「サポートの構成」パネルで、OCM および ASR への登録を構成する方法を決定します。**

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

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

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

詳細は、[35 ページの「Oracle Configuration Manager の使用」](#)を参照してください。

**11. インストール仕様を確認します。**

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

**12. 指定した仕様を使ってシステムをインストールします。**

Oracle Solaris のインストール処理が開始します。



---

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

---

**13. インストールログを確認します。**

「完了」パネルで、インストールログを表示および確認できます。

**14. システムをリブートするか、またはインストーラを終了してシステムをシャットダウンします。**

インストールが成功したら、システムをリブートするか、インストーラを終了してシステムをシャットダウンします。

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

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

## システムがコンソールモードでブートした場合に実行する操作

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

次の 2 つの方法を選択できます。

- Live Media ISO イメージの代わりにテキストインストーライメージを使用します。ローカルコンソールで、ネットワークにアクセスせずにテキストインストーラを実行できます。第4章「テキストインストーラの使用」を参照してください。
- 47 ページの「システムがコンソールモードでブートした場合に Live Media から Oracle Solaris をインストールする方法」の説明に従って、リモートインストールを実行します。

---

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

---

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

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

1. インストール先のシステムにメディアを挿入してから、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
```

8. **GUI** インストーラを実行します。

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

---

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

---

9. インストールが完了したら、ターゲットシステムをリブートします。

## Live Media インストール後のソフトウェアの追加

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

---

**注記** - パッケージのインストール、更新、およびアンインストールには強力な権限が必要です。詳細は、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[インストール権限](#)」を参照してください。

---

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

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

`pkg install` コマンドでは、最初に `-nv` オプションを使用して、実際にパッケージをインストールする前にパッケージのインストールがどのように行われるのかを確認するようにしてください。インストールするパッケージを特定し、`-nv` オプションを指定した `pkg install` コマンドからの出力を確認したあと、次の例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 --be-name new-BE-name package-name
```

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

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

## テキストインストーラの使用

---

個々の SPARC および x86 システムで対話型のテキストインストーラを実行できます。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ラを実行できます。

### テキストインストーラを使用したインストール

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

- 29 ページの「[Live Media およびテキストインストーラのシステム要件](#)」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。
  - ディスク全体またはフォーマットされていないディスクにインストールする場合、インストーラは GPT フォーマットを使用します。ただし、既存の GPT パーティションまたは DOS パーティションがデフォルトで保持され、インストーラに表示されるため、既存のパーティションを保持してそこにインストールできます。詳細は、31 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。
  - このリリースでは、Oracle Solaris インストーラは x86 システムに対して GRUB 2 を使用します。GRUB 2 は、1 つ以上のドライブ上にある複数のオペレーティングシステムのブートをサポートします。GRUB 2 については、『[Oracle Solaris 11.3 システムのブートとシャットダウン](#)』の「[GRUB 2 の概要](#)」を参照してください。

インストールの前にオープンソースまたはサードパーティーのパーティション分割ツールを使用して、新しいパーティションを作成したり、既存のパーティションを調整したりすることもできます。31 ページの「[インストール前にシステムをパーティション分割するためのガイドライン](#)」を参照してください。

- Oracle Solaris インストーラでオペレーティングシステムのアップグレードを行うことはできません。ただし、Oracle Solaris オペレーティングシステムのインストール後に Image Packaging System を使用すると、システム上のパッケージのうち、利用可能な更新があるすべてのパッケージを更新できます。『Oracle Solaris 11.3 ソフトウェアの追加と更新』を参照してください。
- iSCSI ターゲットがブートディスクとして機能し、iSCSI ブートに必要なサポートがシステムに存在する場合は、テキストインストーラを使用して iSCSI ターゲットに Oracle Solaris オペレーティングシステムをインストールできます。システムで iSCSI ディスクの自動検出がサポートされている場合は、インストーラによってそのオプションが提供されます。あるいは、インストール画面で値を手動入力して iSCSI ターゲットを指定することもできます。iSCSI を使用するには、インストール処理を開始する前に、静的 IP アドレスを使用してシステムのネットワークインタフェースを構成する必要があります。iSCSI インストールを実行するときは、次の考慮事項に注意してください。
  - SPARC プラットフォーム上の iSCSI ブートは OpenBoot レベル 4.31 以降でサポートされ、特定の NIC は必要ありません。OpenBoot の boot コマンドは、宛先の iSCSI ターゲットを識別するために一連のキーワードをとるか、network-boot-parameters NVRAM 変数に格納されているパラメータを使用します。このコマンドは boot net:keyword=value という形式を使用します。
  - x86 プラットフォームでは、ブートされるホストは iSCSI Boot Firmware Table (iBFT) 対応の NIC を使用しているか、iBFT 対応のメインボード BIOS を搭載している必要があります。iSCSI ブートを正しく構成するには、使用している特定の NIC ハードウェアのドキュメントを参照してください。

詳細は、53 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。iscsiadm(1M) のマニュアルページも参照してください。

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



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

---

- Live Media には、デスクトップまたはノートパソコンに適したソフトウェアセットが含まれています。テキストインストーラは、汎用的なシステムにより適した小さなソフトウェアセットをインストールします。具体的には、テキストインストーラでは GNOME デスクトップはインストールされません。テキストインストーラを使用して実行したインストールのあとに追加のパッケージをインストールする場合は、61 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処理パネルでは、構成するネットワークインタフェースを選択するよう求められます。構成できるのは、1つの有線ネットワークインタフェースだけです。また、ネットワーク構成プロセスをスキップするよう選択できます。

インタフェースが選択されたら、DHCPにインタフェースの構成を許可するよう選択するか、静的IPv4アドレスを使用してインタフェースを手動で構成するよう選択できます。静的アドレスの場合は、IPv4デフォルトルートも指定できます。どちらの場合も、インタフェース上でIPv6自動構成が有効になります。

## テキストインストーラ内をナビゲートする方法

パネル間をナビゲートするには、各パネルの最下部に一覧表示されているファンクションキーを使用します。特定のパネル内であるフィールドから別のフィールドに移動するには、矢印キーを使用します。キーボードにファンクションキーがない場合や、キーが反応しない場合は、ESCキーを押してナビゲーション用の代替キーを表示します。

インストール中はいつでも前のパネルに戻ることができます。

## テキストインストールのタスク

このセクションには、次のタスクが含まれています。

- [51 ページの「テキストインストーラを準備する方法」](#)
- [53 ページの「テキストインストーラを実行する方法」](#)
- [59 ページの「ネットワーク経由でテキストインストーラを開始する方法」](#)
- [61 ページの「テキストインストーラ後のソフトウェアの追加」](#)

### ▼ テキストインストーラを準備する方法

テキストインストーラを実行する前に、この手順のアクションを完了します。

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

Oracle Solaris テキストインストーラの ISO イメージをダウンロードするには、<http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html> にアクセスします。このサイトには、リムーバブルメディアで使用可能な USB イメージも含まれています。

イメージをダウンロードしたら、USB イメージの場合はリムーバブルメディアに、ISO イメージの場合はアクセス可能なファイルシステムに、イメージをコピーします。USB イメージの場合、`dd` または `usbcopy` コマンドを使用できます。`dd` コマンドでのコピー方法の例を、次に示します。

```
# dd bs=16k conv=sync if=<image path> of=/dev/rdisk/c4t0d0p0
```

---

注記 - `usbcopy` ユーティリティーをシステムに追加するには、`pkg:/install/distribution-creator` パッケージをインストールします。

---

2. 使用しているシステムでインストーラを実行するための要件と制限事項を確認します。
  - a. すべての必要なシステム要件が満たされていることを確認します。

[29 ページの「Live Media およびテキストインストーラのシステム要件」](#) を参照してください。
  - b. 必要なデバイスドライバがすべて揃っていることを確認します。

[35 ページの「適切なデバイスドライバがあることの確認」](#) を参照してください。
3. 複数のオペレーティングシステムをインストールする場合は、必要な環境を設定します。
  - a. [29 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システムの準備」](#) に記載されている仕様を確認します。
  - b. システムをバックアップします。
  - c. インストールの前にシステムをパーティション分割する場合は、[第2章「インストールの準備」](#) のガイドラインを確認してください。

特に、Oracle Solaris をパーティションまたはスライスに設定およびインストールすることを計画していて、まだ実行していない場合は、[31 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。

次の手順 [53 ページの「テキストインストーラを実行する方法」](#) を参照してください。

## ▼ テキストインストールを実行する方法

1. テキストインストールのメディアを挿入し、そのメディアを使ってシステムをブートします。要求された場合は、初期のキーボードおよび言語の選択を行います。  
x86 のインストール処理中には、キーボードおよび言語の選択を求められます。  
SPARC のインストール処理では、これらの値は事前設定されます。

---

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

---

2. (オプション) 必要なドライバをインストールするには、インストールメニューでオプション 2 を選択します。  
デバイスドライバユーティリティーの使用手順については、283 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。
3. (オプション) iSCSI ディスクの検出を使用するには、オプション 3 を選択します。  
シェルのプロンプトで、ネットワークインタフェースを構成するための手順に従います。詳細は、『Oracle Solaris 11.3 でのネットワークコンポーネントの構成と管理』の第 3 章、「Oracle Solaris での IP インタフェースとアドレスの構成および管理」を参照してください。ネットワークインタフェースを構成したら、Ctrl-D を押してシェルを終了します。
4. (オプション) IPoIB を使用している iSCSI デバイスを使用してブートするには、オプション 3 を選択します。  
詳細は、『Oracle Solaris 11.3 でのネットワークコンポーネントの構成と管理』の第 3 章、「Oracle Solaris での IP インタフェースとアドレスの構成および管理」を参照してください。

- a. IPoIB パーティションリンクを作成します。

この例では、使用可能なリンクが表示されたあと、net5 を使用してパーティションリンクが作成されます。最後のコマンドは、このリンクが作成されたことを示しています。

```
# dladm show-ib
LINK          HCAGUID          PORTGUID          PORT STATE  GWNAME  GWPORT  PKEYS
net4          212800013F2EC6  212900013F2EC7  1   down    -        --FFFF
net5          212800013F2EC6  212900013F2EC8  1   up      -        --FFFF
# dladm create-part -l net5 -P 0xFFFF ibd5
# dladm show-part
LINK          PKEY  OVER          STATE  FLAGS
ibd5          FFFF  net5          unknown ----
```

- b. IP インタフェースと IP アドレスを作成します。

この例では、前の手順で作成された IPoIB インストールの `ibd5` を使用します。  
この IP アドレスは静的である必要があります。

```
# ipadm create-ip ibd5
# ipadm create-addr -T static -a 6.6.6.53/24 ibd5/v4
```

この例では、通常 iSCSI 接続に使用される `net0` を使用します。

```
# ipadm create-ip net0
# ipadm create-addr -a 129.144.83.5/8 net0
```

c. **Ctrl + D** キーを押してシェルを終了します。

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

```
Welcome to the Oracle Solaris 11.3 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]:
```

「継続」ファンクションキーを使用して、次のパネルに移動します。

---

**注記** - 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。

---

6. 「検出の選択」パネルで、システムをインストールするディスクの検出方法を選択します。

- 「ローカルディスク」 - これは、コンピュータに接続されているディスク (内蔵および外付けハードディスクを含む) のデフォルトのオプションです。
- 「iSCSI」 - iSCSI 標準を使用してネットワーク経由でアクセスできるリモートディスクをインストーラで検索する場合は、このオプションを選択します。iSCSI を使用する場合は、手順 3 を完了してください。追加のパネルでは、次の情報の入力を求められます。

ターゲット IP            iSCSI ターゲットの IP アドレス。0-255 の範囲の 4 つの数値を指定します。この IP アドレスのシステムがオンラインであり、アクセス可能である必要があります。このフィールドは必須です。

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

ターゲット名	iSCSI ターゲットの iSCSI 修飾名 (IQN) 形式の名前。このフィールドはオプションです。
ポート	iSCSI デバイスを検出するために、指定した IP アドレスと組み合わせて使用されるポート番号。デフォルト値の 3260 は、iSCSI に通常使用されるポートです。このフィールドはオプションです。
イニシエータ名	iSCSI 検出セッション用に設定されるイニシエータノード名。iSCSI ブートの場合は、イニシエータノード名を変更できないため、このフィールドは非表示になっています。このフィールドは自動的に生成されます。
CHAP 名	認証に CHAP を使用する場合は、認証に使用する CHAP (チャレンジハンドシェイク認証プロトコル) 名。このフィールドはオプションです。
CHAP のパスワード	認証用の CHAP シークレットの値。入力する場合は、この値の長さを 12 から 16 文字にする必要があります。このフィールドはオプションです。

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

7. 「ディスク」パネルで、**OS をインストールするディスクを選択**します。  
複数のターゲットディスクがリストされている場合は、いずれかのディスクを選択するか、デフォルトを受け入れます。
8. 「パーティション」パネルで、**オペレーティングシステムをディスク全体またはディスクの一部のどちらにインストールするかを選択**します。  
次の選択肢が表示されます。
  - ディスク全体を使用
  - GPT パーティションを使用

---

注記 - SPARC のインストール中、このパネルではパーティションではなくスライスに関する情報の入力を求められます。

---

9. (オプション) 「パーティションの選択」パネルで、**パーティションのレイアウトを変更**します。  
インストールパネルを進めているときはいつでも、元の設定に戻すことができます。



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

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

次の事項に注意してください。

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

SPARC のインストール処理によって、ディスクスライスに関する情報の入力を求められます。

パーティション分割の詳細な手順については、[31 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。

10. **(オプション)「ブートプール」パネルで、ブートプールデバイスを変更します。**  
このパネルでは、専用のオンボードデバイスが最初に表示されます。デフォルトでは、「専用」および「選択」列に、これらのデバイスに対する「はい」が表示されます。F5 キーを使用すると、ブートプールデバイスを変更できます。
11. 「システム識別情報」パネルで、ネットワーク上でシステムを識別するためのコンピュータ名を指定します。
12. 「ネットワーク」パネルで、有線 Ethernet ネットワーク接続の構成方法を指定します。
  - 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。  
インストーラでは続けて「タイムゾーン」パネルが表示されます。

- 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. 「手動構成」パネルで、接続設定を入力するか、インストーラによって検出および提供されたデフォルトの情報を受け入れます。

---

注記 - 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. DNS ネームサービスをシステムで使用するよう選択した場合は、「DNS ネームサービス」パネルで次の設定を行います。
  - i 「DNS サーバーのアドレス」パネルで、DNS サーバーの IP アドレスを少なくとも 1 つ入力します。
  - ii 「DNS 検索リスト」パネルに、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。
- d. 「代替ネームサービス」パネルで、システムが LDAP ネームサービス、「なし」のどちらを使用すべきであるかを指定します。
  - 前の手順で DNS を選択した場合は、DNS に加えて LDAP 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 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。
  - 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` などの標準の名前ソースファイルを使用してネットワーク名を解決できません。詳細は、[nsswitch.conf\(4\)](#) のマニュアルページを参照してください。
- e. 代替ネームサービスを選択した場合は、「ドメイン名」パネルで、システムが属するドメインを指定します。

---

**注記** - ドメイン名を確認するには、システム管理者にお問い合わせください。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。

---

- f. 「代替ネームサービス」パネルで **LDAP** を選択した場合は、「**LDAP プロファイル**」パネルで、次の **LDAP** 構成の指定項目を設定します。
- 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
  - LDAP プロファイルサーバーの IP アドレス
  - LDAP 検索ベース
  - 「LDAP プロキシ」パネルで、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。  
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。

13. 「**タイムゾーン**」パネルで、**地域、場所、およびタイムゾーン**を選択します。

---

**注記** - デフォルトでは、GMT タイムゾーンが構成されます。

---

14. 「**ロケール**」パネルで、**言語および言語の地域**を選択します。

15. 次のパネルで**日付と時間**を設定します。

16. 次のパネルで**キーボード配列**を選択します。

17. 「**ユーザー**」パネルで**アカウント**を作成します。

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

- このパネルでユーザーアカウントを作成する場合は、**ユーザーのパスワードと root パスワードの両方を指定する必要があります。**  
この場合、root はユーザーに割り当てられた役割になります。  
ユーザーアカウントを作成するには、ユーザー名とパスワードを入力します。この名前は英文字で始まる必要があり、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。

18. 「サポート - 登録」パネルで、**Oracle Configuration Manager** を使用するかどうか、**Oracle Configuration Manager** を使用する方法、または **Oracle Auto Service Request** を開始するかどうかを決定します。  
詳細は、[35 ページの「Oracle Configuration Manager の使用」](#)を参照してください。
19. 「サポート - ネットワーク構成」パネルで、**OCM** および **ASR** へのアクセス方法を選択します。  
次のオプションを指定できます。
  - 「プロキシなし」
  - 「プロキシ」 - セキュアプロキシを使用している場合は、次のパネルでプロキシホスト名、ポート番号、およびユーザー名とパスワードの入力を求められます。
  - 「集約ハブ」 - 次のパネルで、OCM ハブの URL および ASR Manager の URL の入力を求められます。
20. **インストール仕様を確認します。**  
「インストールのサマリー」パネルで、仕様を確認します。インストールを開始する前に、必要に応じて以前の手順に戻って変更を加えます。
21. **指定した仕様を使ってシステムをインストールします。**  
再起動ファンクションキーを使用して、Oracle Solaris のインストール処理を開始します。



---

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。

---

22. **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。
23. **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。**

## ▼ ネットワーク経由でテキストインストールを開始する方法

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システムをブートしてから対話型テキストインストールを開始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。

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、『Oracle Solaris 11.3 システムのインストール』の「インストールサービスの作成」を参照してください。

2. AI クライアントをネットワーク経由でブートします。

- SPARC AI クライアントの場合は、OBP プロンプトで次のコマンドを入力します。

```
# boot net:dhcp
```

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

```
Welcome to the Oracle Solaris 11.3 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. 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、53 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。

---

**注記** - 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。ブート後にネットワーク経由でテキストインストーラを使用すると、より小さなパッケージセット solaris-auto-install がデフォルトでインストールされます。

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

パッケージのインストール、更新、およびアンインストールには強力な権限が必要です。詳細は、『Oracle Solaris 11.3 ソフトウェアの追加と更新』の「インストール権限」を参照してください。

```
# pkg install solaris-desktop
```

```
# pkg install solaris-large-server
```

---

## テキストインストール後のソフトウェアの追加

オペレーティングシステムをインストールしたあとでソフトウェアパッケージを追加するには、[pkg\(1\)](#)のマニュアルページの説明に従って `pkg` 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することもできます。

---

**注記** - パッケージのインストール、更新、およびアンインストールには強力な権限が必要です。詳細は、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[インストール権限](#)」を参照してください。

---

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

`pkg install` コマンドでは、最初に `-nv` オプションを使用して、実際にパッケージをインストールする前にパッケージのインストールがどのように行われるのかを確認するようにしてください。インストールするパッケージを特定し、`-nv` オプションを指定した `pkg install` コマンドからの出力を確認したあと、次の例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 package-name
```

代わりに、新しいブート環境を作成し、その新しいブート環境にパッケージをインストールする場合は、次のコマンドを使用します。

```
# pkg install --require-new-be --be-name new-BE-name package-name
```

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



## メディアからブートする自動インストール

---

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

### AI メディアを使用するインストールの概要

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

- AI 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、診断を実行できます。

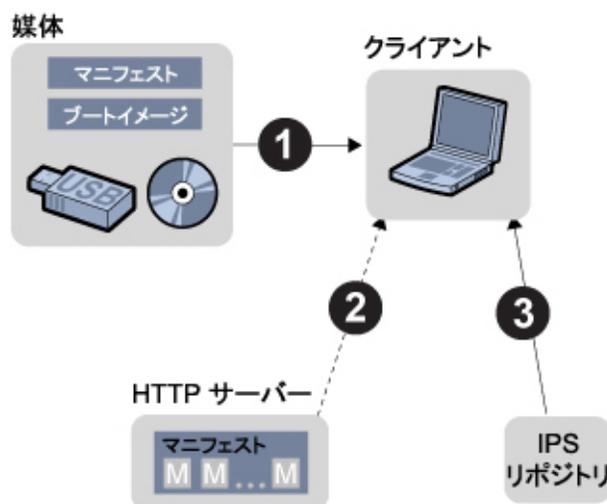
AI メディアを使用するインストールには、次の特徴があります。

- AI 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。[66 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 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

- 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 11.3 ご使用にあたって](#)』を参照してください。
- ディスク容量 - 最新リリースのディスク容量要件を確認するには、『[Oracle Solaris 11.3 ご使用にあたって](#)』を参照してください。
- ネットワークアクセス - インストールされるシステムは、クライアントシステムにインストールするパッケージが含まれる IPS リポジトリにアクセスできる必要があります。また、カスタム AI マニフェストを作成する場合、システムは HTTP サーバー上のそのマニフェストにアクセスできる必要があります。

## ▼ AI メディアを使用してインストールする方法

### 1. AI ブートイメージをダウンロードします。

AI ブートイメージをダウンロードするには、次の場所にアクセスします。<http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html>

### 2. デフォルトの AI マニフェストを確認します。

AI イメージに用意されているデフォルトのマニフェストを使用するか、またはカスタムマニフェストを作成して、クライアントをブートするときはそのカスタムマニフェストの場所を指定できます。66 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。

- ISO イメージ - .iso ファイルを CD または DVD に書き込みます。
- USB イメージ - `usbcopy` ユーティリティーを使用して、イメージを USB フラッシュドライブにコピーします。

---

注記 - このユーティリティーをシステムに追加するには、`pkg:/install/distribution-creator` パッケージをインストールします。

---

### 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、67 ページの「AI メディアからの SPARC システムのブート」および 68 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。

77 ページの「SCI ツールを使用したシステム構成プロファイルの作成」を参照してください。

## ▼ SPARC システムに USB フラッシュドライブの永続的なデバイス別名を作成する方法

SPARC システムに USB フラッシュドライブの永続的なデバイス別名を作成するには、OBP を使用する必要があります。このため、システムを停止する必要があります。別名が作成されると、同じポートを再使用するかぎり、別名を再作成する必要はありません。

1. システムを停止して、ブートプロンプトの状態のままにします。
2. システム上の使用可能なディスクを確認します。  
この例は、2 番目のデバイスを選択したことを示しています。

```
{0} ok show-disks
a) /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@3/disk
b) /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk
c) /pci@400/pci@0/pci@1/scsi@0/disk
d) /iscsi-hba/disk
q) NO SELECTION

Enter Selection, q to quit: b
/pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev
      for creating devalias mydev for
/pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk
```

3. USB フラッシュドライブの別名を設定します。

```
{0} ok nvalias usbdrive ^Y
```

4. USB フラッシュドライブからブートします。

```
{0} ok boot usbdrive
Boot device: /pci@400/pci@0/pci@9/pci@0/usb@0,2/hub@2/storage@2/disk File
and args:
```

## カスタム AI マニフェストの作成

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

.iso AI イメージをダウンロードする場合は、次のサンプルコマンドを使用して、そのイメージ内の AI マニフェストを調べることができます。この例では、/tmp は

AI イメージをダウンロードしたディレクトリで、`/home/username` は AI マニフェストをコピーおよび編集するディレクトリです。AI マニフェストは、イメージ内の `auto_install/manifest/default.xml` にあります。

```
# /usr/sbin/mount -o ro -F hsfs /tmp/sol-11_3-20-ai-x86.iso /mnt
# cp /mnt/auto_install/manifest/default.xml /home/username/custom.xml
# umount /mnt
```

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

インストールするターゲットディスクや追加パッケージなどのインストール仕様を変更する方法については、[aimanifest\(1M\)](#) のマニュアルページを参照してください。

AI マニフェストの変更が終わったら、カスタムマニフェストを HTTP サーバーにコピーします。カスタム AI マニフェストの URL を書き留めて、インストールするシステムをブートするときにその 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 マニフェストを使用して、AI メディアから SPARC システムをブートする

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 マニフェストを使用して、AI メディアから SPARC システムをブートする

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

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

次のプロンプトが表示されます。

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

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

カスタムマニフェスト内の仕様を使って自動インストールが進行します。

## SPARC イメージをブートするときにインストールを行わない

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

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

```
ok> boot cdrom
```

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

## AI メディアからの x86 システムのブート

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

GRUB メニューの選択項目は、次の例のように表示されます。

```
GNU GRUB version 1.99.5.11.0.175.2.0.0.20.0
```

```
Oracle Solaris 11.3 Automated Install custom
Oracle Solaris 11.3 Automated Install
Oracle Solaris 11.3 Automated Install custom ttya
Oracle Solaris 11.3 Automated Install custom ttyb
Oracle Solaris 11.3 Automated Install ttya
Oracle Solaris 11.3 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 マニフェストを使用して、AI メディアから x86 システムをブートする

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 11.3 Automated Install
Oracle Solaris 11.3 Automated Install ttya
Oracle Solaris 11.3 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 マニフェストを使用して、AI メディアから x86 システムをブートする

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 11.3 Automated Install custom
Oracle Solaris 11.3 Automated Install custom ttya
Oracle Solaris 11.3 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 AI manifest location [URL, /filepath, '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してもインストールしない場合があります。たとえば、システムのトラブルシューティングを行ったり、システムを調べたりする場合です。

使用する GRUB2 エントリの `$multiboot` で始まる行に `install=true` が指定されている場合は、自動的にインストールが開始されます。x86 システムをブートして自動インストールをすぐに開始しない場合、使用する予定の GRUB2 エントリのカーネル行に `install=true` が指定されているときは、その行を編集して `install=true` を削除します。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/log/install/install_log` にあります。

## Oracle Solaris インスタンスの構成解除または再構成

---

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

### 機能グループの概要

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は、システム全体または特定のサブシステムの構成データを変更できます。これらのサブシステムは機能グループと呼ばれます。**system** 機能グループは、システム上のすべての機能グループを変更します。または、1つ以上の機能グループを指定して、特定の構成コンポーネントのみを変更できます。

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 5 機能グループ

グループ	コンポーネント	未構成の状態
date_time	システム日付および時間	なし
identity	システムのノード名	不明
keyboard	キーボード	アメリカ英語
location	タイムゾーン	UTC
	ロケール	C ロケール
naming_services	DNS 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
network	ネットワーク	ネットワークなし
support	OCM および ASR のサポート	デフォルト設定は、OCM および ASR への匿名登録です。

グループ	コンポーネント	未構成の状態
system	システム全体	「system」グループには、ほかのすべてのグループが含まれます。
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

## Oracle Solaris インスタンスの構成解除

以前に構成した 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場合は、`unconfigure` サブコマンドを使用します。すべての機能グループまたは一部の機能グループのみを構成解除することを選択できます。詳細は、[sysconfig\(1M\)](#) のマニュアルページを参照してください。

### 例 1 システムの構成解除

次の例のように `sysconfig unconfigure` コマンドを使用して、システムからすべての構成データを削除します。

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

### 例 2 SMF データの構成解除

Oracle Solaris サービス管理機能 (SMF) では、システムが構成解除されても削除されないいくつかの構成データが格納されます。特に、`/etc/svc/profile/site` および管理レイヤーに格納されているデータは、機能グループの一部ではありません。このデータを削除して、システムを構成解除するには、次のコマンドを使用します。

```
# sysconfig unconfigure -g system --include-site-profile
```

削除される XML プロファイルは、`/etc/svc/profile/sysconfig/site-profile.tar` という名前の tar ファイルにアーカイブされます。

### 例 3 特定の機能グループの構成解除

この例では、システムの日付および時間のデータを構成しています。

```
# sysconfig unconfigure -g date_time
```

---

**注記** `-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。

---

**例 4** システムの構成解除後のリポート

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
```

システムがリポートすると SCI ツールが実行されて、システムが再構成されます。

## システムの再構成

`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、非対話型でも対話型でも実行できます。`sysconfig configure` コマンドに `-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。このコマンド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。`-c` を指定せずにコマンドを実行すると、SCI ツールが自動的にアクティブ化されます。

---

注記 - `sysconfig reconfigure` コマンドは `sysconfig configure` の別名です。

---

**例 5** システム構成プロファイルを使用したシステムの再構成

次のコマンドはシステムが `myprofile.xml` という名前の既存のシステム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
```

### ▼ SCI Tool を使用して再構成する方法

1. `root` の役割になります。
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。

---

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

---

3. 初期の開始パネルを過ぎて先に進みます。
4. ネットワーク上でシステムを識別するための名前を指定します。
5. 「ネットワーク」パネルで、有線 Ethernet ネットワーク接続の構成方法を指定します。
  - 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。  
インストーラでは続けて「タイムゾーン」パネルが表示されます。
  - DHCP を使用してネットワーク接続を構成するには、「DHCP」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。
  - 選択したインタフェース上で静的 IPv4 アドレスを手動で構成するには、「静的」を選択します。
    - a. 「手動構成」パネルで、接続設定を入力するか、インストーラによって検出および提供されたデフォルトの情報を受け入れます。

---

注記 - 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。

---

- i DNS ネームサービスをシステムで使用するよう選択した場合は、「DNS ネームサービス」パネルで次の設定を行います。
  - 「DNS サーバーのアドレス」パネルで、DNS サーバーの IP アドレスを少なくとも 1 つ入力します。
  - 「DNS 検索リスト」パネルに、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。
- ii 「代替ネームサービス」パネルで、システムが LDAP ネームサービス、「なし」のどちらを使用すべきであるかを指定します。
  - 前の手順で DNS を選択した場合は、DNS に加えて LDAP 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 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。
- 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` 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。

- b. 代替ネームサービスを選択した場合は、「ドメイン名」パネルで、システムが属するドメインを指定します。

---

注記 - 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。

---

- c. 「代替ネームサービス」パネルで LDAP を選択した場合は、「LDAP プロファイル」パネルで、次の LDAP 構成の指定項目を設定します。

- 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
- LDAP プロファイルサーバーの IP アドレス
- LDAP 検索ベース
- 「LDAP プロキシ」パネルで、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。  
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。

6. 「タイムゾーン」パネルで、地域、場所、およびタイムゾーンを選択します。

---

注記 - デフォルトでは、GMT タイムゾーンが構成されます。

---

7. 「ロケール」パネルで、言語および言語の地域を選択します。
8. 次のパネルで日付と時間を設定します。
9. 次のパネルでキーボード配列を選択します。
10. 「ユーザー」パネルを完了します。  
ユーザーアカウントを作成する必要はありませんが、root パスワードを作成する必要があります。

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

この場合、root はユーザーに割り当てられた役割になります。

ユーザーアカウントを作成するには、ユーザー名とパスワードを入力します。この名前は英文字で始まる必要があり、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。

11. 「サポート - 登録」パネルで、My Oracle Support の電子メールアドレスおよびパスワードを入力します。

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

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

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

詳細については、『Oracle Solaris 11.3 システムのインストール』の「Oracle Configuration Manager の使用」を参照してください。

12. 「サポート - ネットワーク構成」パネルで、OCM および ASR へのアクセス方法を選択します。

次のオプションを指定できます。

- 「プロキシなし」
- 「プロキシ」 - セキュアプロキシを使用している場合は、次のパネルで、プロキシホスト名、ポート番号、およびユーザー名とパスワードの入力を求められます。
- 「集約ハブ」 - 次のパネルで、OCM ハブの URL および ASR Manager の URL の入力を求められます。

**13. インストール仕様を確認します。**

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

**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。**

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。

## SCI ツールを使用したシステム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。

新しいシステム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。

---

**注記** - そのプロファイルを再構成に正しく使用するためには、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「AI クライアントのシステム構成パラメータの定義」も参照してください。

---

### 例 6 デフォルトプロファイルの使用

SCI Tool は、SCI Tool パネルで入力された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 /etc/system/profile/sysconfig/sc_profile.xml
```

### 例 7 プロファイルの作成と使用

`-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5](#)を参照してください。次の例は、プロファイルを作成し、そのプロファイルを使用して非対話形式でシステムを再構成しています。

```
# sysconfig create-profile -o /tmp/myprofiles
# sysconfig configure -g system -c /tmp/myprofiles
```

この例では、`-g` オプションをファイルに適用しています。

```
# sysconfig configure -g system -c /tmp/myprofiles/sc_profile.xml
```

**例 8**                    プロファイルを作成および使用して機能グループを構成する

次の例は、`network` および `naming_services` 機能グループのプロファイルを作成しています。その後、プロファイルを使用して機能グループを非対話形式で再構成しています。

```
# sysconfig create-profile -g network,naming_services -o /tmp/myprofile.xml
# sysconfig configure -g network,naming_services -c /tmp/myprofile.xml
```

## 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。この章の内容は次のとおりです。

- 第7章「複数システムへの自動インストール」
- 第8章「AI サーバーの設定」
- 第9章「AI クライアントへのカスタマイズの割り当て」
- 第10章「AI クライアントのインストールパラメータの定義」
- 第11章「AI クライアントのシステム構成パラメータの定義」
- 第12章「ゾーンのインストールと構成」
- 第13章「初回ブート時のカスタムスクリプトの実行」
- 第14章「AI サーバーを使用した AI クライアントのインストール」
- 第15章「自動インストールのトラブルシューティング」



## 複数システム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 オペレーティングシステム (OS) をネットワーク内の複数の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 AI 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または Oracle Solaris 統合アーカイブ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 OS のインストールを自動化します。ディスクレイアウトおよびソフトウェアパッケージの選択を定義するインストール前手順を使用すると、インストールをカスタマイズできます。ホスト名、ネットワーク構成、ユーザーアカウント、およびその他のインストール後のクライアント固有の手順を定義するカスタムシステム構成パラメータを指定することもできます。すべての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。

---

**注記** - Intel® Virtualization Technology for Directed I/O (VT-d) を備えた Oracle x86 以外のシステムに Oracle Solaris をインストールする場合は、その前に、それらのシステムの Intel VT-d パラメータを **Enabled** に設定しておく必要があります。このパラメータを設定する手順については、それぞれのドキュメントを参照してください。

---

ネットワーク経由でのシステムの自動インストールは、次の概要レベルのステップで構成されます。

1. AI クライアント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 AI サーバーの場所を DHCP サーバーから取得します。SPARC クライアントは、必要に応じて、オープンブート PROM (OBP) に設定された `network-boot-arguments` 変数からネットワーク構成情報と AI サーバーの場所を取得できます。
2. インストールサービスによって AI クライアントにブートイメージが提供されます。
3. AI クライアントの特性によって、AI 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。

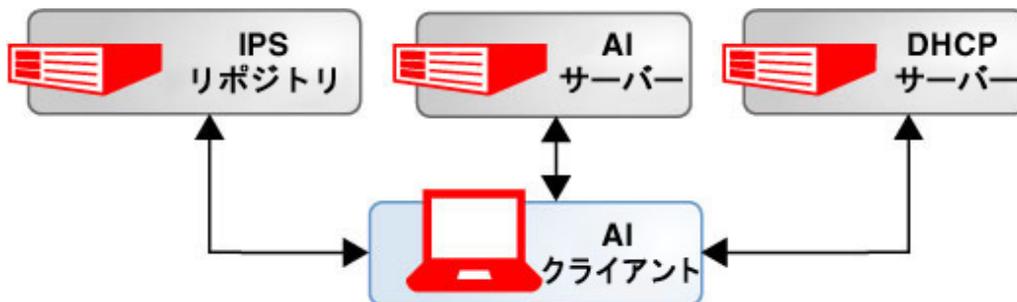
4. Oracle Solaris OS が AI クライアントにインストールされます。インストーラは、インストールサービスのインストール手順に指定されているパッケージリポジトリ内のパッケージまたはアーカイブ内のシステムイメージを取得します。

## 自動インストーラのコンポーネント

AI を使用するネットワークは、次のコンポーネントで構成されます。

- AI クライアントにホスト情報を提供する DHCP サーバー。
- AI クライアントにインストールするソフトウェアパッケージを提供する 1 つ以上の IPS (Image Packaging System) リポジトリ。AI クライアントのシステムイメージは、ネットワーク上に存在するアーカイブからも作成できます。
- AI クライアントの構成手順が格納されている AI サーバー。
- 1 つ以上の AI クライアント。

図 2 AI ネットワークの例



DHCP サーバー、IPS サーバー、および AI サーバーは、別のシステムでホストする必要はありません。特に、DHCP サービスがインストールサービスと同じ場所にある場合は `installadm` コマンドによって DHCP サービスが更新されるため、AI サーバーと DHCP サーバーを同じシステムにインストールすると管理作業が容易になります。

AI サーバー自体には、次のコンポーネントを含めることができます。

- 1 つ以上のインストールサービス。各サービスは、アーキテクチャーおよび AI クライアントにインストールされる OS と一致するように構成されます。
- 1 つ以上の AI マニフェスト。AI マニフェストは、インストール手順 (使用するディスクレイアウト、追加するパッケージなど) を提供します。
- オプションのシステム構成プロファイル。これらのプロファイルは、システム構成情報 (使用するタイムゾーン、ネームサービスなど) を提供します。

また、IPS パッケージを作成すると、インストール処理中に初回ブートスクリプトを AI クライアントに提供できます。このスクリプトは、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して行うことができない追加のインストール手順または構成手順 (AI クライアントへのサードパーティーのユーティリティの追加など) を実行できます。

## AI をサポートする DHCP サーバー

もっとも単純な構成では、AI は DHCP を使って IP アドレス、サブネットマスク、ルーター、ネームサービスサーバー、および AI サーバーの場所を AI クライアントに提供します。DHCP サーバーは AI サーバーと同じシステム上で動作するように構成できますが、それが、もっとも管理の容易な構成となります。DHCP サーバーが別のシステムにある場合は、新しい AI クライアントを追加するときに DHCP サーバーを手動で更新する必要があることがあります。

SPARC クライアントは DHCP を使用せずに AI サーバーを見つけられるように構成できますが、x86 クライアントはそのように構成できません。DHCP を実行せずに x86 クライアントをインストールする代替の方法については、[255 ページの「x86 AI クライアントのインストール」](#)を参照してください。SPARC クライアントが DHCP なしで AI サーバーを使用するように、OBP を使用して必要な情報を提供する手順については、[252 ページの「SPARC AI クライアントのインストール」](#)も参照してください。

## AI をサポートする IPS リポジトリ

インストールするクライアントシステムは、Oracle Solaris Image Packaging System (IPS) ソフトウェアパッケージリポジトリ、または Oracle Solaris 統合アーカイブにアクセスできる必要があります。リポジトリは、ソフトウェアパッケージを取得する場所です。場所は URI (Universal Resource Identifier) によって指定されます。IPS パッケージリポジトリは、AI サーバー上、ローカルネットワークの別のシステム上、またはインターネット上に配置できます。パッケージリポジトリへのアクセスについては、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[パブリッシャーの構成](#)」を参照してください。IPS サーバーは、AI クライアントを完全に構成するために必要な初回ブートスクリプトを提供することもできます。

## AI サーバー

AI を使用してネットワーク経由でシステムを自動インストールするには、まず AI 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。詳細な手順につ

いては、[第8章「AI サーバーの設定」](#)を参照してください。この手順の一部では、AI サーバーの静的ネットワークアドレスを作成する方法を説明しています。これは、AI サーバーの IP アドレスは、各 AI クライアントに作成されるファイルに含められるためです。AI サーバーの IP アドレスが変更された場合は、すべての AI クライアントの構成ファイルを再作成する必要があります。

## インストールサービス

各サーバーには、1つ以上のインストールサービスを含めることができます。サポートする必要がある各バージョンの OS および各クライアントアーキテクチャー用のインストールサービスを作成する必要があります。たとえば、Oracle Solaris 11.2 をブートする SPARC クライアント、Oracle Solaris 11.3 をブートする別の SPARC クライアント、および x86 クライアントに同じサービスを提供する2つのクライアントのインストールサービスを作成する場合があります。各インストールサービスには、SPARC または x86 のブートイメージ、1つ以上のインストール手順のファイル (AI マニフェスト)、およびオプションのシステム構成プロファイルが含まれています。[107 ページの「インストールサービスの作成」](#)では、インストールサービスを作成および保守するための手順を説明しています。

AI サーバーによって提供されるブートイメージは完全なインストールではありません。ブートイメージは、インストールの実行が可能なシステム上で構成を作成します。クライアントシステムは、IPS パッケージリポジトリまたはアーカイブにアクセスしてインストールを完了させる必要があります。

## AI マニフェスト

AI マニフェストには、クライアントのプロビジョニングまたはインストール手順が含まれています。各 AI クライアントは1つの AI マニフェストのみを使用しますが、多数の AI クライアントが1つのマニフェストを共有できます。AI マニフェストによって、AI 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。IPS パッケージの代わりに使用可能なアーカイブを使用することもできます。AI マニフェストには、インストールする追加パッケージの名前と、ターゲットのインストールデバイスやパーティション情報などの情報も含めることができます。2つのクライアントシステムに同じバージョンの Oracle Solaris OS をインストールする必要があるが、異なるインストールを行う必要がある部分がある場合は、1つの AI インストールサービスに関連付けられた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。AI クライアントのブート前に、またはインストール時に動的に、AI マニフェストを作成およびカスタマイズする方法については、[第10章「AI クライアントのインストールパラメータの定義」](#)を参照してください。

## システム構成プロファイル

AI クライアントに異なる構成を適用する必要がある場合は、インストールサービスに対して複数のシステム構成プロファイルを作成します。システム構成プロファイルを別にするによって、たとえば、異なるネットワークまたはロケール設定を指定したり、独自のホスト名や IP アドレスを指定したりできます。タイムゾーンを設定するプロファイルは、複数の AI クライアントが使用できます。プロファイルについては、[第11章「AI クライアントのシステム構成パラメータの定義」](#)を参照してください。

AI クライアントにプロファイルが構成されていない場合は、インストールが完了して AI クライアントがブートされたあとに、対話型ツールによってシステム構成情報の入力を求められます。対話型構成ツールについては、[73 ページの「システムの再構成」](#)を参照してください。

## 初回ブートスクリプト

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。詳細については、[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。

## AI クライアント

AI クライアントをブートするとインストールが開始します。AI クライアントをブートすると、クライアントは AI 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関連付けられている正しい 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9章「AI クライアントへのカスタマイズの割り当て」](#)では、AI クライアントのインストール時に使用する正しい AI マニフェストとシステム構成プロファイルを AI サーバーで特定する方法について説明します。

## AI のセキュリティ保護

Transport Layer Security (TLS) プロトコルを使用して、自動インストールをセキュリティ保護することができます。プライベート証明書、鍵のペア、および証明書発行局 (CA) 証明書を AI サーバーおよび AI クライアントに割り当てることができます。OBP HMAC および暗号化鍵を使用すると、SPARC クライアントをさらにセキュリティ保護できます。詳細は、[118 ページの「自動インストールのセキュリティの向上」](#)を参照してください。

## AI およびゾーン

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。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。

## AI 構成プロセスの概要

このセクションでは、AI の主な使用方法について説明します。カスタマイズを自動的に行う度合いによっては、これらのアクションの一部がオプションとなります。[89 ページの「自動インストーラの使用事例」](#)では、状況に適したアクションについて説明しています。

1. AI クライアントをサポートするために必要な構成が提供されるように AI サーバーを構成します。詳細な手順については、[102 ページの「AI サーバーを設定する方法」](#)を参照してください。AI サーバーを構成する場合:
  - 正しい OS がインストールされます。
  - ネットワークインタフェースが、静的な IP アドレスを使用して構成されます。
  - `installadm` パッケージがインストールされます。必要に応じてマルチキャスト DNS を有効にできます。
2. インストールサービスを作成して、AI クライアントが使用するブートイメージを作成します。インストール可能である必要がある各クライアントのアーキテクチャーおよび各バージョンの OS のインストールサービスを作成する必要があります。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 AI 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、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。詳細は、[107 ページの「インストールサービスの作成」](#)を参照してください。
3. (オプション) AI クライアントをサービスに関連付けます。[114 ページの「インストールサービスへの AI クライアントの関連付け」](#)を参照してください。このステップは、1 つのクライアントアーキテクチャーで複数のバージョンの OS をサポートする場合に必要となります。
4. (オプション) AI マニフェストを作成して、AI クライアントをインストールするために使用するパッケージリポジトリまたはアーカイブがある場所を定義します。詳細については、[第10章「AI クライアントのインストールパラメータの定義」](#)を参照してください。AI マニフェストを作成するときに、条件を指定することに

よって、マニフェストを特定の AI クライアントに関連付けることができます。詳細については、[149 ページの「AI 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」](#)を参照してください。AI マニフェストを作成したら、マニフェストをインストールサービスに関連付ける必要があります。詳細な手順については、[116 ページの「インストールサービスとクライアント固有のインストール手順の関連付け」](#)を参照してください。AI クライアントのインストールをカスタマイズする場合は、マニフェストの作成およびマニフェストとインストールサービスの関連付けが必要となります。カスタマイズ可能な情報の一部を次に示します。

- ディスクレイアウト
  - IPS リポジトリまたはアーカイブの場所
  - インストールするロケール
  - インストールするパッケージ
5. (オプション) 1つ以上のシステム構成プロファイルを作成して、AI クライアントの追加の構成情報を提供します。詳細については、[第11章「AI クライアントのシステム構成パラメータの定義」](#)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、条件を指定することによって、プロファイルを特定の AI クライアントに関連付けることができます。詳細については、[149 ページの「AI 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」](#)を参照してください。システム構成プロファイルを作成したら、プロファイルをインストールサービスに関連付ける必要があります。詳細な手順については、[117 ページの「インストールサービスへのクライアント固有の構成手順の関連付け」](#)を参照してください。次のいずれかを自動的に構成する場合は、1つ以上のシステム構成プロファイルの作成、およびシステム構成プロファイルとインストールサービスの関連付けを行う必要があります。
- SMF サービス (有効化、無効化、およびプロパティの設定を含む)
  - ユーザーアカウントとグループ
  - ノード名
  - タイムゾーンとロケール
  - 端末タイプとキーボード配列
  - ネットワークインタフェース
  - ネームサービス
  - Oracle Configuration Manager と Oracle Auto Service Request (ASR)

システム構成プロファイルを作成することによって、これらの項目のいずれかまたはすべてを構成することを選択できます。AI クライアントがインストールされたら、システム構成プロファイルに指定されていない項目を手動で構成する必要があります。プロファイルを使用しないことを選択した場合は、インストールが完了したあとに、変更を行うように求められます。

6. (オプション) 初回ブートスクリプトを作成して、AI マニフェストまたはプロファイルに追加できない構成要件を指定します。詳細については、[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。

7. 必要に応じて、AI クライアントが DHCP サーバーと、IPS サーバーまたはアーカイブを格納するシステムにアクセスできることを確認します。
8. インストールサービスを使用してネットワーク経由で AI クライアントをブートします。

## AI クライアントのブート

AI クライアントをネットワーク経由でブートすると、次のアクションが行われます。

1. AI クライアントは DHCP サーバーから AI サーバーアドレスを取得します。  
SPARC クライアントは、必要に応じて、OBP の `network-boot-arguments` 変数から AI サーバーのアドレスを取得できます。
2. AI クライアントが AI サーバーからブートプログラムを取得して、ブートプログラムをロードします。
  - SPARC クライアントは AI サーバーから `wanboot` ブートプログラムを取得します。
  - X86 クライアントは AI サーバーから GRUB メニュー (インストールサービス名が含まれています) および `pxegrub` ブートプログラムを取得します。
3. AI クライアントが AI サーバーからブートアーカイブをダウンロードして、カーネルをロードします。
4. AI クライアントが HTTP を使用してインストールプログラムをダウンロードします。  
SPARC クライアントは、インストールプログラムを使用してインストールサービス名を取得します。
5. AI サーバーは選択条件に基づいてマニフェストを選択します。選択したマニフェストに基づいて、AI クライアントでインストールの変更が行われます。
6. AI サーバーが AI クライアントの選択条件に基づいてプロファイルを選択します。
7. AI インストールプログラムが、IPS リポジトリからの AI クライアントパッケージ、またはアーカイブからのイメージをインストールします。
8. ローカルディスク上のイメージを使用するために、AI クライアントがリブートされます。
9. システム構成プロファイルが使用されていない場合は、システムの構成情報の入力を求められます。
10. 該当する場合は、選択されたプロファイルに指定されている情報に基づいて、AI クライアントでシステム構成の変更が行われます。
11. 該当する場合は、初回ブートスクリプトが実行されます。

## AI サーバーの計画

AI サーバーをインストールする前に、このセクションで説明している構成の選択を行う必要があります。

### AI サーバーでのネットワークインタフェースの構成

AI サーバーは複数のインタフェースを持つシステムに構成できます。デフォルトでは、AI サーバーはすべてのネットワークインタフェース上の AI クライアントをサポートするように構成されます。AI サーバーをホストするシステムに複数のネットワークインタフェースがある場合は、[例9「ネットワーク上の AI サポートの無効化」](#)に示されているように、システムがサポートしないネットワークのインストールサービスを無効化できます。

### 必要なインストールインスタンスの識別

各 AI サーバーには1つ以上のインストールサービスが含まれています。各インストールサービスは、アーキテクチャー (SPARC または x86) および指定された OS のリリース (Oracle Solaris 11.2 または 11.3 など) をサポートするように設定されます。必要となるのは、使用可能なオプションの1つまたは2つのみである場合もあります。たとえば、x86 クライアントのみが配置されているが、Oracle Solaris 11.2 または 11.3 をインストールするオプションを提供する場合は、2つのインストールサービス (各バージョンの OS にそれぞれ1つのサービス) を作成する必要があります。x86 クライアントおよび SPARC クライアントの両方があり、Oracle Solaris 11.3 のインストールのみをサポートする場合は、各アーキテクチャーのインストールサービスを作成します。詳細は、[107 ページの「インストールサービスの作成」](#)を参照してください。

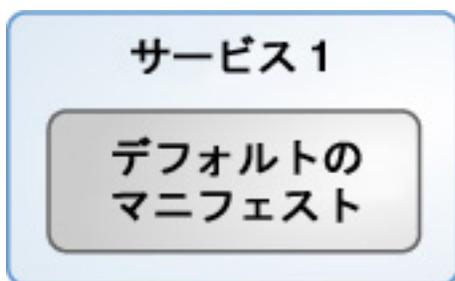
### 自動インストーラの使用事例

このセクションの例は、AI サーバー上でサービスを構成するための主な手順を示しています。この事例では、もっとも簡単なサービス構成から開始して、より高度な構成に発展させています。インストールサービスを作成するためのほとんどの構成手順はオプションであり、それらを使用するかどうかは環境によって異なります。

## 1つのアーキテクチャーと1つのOSをサポートするAIサーバー

この構成は、もっとも簡単に作成できるものの1つです。この構成では、すべての構成データをインストールごとに手動で入力する必要があります。

図 3 1つのアーキテクチャーと1つのOSをサポートするAIサーバー



この環境を作成するには、次の手順に従います。

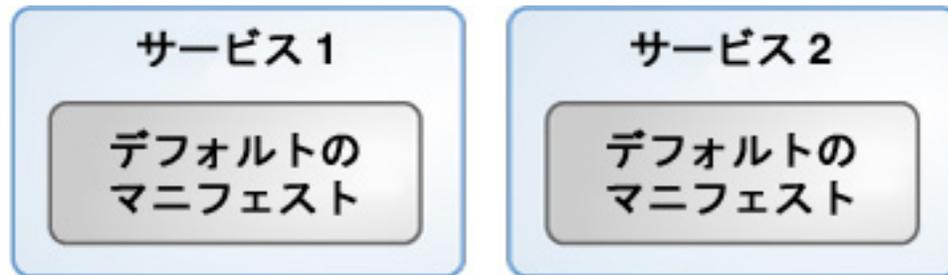
1. AIサーバーを構成します。102 ページの「[AIサーバーを設定する方法](#)」を参照してください。
2. インストールサービスを作成します。107 ページの「[インストールサービスの作成](#)」を参照してください。

これらの手順を完了すると、AIサーバーには、デフォルトのAIマニフェストを含むインストールサービスが含まれています。インストールの完了後にAI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構成されていないため、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。その他の構成手順は手動で行う必要があります。システム構成プロファイルを追加して、システム構成の手順を自動化することができます。

## 2つのアーキテクチャーをサポートするAIサーバー

この例では、AIサーバーがx86およびSPARCクライアントアーキテクチャーをサポートするように構成されます。

図 4 2つのアーキテクチャーをサポートする AI サーバー



この環境を作成するには、次の手順に従います。

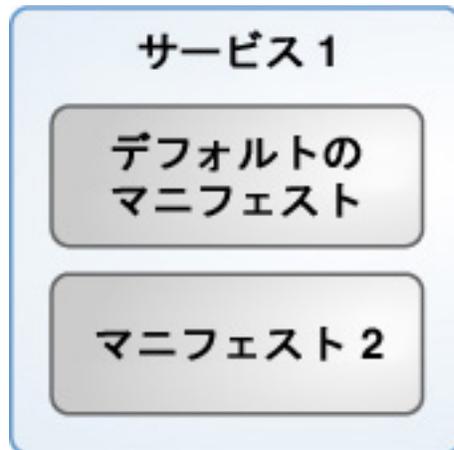
1. AI サーバーを構成します。102 ページの「[AI サーバーを設定する方法](#)」を参照してください。
2. x86 クライアント用のインストールサービスを作成します。107 ページの「[インストールサービスの作成](#)」を参照してください。
3. SPARC クライアント用のインストールサービスを作成します。

これらの手順を完了すると、AI サーバーには2つのインストールサービス(クライアントアーキテクチャーごとに1つ)が含まれています。各サービスには、デフォルトのマニフェストが含まれています。インストールの完了後にAI 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構成されていないため、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。その他の構成手順は手動で行う必要があります。プロファイルを追加して、システム構成の手順を自動化することができます。

## 1つのアーキテクチャーと2つのディスク構成をサポートする AI サーバー

この例では、1つのクライアントアーキテクチャーおよびAI クライアントの2つのディスクレイアウト構成のみをサポートするようにAI サーバーを構成しています。

図 5 1つのアーキテクチャーと2つのディスクレイアウトをサポートする AI サーバー



この環境を作成するには、次の手順に従います。

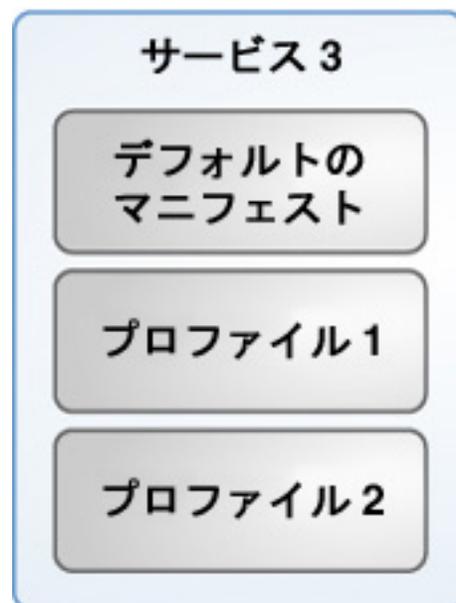
1. AI サーバーを構成します。102 ページの「[AI サーバーを設定する方法](#)」を参照してください。
2. AI クライアント用のインストールサービスを作成します。107 ページの「[インストールサービスの作成](#)」を参照してください。
3. 2 番目のディスクレイアウトの AI マニフェストを作成します。第10章「[AI クライアントのインストールパラメータの定義](#)」を参照してください。
4. 2 番目のマニフェストをインストールサービスに関連付けます。116 ページの「[インストールサービスとクライアント固有のインストール手順の関連付け](#)」を参照してください。

これらの手順を完了すると、AI サーバーには、2つの AI マニフェストを含む1つのインストールサービスが含まれています。デフォルトのマニフェストには1つのディスクレイアウトが含まれています。2 番目のマニフェストには、2 番目のディスクレイアウトが含まれています。2 番目のマニフェストは、そのマニフェストを使用すべき AI クライアントを識別する条件に関連付けられています。ほかのすべての AI 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。インストールの完了後に AI 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構成されていないため、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。その他の構成手順は手動で行う必要があります。プロファイルを追加して、システム構成の手順を自動化することができます。

## 1つのアーキテクチャーと2つのタイムゾーンをサポートする AI サーバー

この例では、AI サーバーが、タイムゾーンを設定する2つのシステム構成プロファイルを含む1つのクライアントアーキテクチャーのみをサポートするように構成されます。

図 6 1つのアーキテクチャーと2つのディスク構成をサポートする AI サーバー



この環境を作成するには、次の手順に従います。

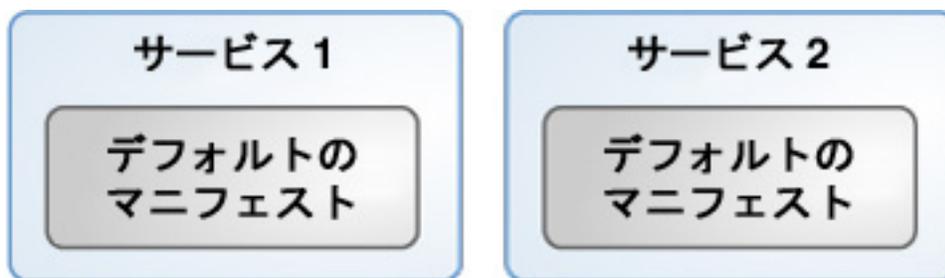
1. AI サーバーを構成します。102 ページの「AI サーバーを設定する方法」を参照してください。
2. AI クライアント用のインストールサービスを作成します。107 ページの「インストールサービスの作成」を参照してください。
3. 1つのタイムゾーンのシステム構成プロファイルを作成します。第11章「AI クライアントのシステム構成パラメータの定義」を参照してください。
4. 2番目のタイムゾーンのシステム構成プロファイルを作成します。
5. 両方のシステム構成プロファイルをインストールサービスに関連付けます。117 ページの「インストールサービスへのクライアント固有の構成手順の関連付け」を参照してください。

これらの手順を完了すると、AI サーバーには、デフォルトのマニフェストと、タイムゾーンの異なる値を設定する 2 つのプロファイルを含む 1 つのインストールサービスが含まれています。各プロファイルは条件に関連付けられ、各プロファイルを使用すべき AI クライアントが識別されます。インストールの完了後に AI クライアントがブートされたら、システム構成プロファイルに含まれていないほかのすべての構成情報を手動で構成する必要があります。

## 1 つのアーキテクチャーと 2 つのリリースをサポートする AI サーバー

この例では、1 つのクライアントアーキテクチャーと 2 つのバージョンの Oracle Solaris OS のみをサポートするように AI サーバーを構成しています。

図 7 1 つのアーキテクチャーと 2 つのリリースをサポートする AI サーバー



この環境を作成するには、次の手順に従います。

1. AI サーバーを構成します。102 ページの「AI サーバーを設定する方法」を参照してください。
2. 最初の OS 用のインストールサービスを作成します。107 ページの「インストールサービスの作成」を参照してください。
3. 2 番目の OS の 2 番目のインストールサービスを作成します。107 ページの「インストールサービスの作成」を参照してください。
4. クライアント定義を作成し、AI クライアントを適切なサービスに関連付けます。114 ページの「インストールサービスへの AI クライアントの関連付け」を参照してください。

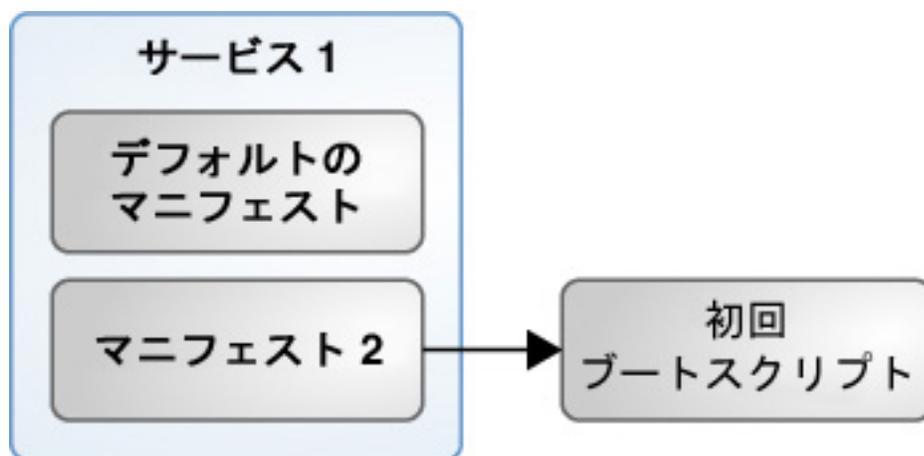
これらの手順を完了すると、AI サーバーには 2 つのインストールサービスが含まれています。クライアントの定義によって、どの AI クライアントがどのサービスを使用するかが決まります。インストールの完了後に AI クライアントがブートすると、システム構成プロファイルが構成されていないため、対話型ツールによってシステム構

成情報の入力を求めるプロンプトが表示されます。その他の構成手順は手動で行う必要があります。プロファイルを追加して、システム構成の手順を自動化することができます。

## 1つのアーキテクチャーと一部の AI クライアント用の追加の構成をサポートする AI サーバー

この例では、AI サーバーが1つのクライアントアーキテクチャーのみをサポートするように構成されますが、1セットの AI クライアントには、マニフェストやプロファイルでは実行できない追加の構成が必要です。

図 8 1つのアーキテクチャーと一部の AI クライアント用の追加の構成をサポートする AI サーバー



この環境を作成するには、次の手順を使用します。

1. AI サーバーを構成します。102 ページの「[AI サーバーを設定する方法](#)」を参照してください。
2. AI クライアント用のインストールサービスを作成します。107 ページの「[インストールサービスの作成](#)」を参照してください。
3. 最初のブート時のスクリプトを作成します。第13章「[初回ブート時のカスタムスクリプトの実行](#)」を参照してください。
4. 初回ブートスクリプトが含まれているパッケージを作成して、そのパッケージをパッケージリポジトリに追加します。241 ページの「[初回ブートスクリプトの IPS パッケージを作成および発行する方法](#)」を参照してください。

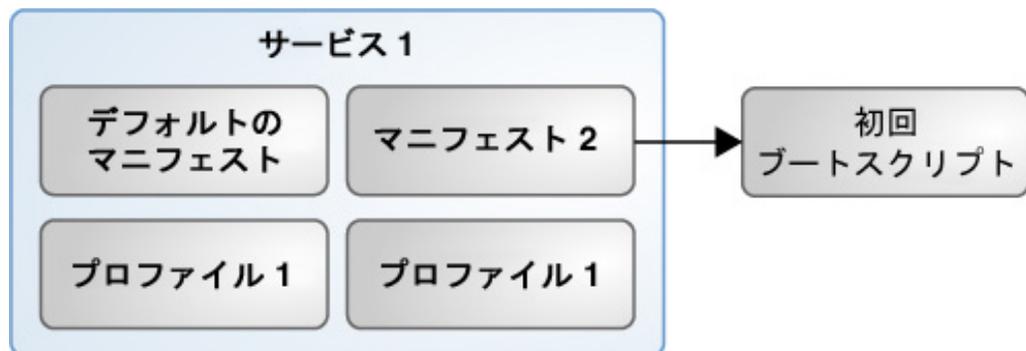
5. 初回ブートパッケージが含まれている 2 番目のマニフェストを作成します。116 ページの「インストールサービスとクライアント固有のインストール手順の関連付け」を参照してください。
6. 2 番目のマニフェストをインストールサービスに関連付けます。116 ページの「インストールサービスとクライアント固有のインストール手順の関連付け」を参照してください。

これらの手順を完了すると、AI サーバーには、2 つのマニフェストを含む 1 つのインストールサービスが含まれています。2 番目のマニフェストには、初回ブートサービスパッケージをインストールするための手順が含まれています。初回ブートサービスパッケージは、AI クライアントが再起動されたときに初回ブートスクリプトを実行します。2 番目のマニフェストには条件が関連付けられ、そのマニフェストを使用すべき AI クライアントが識別されます。ほかのすべての AI 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。インストールの完了後に AI クライアントがブートされると、初回ブートスクリプトが実行されます。次に、システム構成プロファイルが構成されていないため、対話型ツールによってシステム構成情報の入力を求められます。その他の構成手順は手動で行う必要があります。プロファイルを追加して、システム構成の手順を自動化することができます。

## 多くの構成の変更をサポートする AI サーバー

この例では、1 つのアーキテクチャーと 2 つの構成をサポートするように AI サーバーを構成しています。最初の構成は非常に基本的です。2 番目の構成は、別のディスクレイアウトを選択する場合、および追加の構成を実行する場合に使用できます。どちらの構成も、同じシステム構成プロファイルを使用してタイムゾーンを構成します。

図 9 多くの構成の変更をサポートする AI サーバー



この環境を作成するには、次の手順に従います。

1. AI サーバーを構成します。102 ページの「AI サーバーを設定する方法」を参照してください。
2. AI クライアント用のインストールサービスを作成します。107 ページの「インストールサービスの作成」を参照してください。
3. 最初のブート時のスクリプトを作成します。第13章「初回ブート時のカスタムスクリプトの実行」を参照してください。
4. 初回ブートスクリプトが含まれているパッケージを作成して、そのパッケージをパッケージリポジトリに追加します。241 ページの「初回ブートスクリプトのIPS パッケージを作成および発行する方法」を参照してください。
5. 2 番目のマニフェストを作成します。このマニフェストには、初回ブートパッケージが含まれていて、2 番目のディスクレイアウトを定義します。第10章「AI クライアントのインストールパラメータの定義」を参照してください。
6. 2 番目のマニフェストをインストールサービスに関連付けます。116 ページの「インストールサービスとクライアント固有のインストール手順の関連付け」を参照してください。
7. タイムゾーンを設定するシステム構成プロファイルを作成します。第11章「AI クライアントのシステム構成パラメータの定義」を参照してください。
8. システム構成プロファイルをインストールサービスに関連付けます。117 ページの「インストールサービスへのクライアント固有の構成手順の関連付け」を参照してください。

これらの手順を完了すると、AI サーバーには、2つのマニフェストを含む1つのインストールサービスが含まれています。2 番目のマニフェストには、2 番目のディスクレイアウトに関する情報、および初回ブートスクリプトが含まれている初回ブートサービスパッケージをインストールする手順が含まれています。初回ブートスクリプトは、AI クライアントがリブートされると実行されます。2 番目のマニフェストは、そのマニフェストを使用すべき AI クライアントを識別する条件に関連付けられています。ほかのすべての AI 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。どちらのマニフェストを使用する AI クライアントも、プロファイルに関連付けられている条件に従ってプロファイルも使用するよう構成できます。インストールの完了後に AI クライアントがブートされると、初回ブートスクリプトが実行されます。システム構成プロファイルに含まれていない追加のすべての構成情報は、手動で構成する必要があります。



## AI サーバーの設定

---

ネットワーク経由でクライアントを自動インストールするには、AI では AI サーバーとして機能する別個のシステムが必要です。AI サーバー上で、さまざまな AI クライアントに必要な Oracle Solaris リリースをインストールするためのネットイメージと手順を提供するインストールサービスを作成します。この章の内容は次のとおりです。

- 99 ページの「AI サーバー設定タスク」
- 100 ページの「AI サーバーの要件」
- 101 ページの「インストールサービスの管理者権限」
- 101 ページの「AI サーバーの構成」
- 107 ページの「インストールサービスの作成」
- 114 ページの「インストールサービスへの AI クライアントの関連付け」
- 118 ページの「自動インストールのセキュリティーの向上」
- 134 ページの「インストールサービスの情報の表示」
- 140 ページの「インストールサービスの管理」
- 143 ページの「AI マニフェストの管理」
- 145 ページの「システム構成プロファイルの管理」

### AI サーバー設定タスク

AI サーバーを設定する概要レベルのステップは、次のとおりです。

- サーバーを AI サーバーにできる最小要件を満たしているかどうかを確認します。詳細は、100 ページの「AI サーバーの要件」を参照してください。
- AI コマンドを使用可能にするために使用する方法を決定します。完全な説明については、101 ページの「インストールサービスの管理者権限」を参照してください。
- 静的 IP アドレスおよびデフォルトルートを使用し、AI パッケージをインストールし、必要な場合は `svc:/network/dns/multicast` SMF サービスを有効にするよ

うに AI サーバーを構成します。完全な説明については、[101 ページの「AI サーバーの構成」](#)を参照してください。

- 必要に応じて、インストールサービスを有効にするネットワーク、AI Web サーバーのポート番号、すべてのイメージのデフォルトイメージパスなど、AI サーバーに関する追加設定を変更します。[104 ページの「AI サーバーの構成の変更」](#)を参照してください

## AI サーバーの要件

このセクションに記載する要件を満たす、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のすべてのシステムを、AI サーバーとして使用できます。AI サーバーは x86 システム、SPARC システムのいずれかになります。x86 AI 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 AI 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。

### オペレーティングシステム

AI サーバーに Oracle Solaris 2 OS をインストールします。詳細については、[21 ページの「インストールオプションの比較」](#)を参照してください。既存の AI サーバー上でソフトウェアを更新する方法については、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[イメージの更新の概要](#)」を参照してください。AI サーバー上で最新バージョンの OS を使用することにより、古いバージョンのインストールを引き続きサポートしつつ、すべての新しい機能を使用することができます。

---

**注記 - 11.3** クライアントがインストールされている AI サーバー上にインストールする OS として、すべての Oracle Solaris 11 または 12 リリース (更新および SRU を含む) を使用できます。

---

### メモリー

メモリーの最小要件は 1G バイトです。

### ディスク容量

AI 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 AI 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 OS のバージョンごとに、個別のインストールサービスが必要になります。各ネットイメージは約 300-400M バイトです。

## インストールサービスの管理者権限

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

<b>権利プロファイル</b>	<p><code>profiles</code> コマンドを使用して、自分に割り当てられている権利プロファイルを一覧表示します。</p> <p>Software Installation</p> <p>Software Installation 権利プロファイルがある場合、<code>pfexec</code> コマンドを使用して、パッケージをインストールおよび更新できます。</p> <pre>\$ pfexec pkg install install/installadm</pre> <p>Install Service Management</p> <p>Install Service Management 権利プロファイルがある場合、<code>pfexec</code> コマンドを使用して、インストールサービスの作成や、インストールサービスへのシステム構成プロファイルの追加などができます。</p> <pre>\$ pfexec installadm create-service</pre> <p>Service Management</p> <p>Service Management 権利プロファイルがある場合、SMF サービスを構成して有効化できます。Service Management 権利プロファイルには <code>pfexec</code> は必要ありません。</p> <pre>\$ svcadm refresh system/install/server:default</pre>
<b>sudo</b>	<p>サイトのセキュリティーポリシーに応じて、自分のユーザーパスワードで <code>sudo</code> コマンドを使用し、特権コマンドを実行できる場合があります。</p> <pre>\$ sudo pkg install install/installadm</pre>
<b>役割</b>	<p><code>roles</code> コマンドを使用して、自分に割り当てられている役割を一覧表示します。<code>root</code> 役割を持つ場合、<code>root</code> パスワードで <code>su</code> コマンドを使用して、<code>root</code> 役割になることができます。</p>

## AI サーバーの構成

このセクションでは、AI クライアントインストールの準備のために AI 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。

## ▼ AI サーバーを設定する方法

### 1. Oracle Solaris 11.3 をインストールします。

AI サーバーには Oracle Solaris 11 の任意のバージョンをインストールできますが、Oracle Solaris 11.3 リリースには、そのバージョンが AI サーバーにインストールされている場合にのみ使用できる複数のサーバー側の拡張が含まれます。Oracle Solaris AI サーバーがある場合、AI サーバーの構成手順についてのそのリリースのドキュメントを参照してください。どのインストール方法を使用するかは提案については、21 ページの「インストールオプションの比較」を参照してください。

### 2. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

### 3. ネットワークインタフェースの IP アドレスが静的であることを確認します。

この例では net0 は静的 IP アドレスを使用して構成されています。

```
# ipadm
NAME                CLASS/TYPE STATE    UNDER    ADDR
net0                 ip         ok      --        --
  net0/v4            static    ok      --        129.144.83.5/24
lo0                  loopback  ok      --        --
  lo0/v4             static    ok      --        127.0.0.1/24
  lo0/v6             static    ok      --        ::1/128
```

アドレスが静的でない場合、次の手順を実行します。

#### a. IP インタフェースを作成します。

```
# ipadm create-ip net0
```

#### b. インタフェース上で静的 IP を構成します。

```
# ipadm create-addr -T static -a local=129.144.83.5/24 net0
```

#### c. 構成を確認します。

```
# ipadm
NAME                CLASS/TYPE STATE    UNDER    ADDR
net0                 loopback  ok      --        --
  net0/v4            static    ok      --        129.144.83.5/24
```

### 4. (オプション) AI サーバーのデフォルトルートを確立します。

```
# route -p add default 192.144.83.1
```

### 5. AI パッケージをインストールします。

#### a. AI パッケージがまだインストールされていないことを確認します。

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

**b. IPS パッケージリポジトリに AI 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             0.5.11-0.175.1.0.0.24.0  ---
```

**c. AI 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
Create backup boot environment: No
Services to change: 2

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

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

**6. マルチキャスト DNS (mDNS) を有効にします。**

mDNS サービスによって、AI クライアントは同じサブネットの別の AI サーバーにある複製されたインストールサービスを検出できます。mDNS の詳細は、『[Oracle Solaris 11.3 ディレクトリサービスとネームサービスでの作業: DNS と NIS](#)』の「[マルチキャスト DNS およびサービス検出](#)」を参照してください。

**a. 必要に応じて、mDNS パッケージをインストールします。**

```
# pkg install pkg:/service/network/dns/mdns
```

**b. ネームサービススイッチ情報を更新します。**

ローカルホストを解決できるようにするため、name-service/switch サービスの config/host プロパティを変更して、ソースとして mdns を含めます。次に例を示します。

```
# /usr/sbin/svccfg -s svc:/system/name-service/switch
svc:/system/name-service/switch> setprop config/host = astring: "files dns mdns"
svc:/system/name-service/switch> select system/name-service/switch:default
svc:/system/name-service/switch:default> refresh
svc:/system/name-service/switch> quit
```

**c. mDNS サービスを有効にします。**

```
# svcadm enable svc:/network/dns/multicast:default
```

このようにして mDNS を有効にすると、加えた変更が、アップグレードやリブートを行なったあとも維持されるようになります。詳細は、[svcadm\(1M\)](#) のマニュアルページを参照してください。

## AI サーバーの構成の変更

`installadm set-server` コマンドを使用して、AI サーバー上の複数の構成設定を変更できます。これらの構成設定には次のものがあります。

- AI サポートから除外するネットワーク
- AI サポートを含めるネットワーク
- AI Web サーバーのポート

このコマンドを使用するには、管理者になる必要があります。詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

### 例 9 ネットワーク上の AI サポートの無効化

デフォルトでは、AI サーバーはサーバーが接続されているすべてのネットワーク上の AI クライアントにサービスを提供するように構成されます。この例では、`192.168.56.0/24` ネットワークインタフェースは AI 要求に対してサービスを提供しなくなります。

```
# installadm set-server -L 192.168.56.0/24
```

### 例 10 AI サーバーでサポートされるネットワークの組み込み

状況によっては、除外するネットワークの一覧ではなくインストールサービスをサポートするネットワークの一覧を表示の方が簡単です。次のコマンドは、2つのネットワークでインストールサービスを許可する方法を示しています。

```
# installadm set-server -l 205.10.11.0/24, 205.10.12.0/24
```

### 例 11 AI Web サーバーのポート番号の構成

AI サーバーは Web サーバーを使用してインストールサービスをホストします。デフォルトでは、AI 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ファイルは `http://localhost:5555` で表示できます。Web サーバーに使用されるポート番号を変更することもできます。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。

```
# installadm set-server -p 7000
```

**例 12** セキュア AI Web サーバーのポート番号の構成

セキュア 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、セキュア AI 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ファイルは `http://localhost:5556` で安全に表示できます。Web サーバーに使用されるポート番号を変更することもできます。次のコマンドでは、ポート 7001 からセキュア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。

```
# installadm set-server -P 7001
```

**例 13** デフォルトのイメージパスの構成

デフォルトでは、イメージは `/export/auto_install` 内の `service-name` ディレクトリ内に作成されます。したがって、`service-name` サービスのネットイメージは、デフォルトでは `/export/auto_install/service-name` に作成されます。次のコマンドでは、デフォルトで `/export/aimages/service-name` に新しいインストールサービスを作成するように AI サーバーを構成しています。

```
# installadm set-server -d /export/aimages
```

**例 14** AI サーバーに対する AI クライアントの IP アドレスと数の定義

次の例では、ネットワークの DHCP サーバーとして機能する AI サーバーを設定します。DHCP サーバーは、`10.80.239.150` (-i) から始まる 20 個の IP アドレス (-c) を設定します。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。

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

```
# installadm set-server -i 10.80.239.150 -c 20
```

**例 15** AI サーバー上のローカル DHCP サービスの自動更新の無効化

デフォルトでは、ローカル ISC DHCP 構成は、AI サーバー内の AI クライアント構成およびサービス構成が変更されると自動的に更新されます。ローカル ISC DHCP 構成を自動的に保守しないようにする場合は、次のコマンドを使用します。

```
# installadm set-server -M
Changed Server
Disabling SMF service svc:/network/dhcp/server:ip4
Refreshing SMF service svc:/system/install/server:default
```

**例 16** AI サーバーでの DHCP 更新の有効化

ISC DHCP 構成の自動更新が無効化されている場合、次のコマンドを使用してこれを有効化することができます。

```
# installadm set-server -m
Warning: AI server will now manage DHCP
Changed Server
Enabling SMF service svc:/network/dhcp/server:ipv4
```

## Web サーバーのユーザーファイルディレクトリの構成

AI Web サーバーは、`installadm` コマンドを使用して追加されたネットイメージ、AI マニフェスト、およびシステム構成プロファイルを提供します。AI Web サーバーは、AI ユーザーまたは管理者によって提供されるファイルも提供できます。

保護が不要なユーザーファイルは、`svc:/system/install/server:default SMF` サービスの `all_services/webserver_files_dir` プロパティによって指定されるディレクトリに格納できます。このプロパティにはデフォルト値がありません。このプロパティの値を指定する場合、この値はローカルシステム上のディレクトリを指定する必要があります。このディレクトリは AI Web サーバーから次の URL を指定して表示でき、`server` は AI サーバーのホスト名または IP アドレス、`port` は例11「[AI Web サーバーのポート番号の構成](#)」で説明した AI Web サーバーのポート番号です。

```
http://server:port/files
```

保護が不要なユーザーファイルは、`all_services/webserver_secure_files_dir` プロパティによって指定されるディレクトリに格納できます。このプロパティにはデフォルト値がありません。このプロパティの値を指定する場合、この値はローカルシステム上のディレクトリを指定する必要があります。このディレクトリは AI Web サーバーから次の URL を指定して表示でき、`server` は AI サーバーのホスト名または IP アドレス、`secure-port` は上記の例12「[セキュア AI Web サーバーのポート番号の構成](#)」で説明したセキュア AI Web サーバーのポート番号です。

```
https://server:secure-port/secure_files
```

証明書および鍵が必要な IPS パッケージリポジトリを AI マニフェストが指定した場合、これらのパブリッシャー証明書を保管し、この URI を AI マニフェストで指定することができます。割り当てられたセキュリティ資格を持つ AI クライアントのみがこのディレクトリにアクセスできます。

---

**ヒント** - 最高のセキュリティを確保するために、`webserver_secure_files_dir` ディレクトリ内のファイルをユーザー `webservd` およびグループ `webservd` が所有するようにし、どこからでもアクセス可能にしないでください。

---

## インストールサービスの操作

AI サーバーを設定したあと、次のタスクのいくつかを実行する場合があります。[installadm\(1M\)](#) のマニュアルページも参照してください。

- [107 ページの「インストールサービスの作成」](#)
- [114 ページの「インストールサービスへの AI クライアントの関連付け」](#)
- [116 ページの「インストールサービスとクライアント固有のインストール手順の関連付け」](#)
- [117 ページの「インストールサービスへのクライアント固有の構成手順の関連付け」](#)
- [121 ページの「自動インストールのセキュリティーを構成する方法」](#)
- [130 ページの「AI を使用して Kerberos クライアントを構成する方法」](#)
- [134 ページの「インストールサービスの情報の表示」](#)

## インストールサービスの作成

AI 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 OS ごとに、個別のインストールサービスを作成します。

### ▼ インストールサービスを作成する方法

クライアントのアーキテクチャー (SPARC または x86) ごとに、およびインストール可能にする必要のある Oracle Solaris オペレーティングシステムごとに、インストールサービスを作成します。

1. **管理者になります。**  
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. **DHCP サービスを確認します。**  
ローカルまたはリモートの DHCP サポートが適切に設定されていることを確認します。
3. **インストールサービスを作成します。**  
使用できるすべてのオプションについては、[installadm\(1M\)](#) のマニュアルページを参照してください。次に例を示します。

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

**source**                    ネットイメージのデータソースを指定します。この値は IPS AI ネットイメージパッケージの FMRI 識別子とすることができ、デフォルトは `install-image/solaris-auto-install` です。値は AI ISO イメージファイルのフルパス名とすることもできます。

**-y**                         自動生成されたイメージパスの使用を確認するプロンプトを抑制します

**例 17** AI サーバーの DHCP が有効な状態で ISO ファイルを使用した SPARC インストールサービスの作成

この例では、SPARC 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 つのサブネット構成されており、AI サーバーはネットワークの DHCP サーバーとしても動作します。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。

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

このコマンドが x86 ISO ファイルを使用して実行された場合の出力の違いは、ISO ファイルの名前、作成したサービスの名前とアーキテクチャタイプ、および作成されるブートファイルの説明のみです。x86 の出力については、[例18 「IPS パッケージを使用した X86 インストールサービスの作成」](#)を参照してください。

```
# installadm create-service -s /var/tmp/images/sparc/sol-11_3-ai-sparc.iso -y
0% : Service svc:/network/dns/multicast:default is not online. Installation services
will not be advertised via multicast DNS.
0% : Creating service from: /var/tmp/images/sparc/sol-11_3-ai-sparc.iso
36% : Transferring contents
36% : Creating sparc service: solaris11_3-sparc
36% : Image path: /export/auto_install/solaris11_3-sparc
36% : Setting "solaris" publisher URL in default manifest to:
36% : http://pkg.oracle.com/solaris/release/
36% : Creating default-sparc alias
36% : Setting "solaris" publisher URL in default manifest to:
36% : http://pkg.oracle.com/solaris/release/
36% : Setting the default SPARC bootfile(s) in the local DHCP configuration to:
36% : http://10.80.239.2:5555/cgi-bin/wanboot-cgi
100% : Created Service: 'solaris11_3-sparc'
100% : Refreshing SMF service svc:/system/install/server:default
100% : Restarting SMF service svc:/network/dhcp/server:ipv4
# installadm list
```

Service Name	Status	Arch	Type	Secure	Alias	Aliases	Clients	Profiles	Manifests
default-sparc	on	sparc	iso	no	yes	0	0	0	1

```
solaris11_3-sparc      on      sparc iso no      no      1      0      0      1
```

**例 18** IPS パッケージを使用した X86 インストールサービスの作成

この例では、ローカル DHCP サービスを使用せずに x86 AI サーバーを使用します。IPS パッケージからのネットイメージを使用して、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。このコマンドは、オプションが指定されないときのデフォルトの動作も示しています。IPS パッケージを使用するときにはほかの情報が指定されない場合、AI クライアントのアーキテクチャーは AI サーバーのアーキテクチャーと同じであると仮定されます。この AI サーバーが SPARC システムである場合は、`-a i386` オプションを指定して、x86 インストールサービスを作成することを指定する必要があります。

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

このコマンドが SPARC を作成するために実行された場合の出力の違いは、サービスの名前とアーキテクチャータイプ、および作成されるブートファイルの説明のみです。SPARC の出力については、[例17「AI サーバーの DHCP が有効な状態で ISO ファイルを使用した SPARC インストールサービスの作成」](#)を参照してください。

```
# installadm create-service -y
0% : Creating service from: pkg:/install-image/solaris-auto-install
0% : Using publisher(s):
0% :   solaris: http://pkg.oracle.com/solaris/release/
5% : Refreshing Publisher(s)
7% : Startup Phase
15% : Planning Phase
61% : Download Phase
90% : Actions Phase
91% : Finalize Phase
91% : Creating i386 service: solaris11_3-i386
91% : Image path: /export/auto_install/solaris11_3-i386
91% : Setting "solaris" publisher URL in default manifest to:
91% : http://pkg.oracle.com/solaris/release/
91% : DHCP is not being managed by install server.
91% : Creating default-i386 alias
91% : Setting "solaris" publisher URL in default manifest to:
91% : http://pkg.oracle.com/solaris/release/
91% : DHCP is not being managed by install server.
91% : No local DHCP configuration found. This service is the default
91% : alias for all PXE clients. If not already in place, the following should
91% : be added to the DHCP configuration:
91% : Boot server IP: 10.80.239.2
91% : Boot file(s):
91% :   bios clients (arch 00:00): default-i386/boot/grub/pxegrub2
91% :   uefi clients (arch 00:07): default-i386/boot/grub/grub2netx64.efi
91% :
100% : Created Service: 'solaris11_3-i386'
100% : Refreshing SMF service svc:/system/install/server:default

# installadm list
Service Name      Status Arch  Type Secure Alias Aliases Clients Profiles Manifests
-----
default-i386      on      i386  pkg  no    yes   0      0      0      1
solaris11_3-i386  on      i386  pkg  no    no    1      0      0      1
```

**例 19** 異なるアーキテクチャーに対するインストールサービスの作成

デフォルトでは、インストールサービスが作成されると、アーキテクチャーは AI サーバーと同じになります。別のアーキテクチャーのサービスを作成する場合、`-a` オプションを使用します。次の例では SPARC AI サーバー上に x86 サービスが作成されます。

```
# installadm create-service -n solaris11_3-i386 -a i386 -y
```

**例 20** x86 クライアントを自動的にインストールするサービスの作成

x86 クライアント上の GRUB メニューのデフォルトエントリでは、AI が自動的に開始されません。インストールが自動的に開始されるように GRUB メニューをカスタマイズするために、次のコマンドを使用できます。

```
# installadm create-service -s /var/tmp/images/i386/sol-11_3-ai-x86.iso -y -b install=true
```

## インストールサービスが作成されるとき動作

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

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

`default-arch` サービスが別名になっているサービスを変更するには、`set-service` サブコマンドを使用して `aliasof` プロパティを設定します。どちらかのサービスに追加されたマニフェストとプロファイルは、別名のリセット後も変わりません。サービスが使用するネットイメージが変わるだけです。`aliasof` プロパティの設定の詳細は、[140 ページの「インストールサービスの管理」](#)を参照してください。`default-arch` サービスが別名であるサービスのネットイメージを更新するには、[142 ページの「既存のインストールサービスの更新」](#)に示すように、`update-service` サブコマンドを使用します。

`default-arch` 別名が新しいインストールサービスに変更されたときに、ローカル ISC DHCP 構成が見つかった場合、`all_services/manage_dhcp` プロパティ

の値が true であれば、このデフォルトの別名のブートファイルがそのアーキテクチャーのデフォルトの DHCP サーバー全体のブートファイルとして設定されます。all\_services/manage\_dhcp プロパティの詳細については、[例15「AI サーバー上のローカル DHCP サービスの自動更新の無効化」](#)を参照してください。

installadm create-service コマンドでは、ポート 5555 で動作する Web サーバーでネットイメージも提供されます。たとえば、Web サーバーのアドレスは http://10.80.238.5:5555/solaris11\_3-i386 のようになります。別のポートを使用するには、[例11「AI Web サーバーのポート番号の構成」](#)を参照してください。

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名を定義しない場合、名前が生成されます。インストールサービスを作成するときにコマンド行に -n オプションを含めることで、サービス名を指定できます。
2. ネットイメージのソースオプションが指定されていない場合、最新バージョンの install-image/solaris-auto-install パッケージが、AI サーバーのパブリッシャーリスト内でこのパッケージを提供する最初のパブリッシャーから取得されます。
3. デフォルトのインストールサービスのネットイメージディレクトリが作成されます。ディレクトリ名には /export/auto\_install/solaris11\_3-sparc や /export/auto\_install/solaris11\_3-i386 などのサービス名が含まれます。確認プロンプトを抑制するには、-y オプションを指定します。
4. ネットイメージのソースによって、次の 2 つのいずれかの動作が発生します。
  - a. ネットイメージのソースオプションが指定されない場合、install-image/solaris-auto-install パッケージがネットイメージディレクトリ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 サーバーのアーキテクチャーに適合します。AI サーバーが x86 システムで、このサーバー上に SPARC インストールサービスを作成するには、-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、[例19「異なるアーキテクチャーに対するインストールサービスの作成」](#)を参照してください。
  - b. ネットイメージのソースオプションが指定された場合、イメージファイルはネットイメージディレクトリにアンパックまたはインストールされます。
5. インストールサービスアーキテクチャーに従ってファイルが作成されます。
  - SPARC クライアントの場合: このサービス用の wanboot.conf ファイルが /etc/netboot/wanboot.conf に生成されます
  - x86 クライアントの場合: GRUB メニューは /etc/netboot/solaris11\_3-i386/grub.cfg にマウントされます。
6. /export/auto\_install/service-name を /etc/netboot/service-name としてマウントするように、AI SMF サービス system/install/server がリフレッシュされず。

7. これがこの AI 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場合、`default-sparc` サービス別名が自動的に作成されます。また、`/export/auto_install/service-name` が `/etc/netboot/default-sparc` としてマウントされます。  
最初の x86 インストールサービスに対して、`default-i386` サービス別名が作成され、`/etc/netboot/default-i386` マウントポイントが作成されます。
8. SPARC クライアントの場合、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。また、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。
9. 必要に応じて DHCP サービスが作成され、IP アドレスがプロビジョニングされます。このサーバーですでに DHCP サービスが設定されている場合は、`-i` および `-c` オプションによって、DHCP サーバーがこのサービスの新しい IP アドレスで更新されます。`svc:/network/dhcp/server` サービスが `online` になります。
10. ローカル DHCP サービスを使用しない構成の場合は、次のようになります。
  - SPARC クライアントの場合: 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。
  - x86 クライアントの場合: DHCP 構成に必要なブートサーバー IP アドレスが提供されます。DHCP 構成に必要なブートファイル `default-i386/boot/grub/pxegrub2` および `default-i386/boot/grub/grub2netx64.efi` も提供されます。
11. 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` または `default-i386` の別名のブートファイルがすべての一致する AI クライアントのデフォルトのブートファイルとして設定されます。この割り当ては、`-i` および `-c` オプションが使用されているかどうかに関係なく行われます。

## AI クライアントをサポートする DHCP 構成ファイルの例

このセクションでは、`installadm` によって ISC DHCP 構成用の DHCP 構成ファイルにどのように情報が追加されるかを示します。ISC DHCP の構成についての詳細は、『[Oracle Solaris 11.3 での DHCP の作業](#)』の第 2 章、「ISC DHCP サービスの管理」を参照してください。

### Oracle Solaris 11.3 i386 インストールサービスのための ISC DHCP 構成

次の例は、`installadm` によって、`-i` および `-c` オプションを使用して指定した IP アドレスが、以前作成した Oracle Solaris 11.3 i386 インストールサービスの ISC DHCP

構成用の `/etc/inet/dhcd4.conf` ファイルにどのように追加されるかを示しています。

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

次の例は、`installadm` によって、デフォルトの PXE ブートファイルが、以前作成した `default-i386 Oracle Solaris 11.3 i386` インストーラサービスの ISC DHCP 構成用の `/etc/inet/dhcd4.conf` ファイルにどのように設定されるかを示しています。

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

## Oracle Solaris 11 i386 インストーラサービスのための ISC DHCP 構成

Oracle Solaris 11.3 サービスの代わりに Oracle Solaris 11 i386 インストーラサービスを作成した場合は、次の例のような出力が表示されます。

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

次の例は、`installadm` によって、デフォルトの PXE ブートファイルが Oracle Solaris 11 i386 インストーラサービスの ISC DHCP 構成用の `/etc/inet/dhcd4.conf` ファイルにどのように設定されるかを示しています。

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

## Oracle Solaris 11.3 sparc インストーラサービスのための ISC DHCP 構成

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

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

次の例は、`installadm`によって、デフォルトのブートファイルが Oracle Solaris 11.3 `sparc` インストールサービスの ISC DHCP 構成用の `/etc/inet/dhcd4.conf` ファイルにどのように設定されるかを示しています。

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

## インストールサービスへの AI クライアントの関連付け

`installadm create-client` コマンドは、AI 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。また x86 クライアント用のカスタム設定を提供できます。詳しい例と出力例については、[249 ページの「AI クライアントの設定」](#)を参照してください。

`installadm delete-client` コマンドは、インストールサービスに対する AI クライアントの関連付けを削除します。

## インストールサービスへの AI クライアントの関連付け

AI 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 アドレスを指定した場合、その AI クライアントは、最後に指定されたインストールサービスにのみ関連付けられます。このコマンドを実行するには、管理者でなければなりません。

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

### 例 21 インストールサービスへの SPARC クライアントの関連付け

次のコマンドは、MAC アドレスが `00:14:4f:a7:65:70` の AI クライアントを `solaris11_3-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 solaris11_3-sparc
```

**例 22** x86: インストールサービスに対する x86 クライアントの関連付けおよびシリアルラインへの出力のリダイレクト

次の例では x86 クライアントを追加し、クライアント固有の `/etc/netboot/grub.cfg` ファイル内のブートプロパティを変更します。この例では、インストール出力はシリアルコンソールデバイスにリダイレクトされます。

```
# installadm create-client -e c0ffeec0ffee -n solaris11_3-i386 -b console=ttya
```

**例 23** x86: x86 クライアントのブートプロパティの変更

x86 クライアントシステムの場合は、クライアントをブートするときに使用するカスタム GRUB2 メニューを指定する場合に `-G` オプションを使用できます。この例では、`/etc/netboot/grub.custom.cfg` という名前のカスタム GRUB2 メニューを指定します。

```
# installadm create-client -e c0ffeec0ffee -n solaris11_3-i386 -G /etc/netboot/grub.custom.cfg
```

`-b` オプションと `-G` オプションを同時に指定することはできません。

## x86 クライアントに対する自動 AI インストール

AI インストールを自動的に開始する場合は、`create-client` サブコマンドを使用して AI クライアントを作成するときに `-b install=true` オプションを使用します。サービスのすべての AI クライアントにこの設定を適用するために、`create-service` サブコマンドを使用してサービスを作成するときに、このオプションを使用できます。

## インストールサービスからの AI クライアントの削除

`macaddr` クライアントとそのインストールサービスの関連付けを解除するには、`installadm delete-client` コマンドを使用します。

```
installadm delete-client -e mac-addr
```

次のコマンドは、MAC アドレスが `00:14:4f:a7:65:70` の AI クライアントを削除します。AI 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 -e 00:14:4f:a7:65:70
```

## インストール手順のカスタマイズ

AI マニフェストまたは派生マニフェストスクリプトを使用すれば、インストールサービス用の特定のインストール手順を提供できます。システム構成プロファイルは構成手順を提供します。複数のシステム構成プロファイルをサービスまたは AI クライアントに関連付けることができます。また、プロファイルは多くのサービス間で共有できます。

### インストールサービスとクライアント固有のインストール手順の関連付け

カスタム AI マニフェストと特定のインストールサービスを関連付けるには、`installadm create-manifest` コマンドを使用します。インストールサービスに派生マニフェストスクリプトを追加することもできます。各インストールサービスに、複数の AI マニフェストまたは派生マニフェストスクリプトを関連付けることができます。特定のサービスのデフォルトマニフェストとして構成されていないすべてのマニフェストまたはスクリプトにはクライアント条件が定義されるため、各 AI クライアントに対して正しい手順が使用されます。

次のすべての例を使用する前に、まず AI マニフェストを作成する必要があります。AI マニフェストの作成手順については、[第10章「AI クライアントのインストールパラメータの定義」](#)を参照してください。

コマンドの構文は次のようになります。

```
installadm create-manifest -n service -f filename
```

**service**                    マニフェストまたは派生マニフェストスクリプトに関連付けられるサービスを指定します。

**filename**                 サービスに関連付けるマニフェストまたは派生マニフェストスクリプトのパスを識別します。

#### 例 24                    マニフェストとクライアント条件の関連付け

次の例は、`manifest-sparc-ent.xml` マニフェストを `solaris11_3-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 M5000 または M4000 サーバーとして識別されるすべての AI クライアントに、`manifest-sparc-ent.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。`-m` オプションは、マニフェストの AI インスタンス名を `sparc-ent` に設定します。

```
# installadm create-manifest -n solaris11_3-sparc -f ./manifest-sparc-ent.xml \  
-m sparc-ent -c platform="SUNW,SPARC-Enterprise"
```

**例 25** スクリプトとクライアント条件の関連付け

次の例は、manifest-sparc-ent.xml マニフェストを solaris11\_3-sparc インストールサービスに追加します。クライアント条件は criteria-sparc-ent.xml ファイルで定義されます。

```
# installadm create-manifest -n solaris11_3-sparc -f ./manifest-sparc-ent.xml \
-m sparc-ent -C ./criteria-sparc-ent.xml
```

criteria-sparc-ent.xml ファイルの内容は、次のとおりです。

```
<ai_criteria_manifest>
  <ai_criteria name="platform">
    <value>SUNW,SPARC-Enterprise</value>
  </ai_criteria>
</ai_criteria_manifest>
```

**例 26** インストールサービスのデフォルトマニフェストの作成

この例では、-d オプションを使用して、指定されたマニフェストまたはスクリプトをこのサービスの新しいデフォルトに指定します。マニフェストに関連付けられたすべてのクライアント条件は保管されますが、このマニフェストまたはスクリプトがデフォルトである間は無視されます。

```
# installadm create-manifest -n solaris11_3-sparc -f ./manifest-sparc-ent.xml -d
```

## インストールサービスへのクライアント固有の構成手順の関連付け

1 つの AI クライアントで複数のプロファイルを使用できるため、1 回の create-profile コマンドで複数のシステム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または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可能です。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 AI 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。

システム構成プロファイルを最初に作成しておく必要があります。手順については、[第11章「AI クライアントのシステム構成パラメータの定義」](#)を参照してください。

コマンドの構文は次のようになります。

```
# installadm create-profile -n service -f filename
```

*service*                    プロファイルが関連付けられるサービスを指定します。

*filename*                   サービスに関連付けるプロファイルのパス名を識別します。

**例 27** システム構成プロファイルとクライアント条件との関連付け

次のコマンドは、`profile-sparc-ent.xml` プロファイルを `solaris11_3-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る M4000 または M5000 サーバーとして識別されるすべての AI クライアントに、`profile-sparc-ent.xml` のシステム構成情報が割り当てられるように指定します。`-p` オプションは、プロファイルの名前を `sparc-ent` に設定します。

```
# installadm create-profile -n solaris11_3-sparc -f ./profile-sparc-ent.xml \  
-p sparc-ent -c platform="SUNW,SPARC-Enterprise"
```

## 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します。

**例 28** 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 `installadm` コマンドを実行するときにも有効になります。AI SMF サービスを手動で有効化するには、次のようにします。

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

**例 29** AI SMF サービスの無効化

AI SMF サービスを無効にするには、次のようにします。

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

## 自動インストールのセキュリティーの向上

Transport Layer Security (TLS) プロトコルを使用して、自動インストールをセキュリティー保護することができます。TLS で認証するためには、AI サーバーと各 AI クライアントに非公開の証明書と鍵のペアを割り当てる必要があります。さらに、証明書を生成して署名するために使用される認証局 (CA) 証明書を提供する必要があります。SPARC クライアントのセキュリティーを有効にするには、各クライアントの OBP HMAC 鍵および暗号化鍵を生成する必要があります。これらの鍵も初期のネットワークブートファイルのダウンロードを保護します。

x86 クライアントについてもセキュリティーを有効にできますが、x86 クライアントは PXEBoot を使用するため、初期ネットワークブートフェーズはセキュリティー保護されないことに注意してください。x86 クライアントのセキュリティーを有効にするには、CA 証明書とクライアント証明書と鍵ファイルを組み込んだカスタム AI イメージから x86 インストールサービスを作成する必要があります。セキュリティー証明書を含むカスタム AI メディアを構築する方法については、『Oracle Solaris 11.3 カスタムインストールイメージの作成』の第 3 章、「イメージの構築」を参照してください。このイメージからインストールサービスを作成したあとで、AI イメージの作成中に使用されたものと同じセキュリティー証明書を使用して、セキュリティーをインストールサービスに設定する必要があります。

次の方法で自動インストールをセキュリティー保護できます。

- サーバー認証: AI サーバーの識別情報を確認できます。
- クライアント認証: AI クライアントの識別情報を確認できます。
- 自動インストールへのアクセスの制御。
- サーバーデータへのアクセスの制御。
- すべての AI クライアントまたは指定された個々の AI クライアントに対するクライアントデータの保護。
- ネットワーク経由で読み取られないようにするためのデータ暗号化。
- セキュアな IPS パッケージリポジトリへのアクセス。
- Web サーバーによるユーザー指定ディレクトリの安全な公開。このディレクトリにアクセスするには、クライアント認証が必要です。

AI プロセスのセキュリティー保護に加えて、AI を使用して AI クライアントに Kerberos をプロビジョニングすることによってネットワーク内のセキュリティーを向上できます。手順については、130 ページの「AI を使用して Kerberos クライアントを構成する方法」を参照してください。

## セキュリティー資格の構成

AI サーバー、指定された AI クライアント、指定されたインストールサービスの AI クライアント、および資格をまだ持たない AI クライアントに対してセキュリティー資格を構成するには、`installadm` コマンドを使用します。TLS 用の Web サーバーでサーバー資格が必要となるため、クライアント認証の前に AI サーバー認証を構成します。

`installadm` コマンドを使用して、次のタスクを実行できます。

- **資格の自動生成。** ユーザーが提供する資格証明を使用していない場合、`-g` オプションを使用して、非公開の X.509 証明書と鍵のペアである X.509 CA 証明書と OBP 鍵を自動的に生成することができます。詳細は、123 ページの「AI サーバー資格の構成」を参照してください

- **ユーザー指定の資格の入力。** ユーザーが提供する資格を使用している場合、`-c`、`-k`、および `-a` オプションを使用して、ユーザーが提供するこれらの資格を指定します。

CA 証明書のみ指定して (`-a` オプション)、非公開の証明書と鍵を別途指定する (`-c` および `-k` オプション) ことも、3 つすべてのオプションを 1 つのコマンドで指定することもできます。`-c` および `-k` オプションを指定する場合、関連する CA 証明書 (`-a` オプション) は事前に指定されている必要があります。`-c` および `-k` オプションはペアとして指定する必要があります、このうちの 1 つだけを指定することはできません。

`-c` オプションの引数は、PEM エンコードされた X.509 資格ファイルです。

`-k` オプションの引数は、PEM エンコードされた X.509 非公開鍵ファイルです。この鍵ファイルからすべてのパスフレーズを削除する必要があります。

`-a` オプションの引数は、PEM エンコードされた認証局 (CA) 資格ファイルです。CA 証明書には一意のサブジェクト行が含まれている必要があります。各 CA 信頼チェーンは 1 回だけ指定する必要があります。CA チェーンに複数の CA 証明書ファイルが含まれている場合は、1 回の `installadm` コマンドに別々の `-a` オプションを使用します。

OBP 鍵は、それらがまだ存在しない場合に生成されます。OBP 鍵が生成される場合、これらの鍵を設定する OBP コマンドが表示されます。
- **OBP 鍵の生成。** `-g`、`-c`、`-k`、または `-a` オプションを使用すると、OBP 鍵がまだ存在していない場合に OBP 鍵が自動的に生成されます。`-E` および `-H` オプションの使用については、[127 ページの「SPARC クライアントの OBP セキュリティキー」](#) を参照してください。
- **資格の表示。** `installadm list` コマンドを使用して、AI サーバー、インストールサービス、または特定の AI クライアントについての現在の資格証明のセットをいつでも表示できます。

## セキュリティの優先順位

有効となるセキュリティ設定や使用される資格証明を決定するとき、優先順位は次のようになります。

1. AI サーバーは資格証明を持ちません。これがデフォルトの状態です。セキュリティの追加はなく、すべての AI クライアント用のすべてのファームウェア鍵をクリアする必要があります。
2. セキュリティがサーバー全体で無効です。セキュリティの追加はなく、すべての AI クライアント用のすべてのファームウェア鍵をクリアする必要があります。
3. AI クライアントサービスポリシーが `disable` に設定されています。セキュリティの追加はなく、すべての AI クライアント用のすべてのファームウェア鍵をクリアする必要があります。

4. `set-client` サブコマンドで作成されたカスタムのクライアント資格を使用します。
5. `set-service` サブコマンドで作成されたインストールサービス資格証明を使用します。
6. `set-server -D` サブコマンドで作成されたデフォルトのクライアント資格を使用します。
7. AI クライアントのクライアント資格が存在せず、サービスポリシーが `require-server-auth` の場合、デフォルトのクライアント OBP 鍵が使用されます。

## ▼ 自動インストールのセキュリティを構成する方法

### 1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

### 2. AI サーバーのセキュリティ資格を生成します。

次のコマンドは、X.509 ルート CA 証明書と署名 CA 証明書、サーバー証明書と非公開鍵、および AI サーバー認証のための OBP 鍵を自動的に生成します。CA 証明書と OBP 鍵は、それらがまだ存在していない場合にのみ生成されます。OBP 鍵が生成される場合、これらの鍵を設定する OBP コマンドが表示されます。

```
# installadm set-server -g
The root CA certificate has been generated.
The CA signing certificate request has been generated.
The signing CA certificate has been generated.
A new certificate key has been generated.
A new certificate has been generated.
Generating new encryption key...
To set the OBP encryption key for server authentication only, enter
  this OBP command:
  set-security-key wanboot-aes 8d210964e95f2a333c5e749790633273
Generating new hashing key (HMAC)...
To set the OBP hashing (HMAC) key for server authentication only,
  enter this OBP command:
  set-security-key wanboot-hmac-sha1 4088861239fa3f3bed22f8eb885bfa476952fab4
Configuring web server security.
Changed Server
```

AI サーバー資格の構成についての詳細は、[123 ページの「AI サーバー資格の構成」](#)を参照してください。

### 3. (オプション) インストールサービスセキュリティポリシーを設定します。

次の例は、インストールサービスを使用するためにクライアント認証が必要なセキュリティ設定を指定します。特定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 AI クライアントおよびすべてのデータを保護するには、`require-client-auth` インストールサービスセキュリティ設定を使用して、すべての AI クライアントがサーバー認証とクライアント認証の両方でセキュリティ保護されるようにします。

この例では、AI クライアントは `svcname` インストールサービスデータにアクセスするために X.509 資格証明を持っている必要があります。

```
# installadm set-service -p require-client-auth -n svcname
```

インストールサービスセキュリティポリシーの構成についての詳細は、[123 ページの「セキュアインストールサービスの構成」](#)を参照してください。

#### 4. AI クライアントの資格証明を生成します。

次の例は、非公開の X.509 証明書と鍵のペア、および指定された AI クライアントの認証用の X.509 CA 証明書を自動的に生成しますが、`02:00:00:00:00:00` はクライアントの MAC アドレスです。MAC アドレスを指定することによって割り当てられたクライアント資格は、AI クライアントごとに固有です。CA 証明書は、これがまだ存在していない場合に生成されます。AI クライアントが SPARC システムの場合、OBP 鍵もまだそれが存在しない場合は生成され、これらの鍵を設定するための OBP コマンドが表示されます。

```
# installadm set-client -e 02:00:00:00:00:00 -g
Generating credentials for client 02:00:00:00:00:00...
A new certificate key has been generated.
A new certificate has been generated.
Generating new encryption key...
To set the OBP encryption key, enter this OBP command:
  set-security-key wanboot-aes 030fd11c98afb3e434576e886a094c1c
Generating new hashing key (HMAC)...
To set the OBP hashing (HMAC) key, enter this OBP command:
  set-security-key wanboot-hmac-sha1 e729a742ae4ba977254a2cf89c2060491e7d86eb
Changed Client: '02:00:00:00:00:00'
```

クライアント資格の構成についての詳細は、[125 ページの「クライアント資格の構成」](#)を参照してください。

#### 5. SPARC クライアントの OBP 鍵を設定します。

セキュリティ資格を割り当てた SPARC クライアントの場合、AI クライアントをブートして AI インストールを開始するときに、OBP のセキュリティ鍵 (ハッシュ鍵および暗号化鍵) を設定する必要があります。次の例は、SPARC クライアントコンソール上で OBP AES 暗号化鍵を設定します。

```
ok set-security-key wanboot-aes 030fd11c98afb3e434576e886a094c1c
```

次の例は、SPARC クライアントコンソール上で OBP ハッシュ (HMAC) 鍵を設定します。

```
ok set-security-key wanboot-hmac-sha1 e729a742ae4ba977254a2cf89c2060491e7d86eb
```

詳細および例については、[252 ページの「セキュアなダウンロードを使用した SPARC AI クライアントのインストール」](#)を参照してください。

#### 6. セキュアな IPS リポジトリからインストールするように AI マニフェストを変更します。

セキュリティ保護された起点を持つパブリッシャーが AI マニフェストで指定された場合、`publisher` 要素の `credentials` サブ要素で鍵と証明書を指定します。詳

細は、[ai\\_manifest\(4\)](#) のマニュアルページのソフトウェアセクションを参照してください。image 要素の属性で SSL 鍵と証明書を指定できますが、この鍵と証明書はマニフェストで指定された最初のパブリッシャーに対してのみ適用されます。鍵と証明書が image 要素と credentials 要素の両方で指定された場合、credentials 要素で指定された資格証明が使用されます。鍵と証明書のファイルを、AI Web サーバーのユーザー指定のディレクトリに配置することを検討してください。詳細は、[106 ページ](#)の「[Web サーバーのユーザーファイルディレクトリの構成](#)」を参照してください。

## AI サーバー資格の構成

AI サーバーセキュリティには次の利点があります。

- AI クライアントが AI サーバーの識別情報を検証できます。
- AI クライアントは、TLS を介して自動的に暗号化されたデータを受け取るため、ネットワークトラフィックのモニタリングによってデータを読み取ることはできません。

[119 ページ](#)の「[セキュリティ資格の構成](#)」にある -C、-K、および -A オプションを使用する手順を参照してください。

**例 30** ユーザーが提供する資格証明を使用した AI サーバー資格の生成

次の例は、ユーザーが提供する資格証明を指定します。OBP 鍵は、それらがまだ存在しない場合に生成されます。OBP 鍵が生成される場合、これらの鍵を設定する OBP コマンドが表示されます。

```
# installadm set-server -C server.crt -K server.key -A cacert.pem
```

CA 証明書が指定されていない場合、これらのクライアント資格を生成するために使用される CA 証明書が事前に割り当てられている必要があります。

AI クライアントのインストールサービスがポリシー require-server-auth を持ち、クライアント資格またはサービス資格証明が割り当てられていない場合は、デフォルトのクライアント OBP 鍵が使用されます。さらに、デフォルトのクライアント資格が使用されている必要があります。

## セキュアインストールサービスの構成

インストールサービスを作成するときにこれを構成するには、installadm create-service コマンドを使用します。詳細は、[107 ページ](#)の「[インストールサービスを](#)

[作成する方法](#)を参照してください。既存のインストールサービスを再構成するには、`installadm set-service` コマンドを使用します。このセクションでは、インストールサービスのセキュリティーポリシーを設定する方法について説明します。

各インストールサービスには1つのセキュリティーポリシーセットがあります。次の選択が可能です。

#### `require-client-auth`

AI クライアントの識別情報を確認します。指定されたサービスのすべての AI クライアントにクライアントおよびサーバー認証が必要です。このオプションでは暗号化も必要です。

サービスのすべての AI クライアントに、クライアント認証での認証が必要になります。指定されたサービスのすべての AI クライアントに資格証明が割り当てられている必要があり、このサービスのすべての SPARC クライアントに OBP 鍵が定義されている必要があります。クライアント認証用に構成されていないサービスの AI クライアントはこのインストールサービスを使用できません。

#### `require-server-auth`

AI サーバーの識別情報を確認します。指定されたサービスのすべての AI クライアントでサーバー認証を実行する必要があります。このオプションでは暗号化も必要です。

指定されたインストールサービスにアクセスするための AI サーバー認証が最低限必要です。クライアント認証はオプションですが、割り当て済みクライアント資格または帰属済みクライアント資格を提供する必要があります。このサービスのすべての SPARC クライアントの OBP 鍵を定義する必要があります。

#### `optional`

認証された AI クライアントと認証されていない AI クライアントの両方が、インストールサービスにアクセスできるようにします。AI サーバーが資格証明を持つ場合、このオプションでは暗号化も必要です。これはデフォルトの動作です。

割り当て済みクライアント資格を提供する必要があります。割り当て済み資格証明または属性資格証明を持たない AI クライアントは、OBP 鍵またはサーバー認証を使用しません。サーバー認証は、クライアント認証用に構成された AI クライアントでのみ提供されます。

#### `encr-only`

x86 クライアントのみ: 認証を必要としない SSL/TLS のエンドツーエンドの暗号化を有効にします。認証がないため、AI クライアントと AI サーバーの識別情報は保証されません。移動中のデータを第三者がネットワーク上で読み取ることができません。

#### `disable`

指定されたサービスのすべての AI クライアントのすべてのセキュリティーを無効にします。

このサービスのクライアントは認証されません。資格証明は発行されません。このサービスのクライアントは、[106 ページの「Web サーバーのユーザーファイルディレクトリの構成」](#)に記載されている `webserver_secure_files_dir` ディレクトリにアクセスできません。この設定は、慎重に使用してください。認証によって以前保護されていたすべてのインストールサービスファイルが保護されなくなります。クライアントデータは、望ましくないアクセスからセキュリティ保護されません。認証を再度有効にするには、別のセキュリティポリシーの値を使用して、`set-service` サブコマンドを再指定してください。

#### 例 31 インストール時の AI サーバー認証の要求

この例は、インストールサービスを使用するためにサーバー認証が必要なセキュリティ設定を指定します。指定されたサービスの AI クライアントに対して AI サーバーの認証を最低限求めるには、`require-server-auth` インストールサービスセキュリティ設定を使用します。

```
# installadm set-service -p require-server-auth -n install-service
```

#### 例 32 x86: インストール時の暗号化の要求

この例では、暗号化を使用するが認証を必要としないセキュリティ設定を指定します。x86 クライアントについて、特定のインストールサービスについてのデータ転送を保護するが、クライアント認証またはサーバー認証を必要としない場合に、`encr-only` セキュリティ設定を使用します。サーバー証明書はやはり必要です。データはネットワーク上のスヌーピングから保護されますが、AI サーバーは、サーバーへの適切な要求を発行したすべての AI クライアントに対してデータを提供します。

```
# installadm set-service -p encr-only -n install-service
```

## クライアント資格の構成

AI クライアントのセキュリティには次の利点があります。

- AI サーバーは AI クライアントの識別情報を検証できます。
- データはネットワーク上で暗号化されます。
- カスタム資格証明を持つ AI クライアントの場合、クライアントに固有の公開されたすべてのファイルは、ほかの AI クライアントから読み取ることができません。
- 認証済みクライアントのみが、[106 ページの「Web サーバーのユーザーファイルディレクトリの構成」](#)に記載されているユーザー指定のセキュアディレクトリにアクセスできます。

特定の AI クライアント、特定のインストールサービスの AI クライアント、または資格証明をまだ持たないすべての AI クライアントに対して、資格証明を生成または指

定できます。生成された OBP 鍵は、双方向 (クライアントおよびサーバー) 認証に使用されます。SPARC クライアントにセキュリティー資格を割り当てる場合は、AI クライアントをブートしてインストールを開始するときに OBP 鍵を指定する必要があります。252 ページの「セキュアなダウンロードを使用した SPARC AI クライアントのインストール」を参照してください。

---

**注記** - create-client サブコマンドを使用すれば、AI クライアントを 1 つのインストールサービスから別のインストールサービスに移動できます。セキュリティー資格を持つ既存の AI クライアントで使用されるサブコマンドは、そのクライアントの資格に影響を与えません。

---

**例 33** 特定の AI クライアントのためのユーザーが提供する資格証明の使用

この例ではユーザーが提供する資格証明を指定します。AI クライアントが SPARC システムの場合、OBP 鍵がまだ存在しない場合は生成されます。OBP 鍵が生成される場合、これらの鍵を設定する OBP コマンドが表示されます。

```
# installadm set-client -e 02:00:00:00:00:00 -C client.crt -K client.key -A cacert.pem
```

119 ページの「セキュリティー資格の構成」にある、-C、-K、および -A オプションの使用についてのコメントを参照してください。CA 証明書が指定されていない場合、これらのクライアント資格を生成するために使用される CA 証明書が事前に割り当てられている必要があります。

-E および -H オプションの使用については、127 ページの「SPARC クライアントの OBP セキュリティーキー」を参照してください。

**例 34** 特定のインストールサービスの AI クライアントの資格証明

この例は、solaris11\_3-sparc インストールサービスに割り当てられ、資格証明がまだ割り当てられていないすべての AI クライアントに資格証明を提供します。

```
# installadm set-service -g -n solaris11_3-sparc
Generating credentials for service solaris11_3-sparc...
A new certificate key has been generated.
A new certificate has been generated.
Generating new encryption key...
To set the OBP encryption key, enter this OBP command:
  set-security-key wanboot-aes 34bc980ccc8dfef478f89b5acbfd51b4
Generating new hashing key (HMAC)...
To set the OBP hashing (HMAC) key, enter this OBP command:
  set-security-key wanboot-hmac-sha1 b8a9f0b3472e8c3b29443daf7c9d448faad14fee
```

このインストールサービスは SPARC インストールサービスのため、OBP 鍵も生成され、これらの鍵を設定するための OBP コマンドが表示されます。

あとで solaris11\_3-sparc インストールサービスに割り当てられる AI クライアントも、それらの MAC アドレスを指定してそれらのクライアントに資格証明が割り当てられない場合は、これらの資格証明を使用します。

このオプションは、複数の AI クライアント間でアプリケーションの統一されたセットを実行する場合に便利です。ただし、MAC アドレスを指定することによって資格証明が割り当てられていない、このインストールサービスのすべてのクライアントは、同一の資格証明を持ち、互いのインストールデータを表示できます。

119 ページの「[セキュリティー資格の構成](#)」にある、`-C`、`-K`、および `-A` オプションの使用についてのコメントを参照してください。

`-E` および `-H` オプションの使用については、127 ページの「[SPARC クライアントの OBP セキュリティーキー](#)」を参照してください。

### 例 35 デフォルトの AI クライアント資格

この例は、資格証明を割り当てられていないすべての AI クライアントにデフォルトの資格証明セットを提供します。

```
# installadm set-server -D -g
Generating default client credentials...
A new certificate key has been generated.
A new certificate has been generated.
Generating new encryption key...
To set the OBP encryption key, enter this OBP command:
  set-security-key wanboot-aes 7cdbda5b8fc4b10ffbd29fa19d13af77
Generating new hashing key (HMAC)...
To set the OBP hashing (HMAC) key, enter this OBP command:
  set-security-key wanboot-hmac-sha1 14effe2c515da4940ef1db165791e92790163004
```

一部の AI クライアントが SPARC クライアントのため、OBP 鍵も生成され、これらの鍵を設定するための OBP コマンドが表示されます。

デフォルトのクライアント資格が割り当てられると、すべての AI クライアントはクライアント認証およびサーバー認証を実行することが期待され、AI サーバーのすべての SPARC クライアント用にファームウェア鍵が必要になります。また、複数の AI クライアントが同一の資格証明を持つため、これらは互いのインストールデータを表示できるようになります。

119 ページの「[セキュリティー資格の構成](#)」にある、`-C`、`-K`、および `-A` オプションの使用についてのコメントを参照してください。

`-E` および `-H` オプションの使用については、127 ページの「[SPARC クライアントの OBP セキュリティーキー](#)」を参照してください。

## SPARC クライアントの OBP セキュリティーキー

SPARC クライアントが強化されたセキュリティー機能を利用する場合、AI クライアントをブートしてインストールを開始するときに、OBP のセキュリティー鍵を設定する必要があります。

`installadm` コマンドに `set-server`、`set-service`、または `set-client` サブコマンドを指定して使用して、TLS 資格証明を生成または指定するとき、ハッシュ (HMAC) 鍵および暗号化鍵がまだ存在しない場合、これらの鍵は自動的に生成されて表示されます。これらのファームウェア鍵は、同じコマンドを繰り返した場合に自動的に再生成されません。

OBP 鍵を再生成するには `-E` および `-H` オプションを使用できます。OBP 鍵が存在する前に `-E` または `-H` オプションを指定しないでください。すでに存在する暗号化鍵または HMAC が無効化されて置換されます。暗号化鍵を再生成するには `-E` オプションを使用します。ハッシュ鍵を再生成するには `-H` オプションを使用します。`-E` と `-H` の両方のオプションを指定することも、`-E` オプションのみ、または `-H` オプションのみ指定することもできます。コマンドを実行すると、すでに存在する OBP 鍵は無効にされ、新しく生成された値に置き換えられます。これらの鍵を設定する OBP コマンドが表示されます。

OBP セキュリティーキーを設定する OBP コマンドをあとで表示するには、次の例のように `list` サブコマンドに `-v` オプションを付けて使用します。

```
# installadm list -v -e mac-addr
```

このコマンドは、TLS 資格証明がクライアント MAC アドレスを使用して指定されたか、インストールサービス名を使用して指定されたか、あるいはデフォルトのクライアント資格であるかにかかわらず、この AI クライアントの正しい OBP 鍵を表示します。`list` サブコマンドからの出力には、[例49「AI クライアントのセキュリティー情報の一覧表示」](#)に示すように、OBP 鍵が、この特定の AI クライアントに対して、指定されたインストールサービスに対して、あるいはデフォルト AI クライアントに対して定義されたかどうかを示します。

## セキュリティーの無効化および有効化

このセクションでは、セキュリティー構成を削除せずにセキュリティー要件を無効化し、その後、以前構成済みのサーバーおよびクライアントの認証設定を使用してセキュリティー要件を再度有効化するために使用できるオプションについて説明します。

セキュリティーはデフォルトで有効化されています。セキュリティーが無効になっている間は、AI クライアントに資格が発行されず、AI クライアントからも資格が要求されません。セキュリティーが無効になっている間は、AI クライアントに提供されるどの AI ファイルにも HTTPS ネットワークによる保護が提供されません。AI Web サーバーによって提供されるユーザー指定のセキュアファイル ([106 ページの「Web サーバーのユーザーファイルディレクトリの構成」](#)に記載) は、セキュリティーが無効になっている間はアクセスできません。

セキュリティーが無効になっている間も、引き続きセキュリティーを構成できます。変更内容はすべて、セキュリティーがふたたび有効になったときに反映されます。

サーバー全体にわたるセキュリティーの適用を無効にするには、次のコマンドを使用します。

```
# installadm set-server -S
Refreshing web server.
Automated Installer security has been disabled.
```

インストールサービスがすでに構成されているシステムに対してセキュリティーを無効にする場合は、注意してください。セキュリティー保護されているインストールサービスデータはアクセスの認証が不要になり、未認証のクライアントが AI 経由で Oracle Solaris をインストールできるようになります。

`set-security --disable` を使用してセキュリティーの適用を無効にしたあとでセキュリティーの適用を再度有効にするには、次のコマンドを使用します。

```
# installadm set-security -s
Configuring web server security.
Refreshing web server.
Warning: client 02:00:00:00:00:00 of service solaris11_3-i386
is required to have credentials but has none.
Automated Installer security has been enabled.
```

## 資格の削除

セキュリティー資格を削除するには、`installadm` コマンドを使用します。`set-server`、`set-service`、および `set-client` サブコマンドを使用してセキュリティー資格を削除できます。

セキュリティー資格は `delete-client` または `delete-service` サブコマンドを実行するときも削除されます。`delete-client` コマンドは、クライアント固有のすべての資格を削除します。`delete-service` サブコマンドは、サービス固有のすべての資格と、そのサービスおよび別名サービスのすべての AI クライアントについてのクライアント固有の資格を削除します。

**注意** - 削除された資格を回復することはできず、TLS セキュリティープロトコルはサーバー資格がないと機能できません。サーバー資格を削除する前に、AI セキュリティーが無効になります。

### 例 36 1つの AI クライアントの資格の削除

この例では、非公開鍵および証明書、すべての CA 証明書、MAC アドレスを使用して AI クライアントに割り当てられたすべての OBP 鍵を削除します。OBP 鍵がクライアントファームウェアで設定されている場合、[253 ページの「ハッシュ鍵と暗号化鍵の削除」](#)の説明に従ってこれらを設定解除してください。

```
# installadm set-client -e mac-addr -x
```

**例 37** CA 証明書の削除

この例は、その CA 証明書を使用するすべての AI クライアントの指定された CA 証明書を削除します。--hash オプション引数の値は、list サブコマンドによって表示されて例49「AI クライアントのセキュリティ情報の一覧表示」に示されている、証明書の X.509 サブジェクトのハッシュ値です。指定された CA 証明書を使用中のすべての AI クライアントがカウントされ、続行を確認するメッセージとともに表示されます。

```
$ installadm set-client -x --hash b99588cf
Identifier hash: b99588cf
Subject: /C=CZ/O=Oracle Czech s.r.o./OU=install/CN=genca
Issuer: /C=CZ/O=Oracle Czech s.r.o./OU=install/CN=genca
Valid from Apr 27 13:12:27 2012 GMT to Apr 27 13:12:27 2015 GMT
This CA has the following uses:
WARNING: this is the server CA certificate
Deleting this Certificate Authority certificate can prevent
credentials from validating.
Do you want to delete this Certificate Authority certificate [y|N]: y
Deleting all references to Certificate Authority with hash value b99588cf
```

**注意** -この例では、この CA 証明書のすべてのインスタンスが、これを使用するすべての AI クライアントについて削除され、影響を受けるクライアントを認証できなくなります。指定された CA 証明書が証明書の生成に使用されると、installadm コマンドは証明書を生成できなくなります。

**例 38** AI サーバーのセキュリティ資格の削除

この例では、AI サーバーの非公開鍵および証明書、すべての CA 証明書、およびサーバー認証用の OBP 鍵のみを削除します。

```
# installadm set-server -x
```

## ▼ AI を使用して Kerberos クライアントを構成する方法

この手順では、Kerberos クライアントの keytab ファイルはすでに AI サーバーで作成および格納されています。この例では、既存の資格を使用するか、新しいプリンシパルを使用することによって Kerberos クライアントを構成する、自動登録を使用します。自動登録プロセスは、個々の Kerberos クライアント用に keytab ファイルを作成してエンコードする必要がないため簡単です。

1. **管理者になります。**  
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「割り当てられている管理権利の使用」を参照してください。
2. **必要に応じてインストールサービスを作成します。**

```
# installadm create-service -n krb-sparc \
  -d /export/auto_install/krb-sparc \
  -s /export/auto_install/iso/sol-11_3-ai-sparc.iso
Creating service from:
/export/auto_install/iso/sol-11_3-ai-sparc.iso
Setting up the image ...
Creating sparc service: krb-sparc
Image path: /export/auto_install/krb-sparc
Refreshing install services
```

### 3. Kerberos クライアントをサービスに関連付けます。

Kerberos が動作しているインストールする必要があるすべての AI クライアントに対して、この手順を繰り返します。この例では、11:11:11:11:11:11 アドレスを使用中の AI クライアントが、krb-sparc インストールサービスに関連付けられます。

```
# installadm create-client -n krb-sparc -e 11:11:11:11:11:11
Adding host entry for 11:11:11:11:11:11 to local DHCP configuration.
```

### 4. Kerberos クライアントの資格を作成します。

```
# installadm set-client -c 11:11:11:11:11:11 -g
Generating credentials for client 11:11:11:11:11:11...
A new certificate key has been generated.
A new certificate has been generated.
```

### 5. Kerberos 構成ファイルの内容を定義するシステム構成プロファイルを作成します。

この例では、kclient コマンドを対話的に実行してプロファイルを作成します。あるいは、コマンド行オプションを使用するか入力プロファイルを使用するコマンドを呼び出すことができます。詳細は、[kclient\(1M\)](#) のマニュアルページを参照してください。

この例では、KDC は MIT サーバー上で実行されています。Solaris KDC のサンプル出力を表示するには、[例39 「Kerberos クライアントのデプロイ中の既存の鍵のダウンロード」](#) を参照してください。AD クライアントのサンプル出力を表示するには、[例 41 「MS AD ドメインへの Kerberos クライアントの自動参加」](#) を参照してください。

```
# kclient -x /root/krb-sc.xml
Starting client setup
-----
Is this a client of a non-Solaris KDC ? [y/n]: y
Which type of KDC is the server:
  ms_ad: Microsoft Active Directory
  mit: MIT KDC server
  heimdal: Heimdal KDC server
  shishi: Shishi KDC server
Enter required KDC type: mit
Do you want to use DNS for kerberos lookups ? [y/n]: n
  No action performed.
Enter the Kerberos realm: EXAMPLE.COM
Specify the master KDCs for the above realm using a comma-separated list: kdc.example.com
Do you have any slave KDC(s) ? [y/n]: y
Enter a comma-separated list of slave KDC host names: kdc2.example.com
Do you have multiple domains/hosts to map to a realm ? [y/n]: n
```

```
No action performed.
Setting up /root/krb-sc.xml.
```

**6. (オプション) Kerberos クライアントのバイナリ keytab ファイルを XML プロファイルに変換します。**

自動登録によって鍵を取得できるか、Kerberos クライアントに鍵がない場合、この手順は不要です。Kerberos クライアントの keytab ファイルを作成する必要がありますが、これは通常、クライアントが最初に構成されるときに KDC 管理者によって実行されます。

```
# kclient-kt2prof -k ./host1.keytab -p /root/host1.xml
```

**7. Kerberos クライアントの残りの部分を構成するクライアントプロファイルを作成します。**

この手順ではプロファイルを使用する必要があるため、システム構成プロファイルを使用して可能な限り Kerberos クライアントを構成します。

**8. (オプション) プロファイルのセキュリティポリシーを設定します。**

クライアントプロファイルが keytab を含む場合、require-client-auth セキュリティポリシーをサービスに割り当てることで、認証済み AI クライアントのみが keytab ファイルをダウンロードできるようにする必要があります。

```
# installadm set-service -p require-client-auth -n krb-sparc
```

**9. クライアントプロファイルをインストールサービスに関連付けます。**

Kerberos 構成ファイルとクライアント keytab ファイルのプロファイル、および作成したほかのすべてのプロファイルをインストールサービスに関連付けます。

```
# installadm create-profile -n krb-sparc -f /root/krb-sc.xml
Profile krb-sc.xml added to database.
# installadm create-profile -n krb-sparc -f /root/host1.xml -c mac="11:11:11:11:11:11"
Profile host1.xml added to database.
```

**10. Kerberos クライアントをブートして AI プロセスを開始します。**

**例 39** Kerberos クライアントのデプロイ中の既存の鍵のダウンロード

自動登録の使用は、KDC が Solaris KDC または MS AD のいずれかの場合にのみ機能することに注意してください。KDC が MIT、Heimdal、または Shishi の場合、事前生成された keytab 転送のみが可能です。

自動登録を使用して既存の鍵をダウンロードするには、KDC 上で c および i 管理者権限を使用して admin プリンシパルを作成しておく必要があります。この例では、プリンシパルの名前は download/admin です。

この例で、KDC は Oracle Solaris を実行しています。また、Kerberos クライアントの鍵がすでに作成されています。

この例は、Kerberos 構成ファイルのシステム構成プロファイルを作成するときに `download/admin` プリンシパルを追加する方法を示しています。`download/admin` プリンシパルは、Kerberos クライアントがデプロイされるときに KDC サーバーから既存の鍵を転送するために使用される特殊な `admin` プリンシパルです。

```
# kclient -x /root/krb-sc.xml
Starting client setup
-----
Is this a client of a non-Solaris KDC ? [y/n]: n
    No action performed.
Do you want to use DNS for kerberos lookups ? [y/n]: n
    No action performed.
Enter the Kerberos realm: EXAMPLE.COM
Specify the master KDCs for the above realm using a comma-separated
list: kdc.example.com
Do you have any slave KDC(s) ? [y/n]: y
Enter a comma-separated list of slave KDC host names: kdc2.example.com
Do you have multiple domains/hosts to map to realm ? EXAMPLE.COM [y/n]: n
    No action performed.
Should the client automatically join the realm ? [y/n]: y
Enter the krb5 administrative principal to be used: download/admin
Password for download/admin: xxxxxxxx
Do you plan on doing Kerberized nfs ? [y/n]: n
    No action performed.
Is this client a member of a cluster that uses a logical host name ? [y/n]: n
    No action performed.
Do you have multiple DNS domains spanning the Kerberos realm EXAMPLE.COM ? [y/n]: n
    No action performed.
Setting up /root/krb-sc.xml.
```

#### 例 40 Kerberos クライアントのデプロイ中の新しい鍵の作成

自動登録の使用は、KDC が Solaris KDC または MS AD のいずれかの場合にのみ機能することに注意してください。KDC が MIT、Heimdal、または Shishi の場合、事前生成された keytab 転送のみが可能です。

自動登録を使用して新しい鍵をダウンロードするには、KDC 上で `a`、`c`、および `i` 管理者権限を使用して `admin` プリンシパルを作成しておく必要があります。この例では、プリンシパルの名前は `create/admin` です。

この例で、KDC は Oracle Solaris を実行しています。この例では、Kerberos 構成ファイルのシステム構成プロファイルを作成するときに `create/admin` プリンシパルを追加します。`create/admin` プリンシパルは、Kerberos クライアントがデプロイされるときに KDC サーバーから既存の鍵を転送するために使用される特殊な `admin` プリンシパルです。このコマンドには多くのオプションが含まれているため、問われる質問の数は少なくなっています。

```
# kclient -x /root/krb-sc.xml -R EXAMPLE.COM -a create/admin -d none -m kdc.example.com
Starting client setup
-----
Do you have multiple domains/hosts to map to realm ? EXAMPLE.COM [y/n]: n
    No action performed.
Should the client automatically join the realm ? [y/n]: y
Password for create/admin: xxxxxxxx
Setting up /root/krb-sc.xml.
```

**例 41 MS AD ドメインへの Kerberos クライアントの自動参加**

この例では、Kerberos クライアントが AD ドメインに参加します。Kerberos 構成ファイルのシステム構成プロファイルを作成するときに Administrator プリンシパルを追加するには、次のコマンドを使用します。

```
# kclient -x /root/krb-sc.xml
Starting client setup
-----
Is this a client of a non-Solaris KDC ? [y/n]: y
Which type of KDC is the server:
    ms_ad: Microsoft Active Directory
    mit: MIT KDC server
    heimdal: Heimdal KDC server
    shishi: Shishi KDC server
Enter required KDC type: ms_ad
Should the client automatically join AD domain ? [y/n]: y
Enter the Kerberos realm: EXAMPLE.COM
Enter the krb5 administrative principal to be used: Administrator
Password for Administrator: xxxxxxxx
Setting up /root/krb-sc.xml.
```

## インストールサービスの情報の表示

インストールサービスと、そのサービスに関連付けられている AI クライアント、AI マニフェスト、およびシステム構成プロファイルについての情報を表示するには、`installadm list` コマンドを使用します。このセクションの内容は次のとおりです。

[例42「AI サーバー上のすべてのインストールサービスの一覧表示」](#)では、AI 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方法を示します

[例43「指定したインストールサービスの情報の表示」](#)では、特定のインストールサービスに関する情報を一覧表示する方法を示します

[例44「インストールサービスに関連付けられた AI クライアントの一覧表示」](#)では、インストールサービスに関連付けられた AI クライアントを一覧表示する方法を示します

[例45「特定のインストールサービスに関連付けられた AI クライアントの一覧表示」](#)では、特定のインストールサービスに関連付けられた AI クライアントを一覧表示する方法を示します

[例46「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」](#)では、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方法を示します

[例47「指定したインストールサービスに関連付けられているマニフェストとプロファイルの一覧表示」](#)では、特定のインストールサービスに関連付けられた 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方法を示します

[例49「AI クライアントのセキュリティー情報の一覧表示」](#)では、AI クライアントのセキュリティー情報を一覧表示する方法を示します

例48「AI サーバーのセキュリティ情報の一覧表示」では、AI サーバーのセキュリティ情報を一覧表示する方法を示します

例 42 AI サーバー上のすべてのインストールサービスの一覧表示

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

```
$ /usr/sbin/installadm list
```

Service Name	Status	Arch	Type	Alias	Aliases	Clients	Profiles	Manifests
default-i386	on	i386	iso	yes	0	0	0	1
default-sparc	on	sparc	iso	yes	0	0	0	1
solaris11_3-i386	on	i386	iso	no	1	0	1	1
solaris11_3-sparc	on	sparc	iso	no	1	0	2	1

default-i386 サービスは、この AI サーバー上で最初の i386 サービスが作成されたときに自動的に作成されました。default-i386 サービスは、create-client サブコマンドを使用して solaris11\_3-i386 サービスに関連付けられていない x86 クライアントによって使用されます。default-i386 および solaris11\_3-i386 サービスはインストールイメージを共有しますが、異なる AI マニフェストとシステム構成プロファイルを持っています。

default-sparc サービスは、この AI サーバー上で最初の sparc サービスが作成されたときに自動的に作成されました。default-sparc サービスは、create-client サブコマンドを使用して solaris11\_3-sparc サービスに関連付けられていない SPARC クライアントによって使用されます。default-sparc および solaris11\_3-sparc サービスはインストールイメージを共有しますが、異なる AI マニフェストとシステム構成プロファイルを持っています。

例 43 指定したインストールサービスの情報の表示

この例は、-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示します。

```
$ /usr/sbin/installadm list -n solaris11_3-sparc
```

Service Name	Status	Arch	Type	Alias	Aliases	Clients	Profiles	Manifests
solaris11_3-sparc	on	sparc	iso	no	1	0	2	1

例 44 インストールサービスに関連付けられた AI クライアントの一覧表示

この例では、この AI 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 AI クライアントを一覧表示します。AI 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。114 ページの「インストールサービスへの AI クライアントの関連付け」を参照してください。

```
$ /usr/sbin/installadm list -c
```

Service Name	Client Address	Arch	Secure	Custom Args	Custom Grub
solaris11_3-sparc	00:14:4F:A7:65:70	sparc	no	no	no
solaris11_3-i386	08:00:27:8B:BD:71	i386	no	no	no
	01:C2:52:E6:4B:E0	i386	no	no	no

**例 45** 特定のインストーラサービスに関連付けられた AI クライアントの一覧表示

この例では、指定されたインストーラサービスに追加されているすべての AI クライアントを一覧表示します。次の例では、1つの AI クライアントが solaris11\_3-sparc インストーラサービスに関連付けられています。

```
$ /usr/sbin/installadm list -c -n solaris11_3-sparc
```

Service Name	Client Address	Arch	Secure	Custom Args	Custom Grub
solaris11_3-sparc	00:14:4f:a7:65:70	sparc	no	no	no

**例 46** すべての AI マニフェストおよびシステム構成プロファイルの一覧表示

この例では、この AI サーバー上のすべてのインストーラサービスの 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。「Service/Manifest Name」および「Profile Name」列には、マニフェスト、スクリプト、またはプロファイルの内部名が表示されます。「Status」列では、各サービスのデフォルトのマニフェストとアクティブでないマニフェストが特定されます。マニフェストが関連付けられた条件を持たず、デフォルトでもない場合、そのマニフェストは非アクティブです。「Criteria」列には、関連付けられているクライアント条件が表示されます。

orig\_default マニフェストは、インストーラサービスが作成されたときにインストーラ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を使用して作成され、このサービスの新しいデフォルトマニフェストとして指定されています。mem1 はデフォルトのマニフェストなので、その条件は無視されます。別のマニフェストをデフォルトのマニフェストとして作成した場合は、mem1 マニフェストを使用する AI 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 AI 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。AI クライアントは、サービスに関連付けられたほかのすべてのマニフェストの条件に一致しない場合、デフォルトのマニフェスト (この場合は mem1) を使用します。AI マニフェストの選択の詳細については、[第9章「AI クライアントへのカスタマイズの割り当て」](#)を参照してください。

```
$ installadm list -m -p
```

Service Name	Manifest Name	Type	Status	Criteria
default-i386	orig_default	derived	default	one
default-sparc	orig_default	derived	default	none

```

solaris1132-i386          ipv4          xml          active      ipv4 = 10.6.68.1 -
10.6.68.200              mem1          derived      default     (Ignored: mem =
2048 MB - 4095 MB)
                          orig_default  derived      inactive    none
solaris11_3-sparc        sparc-ent    xml          active      mem = 4096 MB -
unbounded
                          platform =
SUNWSPARC-Enterprise
2048 MB - 4095 MB)
                          mem1          derived      default     (Ignored: mem =
                          orig_default  derived      inactive    none

Service Name             Profile Name Environment Criteria
-----
solaris11_3-i386        mac2          system      mac = 08:00:27:8B:BD:71
                          hostname = server2
                          mac3          system      mac = 01:C2:52:E6:4B:E0
                          hostname = server3
                          ipv4          system      ipv4 = 10.0.2.100 - 10.0.2.199
                          mem1          system      mem = 2048 MB - 4095 MB
solaris11_3-sparc        mac1          system      mac = 01:C2:52:E6:4B:E0
                          hostname = server1
                          ipv4          system      ipv4 = 192.168.168.251
                          platform = SUNWSPARC-Enterprise
                          mem = 4096-unbounded
                          sparc-ent    system

```

適切なプロファイルを使用してこのコマンドを実行した場合、マニフェストの一覧の追加の列に、マニフェストのタイプ `xml` または `derived` が示されます。

**例 47** 指定したインストールサービスに関連付けられているマニフェストとプロファイルの一覧表示

この例では、インストールサービス `solaris11_3-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示します。

```

$ installadm list -m -p -n solaris11_3-sparc
Service Name             Manifest Name Type      Status  Criteria
-----
solaris11_3-sparc        sparc-ent    xml      active  mem = 4096 MB - unbounded
                          platform = SUNWSPARC-
Enterprise
                          mem1          derived  default Ignored:
                          mem = 2048 MB - 4095 MB)
                          orig_default  derived  inactive none

Service Name             Profile Name Environment Criteria
-----
solaris11_3-sparc        mac1          system      mac = 01:C2:52:E6:4B:E0
                          hostname = server1
                          ipv4          system      ipv4 = 192.168.168.251
                          platform = SUNWSPARC-Enterprise
                          mem = 4096-unbounded
                          sparc-ent    system

```

例 48 AI サーバーのセキュリティ情報の一覧表示

list サブコマンドに -v および -s オプションを指定すると、次の情報を含む AI サーバーについての情報が表示されます。

- 現在のセキュリティ状態: 有効または無効
- サーバー証明書 X.509 サブジェクトと発行者文字列
- AI サーバー証明書が有効な日付
- AI サーバー証明書の検証結果
- サーバー CA 証明書のハッシュ値、X.509 サブジェクト、および発行者
- クライアント認証用のクライアント CA 証明書
- デフォルトのクライアント証明書

```
# installadm list -v -s
AI Server Parameter Value
-----
Hostname ..... install-svr
Architecture ..... i386
Active Networks .... 10.134.125.170
Http Port ..... 5555
Secure Port ..... 5556
Image Path Base Dir .... /export/auto_install
Multi-Homed? ..... no
Managing DHCP? ..... yes
DHCP IP Range ..... 192.168.100.240 - 192.168.100.249
Boot Server ..... 192.168.100.45
Web UI Enabled? ..... yes
Wizard Saves to Server? no
Security Enabled? ..... yes
Security Key? ..... yes
Security Cert:
    Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=osol-inst
    Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
    Source : Server Certificate
    Valid from: Jan 24 22:53:00 2015 GMT
              to: Jan 24 22:53:00 2025 GMT
    Validates?: yes
CA Certificates:
    d09051e4 Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
            Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
            Source : Server CA Certificate
            Valid from: Jan 24 22:53:00 2015 GMT
                    to: Jan 24 22:53:00 2025 GMT
    f9d73b41 Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
            Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
            Source : Server CA Certificate
            Valid from: Jan 24 22:53:00 2015 GMT
                    to: Jan 24 22:53:00 2025 GMT
Def Client Sec Key? .... yes
Def Client Sec Cert:
    Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Client default
    Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
    Source : Default Client Certificate
    Valid from: Jul 15 19:33:00 2013 GMT
              to: Jul 13 19:33:00 2023 GMT
Def Client CA Certs .... none
Def Client FW Encr Key . adcc858c58ecae04c02282e7245c235c
Def Client FW HMAC Key . cb7bc6213512c8fa3dc7d7283a9e056dc2791f98
```

```
Number of Services ..... 102
Number of Clients ..... 37
Number of Manifests .... 108
Number of Profiles ..... 92
```

**例 49** AI クライアントのセキュリティー情報の一覧表示

list サブコマンドに `-v` および `-e` オプションを指定すると、次の AI クライアントセキュリティー情報が表示されます。

- AI クライアントに使用される資格
- クライアントの資格のソース
- クライアントの証明書の有効性

```
# installadm list -v -e 00:14:4F:83:3F:4A
Service Name      Client Address   Arch  Secure Custom Args Custom Grub
-----
solaris11_3-sparc 00:14:4F:A7:65:70 sparc yes    no      no

Client Credentials? yes
Security Key? ..... yes
Security Cert:
    Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=CID 01020000000000
    Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Signing CA
    Valid from: Jan 24 10:20:00 2015 GMT
    to: Jan 24 10:20:00 2025 GMT

CA Certificates:
d09051e4 Subject: /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
    Issuer : /C=US/O=Oracle/OU=Solaris Deployment/CN=Root CA
    Source : Default CA Certificate
    Valid from: Jan 24 22:53:00 2015 GMT
    to: Jan 24 22:53:00 2025 GMT

FW Encr Key (AES) . 23780bc444636f124ba3ff61bdac32d1
FW HMAC Key (SHA1) 1093562559ec45a5bb5235b27c1d0545ff259d63
Boot Args ..... none
```

export サブコマンドは、AI クライアントに帰属する TLS 資格を表示します。`-c` を追加すると X.509 TLS 証明書が表示されます。

```
# installadm export -e 00:14:4F:83:3F:4A -c
----- certificate: client_00:14:4F:83:3F:4A_cert_de22916b -----
-----BEGIN CERTIFICATE-----
MIICFDCCAX+gAwIBAgIBGTALBgqhkiG9w0BAQswUDELMAKGA1UEBhMCVVMxZDZAN
.....
UiZDA6G0dvE=
-----END CERTIFICATE-----
```

`-k` オプションは X.509 非公開鍵を表示します。

```
# installadm export -e 00:14:4F:83:3F:4A -k
----- key: client_00:14:4F:83:3F:4A_key -----
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDCCJbC5Bd0uMQ0A0k41Ll1Qqw1Qwqkx91pIhH131tF1/WxHi74A
....
```

```
SYoBeKAOPSo7Evund+bHAR010H4QnbSJg11UDuZr3T3h
-----END RSA PRIVATE KEY-----
```

## インストールサービスの管理

既存のインストールサービスを再構成するには、`installadm set-service` コマンドを使用します。このセクションの例では、インストールサービス別名、デフォルト AI マニフェスト、またはインストールサービスのイメージパスを設定する方法と、インストールサービスを更新する方法を示します。

### インストールサービス別名の設定

インストールサービス別名を使用して、新しいサービスが作成されたときに実行する必要のある再構成の量を最小限に抑えることができます。たとえば、`default-arch` インストールサービスは別名です。サービスを作成するとき、`create-service` サブコマンドの `-t` オプションを使用することによって別名を作成できます。`set-service` サブコマンドの `-t` オプションは、指定されたサービスを、別のサービスの別名に変更します。`set-service` サブコマンドを使用するとき、指定されたサービスはすでに別名でなければなりません。

サービスまたは別名のいずれ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条件は、別名のリセット後も変わりません。指定されたサービスが使用するネットイメージが変わるだけです。

新しいインストールイメージに関連付けられた AI DTD と SMF DTD が異なる可能性があるため、別名を設定する前にサービスに追加されたマニフェストとプロファイルは、別名のリセット時に再検証されます。この検証は、`create-manifest` および `create-profile` コマンドによって実行される検証と同じです。

#### 例 50 インストールサービス別名の作成

この例は、新しい `install-sparc` サービスを、既存の `solaris11_3-sparc` インストールサービスの別名として作成します。

```
# installadm create-service -t solaris11_3-sparc -n install-sparc
```

#### 例 51 インストールサービス別名の変更

この例では、`solaris11_3-i386` インストールサービスと `install-i386` インストールサービス別名が両方とも事前に作成されている必要があります。次の例

は、`install-i386` インストーラサービスを `solaris11_3-i386` インストーラサービスの別名として設定します。

```
# installadm set-service -t solaris11_3-i386 -n install-i386
```

## インストーラサービスのデフォルト AI マニフェストの設定

これらの例では、新規インストーラサービスおよび既存のインストーラサービスの両方について、特定のマニフェストまたは派生マニフェストスクリプトをデフォルトとして指定する方法を示します。

**例 52**                    インストーラサービスを作成する際のデフォルト AI マニフェストの設定

この例は、`mem1` マニフェストを新しい `install-sparc` サービスのデフォルトマニフェストとして設定します。ほかのクライアント条件に一致しないこのサービスに関連付けられたすべての AI クライアントは、デフォルトでこのマニフェストを使用します。

```
# installadm create-service -M /tmp/mem1 -n install-sparc
```

**例 53**                    既存のインストーラサービスを変更することによるデフォルト AI マニフェストの設定

この例は、すでに登録済みの `mem1` マニフェストを既存の `install-i386` サービスのデフォルトマニフェストとして設定します。ほかのクライアント条件に一致しないこのサービスに関連付けられたすべての AI クライアントは、デフォルトでこのマニフェストを使用します。

```
# installadm set-service -M mem1 -n install-i386
```

## インストーラサービスのイメージパスの設定

これらの例は、特定のサービスのインストールイメージへのパスを定義またはリセットする方法を示します。

**例 54**                    新しいインストールイメージのイメージパスの設定

この例は、サービスおよびネットイメージを作成するときに、`solaris11_3-i386` サービスのインストールイメージへのパスを定義します。

```
# installadm set-service -d /export/ai-images/solaris11_3.i386 -n solaris11_3-i386
```

**例 55** 既存のインストールイメージのイメージパスの設定

この例は、solaris11\_3-i386 サービスのインストールイメージへのパスを再配置します。

```
# installadm set-service -d /export/ai-images/solaris11_3.i386 -n solaris11_3-i386
```

## 既存のインストールサービスの更新

IPS AI ネットイメージパッケージを使用して作成されたサービスの別名に関連付けられたイメージを更新するには、`update-service` サブコマンドを使用します。更新されたイメージを使用して新規サービスが作成され、別名は新規サービスを使用するように変更されます。

サービスを更新する際に異なるリポジトリを使用するには、`update-service` サブコマンドに `-p` オプションを追加します。`-p` オプションが指定されていない場合、`svc-name` が別名であるサービスのイメージの作成に使用されたパブリッシャーが使用されます。

`-s` オプションが指定されていない場合、パブリッシャーから入手可能な最新バージョンの `install-image/solaris-auto-install` パッケージが使用されます。

**例 56** インストールサービスの更新

この例は新規サービスを作成し、この新規サービスを使用するように `default-i386` 別名を変更します。

```
# installadm update-service -n default-i386
```

**例 57** インストールサービスを更新するときの異なるリポジトリの使用

この例は、solaris11\_3-i386 サービスに関連付けられているパブリッシャーを識別する方法を示しています。最初に `installadm list` サブコマンドを使用してサービスのイメージパスを決定します。次に、イメージパスを使用して、使用されているパブリッシャーを決定することができます。

```
$ installadm list -v -n solaris11_3-i386
Service Name      Status Arch  Type Alias Aliases Clients Profiles Manifests
-----
solaris11_3-i386 on    i386  iso  no   1      0      1      1

    Image Path ..... /export/auto_install/solaris11_3-i386
    ....

$ pkg -R /export/auto_install/solaris11_3-i386 publisher
PUBLISHER      TYPE      STATUS  URI
solaris        origin   online  http://pkg.oracle.com/solaris/release/
```

この例は、インストールサービスを更新するときに、`example.com/solaris/mybuild` のパブリッシャーを使用することを指定します。

```
# installadm update-service -n default-i386 -p solaris=http://example.com/solaris/mybuild
```

**例 58**            インストールサービスを更新するときの異なるネットイメージパッケージの使用

この例は、特定のネットイメージパッケージを指定します。

```
# installadm update-service -n default-i386 -s FMRI
```

## AI マニフェストの管理

このセクションでは、AI マニフェストを更新、削除、検証、またはエクスポートする方法を示します。

### AI マニフェストの更新

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

`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。

マニフェストは指定されたサービスにすでに存在しなければなりません。`installadm list` コマンドを使って確認してください ([例46 「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」](#) を参照)。

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

- 指定されたマニフェスト内の `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名前に一致する場合)。
- 指定されたファイル名のベース名 (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名前に一致する場合)。

この例は、`solaris11_3-sparc` サービスの `sparc-ent` マニフェストの内容を、`./mymanifests/manifest-new-sparc-ent.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 `sparc-ent` です。

```
# installadm update-manifest -n solaris11_3-sparc \  
-f ./mymanifests/manifest-new-sparc-ent.xml -m sparc-ent
```

## AI マニフェストの検証

AI マニフェスト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。

インストールサービスに追加されていないマニフェストを検証するには `-M` オプションを使用します。`-M` 引数の値はマニフェストへのパス名です。

指定された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を検証するには `-m` オプションを使用します。マニフェスト名の可能な値を表示するには、[例 46 「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」](#) に示すように、`installadm list` コマンドを使用します。`create-manifest` サブコマンドは、AI マニフェストを検証してから、それらをインストールサービスに追加します。`validate -m` サブコマンドは、マニフェストが追加されて以降、壊れていないことを検証します。

インストールサービスに追加されたマニフェストと、まだ追加されていないマニフェストのサービス名を指定する必要があります。それぞれのバージョンの OS で 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マニフェストにはサービス名が必要です。AI 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 AI クライアントで使用される DTD に対してマニフェストを検証する必要があります。詳細は、[ai\\_manifest\(4\)](#) のマニュアルページを参照してください。

検証されたマニフェストは `stdout` に出力されます。エラーは `stderr` に出力されます。

## AI マニフェストの削除

指定された AI マニフェストまたは派生マニフェストスクリプトを指定された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。マニフェスト名は、`installadm list` コマンドから返されるのと同じ名前です ([例 46 「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」](#) を参照)。

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、`sparc-ent` AI マニフェストを `solaris11_3-sparc` インストールサービスから削除します。

```
# installadm delete-manifest -m sparc-ent -n solaris11_3-sparc
```

## システム構成プロファイルの管理

このセクションでは、システム構成プロファイルを更新、削除、検証、およびエクスポートするための手順を示します。

### システム構成プロファイルの更新

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

プロファイルと、サービスを含むインストールサービスの名前を指定できます。インストールサービスとともにプロファイルが指定されない場合、更新されるプロファイルの名前はファイルのベース名です。

次のコマンドは、`solaris11_3-sparc` サービスの `sparc-ent` プロファイルの内容を、`./myprofiles/profile-new-sparc-ent.xml` の内容で更新します。

```
# installadm update-profile -n solaris11_3-sparc \  
-f ./myprofiles/profile-new-sparc-ent.xml -p sparc-ent
```

### システム構成プロファイルの検証

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

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。 `-p` 引数の値は、プロファイルへのパス名です。

指定された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。プロファイル名の可能な値を表示するには、[例46「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」](#) に示すように、`installadm list` コマンドを使用します。`create-profile` サブコマンドは、システム構成プロファイルを検証してから、それらをインストールサービスに追加します。`validate -p` サブコマンドは、プロファイルが追加されて以降、壊れていないことを検証します。

インストールサービスに追加されたプロファイルと、まだ追加されていないプロファイルのサービス名を指定する必要があります。それぞれのバージョンの OS で 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。AI 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 AI 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。詳細は、[service\\_bundle\(4\)](#) のマニュアルページを参照してください。

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` に出力されません。

## システム構成プロファイルの削除

*profile* システム構成プロファイルを *svcname* インストールサービスから削除するには、`installadm delete-profile` コマンドを使用します。*profile* 引数の値は、`installadm list` コマンドによって返されるプロファイル名です。[例46「すべての AI マニフェストおよびシステム構成プロファイルの一覧表示」](#)を参照してください。

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

次のコマンドは、`sparc-ent` システム構成プロファイルを `solaris11_3-sparc` インストールサービスから削除します。

```
# installadm delete-profile -p sparc-ent -n solaris11_3-sparc
```

## AI マニフェストまたはシステム構成プロファイルのエクスポート

指定された AI マニフェストまたはシステム構成プロファイルの内容を、指定されたインストールサービスから指定された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。

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

指定されたマニフェストは、XML AI マニフェストまたは派生マニフェストスクリプトの名前です。マニフェストや派生マニフェストスクリプトの作成方法については、[第10章「AI クライアントのインストールパラメータの定義」](#)を参照してください。

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

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



## AI クライアントへのカスタマイズの割り当て

---

インストールをカスタマイズするには、AI マニフェスト内のインストール手順とシステム構成プロファイル内の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 AI 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、インストール手順が指定されている 1 つ以上の AI マニフェストと、構成手順が指定されている 0 個以上の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 AI クライアントで 1 つのみ使用します。各 AI 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。AI クライアントがプロファイルを使用しない場合は、その AI クライアントの構成を行うために、対話型ツールがインストール後の初回ブート時にそのクライアント上で開きます。

### AI 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 AI サーバーを設定します。AI クライアントはネットワーク経由でブートするとき、AI サーバーのインストールサービスを使用します。

AI クライアントは、そのクライアントのアーキテクチャー用のデフォルトのインストールサービスか、または割り当てられたインストールサービスを使用します。インストールサービスは、この章で説明する方法を使用して、AI 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。

ある AI 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 AI クライアントをインストールサービスに追加します (第14章「AI サーバーを使用した AI クライアントのインストール」を参照)。AI クライアントの MAC アドレスと、このク

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 AI クライアントがブートすると、そのクライアントは AI 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 アドレスを確認するには、[dladm\(1M\)](#) のマニュアルページに説明されている `dladm` コマンドを使用します。

1 つのインストールサービス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 マニフェストおよびシステム構成プロファイルを作成します。新しい AI マニフェストおよびプロファイルを AI インストールサービスに追加します。どの AI クライアントがどの AI マニフェストや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。[116 ページの「インストールサービスとクライアント固有のインストール手順の関連付け」](#) を参照してください。

カスタムの AI マニフェストを作成する方法については、[第10章「AI クライアントのインストールパラメータの定義」](#) 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方法については、[第11章「AI クライアントのシステム構成パラメータの定義」](#) を参照してください。

## AI マニフェストの選択

各 AI 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、AI 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、[189 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、AI 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 AI 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。
- AI 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、AI 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6](#)に示されている順序でクライアントの特性が評価されます。`installadm` ツールは、同じタイプの条件が重複していないことを確認します。詳細は、[116 ページの「インストールサービスとクライアント固有のインストール手順の関連付け」](#) を参照してください。

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 59 AI 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。sparc-ent.xml の AI マニフェストは、AI クライアントプラットフォーム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="platform">
    <value>SUNW, SPARC-Enterprise</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>
```

MAC アドレス 00:14:4f:a7:65:70 を持つ AI 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。

AI クライアントが M4000 または M5000 の場合、sparc-ent.xml が割り当てられます。

AI クライアントがいずれの AI マニフェストの条件にも一致しない場合、インストールサービスのデフォルトマニフェストがクライアントに割り当てられます。

## システム構成プロファイルの選択

AI クライアントのシステム構成プロファイルの選択にも、AI マニフェストの選択に使用されるのと同じ条件のキーワードが使用されます。表6を参照してください。

特定の AI 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、両方のプロファイルがその AI クライアントの構成に使用されます。

## 選択条件

表6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 AI クライアントを示すために使用できる条件のキーワードを示しています。例の欄には、使用可能な値がいくつか示されています。条件のキーワードと値は、`installadm` サブコマンドの `create-manifest`、`create-profile`、および `set-criteria` とともに使用できます。

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

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

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

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

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

表 6 条件のキーワードと条件の階層

条件のキーワード	説明	コマンド行と XML ファイルの例
<code>arch</code>	<code>uname -m</code> によって返されるアーキテクチャー  値: <code>i86pc</code> 、 <code>sun4u</code> 、または <code>sun4v</code>	<b>CLI:</b> <code>-c arch="i86pc"</code>  <b>XML:</b> <pre>&lt;ai_criteria name="arch"&gt;   &lt;value&gt;i86pc&lt;/value&gt; &lt;/ai_criteria&gt;</pre>
<code>cpu</code>	<code>uname -p</code> によって返される CPU クラス  値: <code>i386</code> または <code>sparc</code>	<b>CLI:</b> <code>-c cpu="sparc"</code>  <b>XML:</b> <pre>&lt;ai_criteria name="cpu"&gt;   &lt;value&gt;sparc&lt;/value&gt; &lt;/ai_criteria&gt;</pre>
<code>hostname</code>	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。	<b>CLI、単一のホスト名:</b> <code>-c hostname="host3"</code>  <b>CLI、ホスト名のリスト:</b> <code>-c hostname="host1 host2 host6"</code>

条件のキーワード	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	<p><b>XML、単一のホスト名:</b></p> <pre>&lt;ai_criteria name="hostname"&gt;   &lt;value&gt;host3&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p><b>XML、ホスト名のリスト:</b></p> <pre>&lt;ai_criteria name="hostname"&gt;   &lt;value&gt;host1 host2 host6&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p><b>CLI、単一の IP アドレス:</b></p> <pre>-c ipv4="10.6.68.127"</pre> <p><b>CLI、IP アドレスの範囲:</b></p> <pre>-c ipv4="10.6.68.1-10.6.68.200"</pre> <p><b>XML、単一の IP アドレス:</b></p> <pre>&lt;ai_criteria name="ipv4"&gt;   &lt;value&gt;10.6.68.127&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p><b>XML、IP アドレスの範囲:</b></p> <pre>&lt;ai_criteria name="ipv4"&gt;   &lt;range&gt;     10.6.68.1     10.6.68.200   &lt;/range&gt; &lt;/ai_criteria&gt;</pre>
mac	コロン (;) 区切り文字による 16 進の MAC アドレスまたは MAC アドレスの範囲。	<p><b>CLI、単一の MAC アドレス:</b></p> <pre>-c mac="0:14:4F:20:53:97"</pre> <p><b>CLI、MAC アドレスの範囲:</b></p> <pre>-c mac=0:14:4F:20:53:94-0:14:4F:20:53:A0</pre> <p><b>XML、単一の MAC アドレス:</b></p> <pre>&lt;ai_criteria name="mac"&gt;   &lt;value&gt;0:14:4F:20:53:97&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p><b>XML、MAC アドレスの範囲:</b></p> <pre>&lt;ai_criteria name="mac"&gt;   &lt;range&gt;     0:14:4F:20:53:94     0:14:4F:20:53:A0   &lt;/range&gt; &lt;/ai_criteria&gt;</pre>
mem	prtcnf によって返される M バイト単位のメモリーサイズ、またはメモリーサイズの範囲  unbounded キーワードは、範囲に上限がないことを示します。	<p><b>CLI、1 つのメモリーサイズ:</b></p> <pre>-c mem="4096"</pre> <p><b>CLI、メモリーサイズの範囲:</b></p> <pre>-c mem="2048-unbounded"</pre> <p><b>XML、1 つのメモリーサイズ:</b></p> <pre>&lt;ai_criteria name="mem"&gt;   &lt;value&gt;4096&lt;/value&gt; &lt;/ai_criteria&gt;</pre>

条件のキーワード	説明	コマンド行と XML ファイルの例
		<p><b>XML、メモリーサイズの範囲:</b></p> <pre>&lt;ai_criteria name="mem"&gt;   &lt;range&gt;     2048     unbounded   &lt;/range&gt; &lt;/ai_criteria&gt;</pre>
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	<p><b>CLI、単一の IP アドレス:</b></p> <pre>-c network="10.0.0.0"</pre> <p><b>CLI、IP アドレスの範囲:</b></p> <pre>-c network="11.0.0.0-12.0.0.0"</pre> <p><b>XML、単一の IP アドレス:</b></p> <pre>&lt;ai_criteria name="network"&gt;   &lt;value&gt;10.0.0.0&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p><b>XML、IP アドレスの範囲:</b></p> <pre>&lt;ai_criteria name="network"&gt;   &lt;range&gt;     11.0.0.0     12.0.0.0   &lt;/range&gt; &lt;/ai_criteria&gt;</pre>
platform	uname -i (x86 システム) および prtconf -b (SPARC システム) によって返されるプラットフォーム名 値には次が含まれます。  i86pc M4000 および M5000 サーバーの場合は SUNW, SPARC-Enterprise T4 サーバーの場合は ORCL, SPARC-T4-2	<p><b>CLI:</b></p> <pre>-c platform="SUNW, SPARC-Enterprise"</pre> <p><b>XML:</b></p> <pre>&lt;ai_criteria name="platform"&gt;   &lt;value&gt;SUNW, SPARC-Enterprise&lt;/value&gt; &lt;/ai_criteria&gt;</pre>
zonename	zoneadm list によって示されるゾーンの名前または名前のリスト。第12章「ゾーンのインストールと構成」を参照してください。	<p><b>CLI、単一のゾーン名:</b></p> <pre>-c zonename="myzone"</pre> <p><b>CLI、ゾーン名のリスト:</b></p> <pre>-c zonename="zoneA zoneB zoneC"</pre> <p><b>XML、単一のゾーン名:</b></p> <pre>&lt;ai_criteria name="zonename"&gt;   &lt;value&gt;myzone&lt;/value&gt; &lt;/ai_criteria&gt;</pre> <p><b>XML、ゾーン名のリスト:</b></p> <pre>&lt;ai_criteria name="zonename"&gt;   &lt;value&gt;zoneA zoneB zoneC&lt;/value&gt; &lt;/ai_criteria&gt;</pre>

## AI クライアントのインストールパラメータの定義

---

インストールサービスを作成すると、AI 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。デフォルトの 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派生マニフェストです。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を参照してください。

この章では、特定の AI クライアントのカスタム AI マニフェストを作成する方法について説明します。

- カスタムの XML AI マニフェストファイルを作成します。この方法は、カスタム プロビジョニングを必要とするシステムがほとんどない環境にもっとも適しています。インストールされるほとんどのシステムは同一または類似したハードウェアを備えており、同じようにプロビジョニングされます。[156 ページの「XML AI マニフェストファイルをカスタマイズする方法」](#)を参照してください。
- インストール時に AI 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特性に基づいて AI 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。[159 ページの「派生マニフェストスクリプトを作成して適用する方法」](#)を参照してください。
- AI マニフェストウィザードを使用して、XML ファイルを編集する必要なしにマニフェストを作成します。[176 ページの「AI マニフェストウィザードを使用して AI マニフェストを作成する方法」](#)を参照してください。
- Oracle Solaris 11.3 リリースを実行している場合、XML ファイルを編集することなく新しいマニフェストを作成したり、既存のマニフェストを編集したりするには、`installadm` コマンドを使用します。この方法は、コマンド行から変更を行うために使用します。[178 ページの「installadm コマンドを使用した AI マニフェストの作成」](#)を参照してください。

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。どの特定の AI 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 AI クライアントがどの AI マニフェストを使用するか

は、そのマニフェストをインストールサービスに追加するときに指定する条件によって決まります。AI クライアントがカスタムの AI マニフェストの使用条件に一致しない場合は、デフォルトのマニフェストが使用されます。サービス内の任意の AI マニフェストをそのサービスのデフォルトになるように指定できます。

## XML AI マニフェストファイルのカスタマイズ

カスタムの XML AI マニフェストファイルを作成して適用するには、次の手順を使用します。

### ▼ XML AI マニフェストファイルのカスタマイズする方法

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. 既存の AI マニフェストをコピーします。

AI インストールサービスを作成すると、そのインストールサービスにデフォルトの AI マニフェストが用意されます。インストールサービスの作成方法については、[第8章「AI サーバーの設定」](#)を参照してください。

a. 既存のマニフェストを一覧表示します。

`installadm list` コマンドを使用して、特定のインストールサービスにすでに関連付けられている AI マニフェストを表示します。

```
$ installadm list -m -n solaris11_3-i386
Service Name      Manifest Name Type   Status Criteria
-----
solaris11_3-i386  orig_default  derived default none
```

b. 特定のマニフェストのコピーを取得します。

`installadm export` コマンドを使用して、このデフォルトのマニフェストまたはこのサービスに追加されているその他の AI マニフェストの内容を抽出します。

```
# installadm export -n solaris11_3-i386 -m orig_default -o mem1
```

これで `orig_default` のコピーがファイル `mem1` に入ります。

3. マニフェストのコピーを変更します。

mem1 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。

#### 4. 新しいマニフェストをインストールサービスに追加します。

新しい AI マニフェストを適切な AI インストールサービスに追加し、どの AI 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 solaris11_3-i386 -f ./mem1 -m mem1 \
-c mem="2048-unbounded"
```

-c オプションは複数指定できます。または、-c を指定して、多数のクライアント条件が含まれているファイルを使用します。クライアント条件の指定方法については、[第9章「AI クライアントへのカスタマイズの割り当て」](#) および set-criteria サブコマンドを参照してください。

このコマンドの実行後に list サブコマンドを実行すると、次のように表示されます。

```
# installadm list -m -n solaris11_3-i386
Service Name          Manifest Name      Type   Status   Criteria
-----
solaris11_3-i386     mem1               derived active   mem = 2048 MB -
unbounded
orig_default         derived default   none
```

#### ■ 新しいマニフェストをデフォルトにします。

任意のマニフェストファイルまたは派生マニフェストスクリプトをサービスのデフォルトのマニフェストまたはスクリプトになるように指定できます。サービスにすでに追加したマニフェストおよびスクリプトの中でデフォルトを変更するには、set-service サブコマンドで -M オプションを使用します。

```
# installadm set-service -M mem1 -n solaris11_3-i386
# installadm list -m -n solaris11_3-i386
Service Name          Manifest Name      Type   Status
Criteria
-----
solaris11_3-i386     mem1               derived default / active   mem =
2048 MB - unbounded
orig_default         derived inactive   none
```

この例で元のデフォルトは、どの AI クライアントがそれを使用するかを指定した条件がないため、非アクティブになっています。クライアントの選択条件がなくてもアクティブな状態でいられるのは、デフォルトのマニフェストまたはスクリプトのみです。

#### ■ 新しいマニフェストをデフォルトとして追加します。

このサービス用に新しいデフォルトのマニフェストまたはスクリプトを追加する場合は、create-manifest で -d オプションを使用します。指定した条件は保存されますが、別のマニフェストがデフォルトにされるまで無視されます。

```
# installadm create-manifest -n solaris11_3-i386 -d \
-f ./region1.xml -m region1
# installadm list -m -n solaris11_3-i386
```

Service Name	Manifest Name	Type	Status	Criteria
solaris11_3-i386	mem1	derived	active	mem = 2048 MB -
unbounded	region1	xml	default	none
	orig_default	derived	inactive	none

- **既存のマニフェストをカスタマイズします。**  
新しいマニフェストまたはスクリプトを追加せずに、既存のマニフェストまたはスクリプトの内容を変更するには、`installadm update-manifest` コマンドを使用します。更新の結果として、条件、デフォルトのステータス、およびマニフェスト名またはスクリプト名は変更されません。

```
# installadm update-manifest -n solaris11_3-i386
-f ./newregion1.xml -m region1
```

5. **カスタマイズしたマニフェストを検証します。**  
`create-manifest` および `update-manifest` サブコマンドは、XML マニフェストファイルを構文的に検証してから、それらをインストールサービスに追加します。AI では、インストール時に AI マニフェストが意味的に検証されます。

---

**注記** - AI 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 AI クライアント上の `/system/volatile/install_log` を参照してください。

---

`installadm` の `list`、`export`、`create-manifest`、`set-criteria`、`update-manifest`、および `set-service` サブコマンドの詳細は、[107 ページの「インストールサービスの操作」](#) も参照してください。

## クライアントインストール時の AI マニフェストの作成

インストールに先だってカスタム AI マニフェストを作成する代わりに、インストール時に AI 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もできます。このスクリプトでは、AI 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 AI 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、各システムをインス

トールするときに、AI 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、インストール時に各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、AI クライアントごとにカスタムのディスクレイアウトを指定するように AI マニフェストを変更できます。

## ▼ 派生マニフェストスクリプトを作成して適用する方法

### 1. 変更するマニフェストを選択します。

変更するベースマニフェストとして使用する既存の AI マニフェストを特定します。

スクリプトを作成してテストする場合は、ローカルコピーを使って作業できます。インストール時に、この派生マニフェストスクリプトを使用する各 AI クライアントがベースマニフェストにアクセスできる必要があります。

### 2. マニフェストを変更するスクリプトを記述します。

インストールしている AI クライアントの属性に基づいてインストール時にベースマニフェストを動的に変更するスクリプトを作成します。

### 3. スクリプトをインストーラサービスに追加します。

派生マニフェストスクリプトを適切な AI インストールサービスに追加し、インストール時にどの AI クライアントがこのスクリプトを使用してインストール手順を作成するかを定義した条件を指定します。クライアントの選択条件を指定しない場合は、このスクリプトをサービスのデフォルトの AI マニフェストとして追加できます。

AI は、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。

---

**注記** - 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、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 AI クライアント上の `/system/volatile/install_log` を参照してください。

---

インストールが成功した場合は、派生マニフェストが AI クライアント上の `/var/log/install/derived/manifest.xml` にコピーされ、マニフェストの派生に使われたスクリプトが `/var/log/install/derived/manifest_script` にコピーされます。

## 派生マニフェストスクリプトの作成

一般に、派生マニフェストスクリプトでは、クライアントから情報を取得し、その情報を使ってベース AI マニフェストを変更して、この AI クライアント専用のカスタム

AI マニフェストを作成します。派生マニフェストスクリプトでは、複数の部分的な AI マニフェストを組み合わせることもできます。最終的な派生マニフェストは、完全で、検証に合格する必要があります。

派生マニフェストスクリプトは、イメージでサポートされているどの種類のスクリプトにもできます。たとえば、デフォルトでは `ksh93` と `python` がイメージに含まれています。別の種類のスクリプトを使用する場合は、必要なサポートがイメージに含まれていることを確認します。

## クライアント属性の取得

派生マニフェストスクリプトは、システムの属性を読み取るコマンドを実行できません。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、および権限の詳細については、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』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 AI クライアントの属性を使用できます。

表 7 AI 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 AI 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_CONFIG_PROFILE_DIR	ユーザーが作成したシステム構成プロファイルが格納され、インストールサービスによって使用されるディレクトリ。
SI_CPU	インストールされる AI 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_DISKNAME_#	AI クライアントで見つかったディスクの名前を表す平坦な変数セット。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。
SI_DISKSIZE_#	AI 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 AI クライアントの IP アドレス。

環境変数名	説明
SI_HOSTNAME	インストール環境で設定された AI クライアントのホスト名。
SI_STALL_PROFILE_DIR	インストール環境用のユーザー指定のシステム構成プロファイルが格納され、インストールサービスによって使用される可能性のあるディレクトリ。
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_KARCH	AI 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_MEMSIZE	AI 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_NATISA	AI 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	AI 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_NUMDISKS	AI クライアントのディスク数。
SI_PLATFORM (または SI_MODEL)	AI クライアントのプラットフォーム。uname -i (x86 システム) および prtconf -b (SPARC システム) の出力と同等です。
SI_SYSPKG	AI クライアント上の Oracle Solaris incorporation パッケージのリリース (現在は entire という名前です)。AI クライアントのパッケージ全体が pkg://solaris/entire@0.5.11,5.11-0.175.0.0.0.2.0:20111020T143822Z である場合、SI_SYSPKG の値は pkg:/entire@0.5.11-0.175.0 となります。アップデートまたは SRU リリースの場合、AI クライアントのパッケージ全体が pkg://solaris/entire@0.5.11,5.11-0.175.1.19.0.6.0:20140508T221351Z であれば、SI_SYSPKG の値は pkg:/entire@0.5.11-0.175.1 となります。

## AI マニフェストのカスタマイズ

AI マニフェストの XML 要素を追加または変更するには、`/usr/bin/aimanifest` コマンドを使用します。

`aimanifest` によって変更するファイルには、少なくとも次の内容が含まれている必要があります。

- 作成している XML マニフェストで有効な DTD への `!DOCTYPE` 参照。
- この DTD のルート要素。

次の例は、この派生マニフェストスクリプトが追加されるインストールサービスの AI DTD ファイルの指定など、AI マニフェストのために最低限必要なベースマニフェストファイルを示しています

```
<!DOCTYPE auto_install SYSTEM "file:///imagepath/auto_install/ai.dtd.1">
<auto_install/>
```

`imagepath` 引数の値は、次のコマンドによって返されるパスです (`svcname` はこの派生マニフェストスクリプトが追加されるインストールサービスの名前です)。

```
$ installadm list -v -n svcname
```

---

**注記** - スクリプトを使用して AI クライアントをインストールする前に、イメージパスを `///usr/share` に戻してください。

---

`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。ロードされるファイルはすべて、インストール時に AI クライアントからアクセス可能でなければいけません。たとえば、ターゲットインストールサービスの `imagepath/auto_install/manifest/` からマニフェストを読み込むことができます。

この章の例では、ファイル `/usr/share/auto_install/manifest/default.xml` が読み込まれます。`/usr/share/auto_install/manifest/` 内のマニフェスト例は、ターゲットインストールサービスのマニフェストとは異なっている可能性があります。本稼働では、`/usr/share/auto_install/manifest/` からマニフェストを読み込まないようにしてください。

`load` サブコマンドは、部分的なマニフェストの読み込みまたは挿入にも使用できます。

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。マニュアルページと以下のスクリプト例には、`aimanifest` コマンドの使用例が含まれています。

---

**注記** - `aimanifest` コマンドで指定した値に次のいずれかの文字が含まれている場合は、これらの文字が XML のパス名の一部として解釈されないように、その値を一重引用符または二重引用符で囲む必要があります。

```
/"@[]=
```

場合によっては、使用するシェルの規則に従って引用符の前にバックスラッシュ文字 (`\`) を付けてエスケープし、シェルが引用符を削除または解釈しないようにする必要があります。

---

次の例では、パッケージ名 `pkg:/entire` が含まれる `software_data` 要素のアクションが返されます。この例では、スラッシュ文字が特殊文字であるため、`pkg:/entire` の前後に引用符を付ける必要があります。このコマンドが `ksh93` スクリプトなどのシェルスクリプト内で呼び出される場合は、バックスラッシュ文字でこれらの引用符をエスケープする必要があります。

```
# /usr/bin/aimanifest get software_data[name="\pkg:/entire\"]@action
```

---

**ヒント** - ベストプラクティスに基づき、エラー発生時に停止するトラップを設定してください。

---

次のスクリプトの一部は、派生マニフェストスクリプトの良い見本です。

```
#!/bin/ksh93
```

```

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load baseAImanifest.xml

# Customize AI manifest. For example:
/usr/bin/aimanifest load -i manifest_fragment.xml
/usr/bin/aimanifest set origin@name file:///net/myserver/myrepo/repo.redist

exit $SCRIPT_SUCCESS

```

## 派生マニフェストスクリプトの例

このセクションでは、クライアントの属性を判別し、その情報を使って AI マニフェストをカスタマイズする派生マニフェストスクリプトを作成する方法を示します。これらの例には、有効な AI マニフェストの作成に必要な情報が必ずしもすべて含まれているわけではありません。

これらの例を試すには、次の設定手順を実行します。

1. **AIM\_MANIFEST** 環境変数を、スクリプトによって AI マニフェストが作成される場所に設定します。  
**\$AIM\_MANIFEST** ファイルは、そのファイルを変更する **aimanifest** コマンドごとに書き換えられます。**aimanifest** を **load**、**add**、**delete**、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\_LOGFILE** 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 AI 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7](#)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 60** ディスクサイズに基づいたディスクパーティション分割の指定

この例では、ターゲットディスクのサイズが 1T バイトを超える場合は Oracle Solaris fdisk パーティションにディスクの半分だけを使用するように AI マニフェストをカスタマイズします。このスクリプトの異なる実行に対して、`SI_DISKSIZE_1` を 1T バイト未満に設定してから、1T バイトを超えるサイズに設定してみます。また、スクリプトを実行する前に `SI_NUMDISKS` と `SI_DISKNAME_1` も設定します。指定されたパーティション分割は x86 クライアントのみに適用されるため、このスクリプトは x86 クライアント専用です。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# Check that there is only one disk on the system.
if [[ $SI_NUMDISKS -gt "1" ]] ; then
    print -u2 "System has too many disks for this script."
    exit $SCRIPT_FAILURE
fi

/usr/bin/aimanifest add \
    /auto_install/ai_instance/target/disk/disk_name@name $SI_DISKNAME_1

if [[ $SI_DISKSIZE_1 -gt "1048576" ]] ; then
    typeset -i PARTN_SIZE=$SI_DISKSIZE_1/2

    # Default action is to create.
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk[disk_name@name="\$SI_DISKNAME_1\"]/
partition@name 1
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk/partition[@name=1]/size@val \
        ${PARTN_SIZE}mb
else
    /usr/bin/aimanifest add \
        /auto_install/ai_instance/target/disk[disk_name@name="\$SI_DISKNAME_1\"]/
partition@action \
        use_existing_solaris2
fi
exit $SCRIPT_SUCCESS
```

`SI_DISKSIZE_1` の値が 1048576 以下である AI クライアントでは、次の要素が `$AIM_MANIFEST` に追加されます。

```
<target>
  <disk>
    <disk_name name="/dev/dsk/c0t0d0s0"/>
    <partition action="use_existing_solaris2"/>
  </disk>
<!-- <logical> section -->
```

```
</target>
```

SI\_DISKSIZE\_1 が 1048576 より大きい AI クライアントでは、SI\_DISKSIZE\_1 の値に応じて、次のような要素が \$AIM\_MANIFEST に追加されます。

```
<target>
  <disk>
    <disk_name name="/dev/dsk/c0t0d0s0"/>
      <partition name="1">
        <size val="524288mb"/>
      </partition>
    </disk>
    <!-- <logical> section -->
  </target>
```

パーティションを追加するためのコマンドには、そのパーティションに対して別個のディスク指定が作成されないように、disk\_name が指定されています。この例のスクリプトは、パーティションが (ほかのディスクではなく) \$SI\_DISKNAME\_1 ディスク上にあることを指定しています。この例の該当する行が次の行に置き換わると、意図した結果が得られません。

```
    /usr/bin/aimanifest add \
      /auto_install/ai_instance/target/disk/partition@name 1
  /usr/bin/aimanifest add \
    /auto_install/ai_instance/target/disk/partition[@name=1]/size@val \
    ${PARTN_SIZE}mb
else
  /usr/bin/aimanifest add \
    /auto_install/ai_instance/target/disk/partition@action \
    use_existing_solaris2
```

このスクリプトは、上記の出力ではなく、次の不正な出力を提供します。

```
<target>
  <disk>
    <disk_name name="c0t0d0s0"/>
  </disk>
  <disk>
    <partition name="1">
      <size val="524288mb"/>
    </partition>
  </disk>
</target>
```

#### 例 61 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 という名前の 2 台のディスクを含むシステムの場合、この例の出力は次の XML 要素になります。

```

<target>
  <disk in_vdev="mirror_vdev" in_zpool="rpool" whole_disk="true">
    <disk_name name="c0t0d0" name_type="ctd"/>
  </disk>
  <disk in_vdev="mirror_vdev" in_zpool="rpool" whole_disk="true">
    <disk_name name="c0t1d0" name_type="ctd"/>
  </disk>
  <logical>
    <zpool name="rpool" is_root="true">
      <vdev name="mirror_vdev" redundancy="mirror"/>
      <filesystem name="export" mountpoint="/export"/>
      <filesystem name="export/home"/>
      <be name="solaris"/>
    </zpool>
  </logical>
</target>

```

#### 例 62 特定のサイズのディスクが 2 つ以上ある場合のミラー化構成の指定

この例では、システムに 200G バイトのディスクが 2 つ以上ある場合にミラー化構成を指定するように AI マニフェストをカスタマイズします。少なくとも 200G バイトであることがわかった最初の 2 つのディスクを使用します。スクリプトを実行する前

に、テスト環境で `SI_NUMDISKS`、`SI_DISKNAME_1`、および `SI_DISKSIZE_1` を設定します。また、`SI_NUMDISKS` に設定した値に応じて、`SI_DISKNAME_2`、`SI_DISKSIZE_2` などを必要に応じて設定します。AI のインストール中に、これらの環境変数が設定され、派生マニフェストスクリプトで使用できるようになります。

この例では、同じパスを持つノードが複数存在している場合にノードを変更する方法を示しています。シェルの実装では、`aimanifest` の戻りパス (`-r`) オプションを使用して特定のノードへのパスを返し、そのパスを使用して同じノードに追加の変更を行います。Python の実装では、サブパス (ノードパスの内側で `[]` を使用) を使って同じノードに追加の変更を行う方法を示しています。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

# Find the disks first.
typeset found_1
typeset found_2
typeset -i disk_num

for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)); do
    eval curr_disk="$SI_DISKNAME_${disk_num}"
    eval curr_disk_size="$SI_DISKSIZE_${disk_num}"
    if [[ $curr_disk_size -ge "204800" ]]; then
        if [ -z $found_1 ]; then
            found_1=$curr_disk
        else
            found_2=$curr_disk
            break
        fi
    fi
done

# Now, install them into the manifest.
# Let the installer take the default action if two large disks are not found.

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

if [[ -n $found_2 ]]; then
    # Turn on mirroring.
    vdev=$(/usr/bin/aimanifest add -r \
        /auto_install/ai_instance/target/logical/zpool/vdev@redundancy mirror)
    /usr/bin/aimanifest set ${vdev}@name mirror_vdev

    disk=$(/usr/bin/aimanifest add -r \
        /auto_install/ai_instance/target/disk@in_vdev mirror_vdev)
    disk_name=$(/usr/bin/aimanifest add -r ${disk}/disk_name@name $found_1)
    /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
```

```
fi

exit $SCRIPT_SUCCESS

次のスクリプトは、前の Korn shell 版を Python 版にしたものです。

#!/usr/bin/python2.6

import os
import sys

from subprocess import check_call, CalledProcessError

SCRIPT_SUCCESS = 0
SCRIPT_FAILURE = 1

def main():

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

    si_numdisks = int(os.environ["SI_NUMDISKS"])
    for disk_num in range(1, si_numdisks + 1):
        curr_disk_var = "SI_DISKNAME_" + str(disk_num)
        curr_disk = os.environ[curr_disk_var]
        curr_disk_size_var = "SI_DISKSIZE_" + str(disk_num)
        curr_disk_size = os.environ[curr_disk_size_var]
        if curr_disk_size >= "204800":
            if not len(found_1):
                found_1 = curr_disk
            else:
                found_2 = curr_disk
            break

    # Now, write the disk specifications into the manifest.
    # Let the installer take the default action if two large disks are not found.

    try:
        check_call(["/usr/bin/aimanifest", "load",
                  "/usr/share/auto_install/manifest/default.xml"])
    except CalledProcessError as err:
        sys.exit(err.returncode)

    if len(found_2):
        try:
            check_call(["/usr/bin/aimanifest", "add",
                      "target/logical/zpool[@name=rpool]/vdev@redundancy", "mirror"])
            check_call(["/usr/bin/aimanifest", "set",
                      "target/logical/zpool/vdev[@redundancy='mirror']@name", "mirror_vdev"])

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

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

```

        "target/disk[disk_name@name='" + found_2 + "'" + "@in_vdev",
"mirror_vdev"])
    except CalledProcessError as err:
        sys.exit(err.returncode)

    sys.exit(SCRIPT_SUCCESS)

if __name__ == "__main__":
    main()

```

#### 例 63 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、AI 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、AI クライアントの IP アドレスが別の範囲にある場合は別のパッケージをインストールするように AI マニフェストをカスタマイズします。スクリプトを実行する前に、テスト環境で `SI_HOSTADDRESS` を設定します。AI のインストール中に、この環境変数が設定され、派生マニフェストスクリプトで使用できるようになります。

```

#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

# First determine which range the host IP address of the client is in.
echo $SI_HOSTADDRESS | sed 's/\./ /g' | read a b c d

# Assume all systems are on the same class A and B subnets.

# If the system is on class C subnet = 100, then install the /pkg100 package.
# If the system is on class C subnet = 101, then install the /pkg101 package.
# 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 64 特定のサイズ以上のディスクのみをターゲットディスクにする指定

この例では、50G バイト以上のディスクにのみインストールするように AI マニフェストをカスタマイズします。それよりも小さいディスクは無視します。

スクリプトを実行する前に、テスト環境で `SI_NUMDISKS`、`SI_DISKNAME_1`、および `SI_DISKSIZE_1` を設定します。また、`SI_NUMDISKS` に設定した値に応じて、`SI_DISKNAME_2`、`SI_DISKSIZE_2` などを必要に応じて設定します。AI のインストール中に、これらの環境変数が設定され、派生マニフェストスクリプトで使用できるようになります。

```
#!/bin/ksh93

SCRIPT_SUCCESS=0
SCRIPT_FAILURE=1

function handler
{
    exit $SCRIPT_FAILURE
}

trap handler ERR

/usr/bin/aimanifest load /usr/share/auto_install/manifest/default.xml

typeset found
typeset -i disk_num
for ((disk_num = 1; disk_num <= $SI_NUMDISKS; disk_num++)) ; do
    eval curr_disk="$SI_DISKNAME_${disk_num}"
    eval curr_disk_size="$SI_DISKSIZE_${disk_num}"
    if [[ $curr_disk_size -ge "512000" ]] ; then
        found=$curr_disk
        /usr/bin/aimanifest add \
            /auto_install/ai_instance/target/disk/disk_name@name $found
        break
    fi
done

if [[ -z $found ]] ; then
    exit $SCRIPT_FAILURE
fi

exit $SCRIPT_SUCCESS
```

#### 例 65 システム構成プロファイルの追加

各 AI クライアントのシステム構成を変更する必要がある場合があります。派生マニフェストスクリプトを構成してプロファイルを作成すれば、各 AI クライアントの AI サーバーに個別のシステム構成プロファイルを作成する必要はありません。インストールサービスが使用できるようにするには、このプロファイルを `/system/volatile/profile` に格納する必要があります。この例では、AI クライアントが再構成されるときに、ローカルのデフォルトルーターの設定が使用されています。

```
ROUTER-CONFIG=/system/volatile/profile/router-config.xml
ROUTER=`netstat -rn | grep "^default" | awk '{print $2}'`

cat<<EOF>${ROUTER-CONFIG}
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="router">
  <service name="network/install" version="1" type="service">
    <instance name="default" enabled="true">
      <property_group name="install_ipv4_interface" type="application">
        <propval name="default_route" type="net_address_v4" value="${ROUTER}"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

```

        </property_group>
    </instance>
</service>
</service_bundle>
EOF

```

#### 例 66 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ています。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、変更するパッケージを特定できないため、あいまいエラーが発生します。この例の最初の `set` サブコマンドは `add` サブコマンドにするべきでした。
2. この例の2番目の `set` サブコマンドでは、値 `pkg:/driver/pcmcia` を持つ要素 `name` が `@` 記号を前に付けて指定されています。属性値は `@` 記号を前に付けて指定しますが、要素値はそのようにしません。
3. 値 `pkg:/driver/pcmcia` は引用符で囲むようにしてください。スラッシュなどの特殊文字が含まれる値には引用符を付ける必要があります。

この例の2つの `set` 行は、次の行で置き換えるようにしてください。

```

/usr/bin/aimanifest add \
    software[@type="IPS"]/software_data@action uninstall
/usr/bin/aimanifest add \
    software/software_data[@action=uninstall]/name pkg:/driver/pcmcia

```

これらの2つの `add` サブコマンドによって、作成中のマニフェストの `software` セクションの最後に次の行が追加されます。

```
<software_data action="uninstall">
  <name>pkg:/driver/pcmcia</name>
</software_data>
```

## 派生マニフェストスクリプトのテスト

派生マニフェストスクリプトをテストするには、AI インストール環境に似ている環境でスクリプトを実行します。

1. 変更するスクリプト用のベース AI マニフェストを設定します。

- a. スクリプト内の最初の `aimanifest` コマンドが `aimanifest load` コマンドであることを確認します。読み込まれているファイルに、ターゲットインストールサービスの AI マニフェストの検証に使用する適切な DTD を指定した `<!DOCTYPE>` 定義が含まれていることを確認します。次の例は、この派生マニフェストスクリプトが追加されるインストールサービスの AI DTD ファイルの指定など、AI マニフェストのために最低限必要なベースマニフェストファイルを示しています

```
<!DOCTYPE auto_install SYSTEM "file:///imagepath/auto_install/ai.dtd.1">
<auto_install/>
```

`imagepath` 引数の値は、次のコマンドによって返されるパスです (`svcname` はこの派生マニフェストスクリプトが追加されるインストールサービスの名前です)。

---

**注記** - AI クライアントでスクリプトの使用を試みる前に、イメージパスをデフォルトのパス `///usr/share` にリセットしてください。

---

```
$ installadm list -v -n svcname| grep Image
```

- 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。

---

**注記** - 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。

---

**注記** - 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. 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7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。

5. `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くために、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。

6. 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。

意図されたシステムが、派生マニフェストスクリプトをテストする AI サーバーやその他のシステムと大きく異なる場合があります。スクリプト内で呼び出すコマンドが使用できない場合や、動作の異なる別のバージョンである場合があります。システムのアーキテクチャーが異なる場合や、ディスクの数とサイズが異なる場合があります。説明したようにテスト環境で環境変数を設定することによって、これらの差異のいくつかに対処できます。

## ▼ インストール環境で派生マニフェストスクリプトをテストする方法

この手順は、対象となるシステムの1つで、すべてのインストール処理を実行することなく、派生マニフェストスクリプトをテストする方法を説明しています。

1. システムで AI イメージをブートします。  
システム上で、AI イメージを「Text Installer and command line」モードでブートします。
2. インストーラの初期メニューから「シェル」を選択します。
3. AI サーバーからスクリプトをコピーします。  
wget または sftp を使用して、AI サーバーからスクリプトをコピーします。
4. スクリプトをデバッグします。  
次のいずれかの方法を使用して、スクリプトをデバッグします。
  - スクリプトを手動で実行します。
  - AI をテストモードで実行します。  
次のコマンドを実行して、AI をテストモードで実行します。

```
$ auto-install -m script -i
```

AI ログファイル /system/volatile/install\_log を調べます。ログファイルに、スクリプトが有効であることを示す次の行が含まれているはずですが。

```
Derived Manifest Module: XML validation completed successfully
```
5. スクリプトをコピーして AI サーバーに戻します。  
変更が行われた場合は、スクリプトをコピーして AI サーバーに戻します。

## インストールサービスへの派生マニフェストスクリプトの追加

XML マニフェストをインストールサービスに追加するのと同じ方法で、派生マニフェストスクリプトを AI インストールサービスに追加します。同じオプションを使用して、インストールの際にこのスクリプトを使ってマニフェストを作成する AI クライアントの選択条件を指定します。また、XML マニフェストを更新するのと同様と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、スクリプトおよび XML マニフェストが表示されます。スクリプトはマニフェストタイプ `derived` として表示されます。スクリプトの内容は、XML マニフェストをエクスポートする場合と同じように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。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ときに、そのスクリプトは検証されません。

意図されたシステムに似ている環境でスクリプトを実行します。詳細な手順については、[172 ページの「派生マニフェストスクリプトのテスト」](#)を参照してください。

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 AI クライアントがこれらのインストール手順を使用するかを定義する条件を指定します。クライアントの選択条件を指定しない場合は、`-d` オプションを使用して、このスクリプトをサービスのデフォルトの AI マニフェストとして追加できます。

```
# installadm create-manifest -n solaris11_3-i386 -f ./mac1.ksh -m mac1 \  
-c mac=BB:AA:AA:AA:AA:AA
```

複数の `-c` オプションまたは 1 つの `-c` ファイルを指定できます。 `set-criteria` サブコマンドも参照してください。クライアント条件の指定方法については、[第9章「AI クライアントへのカスタマイズの割り当て」](#)を参照してください。

`installadm` の `list`、`export`、`create-manifest`、`set-criteria`、`update-manifest`、および `set-service` サブコマンドについては、[107 ページの「インストールサービスの操作」](#)を参照してください。

## AI マニフェストウィザードを使用した AI マニフェストの作成

AI マニフェストウィザードは、AI マニフェストを作成するために使用できる BUI (Browser User Interface) Web アプリケーションであり、XML ファイルを手動で編集する必要はありません。このアプリケーションには、AI サーバーの URL を使用してアクセスします。このウィザードには、ユーザーが AI マニフェストの多くのセクションを構成できる 8 個のメイン画面が含まれています。

## AI マニフェストウィザードのための AI サーバーの構成

次の例は、AI マニフェストウィザードを無効にする方法、およびユーザーがマニフェストファイルを AI サーバーに保存できるようにする方法を示しています。

**例 67** AI マニフェストウィザードの無効化

デフォルトでは、AI サーバーを有効にすると、AI マニフェストウィザードが有効になります。AI マニフェストウィザードを無効にするには、次のコマンドを使用します。

```
# installadm set-server -U
```

**例 68** マニフェストファイルを AI サーバーに保存できるようにする

デフォルトでは、AI マニフェストウィザードによって作成された AI マニフェストは AI サーバーに保存できません。ファイルはユーザーのデスクトップに保存する必要があります。ファイルを AI サーバーに保存できるようにするには、次のコマンドを使用します。

```
# installadm set-server -z
```

マニフェストを保存すると、ファイルは `/var/ai/wizard-manifest/` に格納されます。その後、`installadm` コマンドを実行して、このマニフェストをインストールサービスに関連付けることができます。

## ▼ AI マニフェストウィザードを使用して AI マニフェストを作成する方法

このウィザードは、AI サーバーで `/usr/bin/ai-wizard` を実行して開始することもできます。

**始める前に** AI マニフェストウィザードを使用して作成されたマニフェストを追加しやすくするには、生成されたマニフェストを AI サーバー上の一時的な場所に保存する機能を有効にします。詳細については、[例68「マニフェストファイルを AI サーバーに保存できるようにする」](#)を参照してください。

### 1. AI マニフェストウィザードを開始します。

このアプリケーションには、AI サーバーの URL を使用してアクセスします。デフォルトでは、`ai-server` という名前の AI サーバーの URL は `http://ai-server.domain:5555` です。

### 2. 「ようこそ」画面で、マニフェストに関連付けるサービスを識別します。

この画面には、構成されているすべてのインストールサービス、および各インストールサービスのステータスとクライアントアーキテクチャーが一覧表示されます。このリストの最初の項目は AI サーバー自体に関連しており、常に表示されます。サービスが 1 つも構成されていない場合は、この項目を選択することによってマニフェストを作成できます。AI マニフェストに関連付けるインストールサービスを選択して、「開始」をクリックします。

### 3. 「はじめに」画面で、マニフェスト名とターゲットを選択します。

AI マニフェストの名前を入力するか、`default` を使用することを選択します。また、AI マニフェストが大域ゾーン用または非大域ゾーン用のいずれであるかを選択します。この最後の値はターゲットと呼ばれます。「次へ」をクリックして続行します。

**4. 「ルートプール」画面で、ルートプールに関する情報を入力します。**

この画面では、ルートプール名とブート環境名の設定、ルートプールをミラー化するかどうかの選択、およびスワップデバイスとダンプデバイスの構成パラメータの定義を行うことができます。「次へ」をクリックして続行します。

**5. 「データプール」画面で、追加の ZFS ストレージプールに関する情報を入力します。**

最大 5 個のデータプールに指定できます。各データプールには、プール名とマウントポイントを指定する必要があります。データプールごとに冗長性レベル（「なし」、「ミラー」、「RAID-Z」、「RAID-Z1」、「RAID-Z2」、または「RAID-Z3」）を選択できます。「次へ」をクリックして続行します。

**6. 「ディスク」画面で、ルートプールおよびストレージプールのディスクを割り当てます。**

この画面には、前の画面で定義したすべてのプールが一覧表示されます。各プールに対して次のいずれかを構成できます。

- ディスクのキーワード（「自動」または「ブートディスク」）
- ディスクのプロパティ（「デバイスサイズ」、「デバイスタイプ」など）
- ディスク名（「CTD 名」、「ボリューム識別子」など）

これらのプロパティを一度に複数構成する必要がある場合は、ウィザードを使用するときにそれらのいずれかを選択して追加し、マニフェストが作成されたあとに、ファイルを編集するか、`aimanifest` コマンドを使用してほかのプロパティを追加します。「次へ」をクリックして続行します。

**7. 「リポジトリ」画面で、IPS リポジトリを定義します。**

デフォルトの Oracle Solaris サポートリポジトリおよびリリースリポジトリは自動的に定義されます。また、AI クライアントに追加するパッケージが含まれている最大 5 個の追加のリポジトリに関する情報を入力できます。各リポジトリには、リポジトリ名と起点 URI が必要となります。

「詳細を追加」をクリックして、リポジトリごとに SSL 証明書ファイル、SSL 鍵ファイル、およびリポジトリのバックアップ起点 URI を示すこともできます。「次へ」をクリックして続行します。

**8. 「ソフトウェア」画面で、インストールするソフトウェアパッケージを選択します。**

必要に応じて、大規模サーバー、小規模サーバー、およびデスクトップグループのパッケージをインストールすることを選択できます。また、AI クライアントの作成時に追加する必要がある追加のパッケージまたはカスタムパッケージのパッケージ FMRI を追加できます。「次へ」をクリックして続行します。

9. 「ゾーン」画面で、ゾーン名およびゾーン構成ファイルの URI を定義します。  
AI クライアントの作成時に追加する各ゾーンのゾーン名とゾーン構成ファイルの URI を入力します。「次へ」をクリックして続行します。
10. 「確認」画面で、入力した情報を確認し、「保存」をクリックしてマニフェストを作成します。  
サーバー側に保存できるように AI サーバーが構成されている場合、マニフェストは AI サーバーに保存されますが、ファイルをローカルに保存するオプションもあります。そのように構成されていない場合、マニフェストはローカルにのみ保存できます。

## installadm コマンドを使用した AI マニフェストの作成

Oracle Solaris 11.3 リリースでは、`installadm create-manifest` および `installadm update-manifest` コマンドが、作成または更新されるマニフェストをユーザーが編集できるように拡張されています。さらに、インストールサービスで既存のマニフェストから新しいマニフェストを作成できるようになりました。次に例を示します。

`create-manifest -n servicename -m new-manifest` は、妥当なデフォルト値に基づいた `installadm` 対話型モードでマニフェストを表示します。

`create-manifest -n servicename -f file` は、指定された `file` の内容を使用してマニフェストを作成します。

`create-manifest -n servicename -f file -e` は、指定された `file` の内容を使用してマニフェストを編集できるようにします。このファイルが派生マニフェストスクリプトの場合は、エディタが表示されます。選択されるエディタは、`VISUAL` または `EDITOR` 環境変数によって定義されます。どちらの変数も定義されていない場合は、`vi` エディタが使用されます。このファイルが派生マニフェストスクリプトでない場合は、`installadm` 対話型モードでマニフェストを編集します。

`create-manifest -n servicename -M manifest -m new-manifest` は、既存の `manifest` の内容を使用して新しいマニフェストを作成します。

`create-manifest -n servicename -M manifest -e -m new-manifest` は、既存の `manifest` の内容を使用して新しいマニフェストを作成します。既存のマニフェストが派生マニフェストスクリプトの場合は、エディタが表示されます。選択されるエディタは、`VISUAL` または `EDITOR` 環境変数によって定義されます。どちらの変数も定義されていない場合は、`vi` エディタが使用されます。既存のマニフェストが派生マニフェストスクリプトでない場合は、`installadm` 対話型モードでマニフェストを編集します。

`update-manifest -n servicename -m new-manifest` は、マニフェストを更新します。このマニフェストが派生マニフェストスクリプトの場合は、エディタが表示されます。選択されるエディタは、`VISUAL` または `EDITOR` 環境変数によって定義されます。どちらの変数も定義されていない場合は、`vi` エディタが使用されます。こ

のマニフェストが派生マニフェストスクリプトでない場合は、`installadm` 対話型モードでマニフェストを編集します。

`update-manifest -n servicename -f file -m manifest` は、指定された `file` の内容を使用してマニフェストを更新します。

`update-manifest -n servicename -f file -m manifest -e` を使用すると、`file` の内容を使用して更新されたマニフェストを編集できます。このファイルが派生マニフェストスクリプトの場合は、エディタが表示されます。選択されるエディタは、`VISUAL` または `EDITOR` 環境変数によって定義されます。どちらの変数も定義されていない場合は、`vi` エディタが使用されます。このファイルが `xml` ファイルの場合は、`installadm` 対話型モードでマニフェストを編集します。

## installadm 対話型エディタコマンド

コマンド行で `-e` オプションを使用すると、対話型エディタモードに移行できます。対話型編集モードになると、次のコマンドを使用できます。

<code>add</code>	オブジェクトまたはプロパティを追加します。 <code>-w</code> オプションを追加すると、コマンドから、オブジェクトやプロパティの入力が求められます。下の <code>walk</code> の説明を参照してください。
<code>cancel</code>	現在のレベルで行なった変更を破棄し、1つ上のレベルに移動します。
<code>commit</code>	最上位レベルで、変更を検証し、マニフェストを保存して編集を続行します。
<code>delete</code>	オブジェクトまたはプロパティを削除します。
<code>end</code>	現在のレベルで行われた変更を検証し、エラーが発生しなかった場合は1つ上のレベルに移動します。
<code>exit</code>	変更が検証されたあとにマニフェストを保存して終了するか、コミットされていない変更を保存しないで終了するか、または編集を続行するか確認するよう要求します。
<code>info</code>	1つ下のレベルまでのすべてのオブジェクトとプロパティを表示します。
<code>move</code>	複数のオブジェクト (ソフトウェアパブリッシャーなど) の順序を変更します。
<code>select</code>	オブジェクトを選択し、そのレベルに移動します。
<code>set</code>	プロパティの値を設定します。

walk 選択されたオブジェクトの設定可能な各プロパティおよび設定可能なサブオブジェクトまたはサブプロパティ (存在する場合) の入力を要求します。

## installadm 対話型編集モードの使用の例

[installadm\(1M\)](#) のマニュアルページには、対話型モードを示すいくつかの例が記載されています。次の各セクションでは、追加の例を示します。

**例 69** walk モードでの installadm によるマニフェストエントリの作成

この例では、test という名前のマニフェストに 2 番目のパブリッシャーが追加されます。

```
# installadm create-manifest -n default-sparc -m test
installadm:test> select software
installadm:test:software> info
  type: IPS
  name: <not specified>
  facet[1]:
  ...
  facet[20]:
    name: facet.locale.zh_TW
    value: true
  publisher:
    name: solaris
    key: <not specified>
    cert: <not specified>
    ca-cert: <not specified>
    origin: http://pkg.oracle.com/solaris/release
    mirror: <not specified>
    cmd-options: <not specified>
  pkg-list:
    action: install
    name: pkg:/entire@0.5.11-0.175.3
    name: pkg:/group/system/solaris-large-server
    reject: <not specified>
installadm:test:software> add -w publisher
*** To terminate walk, use Ctrl-D ***
  name [<not specified>]: firstboot
  key [<not specified>]:
  cert [<not specified>]:
  ca-cert [<not specified>]:
  origin [<not specified>]: file:///net/host1/export/firstbootrepo
  origin [<not specified>]:
  mirror [<not specified>]:
  cmd-options [<not specified>]:
installadm:test:software:publisher> info
  name: firstboot
  key: <not specified>
  cert: <not specified>
  ca-cert: <not specified>
  origin: file:///net/host1/export/firstbootrepo
  mirror: <not specified>
  cmd-options: <not specified>
installadm:test:software:publisher> end
```

```
installadm:test:software> exit
1. Save manifest and exit
2. Exit without saving uncommitted changes
3. Continue editing
Please select choice: 1
Created Manifest: 'test'
```

**例 70** installadm によるオブジェクトの順序の変更

この例では、パブリッシャーの順序が切り替えられます。

```
# installadm update-manifest -n default-sparc -m test
installadm:test> select software
installadm:test:software> info
  type: IPS
  name: <not specified>
  facet[1]:
  ...
  facet[20]:
    name: facet.locale.zh_TW
    value: true
  publisher[1]:
    name: solaris
    key: <not specified>
    cert: <not specified>
    ca-cert: <not specified>
    origin: http://pkg.oracle.com/solaris/release
    mirror: <not specified>
    cmd-options: <not specified>
  publisher[2]:
    name: firstboot
    key: <not specified>
    cert: <not specified>
    ca-cert: <not specified>
    origin: http://example.com/solaris/mybuild
    mirror: <not specified>
    cmd-options: <not specified>
  pkg-list:
    action: install
    name: pkg:/entire@0.5.11-0.175.3
    name: pkg:/group/system/solaris-large-server
    reject: <not specified>
installadm:test:software> move publisher 2 1
installadm:test:software> info
  type: IPS
  name: <not specified>
  ...
  publisher[1]:
    name: firstboot
    key: <not specified>
    cert: <not specified>
    ca-cert: <not specified>
    origin: http://example.com/solaris/mybuild
    mirror: <not specified>
    cmd-options: <not specified>
  publisher[2]:
    name: solaris
    key: <not specified>
    cert: <not specified>
    ca-cert: <not specified>
    origin: http://pkg.oracle.com/solaris/release
    mirror: <not specified>
```

```
cmd-options: <not specified>
pkg-list:
  action: install
  name: pkg:/entire@0.5.11-0.175.3
  name: pkg:/group/system/solaris-large-server
  reject: <not specified>
installadm:test:software> exit
1. Save manifest and exit
2. Exit without saving uncommitted changes
3. Continue editing
Please select choice: 1
Updated Manifest: 'test'
```

## AI マニフェストの例

このセクションの例は、完成した AI マニフェストが規定された結果を実現するために持っている必要がある XML 要素を示しています。これらのマニフェストを作成するには、XML を直接編集するか、または派生マニフェストスクリプトを使用します。

このセクションに示されているすべてのマニフェストは、`/image-path/auto_install/manifest/default.xml`にあるサンプルのデフォルト XML ファイルに必要な変更を加えたものに基づいています。説明を簡単にするため、`software` 要素内の `destination` 要素は省略しています。



---

**注意** - `entire` パッケージはシステムパッケージのバージョンを同じビルドに制約するため、各マニフェストにこのパッケージが含まれている必要があります。適切なシステム更新と正しいパッケージ選択は、このパッケージの存在に依存します。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。このパッケージを削除すると、サポートされないシステムになります。

---

次の例が含まれます。

- [183 ページの「iSCSI ターゲットデバイスの指定」](#)
- [183 ページの「AI マニフェストでのルートプールとブートプールの指定」](#)
- [184 ページの「RAID 構成の指定」](#)
- [185 ページの「SVR4 パッケージのインストール」](#)
- [186 ページの「複数の SVR4 パッケージのインストール」](#)
- [186 ページの「既存のディスクスライスまたはパーティションの再使用」](#)
- [187 ページの「SSL クライアント認証を使用した統合アーカイブへのアクセス」](#)
- [187 ページの「HTTP 認証トークンを使用した統合アーカイブへのアクセス」](#)
- [188 ページの「セキュアな IPS リポジトリへのアクセス」](#)
- [188 ページの「solaris-minimal-server パッケージインストール時のローカルの変更」](#)

## iSCSI ターゲットデバイスの指定

この例では、インストールのターゲットは iSCSI デバイスです。target 要素内の disk 要素内の iscsi 要素を使用します。disk 要素の whole\_disk 属性は、iSCSI ディスクで一般的な true に設定されています。target\_name、target\_lun、および target\_ip 属性の説明については、[ai\\_manifest\(4\)](#) のマニュアルページを参照してください。

```
<auto_install>
  <ai_instance name="default">
    <target>
      <disk whole_disk="true">
        <iscsi target_name="iqn.1986-03.com.sun:02:1234567890abcdef" \
          target_lun="1" target_ip="129.158.144.200"/>
      </disk>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@0.5.11-0.175.2</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

## AI マニフェストでのルートプールとブートプールの指定

次のサンプル AI マニフェストでは、iSCSI デバイスをルートプールとして定義し、ブートプールを mybpool という名前に構成します。ブートプールのデフォルトの名前は bpool です。ブートプールの説明にはディスクデバイスが含まれていないため、専用の eUSB ディスクがデフォルトで使用されます。

```
<auto_install>
  <ai_instance name="default">
    <target>
      <disk in_zpool="rpool" whole_disk="true">
```

```

        <iscsi target_port="326" target_ip="6.6.6.53"\
            target_lun="1">
    </disk>
    <logical>
        <zpool name="rpool" is_root="true">
            <filesystem name="export" mountpoint="/export"/>
            <filesystem name="export/home"/>
            <be name="solaris"/>
        </zpool>
        <zpool name="mybpool" is_boot="true">
        </zpool>
    </logical>
</target>
<software type="IPS">
    <source>
        <publisher name="solaris">
            <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
    </source>
    <software_data action="install">
        <name>pkg:/entire@0.5.11-0.175.2</name>
        <name>pkg:/group/system/solaris-large-server</name>
    </software_data>
</software>
</ai_instance>
</auto_install>

```

## RAID 構成の指定

この例では、2 台のディスク `c0t0d0` および `c0t1d0` を使用した RAID 構成を指定しています。このマニフェストは、[例62 「特定のサイズのディスクが2つ以上ある場合のミラー化構成の指定」](#) で示したミラー化構成のマニフェストに似ています。2つのマニフェストの違いの1つは、`redundancy` 属性の値が `mirror` ではなく `raidz` であることです。冗長性タイプについては、[zpool\(1M\)](#) のマニュアルページを参照してください。もう1つの違いは、ZFS プールの名前が `rpool` ではないことです (`rpool` はルートプールを意味するためです)。デフォルトでは、`zpool` 要素の `is_root` 属性の値は `false` であるため、この例では割り当てを省略できます。ルートプールが指定されていないため、このインストールの初期ユーザーを構成しないでください。

```

<auto_install>
  <ai_instance name="default">
    <target>
      <disk in_vdev="raid_vdev" in_zpool="raidpool" whole_disk="true">
        <disk_name name="c0t0d0" name_type="ctd"/>
      </disk>
      <disk in_vdev="raid_vdev" in_zpool="raidpool" whole_disk="true">
        <disk_name name="c0t1d0" name_type="ctd"/>
      </disk>
    <logical>
      <zpool name="raidpool" is_root="false">
        <vdev name="raid_vdev" redundancy="raidz"/>
      </zpool>
    </logical>
  </target>
  <software type="IPS">
    <source>

```

```

    <publisher name="solaris">
      <origin name="http://pkg.oracle.com/solaris/release"/>
    </publisher>
  </source>
  <software_data action="install">
    <name>pkg:/entire@0.5.11-0.175.2</name>
    <name>pkg:/group/system/solaris-large-server</name>
  </software_data>
</software>
</ai_instance>
</auto_install>

```

## SVR4 パッケージのインストール

この例は、SVR4 パッケージをインストールする方法を示しています。SVR4 パッケージの名前は、`type` が SVR4 である `software` 要素内で指定する必要があります。パブリッシャーの起点の `name` 属性の値は、SVR4 パッケージのサブディレクトリを含むディレクトリまたは SVR4 パッケージのデータストリームファイルです。SVR4 パッケージサブディレクトリのこの起点名は、ファイルのフルディレクトリパスまたはファイルの URI にすることができます。SVR4 パッケージデータストリームファイルのこの起点名は、ファイルのフルディレクトリパス、ファイルの URI、または HTTP URI にすることができます。

**ヒント** - インストールの一部として、ユーザー入力が必要なパッケージをインストールしないでください。

```

<auto_install>
  <ai_instance name="default">
    <target>
      <logical>
        <zpool name="rpool" is_root="true">
          <filesystem name="export" mountpoint="/export"/>
          <filesystem name="export/home"/>
          <be name="solaris"/>
        </zpool>
      </logical>
    </target>
    <software type="IPS">
      <source>
        <publisher name="solaris">
          <origin name="http://pkg.oracle.com/solaris/release"/>
        </publisher>
      </source>
      <software_data action="install">
        <name>pkg:/entire@0.5.11-0.175.2</name>
        <name>pkg:/group/system/solaris-large-server</name>
      </software_data>
    </software>
    <software type="SVR4">
      <source>
        <publisher>
          <origin name="/net/host2/usr/dist"/>

```

```
    </publisher>
  </source>
  <software_data>
    <name>SUNWpackage</name>
  </software_data>
</software>
</ai_instance>
</auto_install>
```

## 複数の SVR4 パッケージのインストール

複数の SVR4 パッケージをインストールするには、次のように各パッケージにソフトウェアタグを指定する必要があります。

```
<software type="SVR4">
  <source>
    <publisher>
      <origin name="/net/192.168.56.2/svr4/app1.pkg"/>
    </publisher>
  </source>
  <software_data>
    <name>application1</name>
  </software_data>
</software>
<software type="SVR4">
  <source>
    <publisher>
      <origin name="/net/192.168.56.2/svr4/app2.pkg"/>
    </publisher>
  </source>
  <software_data>
    <name>application2</name>
  </software_data>
</software>
```

## 既存のディスクスライスまたはパーティションの再利用

この例は、SPARC クライアントの既存のディスクスライスを使用するように指定する方法を示しています。ディスクスライスの場合は、既存のスライスの設定 (`start_sector` および `size`) が再使用されます。構成処理は、Solaris のバージョンがすでにインストールされているかどうかを確認するためにスライスを検索しません。

```
<disk>
  <disk_name name="c1t0d0" name_type="ctd"/>
  <slice name="0" action="use_existing" force="true" in_zpool="rpool">
</disk>
```

次の例は、AI 処理中に x86 クライアントでディスク上にある既存のパーティションを再使用するように指定する方法を示しています。パーティションの場合は、指定されたスライスの既存の設定が再使用されます。この場合、再使用されるパーティションは構成処理中に自動的に判断されます。

```
<partition action="use_existing_solaris2">
  <slice action="use_existing" name="0" force="true"/>
</partition>
```

## SSL クライアント認証を使用した統合アーカイブへのアクセス

SSL を使用して統合アーカイブにアクセスするには、AI マニフェストに `credentials` 要素を含めて、鍵、証明書、および CA 証明書情報を指定します。この例では、統合アーカイブの名前は `sslarchive.ua` です。

```
<software type="ARCHIVE">
  <source>
    <file uri="https://example.com/sslarchive.ua"/>
      <credentials>
        <key src="file://root/key.pem"/>
        <cert src="file://root/cert.pem"/>
        <cacert src="file://root/cacert.pem"/>
      </credentials>
    </file>
  </source>
</software>
```

AI サーバー上での SSL の設定については、[118 ページの「自動インストールのセキュリティの向上」](#)を参照してください。

## HTTP 認証トークンを使用した統合アーカイブへのアクセス

統合アーカイブへのアクセス時に HTTP 認証トークンを使用するには、`credentials` 要素にそのトークンを追加します。また、統合アーカイブを定義する行も追加します。

```
<software type="ARCHIVE">
  <source>
    <file uri="http://example.com/httparchive.ua"/>
      <credentials>
        <http_auth_token>my-specifically-granted-auth-token</http_auth_token>
      </credentials>
    </file>
  </source>
</software>
```

```

        </credentials>
      </file>
    </source>
  </software>

```

## セキュアな IPS リポジトリへのアクセス

セキュアな IPS リポジトリにアクセスするには、そのリポジトリの鍵と証明書を指定する必要があります。

```

<publisher name="solaris">
  <origin name="https://pkg.oracle.com/solaris/support/" />
  <credentials>
    <key src="/var/pkg/ssl/Oracle_Solaris_11_Support.key.pem" />
    <cert src="/var/pkg/ssl/Oracle_Solaris_11_Support.certificate.pem" />
  </credentials>
</publisher>

```

## solaris-minimal-server パッケージインストール時のロケールの変更

solaris-minimal-server グループをインストールする場合、C/POSIX ロケールだけがシステムにインストールされます。UTF-8 ロケールのいずれかを使用する場合は、system/locale パッケージをインストールし、IPS の facet.locale 属性設定を必要に応じてカスタマイズする必要があります。facet.locale.\* を false に設定しなかった場合、使用可能な UTF-8 ロケールがすべてインストールされます。たとえば、システム構成プロファイル内でシステムのデフォルトロケールが en\_US.UTF-8 に設定されている場合、en\_US ロケールを追加するには、次の行をマニフェストに追加すべきです。

```

<software type="IPS">
  <destination>
    <image>
      <!-- Specify locales to install -->
      <facet set="false">facet.locale.*</facet>
      <facet set="true">facet.locale.en</facet>
      <facet set="true">facet.locale.en_US</facet>
    </image>
  </destination>
  ...
  <software_data action="install">
    <name>pkg:/entire@5.12-5.12.0</name>
    <name>pkg:/group/system/solaris-minimal-server</name>
    <name>pkg:/system/locale</name>
  </software_data>
</software>

```

## solaris-minimal-server パッケージでマニュアルページパッケージをインストールしない

solaris-minimal-server グループには man コマンドは含まれていませんが、デフォルトでマニュアルページが含まれています。マニュアルページのインストールを防ぐには、次のように doc.man ファセットを false に設定する必要があります。

```
<software type="IPS">
  <destination>
    <image>
      <!-- Specify that man pages should not be installed -->
      <facet set="false">doc.man</facet>
    </image>
  </destination>
  ...
  <software_data action="install">
    <name>pkg:/entire@5.12-5.12.0</name>
    <name>pkg:/group/system/solaris-minimal-server</name>
    <name>pkg:/system/locale</name>
  </software_data>
</software>
```

## パッケージグループのインストール時における特定のパッケージの追加

solaris-minimal-server グループには man コマンドは含まれていませんが、デフォルトでマニュアルページが含まれています。インストール時に man コマンドをシステムに追加するには、次のように /text/doctools パッケージを含める必要があります。

```
<software type="IPS">
  ...
  <software_data action="install">
    <name>pkg:/entire@5.12-5.12.0</name>
    <name>pkg:/group/system/solaris-minimal-server</name>
    <name>pkg:/text/doctools</name>
  </software_data>
</software>
```

## デフォルトの AI マニフェスト

インストールサービスのデフォルトの AI マニフェストは派生マニフェストです。インストールサービスを作成すると、orig\_default というデフォルトのマニフェストがサービスに作成されます。インストールイメージが異なると、作成されるデフォルト

トのマニフェストがわずかに異なる場合があります。このファイルの XML の内容に関する詳細は、[ai\\_manifest\(4\)](#) のマニュアルページを参照してください。

## AI クライアントのシステム構成パラメータの定義

---

この章では、インストール後に AI クライアント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、サービス管理機能 (SMF) プロパティを使用して指定できます。

### 構成プロファイルの指定

システム構成プロファイルは、システムの構成を一連の構成パラメータとして SMF プロファイルの形式で指定します。システム構成プロファイルは、適切な SMF サービスに対して SMF プロパティを設定します。

システム構成プロファイルは、AI インストール後のシステムの初回ブート時に適用されます。特定の構成領域を担当する SMF サービスは、SMF プロパティを処理し、それに応じてシステムを構成します。

各 AI 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 AI 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 AI クライアントとほかの多くのクライアントに、より広範囲に適用できるプロパティ値を設定するほかのプロファイルを割り当てることができます。

特定の AI 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、インストール後の初回ブート時に、そのシステム上で対話型の構成ツールが開きます。この構成ツールを対話形式で使用する方法については、[73 ページの「システムの再構成」](#)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、77 ページの「[SCI ツールを使用したシステム構成プロファイルの作成](#)」を参照してください。

- 194 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 208 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を指定した場合、その指定は無視されません。

特定のプロパティを何度も指定しないでください。

- 派生マニフェストスクリプトは、スクリプトの実行時にシステム構成プロファイルを作成できます。[例65「システム構成プロファイルの追加](#)」を参照してください。

システム構成プロファイルでは、プロパティと属性の値を 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仕様を使って、プロファイルがインストールサービスに追加される前に値を明示的に入力できます。
- システム構成プロファイルには、そのプロファイルを使用してシステムをインストールするときに有効な値に置き換えられる変数を含めることができます。

## 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する必要があります。[145 ページの「システム構成プロファイルの検証](#)」サブコマンドの詳細は、Validating a System Configuration Profileを参照してください。

## インストールサービスへのシステム構成プロファイルの追加

システム構成プロファイルをインストールサービスに追加するには、`installadm create-profile` コマンドを使用します。`create-profile` サブコマンドは、プロファイルを検証してから、それらをインストールサービスに追加します。

適切な AI 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 AI クライアントがこのプロファイルを使用します。

このプロファイルが AI クライアントに適用される環境を、`-e` オプションで指定します。この環境は、インストール環境で使用されるプロファイルの場合は `install`、インストール後に使用されるプロファイルの場合は `system`、どちらの環境でも使用できるプロファイルの場合は `all` のいずれかに設定できます。

単一の AI クライアントが複数のシステム構成プロファイルに一致し、複数のシステム構成プロファイルを使用する場合があります。特定のプロパティが何度も指定されるような一連のプロファイルを使用する AI クライアントがないようにしてください。AI クライアントが特定のプロパティの複数の指定を受け取る場合は、そのプロパティの値が各指定で同じであっても、構成している SMF サービスの動作は未定義です。

AI 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、インストール後の最初のブート時に、対話型構成ツールがそのシステムで開きます。

特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と環境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。

このプロファイルがインストール環境で使用されるようにプロファイルの名前を変更したり、プロファイルの環境値を設定したりするには、`installadm set-profile` コマンドを使用します。

インストールサービスに追加されているプロファイルの内容のコピーを取得するには、`installadm export` コマンドを使用します。そのコピーを変更して別のプロファイルを作成できます。

インストールサービスにすでに追加されているプロファイルの内容を置き換えるには、`installadm update-profile` コマンドを使用します。

これらのサブコマンドの詳細は、[107 ページの「インストールサービスの操作」](#) および [installadm\(1M\)](#) のマニュアルページを参照してください。

## システム構成プロファイル内での構成の指定

構成可能なシステム特性の構成は、SMF 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、キーボード配列、端末タイプ、IPv4 ネットワークインタフェース (静的または DHCP) とデフォルトルート、および IPv6 ネットワークインタフェース (静的または `addrconf`) とデフォルトルート、およびネームサービス (ネームサーバーリスト、検索リスト、ドメイン) を構成できます。該当しないサービスまたはプロパティーを指定した場合、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ってない場合は、`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と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] [property-group/property]
```

プロパティーグループまたは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、または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(5).
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(5).
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(5).
    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
        Overwork configuration for network database lookups.
...
$ svccfg -s system/config-user describe root_account
root_account          application
root_account/expire  astring
root_account/password astring
root_account/read_authorization astring          solaris.smf.read.system-config
root_account/stability astring          Evolving
root_account/type     astring

```

## root アカウントとユーザーアカウントの構成

root ユーザーと初期ユーザーを構成する有効なプロファイルを生成するには、次の `sysconfig create-profile` コマンドを `users` グループとともに入力します。

```
# sysconfig create-profile -g users -o sc_users.xml
```

`svc:/system/config-user` SMF サービスは、ユーザーアカウントと `root` アカウントを構成します。このサービスは、2つのプロパティグループを認識します。

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

---

**ヒント** - `pwhash` コマンドを使用して、暗号化されたパスワード、つまりパスワードハッシュを生成します。結果のハッシュを任意のアカウントに提供できます。

---

## root アカウムの構成

root\_account プロパティグループには、次の表に示すプロパティが含まれています。

表 8 root\_account プロパティグループのプロパティ

プロパティ	タイプ	必須	説明
password	astring	必須	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	オプション	アカウントタイプ: normal または role。
expire	string	オプション	ログインの有効期限。0 (ゼロ) に設定した場合、ユーザーは次のログイン時に root パスワードを変更するように要求されます。

例 71 パスワードが必ず期限切れになる root アカウムの構成

```
<service name="system/config-user" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="root_account" type="application">
      <propval name="password" value="encrypted_password"/>
      <propval name="type" value="normal"/>
      <propval name="expire" value="0"/>
    </property_group>
  </instance>
</service>
```

## ユーザーアカウントの構成

このセクションでは、次の内容について説明します。

- [196 ページの「オートマウントに依存しないユーザーアカウントの作成」](#)
- [197 ページの「ユーザーアカウントプロパティ」](#)
- [198 ページの「複数の初期ユーザーの構成」](#)

### オートマウントに依存しないユーザーアカウントの作成

デフォルトでは、初期ユーザーアカウントが作成されると、ホームディレクトリはオートマウントによって管理され、/home/login ディレクトリの下にアクセスされません。オートマウントに依存せずに初期ユーザーアカウントを作成するには、構成プロファイル内の user\_account/autohome プロパティを空の文字列 ("") に設定します。

`user_account/autohome` プロパティを空の文字列に設定すると、次の効果があります。

- `/etc/passwd` ファイル内のホームディレクトリエントリが `/home/login` ではなくホーム ZFS データセットのマウントポイントに設定されます。ホーム ZFS データセットのデフォルトのマウントポイントは `/export/home/login` です。
- `/etc/auto_home` ファイルにマッピングエントリは追加されません。

## ユーザーアカウントプロパティ

`user_account` プロパティグループには、次の表に示すプロパティが含まれています。

表 9 `user_account` プロパティグループのプロパティ

プロパティ	タイプ	必須	説明
<code>login</code>	<code>astring</code>	<code>required</code>	ユーザーのログイン。
<code>password</code>	<code>astring</code>	<code>required</code>	暗号化されたユーザーパスワード。
<code>description</code>	<code>astring</code>	<code>optional</code>	通常はユーザーのフルネーム。
<code>shell</code>	<code>astring</code>	<code>optional</code>	ユーザーのログインシェルとして使用するプログラムのフルパス名を指定します。
<code>uid</code>	<code>count</code>	<code>optional</code>	新しいユーザーの UID。UID はデフォルトで、現在割り当てられているもっとも大きな番号の次の番号に設定されます。
<code>gid</code>	<code>count</code>	<code>optional</code>	ユーザーのプライマリグループのメンバーシップ。デフォルトの GID は 10 (staff グループ) です。
<code>type</code>	<code>astring</code>	<code>optional</code>	アカウントタイプ: <code>normal</code> または <code>role</code> 。
<code>profiles</code>	<code>astring</code>	<code>optional</code>	<code>prof_attr(4)</code> のマニュアルページで定義された、コンマで区切られた 1 つ以上の実行プロファイル。
<code>roles</code>	<code>astring</code>	<code>optional</code>	<code>user_attr(4)</code> のマニュアルページで定義された、コンマで区切られた 1 つ以上の役割。
<code>sudoers</code>	<code>astring</code>	<code>optional</code>	<code>/etc/sudoers.d/svc-system-config-user</code> ファイルに <code>login</code> とともに追加されるエントリ。
<code>expire</code>	<code>astring</code>	<code>optional</code>	ログインの有効期限。0 (ゼロ) に設定した場合、ユーザーは次のログイン時にパスワードを変更するように要求されます。
<code>home_zfs_dataset</code>	<code>astring</code>	<code>optional</code>	ユーザーのホームディレクトリの ZFS データセット。
<code>home_mountpoint</code>	<code>astring</code>	<code>optional</code>	ユーザーのホームディレクトリのマウントポイント。
<code>autohome</code>	<code>astring</code>	<code>optional</code>	ユーザーの自動ホームディレクトリのマウントポイント。デフォルト値は、ローカルホストの <code>/export/home/login</code> ディレクトリです。 <code>autohome</code> プロパティが空の文字列 ("") に設定されると、オートマウントに依存せずにユーザーアカウントが作成されます。

プロパティ設定は、構成されるユーザーの `/etc/auto_home` ファイルに格納されます。

**例 72** パスワードを使用しないアカウントの構成

この例では、パスワードを NP に設定することにより、特権ユーザーとして su コマンドを使用するか鍵ベース認証で ssh コマンドを使用する場合にのみ、アカウントにアクセスできるようにしています。

```
<service name="system/config-user" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="user_account">
      <propval name="login" value="jack"/>
      <propval name="password" value="NP"/>
      <propval name="description" value="default_user"/>
      <propval name="shell" value="/usr/bin/bash"/>
      <propval name="gid" value="10"/>
      <propval name="uid" value="1001"/>
      <propval name="type" value="normal"/>
      <propval name="roles" value="root"/>
      <propval name="profiles" value="System Administrator"/>
    </property_group>
  </instance>
</service>
```

**複数の初期ユーザーの構成**

新規にインストールされたシステムに複数のユーザーを構成するには、スクリプト内で useradd コマンドを使用してユーザーを指定します。次に、1 度だけ実行する SMF サービスを使用して初回ブート時にスクリプトを実行します。手順については、[第13章「初回ブート時のカスタムスクリプトの実行」](#)を参照してください。

**SSH 鍵の構成**

ssh\_public\_keys プロパティグループは、事前生成された SSH 鍵を保持します。この鍵は、システムが構成されるときにユーザーの \$HOME/.ssh/authorized\_keys ファイルに書き込まれます。

**例 73** SSH 鍵の構成

```
<property_group name="user_account" type="application">
  <...>
  <property type="astring" name="ssh_public_keys">
    <astring_list>
      <value_node value='[<options>] <key-type> <base64-encoding-key> [<comment>]'\
      <value_node value='[<options>] <key-type> <base64-encoding-key> [<comment>]'\
    </astring_list>
  </property>
</property_group>
```

## システム識別情報の設定

システムノード名を構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `identity` グループとともに使用します。

```
# sysconfig create-profile -g identity -o sc_identity.xml
```

`svc:/system/identity:node` SMF サービスは、システムのホスト名を設定します。`node` は `svc:/system/identity` のインスタンスです。

`identity` プロパティグループには、次の表に示すプロパティが含まれています。

表 10 config プロパティグループのプロパティ

プロパティ	タイプ	必須	説明
<code>nodename</code>	<code>astring</code>	オプション	システムのホスト名。
<code>enable_mapping</code>	<code>boolean</code>	オプション	ノード名マッピングを無効にするために使用する値。
<code>loopback</code>	<code>astring</code>	オプション	<code>loopback</code> にマップされるホスト名。

### 例 74 ホスト名の構成

この例では、システムのホスト名を `solaris` に設定しています。

```
<service name="system/identity" version="1" type="service">
  <instance name="node" enabled="true">
    <property_group name="config" type="application">
      <propval name="nodename" value="solaris"/>
    </property_group>
  </instance>
</service>
```

### 例 75 ノード名マッピングの無効化

Oracle Solaris 11 OS または Oracle Solaris 11 アップデートリリースをインストールする場合、デフォルトではシステムノード名はインストールの一部として構成されたインタフェースのループバックまたは IP アドレスにマップされます。このデフォルトのマッピングを無効にするには、次の例に示すように、`enable_mapping` プロパティを `false` に設定します。

```
<service name="system/identity" version="1" type="service">
  <instance name="node" enabled="true">
    <property_group name="config" type="application">
```

```

        <propval name="nodename" value="solaris"/>
        <propval name="enable_mapping" value="false"/>
    </property_group>
</instance>
</service>

```

## タイムゾーンとロケールの設定

タイムゾーンとロケールを構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `location` グループとともに使用します。

```
# sysconfig create-profile -g location -o sc_location.xml
```

`svc:/system/timezone` SMF サービスは、システムのタイムゾーンを設定します。

`timezone` プロパティグループには、次の表に示すプロパティが含まれています。

表 11 `timezone` プロパティグループのプロパティ

プロパティ	タイプ	必須	説明
<code>localtime</code>	<code>astring</code>	オプ ション	システムのタイムゾーン。

### 例 76 `timezone` の構成

この例では、タイムゾーンを中央ヨーロッパ標準時/プラハ、チェコに設定しています。

```

<service name='system/timezone' version='1'>
  <instance name='default' enabled='true'>
    <property_group name='timezone'>
      <propval name='localtime' value='Europe/Prague' />
    </property_group>
  </instance>
</service>

```

`svc:/system/environment:init` SMF サービスは、システムのロケールを設定します。

`environment` プロパティグループは、次の環境変数を定義できます。環境変数については、[environ\(5\)](#) のマニュアルページを参照してください。

表 12 `environment` プロパティグループのプロパティ

環境変数	タイプ	必須	デフォルト値
<code>LC_CTYPE</code>	<code>astring</code>	オプ ション	<code>C</code>

環境変数	タイプ	必須	デフォルト値
LC_NUMERIC	astring	オプション	C
LC_TIME	astring	オプション	C
LC_COLLATE	astring	オプション	C
LC_MONETARY	astring	オプション	C
LC_MESSAGES	astring	オプション	C
LC_ALL	astring	オプション	C
LANG	astring	オプション	C

#### 例 77 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>
```

## 端末タイプとキーボード配列の設定

次の例は、システム構成プロファイル内でコンソールの端末タイプとキーボードレイアウトを設定する方法を示したものです。

#### 例 78 端末タイプの構成

svc:/system/console-login SMF サービスは、端末タイプを構成します。関連する SMF プロパティの定義については、[ttymon\(1M\)](#) のマニュアルページを参照してください。

この例では、端末タイプを vt100 に設定しています。

```
<service name="system/console-login" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="ttymon" type="application">
      <propval name="terminal_type" value="vt100" />
    </property_group>
  </instance>
</service>
```

**例 79** キーボード配列の構成

キーボード配列を構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `keyboard` グループとともに使用します。

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

`svc:/system/keymap` SMF サービスは、キーボード配列を構成します。関連する SMF プロパティの定義については、[kbd\(1\)](#) のマニュアルページを参照してください。

この例では、キーボード配列をチェコ語に設定しています。

```
<service name='system/keymap' version='1' type='service'>
  <instance name='default' enabled='true'>
    <property_group name='keymap' type='system'>
      <propval name='layout' value='Czech' />
    </property_group>
  </instance>
</service>
```

## ネットワークインタフェースの構成

ネットワークを構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `network` グループとともに使用します。このコマンドによって SCI ツールが開始され、インタフェースの構成に必要な情報の入力を求められます。

```
# sysconfig create-profile -g network -o sc_network.xml
```

`svc:/network/install` SMF サービスは、初期の物理ネットワークインタフェースを構成します。このサービスは、システム構成が行われないプロパティ値で最初は無効になっています。

---

**注記** - インストールのターゲットが iSCSI デバイスである場合は、そのインストールのシステム構成プロファイルでそのネットワークインタフェースを構成しないでください。iSCSI ブートの場合は、iSCSI デバイスのネットワークインタフェースがクライアントのブートプロセスの早い段階で構成されます。その同じインタフェースを再度構成すると、インタフェースの `network/install` サービスが保守状態になります。

---

複数のネットワークインタフェースを構成するには、スクリプトにその構成を指定し、1度だけ実行する SMF サービスを使用して初回ブート時にそのスクリプトを実行します。手順やサンプルスクリプトについては、[第13章「初回ブート時のカスタムスクリプトの実行」](#)を参照してください。

`svc:/network/install` サービスは、複数の IPv4 インタフェースと IPv6 インタフェース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をサポートしています。このサービスを使用すると、IPv4 インタフェースおよび IPv6 インタフェースを構成できます。このサービスは、それ自体のプロパティおよび `ipadm` コマンド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、このサービスはそれ自体のプロパティおよび `route` コマンドを使用してデフォルトルートを実行します。

210 ページの「静的なネットワーク構成の指定」の例を参照してください。

`install_ipv4_interface` プロパティグループでは1つのインタフェースのみを構成できますが、`ipv4_interface` プロパティグループでは複数のインタフェースを構成できます。どちらの IPv4 プロパティグループにも、次の表にリストされているプロパティが含まれています。

表 13 IPv4 ネットワークインタフェースのプロパティグループのプロパティ

プロパティ	タイプ	必須	説明
<code>name</code>	<code>astring</code>	<code>required</code>	ネットワークインタフェースの名前。
<code>address_type</code>	<code>astring</code>	<code>required</code>	<code>ipadm create-addr</code> サブコマンドの <code>-T</code> オプションの構築に使用する値。有効な値は <code>static</code> または <code>dhcp</code> です。
<code>static_address</code>	<code>net_address_v4</code>	<code>optional</code>	<code>static</code> の <code>address_type</code> でのみ必要です。 <code>ipadm create-addr</code> 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。
<code>dhcp_wait</code>	<code>astring</code>	<code>optional</code>	<code>dhcp</code> の <code>address_type</code> でのみ適用されます。このプロパティを定義した場合は、 <code>ipadm create-addr</code> サブコマンドの <code>-w seconds</code> (または <code>forever</code> ) の部分の構築に使用されます。
<code>default_route</code>	<code>net_address_v4</code>	<code>optional</code>	<code>route</code> コマンドを使用したデフォルトルートの定義に使用されます。  <pre># /usr/sbin/route \ -p add default default-route \ -ifp ifname</pre> <code>ifname</code> の値は、 <code>name</code> プロパティのインタフェース名の部分です。

`install_ipv6_interface` プロパティグループでは1つのインタフェースのみを構成できますが、`ipv6_interface` プロパティグループでは複数のインタフェースを構成できます。IPv6 インタフェースのプロパティグループには、次の表にリストされているプロパティが含まれています。

表 14 IPv6 ネットワークインタフェースのプロパティグループのプロパティ

プロパティ	タイプ	必須	説明
<code>name</code>	<code>astring</code>	<code>required</code>	ネットワークインタフェースの名前。
<code>address_type</code>	<code>astring</code>	<code>required</code>	<code>ipadm create-addr</code> サブコマンドの <code>-T</code> オプションの構築に使用する値。有効な値は <code>static</code> または <code>addrconf</code> です。
<code>static_address</code>	<code>net_address_v6</code>	<code>optional</code>	<code>static</code> の <code>address_type</code> でのみ必要です。 <code>ipadm create-addr</code> 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。

プロパティ	タイプ	必須	説明
interface_id	net_address_v6	optional	addrconf の address_type でのみ適用されま す。ipadm 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	optional	addrconf の address_type でのみ適用されま す。ipadm 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。
stateful	astring	optional	addrconf の address_type でのみ適用されま す。ipadm 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	optional	route コマンドを使用したデフォルトルート の定義に使用されます。  # /usr/sbin/route \ -p add default default-route \ -ifp ifname  ifname の値は、name プロパティのインタフェ ース名の部分です。

## ネームサービスの構成

DNS、NIS、および LDAP クライアントとネームサービススイッチを構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `naming_services` グループとともに使用します。

```
# sysconfig create-profile -g naming_services -o sc_ns.xml
```

`naming_services` グループには、2 つの SMF サービスが含まれています。

- `svc:/system/name-service/switch` サービスはネームサービスを管理します。
- `svc:/network/dns/client` サービスは DNS サービスを管理します。

`svc:/system/name-service/switch` SMF サービスは、ネームサービスのスイッチを構成します。このサービスは、システム構成が行われないプロパティ値で最初は無効になっています。215 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。`svc:/system/name-service/switch` サービスの `config` プロパティグループには、次の表にリストされているプロパティが含まれています。すべてのプロパティの完全な一覧については、`nsswitch.conf(4)` のマニュアルページを参照してください。

表 15 `svc:/system/name-service/switch` プロパティグループの `config` プロパティ

プロパティ	タイプ	必須	説明
default	astring	オプション	すべてのネームサービススイッチデータベースのデフォルトのソース構成を設定します
bootparam	astring	オプション	bootparams データベースのデフォルトのソース構成をオーバーライドします

プロパティ	タイプ	必須	説明
ether	astring	オプション	ethers データベースのデフォルトのソース構成をオーバーライドします
group	astring	オプション	group データベースのデフォルトのソース構成をオーバーライドします
host	astring	オプション	host データベースのデフォルトのソース構成をオーバーライドします
netmask	astring	オプション	netmask データベースのデフォルトのソース構成をオーバーライドします
network	astring	オプション	network データベースのデフォルトのソース構成をオーバーライドします
password	astring	オプション	passwd データベースのデフォルトのソース構成をオーバーライドします
protocol	astring	オプション	protocol データベースのデフォルトのソース構成をオーバーライドします
rpc	astring	オプション	rpc データベースのデフォルトのソース構成をオーバーライドします

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つのプロパティグループ (config) を定義します。このサービスは、それ自体のプロパティを使用して DNS サービスの構成情報を作成します。215 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。

config プロパティグループには、次の表に示すプロパティが含まれています。

表 16 config プロパティグループのプロパティ

プロパティ	タイプ	必須	説明
domain	astring	optional	ローカルドメイン名。resolv.conf の domain ディレクティブの構築に使用されます。
nameserver	net_address_list	required	IPv4 および IPv6 アドレスのリスト。resolv.conf の nameserver ディレクティブの構築に使用されます。
search	astring_list	optional	ホスト名検索用の検索リストのためのドメイン値のリスト。resolv.conf の search ディレクティブの構築に使用されます。

## Kerberos の構成

AI クライアントの Kerberos 構成情報が含まれているシステム構成プロファイルは、kclient コマンドを使用して作成してください。このプロファイルは表示できませんが、ファイルを手動で編集することは推奨しません。詳細は、130 ページの「[AI を使用して Kerberos クライアントを構成する方法](#)」を参照してください。

## Oracle Configuration Manager と Oracle Auto Service Request の設定

Oracle Configuration Manager では、システム構成を My Oracle Support に登録でき、Oracle Auto Service Request では、特定のハードウェア障害に対するサービスリクエストを自動的に生成できます。

Oracle Configuration Manager と Oracle Auto Service Request を構成する有効なプロファイルを生成するには、`sysconfig create-profile` コマンドを `support` グループとともに使用します。

```
# sysconfig create-profile -g support -o sc_support.xml
```

出力のプロファイルは登録の最初のフェーズを設定しますが、これは次の条件に一致するすべての AI クライアントで同じになります。

- システムは、同じ My Oracle Support 資格を使用して登録します。このプロファイルを使用するすべての AI クライアントは、同じ方法で My Oracle Support に登録します。これらすべての AI クライアントのデータは、同じ My Oracle Support アカウントに関連付けられます。
- システムは、同じネットワーク構成を介して My Oracle Support にアクセスします。このプロファイルを使用するすべての AI クライアントシステムは、たとえば同じプロキシサーバーとアグリゲーションハブを介して My Oracle Support にアクセスします。

AI クライアントの別のグループのために追加のプロファイルを作成する必要がある場合は、既存のプロファイルをコピーして編集せずに、`sysconfig create-profile` コマンドを再実行するようにしてください。プロキシサーバーにユーザー名とパスワードがある場合は、パスワードが暗号化されているため、`sysconfig create-profile` を再実行する必要があります。

## システム構成プロファイルテンプレートの使用

プロファイルには、インストール処理中に AI クライアントのインストール環境から得られた値で置換される変数を含めることができます。このように、単一のプロファイルファイルで、さまざまな AI クライアントにさまざまな構成パラメータを設定できます。使用可能な変数の一覧については、[表17](#)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 AI クライアントのホスト名のプレースホルダであり、`AI_IPV4` は AI クライアントの IP アドレスのプレースホルダです。

```
<?xml version='1.0'?>
```

```
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service name="system/identity" version="1" type="service">
    <instance name="node" enabled="true">
      <property_group name="config" type="application">
        <propval name="nodename" value="{AI_HOSTNAME}"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="network/install">
    <instance enabled="true" name="default">
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" value="net0/v4"/>
        <propval name="address_type" value="static"/>
        <propval name="static_address" type="net_address_v4" value="{AI_IPV4}/8"/>
        <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

次のコマンドでは、入力の `hostandIP.xml` ファイルを変更せずに、インストール AI クライアントごとにカスタマイズされるシステム構成プロファイルをインストール サービス内に作成しています。

```
# installadm create-profile -n solaris11-3-i386 -f /export/hostIPnet.xml
```

`hostandIP.xml` ファイルは変更されませんが、AI クライアントに適用されるプロファイルがカスタマイズされます。たとえば、`server1` というホスト名を持つ AI クライアントにインストールする場合、`hostandIP.xml` プロファイルの内容は次のようになる可能性があります。

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service name="system/identity" version="1" type="service">
    <instance name="node" enabled="true">
      <property_group name="config" type="application">
        <propval name="nodename" value="server1"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="network/install">
    <instance enabled="true" name="default">
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" value="net0/v4"/>
        <propval name="address_type" value="static"/>
        <propval name="static_address" type="net_address_v4" value="10.0.0.2/8"/>
        <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

次の表に、テンプレートプロファイル内でプレースホルダとして使用できる変数を示します。

---

**注記** - プロファイルテンプレート変数は、ゾーンプロファイルでサポートされていません。

---

表 17 システム構成テンプレートプロファイルの変数

変数名	説明
AI_ARCH	uname -m からのカーネルアーキテクチャー
AI_CPU	uname -p からのプロセッサタイプ
AI_HOSTNAME	AI クライアントの DNS 名
AI_IPV4	IP バージョン 4 のネットワークアドレス
AI_IPV4_PREFIXLEN	IPv4 ネットワークアドレスの接頭辞の長さ
AI_MAC	コロン (:) で区切られた 16 進数の MAC アドレス
AI_MEM	prtcnf によって返される M バイト単位のメモリーサイズ
AI_NETLINK_DEVICE	ネットワークインタフェースの物理デバイスの名前
AI_NETLINK_VANITY	ネットワークインタフェースのデフォルトのバニティー名
AI_NETWORK	IP バージョン 4 のネットワーク識別子
AI_ROUTER	AI クライアントのデフォルトルーターの IP バージョン 4 ネットワークアドレス
AI_ZONENAME	AI クライアントのゾーン名

## 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。これらのサンプルの各部分を必要に応じて組み合わせて使用できます。次のサンプルシステム構成プロファイルは `/usr/share/auto_install/sc_profiles` に含まれています。

- `enable_sci.xml` - ブートプロセス中に対話型のシステム構成ツールを起動します。
- `sc_sample.xml` - ユーザーアカウント、`root` 役割、コンソール端末タイプ、キーボード、ロケール、タイムゾーン、および単一の IPv4 ネットワークインタフェースを構成する方法を示します。次を参照してください。
- `static_network.xml` - `sc_sample.xml` とほぼ同じですが、ネットブートされた AI クライアントのネットワーク構成を使用できるように、プロファイルテンプレート変数がネットワーク構成で使用されている点が異なります。このプロファイルにはネームサービスの構成例も含まれています。
- `unconfig.xml` - ブートプロセス中に AI クライアントの構成を解除します。
- `install_env/dns.xml` - DNS クライアントを構成するための構成情報を含む、部分的なプロファイル。
- `install_env/network.xml` - 静的 IP アドレス、および静的 IP を含む InfiniBand データリンクの構成情報を含む、部分的なプロファイル。

## 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すると、このサンプルプロファイルが`image-path/auto_install/sc_profiles/sc_sample.xml`に作成されています。

```
<?xml version="1.0"?>
<!--
Copyright (c) 2011, 2012, Oracle and/or its affiliates. All rights reserved.
-->

<!--
Sample system configuration profile for use with Automated Installer

Configures the following:
* User account name 'jack', password 'jack', GID 10, UID 101, root role, bash shell
* 'root' role with password 'solaris'
* Keyboard mappings set to US-English
* Time zone set to UTC
* Network configuration is automated with Network Auto-magic
* DNS name service client is enabled

See the installadm(8) man page for usage of 'create-profile' subcommand.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="system configuration">
  <service name="system/config-user" version="1">
    <instance name="default" enabled="true">
      <property_group name="user_account">
        <propval name="login" value="jack"/>
        <propval name="password" value="9Nd/cwBcNWFZg"/>
        <propval name="description" value="default_user"/>
        <propval name="shell" value="/usr/bin/bash"/>
        <propval name="gid" value="10"/>
        <propval name="uid" value="101"/>
        <propval name="type" value="normal"/>
        <propval name="roles" value="root"/>
        <propval name="profiles" value="System Administrator"/>
      </property_group>
      <property_group name="root_account">
        <propval name="password" value="encrypted_password"/>
        <propval name="type" value="role"/>
      </property_group>
    </instance>
  </service>

  <service version="1" name="system/identity">
    <instance enabled="true" name="node">
      <property_group name="config">
        <propval name="nodename" value="solaris"/>
      </property_group>
    </instance>
  </service>

  <service name="system/console-login" version="1">
    <instance name="default" enabled="true">
      <property_group name="ttymon">
        <propval name="terminal_type" value="sun"/>
      </property_group>
    </instance>
  </service>
```

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

<service name="system/keymap" version="1">
  <instance name="default" enabled="true">
    <property_group name="keymap">
      <propval name="layout" value="US-English"/>
    </property_group>
  </instance>
</service>

<service name="system/timezone" version="1">
  <instance name="default" enabled="true">
    <property_group name="timezone">
      <propval name="localtime" value="UTC"/>
    </property_group>
  </instance>
</service>

<service name="system/environment" version="1">
  <instance name="init" enabled="true">
    <property_group name="environment">
      <propval name="LANG" value="en_US.UTF-8"/>
    </property_group>
  </instance>
</service>

<service name="network/nstall" version="1">
  <instance name="default" enabled="true">
    <property_group name="install_ipv4_interface" type="application">
      <propval name="name" type="astring" value="{AI_NETLINK_VANITY}/v4"/>
      <propval name="address_type" type="astring" value="dhcp"/>
    </property_group>
  </instance>
</service>
</service_bundle>
```

## 静的なネットワーク構成の指定

このプロファイル例のバージョンは、`/usr/share/auto_install/sc_profiles/static_network.xml` で利用できます。このプロファイルの以下に示すバージョンは、次のパラメータを構成するために変更されています。

- IPv4 の静的アドレスが `10.0.0.10` でネットマスクが `255.0.0.0` の `bge0`
- `10.0.0.1` の IPv4 デフォルトルート
- IPv6 のアドレスタイプが `addrconf` の `bge1`
- DNS `8.8.8.8` ネームサーバー
- `example1.com` と `example2.com` (ホスト名検索の DNS 検索リストとして)

ネットマスクは、`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。

<i>netmask</i> の値	ネットマスクの例
8	255.0.0.0
16	255.255.0.0
24	255.255.255.0

```

<?xml version="1.0"?>
<!--
Copyright (c) 2010, 2011, Oracle and/or its affiliates. All rights reserved.
-->

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="system configuration">
  <service name="system/config-user" version="1">
    <instance name="default" enabled="true">
      <property_group name="user_account">
        <propval name="login" value="jack"/>
        <propval name="password" value="9Nd/cwBcNWFZg"/>
        <propval name="description" value="default_user"/>
        <propval name="shell" value="/usr/bin/bash"/>
        <propval name="gid" value="10"/>
        <propval name="type" value="normal"/>
        <propval name="roles" value="root"/>
        <propval name="profiles" value="System Administrator"/>
      </property_group>
      <property_group name="root_account">
        <propval name="password" value="$5$dnRfcZse
$Hx4aBQ161Uvn9ZxJFKMdRiy8tCf4gMT2s2rtkFba2y4"/>
        <propval name="type" value="role"/>
      </property_group>
    </instance>
  </service>

  <service version="1" name="system/identity">
    <instance enabled="true" name="node">
      <property_group name="config">
        <propval name="nodename" value="solaris"/>
      </property_group>
    </instance>
  </service>

  <service name="system/console-login" version="1">
    <instance name="default" enabled="true">
      <property_group name="ttymon">
        <propval name="terminal_type" value="sun"/>
      </property_group>
    </instance>
  </service>

  <service name="system/keymap" version="1">
    <instance name="default" enabled="true">
      <property_group name="keymap">
        <propval name="layout" value="US-English"/>
      </property_group>
    </instance>
  </service>

  <service name="system/timezone" version="1">
    <instance name="default" enabled="true">
      <property_group name="timezone">
        <propval name="localtime" value="UTC"/>
      </property_group>
    </instance>
  </service>

```

```
</instance>
</service>

<service name="system/environment" version="1">
  <instance name="init" enabled="true">
    <property_group name="environment">
      <propval name="LANG" value="en_US.UTF-8"/>
    </property_group>
  </instance>
</service>

<service name="network/physical" version="1">
  <instance name="default" enabled="true">
    <property_group name="install_ipv4_interface" type="application">
      <propval name="name" type="astring" value="{{AI_NETLINK_VANITY}}/v4"/>
      <propval name="address_type" type="astring" value="static"/>
      <propval name="static_address" type="net_address_v4" value="{{AI_IPV4}}/
{{AI_IPV4_PREFIXLEN}}"/>
      <propval name="default_route" type="net_address_v4" value="{{AI_ROUTER}}"/>
    </property_group>
  </instance>
</service>

<service name="network/install" version="1" type="service">
  <instance name="default" enabled="true">
    <property_group name="install_ipv4_interface" type="application">
      <propval name="name" type="astring" value="bge0/v4"/>
      <propval name="address_type" type="astring" value="static"/>
      <propval name="static_address" type="net_address_v4" value="10.0.0.10/8"/>
      <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
    </property_group>

    <property_group name="install_ipv6_interface" type="application">
      <propval name="name" type="astring" value="bge1/v6"/>
      <propval name="address_type" type="astring" value="addrconf"/>
      <propval name="stateless" type="astring" value="yes"/>
      <propval name="stateful" type="astring" value="yes"/>
    </property_group>
  </instance>
</service>

<service name="network/dns/client" version="1">
  <property_group name="config">
    <property name="nameserver">
      <net_address_list>
        <value_node value="8.8.8.8"/>
      </net_address_list>
    </property>
    <property name="search">
      <astring_list>
        <value_node value="example1.com example2.com"/>
      </astring_list>
    </property>
  </property_group>
  <instance name="default" enabled="true"/>
</service>

<service version="1" name="system/name-service/switch">
  <property_group name="config">
    <propval name="default" value="files"/>
    <propval name="host" value="files dns mdns"/>
    <propval name="printer" value="user files"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
```

```

</service>

<service version="1" name="system/name-service/cache">
  <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

## システム構成プロファイル内での IB リンクの指定

次の例は、システム構成プロファイル内で Infiniband (IB) リンクを定義する方法を示しています。

```

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="system configuration">

  <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="net5_partlink" type="ib_partlink">
        <propval name="name" type="astring" value="FFFF.net5"/>
        <propval name="link" type="astring" value="net5"/>
        <propval name="pkey" type="astring" value="FFFF"/>
      </property_group>

      <property_group name="FFFF_net5_ipv4" type="ipv4_interface">
        <propval name="name" type="astring" value="FFFF.net5/v4"/>
        <propval name="address_type" type="astring" value="static"/>
        <propval name="static_address" type="net_address_v4" value="6.6.6.49/24"/>

      <property_group name="net0_ipv4" type="ipv4_interface">
        <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="10.10.6.49/24"/>
      >
        <propval name="default_route" type="net_address_v4" value="10.10.6.1"/>
      </property_group>

    </instance>
  </service>

  <service name="network/dns/client" version="1">
    <property_group name="config">
      <property name="nameserver">
        <net_address_list>
          <value_node value="192.1.1.132"/>
          <value_node value="130.1.1.41"/>
          <value_node value="130.1.1.52"/>
        </net_address_list>
      </property>
      <property name="search">
        <astring_list>

```

```

        <value_node value="example.com"/>
    </astring_list>
</property_group>
</property_group>
<instance name="default" enabled="true"/>
</service>

<service version="1" name="system/name-service/switch">
  <instance enabled="true" name="default"/>
  <property_group name="config">
    <propval name="default" value="files"/>
    <propval name="host" value="files dns mdns"/>
  </property_group>
</service>

<service version="1" name="system/name-service/cache">
  <instance enabled="true" name="default"/>
</service>

```

## 複数の IPv4 インタフェースの構成

この例では `ipv4_interface` プロパティグループタイプを使用しているため、複数のインタフェースを構成できます。 `ipv6_interface` プロパティグループタイプを使用することもできます。

```

<?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="network/install">
      <instance enabled="true" name="default">
        <property_group name="install_ipv4_interface_0" type="ipv4_interface">
          <propval type="net_address_v4" name="static_address" value="10.0.0.10
/8"/>

          <propval type="astring" name="name" value="net0/v4"/>
          <propval type="astring" name="address_type" value="static"/>
        </property_group>
        <property_group name="install_ipv4_interface_1" type="ipv4_interface">
          <propval type="net_address_v4" name="static_address" value="10.0.0.11
/8"/>

          <propval type="astring" name="name" value="net1/v4"/>
          <propval type="astring" name="address_type" value="static"/>
        </property_group>
      </instance>
    </service>
  </service_bundle>
  <service name="network/install" version="1" type="service">
    <instance name="default" enabled="true">
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" type="astring" value="bge0/v4"/>
        <propval name="address_type" type="astring" value="static"/>
        <propval name="static_address" type="net_address_v4" value="10.0.0.10/8"/>
        <propval name="default_route" type="net_address_v4" value="10.0.0.1"/>
      </property_group>
    </instance>
  </service>

```

## ユーザーの SSH 鍵の追加

この例は、自動インストールセッション中に、`ssh_public_keys` を使用してユーザーに SSH 鍵を追加する方法を示しています。各鍵は、指定されたユーザーの `$HOME/.ssh/authorized_keys` に追加されます。

```
<property_group type="application" name="user_account">
  ...
  <property type="astring" name="ssh_public_keys">
    <astring_list>
      <value_node value='[<options>] <key-type> <base64-encoding-key> [<comment>]' />
      <value_node value='[<options>] <key-type> <base64-encoding-key> [<comment>]' />
    </astring_list>
  </property>
</property_group>
```

## 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、[77 ページの「SCI ツールを使用したシステム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 80** 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

```
<?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(5)) -->
  <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 81 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' />
        </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>

```

## DNS ネームサービスの構成

例 82 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
- 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: my.domin.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(5)) -->
  <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>
  <!-- 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="my.domain.com domain.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 83 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。

- LDAP ネームサーバーの IP アドレス: 10.0.0.10
- サービス system/nis/domain に指定されたドメイン my.domain.com
- LDAP 検索ベース (必須)、dc=my,dc=domain,dc=com

```

<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/name-service/switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files ldap"/>
      <propval type="astring" name="printer" value="user files ldap"/>
      <propval type="astring" name="netgroup" value="ldap"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>
  <service version="1" type="service" name="network/ldap/client">
    <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>
  </service>
</service_bundle>

```

```

    </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 84 セキュアな LDAP サーバーを使用した LDAP の構成

このプロファイル例では、次のパラメータを構成します。

- LDAP サーバーの IP アドレス: 10.0.0.10
- サービス system/nis/domain に指定されたドメイン my.domain.com
- LDAP 検索ベース (必須)、dc=my,dc=domain,dc=com
- LDAP プロキシバインド識別名 cn=proxyagent,ou=profile,dc=my,dc=domain,dc=com
- LDAP プロキシバインドパスワード。セキュリティ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>
  <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"/>
    </property_group>
  </service>

```

```

        <!-- note that the password below is encrypted -->
        <propval type="astring" name="bind_passwd" value="{NS1}c2ab873ae7c5ceefa4b9"/>
    </property_group>
    <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
    <property_group type="application" name="config">
        <propval type="hostname" name="domainname" value="my.domain.com"/>
    </property_group>
    <instance enabled="true" name="default"/>
</service>
</service_bundle>

```

## DNS を LDAP とともに使用する

DNS ネームサービスを LDAP ネームサービスとともに使用できます。一般的な使用方法では、DNS でノード名 (LDAP サーバー名を含む) を解決し、LDAP でほかのすべての名前を解決します。この例の最初の service 要素に示したように、サービス `system/name-service/switch` を使用して、DNS をノード名の検索に指定し、LDAP をほかの名前解決に指定します。

```

<?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>
        </property_group>
    </service>

```

```
        </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>
```



## ゾーンのインストールと構成

---

この章では、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り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、[224 ページの「大域ゾーンの AI マニフェストでの非大域ゾーンの指定」](#)を参照してください。
2. 大域ゾーンのインストール後にシステムが最初にブートしたときに、ゾーンの自己アセンブリ SMF サービス (`svc:/system/zones-install:default`) によって、大域ゾーンの AI マニフェストに定義されている各非大域ゾーンが構成され、インストールされます。非大域ゾーンのインストールに使用されるデータについては、[224 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モニターするには、`system/zones-install` サービス、または `zoneadm list -cv` の出力をモニターします。

次のいずれかのエラーが発生した場合、ゾーンはインストールされません。

- ゾーンの `config` ファイルが構文的に正しくありません

- インストールされるゾーンセット内のゾーン名、ゾーンパス、または委任された ZFS データセットの間に競合が存在します。
- 必要なデータセットが大域ゾーンに構成されていません

## 大域ゾーンの AI マニフェストでの非大域ゾーンの指定

非大域ゾーンを指定するには、AI クライアントの 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 AI 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 AI クライアントにこの 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

オプション。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、パブリッシャー情報、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、[226 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85「ゾーンのデフォルトの AI マニフェスト」](#)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。

## システム構成プロファイル

オプション。非大域ゾーンに 0 個以上のシステム構成プロファイルを指定できます。これらのプロファイルは、大域ゾーンを構成するためのプロファイルに似ています。システム構成プロファイルについては、[第11章「AI クライアントのシステム構成パラメータの定義」](#)を参照してください。プロ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。非大域ゾーンのプロファイル例については、[228 ページの「非大域ゾーンのシステム構成プロファイル」](#)を参照してください。

システム構成プロ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プロファイルを追加します。create-profile コマンドで、この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。

システム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、[73 ページの「システムの再構成」](#)を参照してください。

次の例では、/tmp/zmanifest.xml AI マニフェストを solaris11\_3-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 solaris11_3-sparc -f /tmp/zmanifest.xml \
-m zmanifest -c zonename="zone1 zone2"
```

次の例では、/tmp/z1profile.xml プロファイルを solaris11\_3-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 solaris11_3-sparc -f /tmp/z1profile.xml \
-p z1profile -c zonename="zone1 zone2"
```

次の例では、/tmp/z2profile.xml プロファイルを solaris11\_3-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 solaris11_3-sparc -f /tmp/z2profile.xml \
-p z2profile -c zonename=zone2
```

次の例は、solaris11\_3-sparc インストールサービスに追加されている AI マニフェストとシステム構成プロファイルを示しています。

```
$ installadm list -n solaris11_3-sparc -m -p
```

Service Name	Manifest Name	Type	Status	Criteria
solaris11_3-sparc	line1-netra2000	xml	active	mac = 00:14:4F:2D:7A:DC
	zmanifest	xml	active	zonename = zone1,zone2
	orig_default	derived	default	none

Service Name	Profile Name	Environment	Criteria
solaris11_3-sparc	z1profile	system	zonename = zone1,zone2
	z2profile	system	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 要素には 1 つの zpool 子要素のみ指定できます。

zpool 要素では、非大域ゾーンの AI マニフェストの 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 85**                    ゾーンのデフォルトの AI マニフェスト

次のファイルは、非大域ゾーンのデフォルトの AI マニフェストを示しています。このマニフェストは、カスタム AI マニフェストをゾーンに指定しない場合に使用されます。このマニフェストは、/usr/share/auto\_install/manifest/zone\_default.xml で入手できます。

target セクションは、ゾーンの ZFS ファイルシステムを定義します。destination セクションは、インストールするロケールを指定します。software\_data セクションは、solaris-small-server パッケージのインストールを指定します。solaris-small-server パッケージは、インストールするほとんどの非大域ゾーンで推奨されるツールとデバイスドライバのグループパッケージです。solaris-small-server グループパッケージに含まれるパッケージの完全なリストについては、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[グループパッケージ内のすべてのインストール可能なパッケージの一覧表示](#)」で説明されているように、pkg contents コマンドを使用します。

パッケージソースが指定されていないことに注意してください。システムリポジトリについては、[pkg.sysrepo\(1M\)](#) を参照してください。

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

Copyright (c) 2011, 2013, Oracle and/or its affiliates. All rights reserved.

-->
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">

<auto_install>
  <ai_instance name="zone_default">
    <target>
      <logical>
        <zpool name="rpool">
          <!--
            Subsequent <filesystem> entries instruct an installer
            to create following ZFS datasets:

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

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

            In rare cases, if there is a need to deploy a zone
            without these datasets, either comment out or remove
            <filesystem> entries. In such scenario, it has to be also
            assured that in case of non-interactive post-install
            configuration, creation of initial user account is
            disabled in related system configuration profile.
            Otherwise the installed zone would fail to boot.
          -->
          <filesystem name="export" mountpoint="/export"/>
```

```

        <filesystem name="export/home"/>
        <be name="solaris">
            <options>
                <option name="compression" value="on"/>
            </options>
        </be>
    </zpool>
</logical>
</target>

<software type="IPS">
    <destination>
        <image>
            <!-- Specify locales to install -->
            <facet set="false">facet.locale.*</facet>
            <facet set="true">facet.locale.de</facet>
            <facet set="true">facet.locale.de_DE</facet>
            <facet set="true">facet.locale.en</facet>
            <facet set="true">facet.locale.en_US</facet>
            <facet set="true">facet.locale.es</facet>
            <facet set="true">facet.locale.es_ES</facet>
            <facet set="true">facet.locale.fr</facet>
            <facet set="true">facet.locale.fr_FR</facet>
            <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>

```

## 非大域ゾーンのシステム構成プロファイル

システム構成プロファイルをゾーンに指定して、言語、ロケール、タイムゾーン、端末、ユーザー、ゾーン管理者の root パスワードなどのゾーンパラメータを構成できます。タイムゾーンは構成できますが、時間を設定することはできません。ネームサービスを構成できます。

ゾーンで許可されていない構成を指定した場合、それらのプロパティ設定は無視されます。

次のファイルは、非大域ゾーンのシステム構成プロファイルファイル例を示しています。

```
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
```

```
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/config-user">
    <instance enabled="true" name="default">
      <property_group type="application" name="root_account">
        <propval type="astring" name="login" value="root"/>
        <propval type="astring" name="password" value="encrypted_password"/>
        <propval type="astring" name="type" value="normal"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/timezone">
    <instance enabled="true" name="default">
      <property_group type="application" name="timezone">
        <propval type="astring" name="localtime" value="UTC"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/environment">
    <instance enabled="true" name="init">
      <property_group type="application" name="environment">
        <propval type="astring" name="LC_ALL" value="C"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/identity">
    <instance enabled="true" name="node">
      <property_group type="application" name="config">
        <propval type="astring" name="nodename" value="z2-test"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/keymap">
    <instance enabled="true" name="default">
      <property_group type="system" name="keymap">
        <propval type="astring" name="layout" value="US-English"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/console-login">
    <instance enabled="true" name="default">
      <property_group type="application" name="ttymon">
        <propval type="astring" name="terminal_type" value="vt100"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="network/physical">
    <instance enabled="true" name="default">
      <property_group type="application" name="netcfg"/>
    </instance>
  </service>
</service_bundle>
```



## 初回ブート時のカスタムスクリプトの実行

---

AI マニフェストまたはシステム構成プロファイルで実行できない追加のインストールまたは構成を行う場合は、一度だけ実行する 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。初回ブートスクリプトを実行するサービスと、そのサービスを含むパッケージを作成するには、次のいずれかの方法を使用します。

- 231 ページの「初回ブートサービスおよびパッケージの自動作成」
- 232 ページの「初回ブートサービスパッケージの対話式での作成」
- 次の手順を含む手動プロセス:
  - 233 ページの「初回ブート時に実行するスクリプトの手動作成」
  - 236 ページの「初回ブートサービスパッケージの手動作成」

AI 初回ブートスクリプトパッケージの作成が完了したら、次の手順のいずれかに従うことができます。

- 244 ページの「初回ブートスクリプトまたはサービスを更新する方法」
- 245 ページの「初回ブートサービスのテスト」

## 初回ブートサービスおよびパッケージの自動作成

このセクションでは、`svc-create-first-boot` コマンドを実行して初回ブートサービスおよびパッケージを作成する場合の、いくつかのオプションについて説明します。次の例では、変更はすべてコマンド行オプションによって処理されます。手動編集の必要はありません。

**例 86**                    コマンド行モードでの初回ブートサービスおよびパッケージの作成

次の例では、`first-boot-script.sh` スクリプトを使ってサービスが作成されています。任意の実行可能スクリプトを使用できます。アーカイブは `myp5p.p5p` に格納されます。

```
% /usr/sbin/svc-create-first-boot -s first-boot-script.sh -d myp5p.p5p
```

**例 87** コマンド行モードでの初回ブートサービスおよびパッケージの作成 (カスタマイズあり)

次の例では、`first-boot-script.sh` スクリプトを使ってサービスが作成されます。ネットワークマイルストーンが依存関係として追加されます。アーカイブは `myp5p.p5p` に格納されます。

```
% /usr/sbin/svc-create-first-boot -s first-boot-script.sh -o service-dependency=milestone/network -d myp5p.p5p
```

**例 88** 初回ブートサービスで使用するシェルスクリプト

初回ブートサービスに含めるスクリプトの内容は、正規の実行可能スクリプトであれば、どのようなものでもかまいません。たとえば、2つのインタフェースを構成するには、次のようなスクリプトを使用できます。

```
#!/bin/ksh
# Create and configure addresses on two IP interfaces
/usr/sbin/ipadm create-ip net0
/usr/sbin/ipadm create-ip net1
/usr/bin/ipadm create-addr -a 10.153.125.222/24 net0
/usr/bin/ipadm create-addr -a 10.169.254.182.77/24 net1
```

## 初回ブートサービスパッケージの対話式での作成

初回ブートスクリプト実行用の SMF サービスを含む IPS パッケージを対話式で作成するには、`svc-create-boot-script` コマンドを引数なしで実行します。次の情報を提供する必要があります。

- 初回ブートスクリプトへのパス。
- 初回ブートサービスの FMRI - デフォルトは `svc:/site/first-boot-svc` です。
- 依存関係をカスタマイズする必要があるかどうかを選択します - 「いいえ」を選択した場合、マルチユーザーマイルストーンが選択されます。「はい」を選択した場合、次の質問が提示されます。
  - 初回ブートサービスがネットワークを構成したり、ネットワークへのアクセスを必要とするか - 「はい」を選択した場合、ネットワークマイルストーンが依存関係として追加されます。
  - 初回ブートサービスが非ローカルのホストやユーザーの名前を検索する必要があるか - 「はい」を選択した場合、`name-services` マイルストーンが依存関係として追加されます。
  - 初回ブートサービスがルートファイルシステムの外側にあるファイルにアクセスする必要があるか - 「はい」を選択した場合、`multi-user` マイルストーンが依存関係として追加されます。
  - 追加の依存関係があれば入力します。

- クローンされたインスタンス上で初回ブートスクリプトを再実行すべきかどうかを選択します。
- メソッドスクリプトのタイムアウトを秒単位で入力します - デフォルトは 60 秒です。
- パッケージの FMRI を入力します - デフォルトの FMRI は `first-boot-svc` です。
- パブリッシャーの名前を入力します - デフォルトの名前は `firstboot` です。
- 既存のリポジトリへの URI、生成された p5p アーカイブへのパス、のいずれかを入力します。

## 初回ブート時に実行するスクリプトの手動作成

スクリプトに使用可能なソースを知るには、初回ブート時にどのツールが AI クライアントにインストールされているかを知る必要があります。デフォルトでは、`solaris-large-server` パッケージがインストールされています。そのグループパッケージをインストールした場合は、`Python`、`bash`、`ksh`、およびその他のツールが初回ブート時に使用可能です。`solaris-large-server` グループパッケージに含まれるパッケージの完全なリストについては、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[グループパッケージ内のすべてのインストール可能なパッケージの一覧表示](#)」で説明されているように、`pkg contents` コマンドを使用します。`solaris-large-server` パッケージに用意されていないソースをスクリプトに使用する場合は、必要なパッケージを特定し、それを AI マニフェストに指定します。インストールする可能性があるほかのパッケージの名前を検索する方法については、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』を参照してください。

- 異なるスクリプト内のそれぞれのコマンドが互いに競合しないようにするため、使用する初回ブートスクリプトは 1 つのみにしてください。
- 初回ブートスクリプト内ではリブートしないでください。

すべての初回ブートスクリプトに次のテキストを含めるようにしてください。

- 初回ブートスクリプトでは、SMF メソッドの終了コードなどの定義を使用するために `/lib/svc/share/smf_include.sh` を読み込む必要があります。
- スクリプトでは、前のブートですでに実行されたかどうかを検査するようにしてください。`completed` プロパティがすでに `true` に設定されている場合、スクリプトは `start` メソッドを終了し、サービスを一時的に無効にすべきです。

スクリプトの次の行は、`site/first-boot-script-svc:default` サービスインスタンスの `config` プロパティグループ内の `completed` プロパティの値を取得し、その値をローカルの `completed` 変数に割り当てます。

```
completed=`svccprop -p config/completed site/first-boot-script-svc:default`
```

completed 変数が true に設定された場合、スクリプトの次の行は、SMF\_EXIT\_TEMP\_DISABLE 終了コードとともに、終了の簡単な理由として method\_completed を、終了の理由を示す長い説明として "Configuration completed" を、サービスの start メソッドに送ります。

```
smf_method_exit $SMF_EXIT_TEMP_DISABLE script_completed "Configuration completed"
```

- 初回ブートスクリプトでは、AI インストールによって作成されたばかりのブート環境 (BE) のコピーを保存するようにしてください。初回ブートスクリプトで変更される前に BE のコピーを保存しておく、スクリプトによって何らかの問題が発生した場合に、保存された BE にブートするだけで簡単に回復できます。次のテキストは、beadm list 出力の R フラグによって実行中の BE を特定したあと、その BE をコピーします。

```
bename=`beadm list -Hd|nawk -F ';' '$3 ~ /R/ {print $1}`
beadm create ${bename}.orig
echo "Original boot environment saved as ${bename}.orig"
```

- スクリプトでは、その作業が完了したときに、completed プロパティの値を true に設定し、新しいプロパティ値でサービスをリフレッシュし、start メソッドを終了してサービスを一時的に無効にする必要があります。svccfg コマンドを使用して completed プロパティを true に設定し、svccadm コマンドを使用してサービスをリフレッシュします。これにより、以降のリブート時にスクリプトが再実行されないようになります。

```
smf_method_exit $SMF_EXIT_TEMP_DISABLE script_completed "Configuration completed"
```

#### 例 89 テンプレート初回ブートスクリプト

デフォルトでは sh が ksh93 であることを覚えておいてください。

```
#!/bin/sh

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

# If nothing to do, exit with temporary disable
completed=`(svccprop -p config/completed site/first-boot-script-svc:default`
[ "${completed}" = "true" ] && \
    smf_method_exit $SMF_EXIT_TEMP_DISABLE completed "Configuration completed"

# Obtain the active BE name from beadm: The active BE on reboot has an R in
# the third column of 'beadm list' output. Its name is in column one.
bename=`beadm list -Hd|nawk -F ';' '$3 ~ /R/ {print $1}`
beadm create ${bename}.orig
echo "Original boot environment saved as ${bename}.orig"

# Place your one-time configuration tasks here

# Record that this script's work is done
svccfg -s site/first-boot-script-svc:default setprop config/completed = true
```

```
svcadm refresh site/first-boot-script-svc:default
smf_method_exit $SMF_EXIT_TEMP_DISABLE method_completed "Configuration completed"
```

---

ヒント - スクリプト内の構文エラーをチェックするには、`-n` オプションを使用します。

```
$ ksh -n first-boot-script.sh
```

---

#### 例 90 複数の IP インタフェースを構成する初回ブートスクリプト

この例は、2つの IP インタフェースのアドレスを構成し、デフォルトルートを追加する `first-boot-script.sh` という名前の初回ブートスクリプトを示しています。

```
#!/bin/sh

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

# If nothing to do, exit with temporary disable
completed=`svccprop -p config/completed site/first-boot-script-svc:default`
[ "${completed}" = "true" ] && \
    smf_method_exit $SMF_EXIT_TEMP_DISABLE completed "Configuration completed"

# Obtain the active BE name from beadm: The active BE on reboot has an R in
# the third column of 'beadm list' output. Its name is in column one.
bename=`beadm list -Hd|nawk -F ' '; ' $3 ~ /R/ {print $1}'`
beadm create ${bename}.orig
echo "Original boot environment saved as ${bename}.orig"

# Create and configure addresses on two IP interfaces
/usr/sbin/ipadm create-ip net0
/usr/sbin/ipadm create-ip net1
/usr/sbin/ipadm create-addr -a 10.153.125.222/24 net0
/usr/sbin/ipadm create-addr -a 169.254.182.77/24 net1

# Add a default route with net0 as the gateway
/usr/sbin/route add default 10.153.125.1 -ifp net0

# Record that this script's work is done
svccfg -s site/first-boot-script-svc:default setprop config/completed = true
svcadm refresh site/first-boot-script-svc:default

smf_method_exit $SMF_EXIT_TEMP_DISABLE method_completed "Configuration completed"
```

初回ブートスクリプトの別の適切な使用法は、`useradd` コマンドを使用して複数のユーザーをシステム上に構成する方法です。

## 初回ブートサービスパッケージの手動作成

次の手順では、初回ブートサービスパッケージで使用される情報を手動で作成する方法を示します。

1. [236 ページの「初回ブートスクリプトを含む SMF マニフェストファイルをマニフェスト作成ツールを使って作成する」](#) - AI サービスの作成が完了したら、次を行うこともできます。
  - [238 ページの「初回ブートスクリプトが一度しか実行されないようにする方法」](#)
  - [239 ページの「svcbundle によって作成された生成済みマニフェストのカスタマイズ」](#)
2. [241 ページの「初回ブートスクリプトの IPS パッケージを作成および発行する方法」](#)
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。

## 初回ブートスクリプトを含む SMF マニフェストファイルをマニフェスト作成ツールを使って作成する

スクリプトを実行するサービスを定義した SMF マニフェストファイルを作成します。

- サービスの `start` メソッドによって初回ブートスクリプトが実行されます。
- この例では、`multi-user` 依存性を指定して、初回ブート後の起動シーケンスの終わりに初回ブートスクリプトが実行されるようにしています。初回ブートスクリプトの実行内容によっては、このような依存性は必要ない場合もあります。このような依存性を指定しなかった場合は、システムが適切に構成される前にスクリプトが実行される可能性があります。

---

ヒント - スクリプトの依存性を評価し、その依存性が満たされたあとでスクリプトを実行するようにサービスを作成してください。

---

- `completed` プロパティは `false` の値で定義されます。

`svcbundle` コマンドを使用して、有効なサービスマニフェストを生成できます。次の例では、デフォルトで、`svcbundle` コマンドによって生成されたマニフェストが一時的なサービスを指定し、`multi-user` 依存性を指定しています。

## 例 91 生成された SMF サービスマニフェスト

次のコマンドでは、233 ページの「初回ブート時に実行するスクリプトの手動作成」に示したスクリプトの名前が `start-method` の値として指定されています。スクリプトの名前は `/opt/site/first-boot-script.sh` として指定されていますが、これは、241 ページの「初回ブートスクリプトの IPS パッケージを作成および発行する方法」で作成されるパッケージによって、`first-boot-script.sh` スクリプトが `/opt/site/first-boot-script.sh` にインストールされるからです。

次のコマンドでは、`completed` プロパティが、プロパティグループ名、プロパティ名、プロパティのタイプ、およびプロパティの初期値をコロンで区切ったリストで指定されています。

```
$ svcbundle -s service-name=site/first-boot-script-svc \
-s start-method=/opt/site/first-boot-script.sh \
-s instance-property=config:completed:boolean:false \
> first-boot-script-svc-manifest.xml
```

次に示す生成されたサービスマニフェストでは、初回ブートスクリプト `/opt/site/first-boot-script.sh` は `start` メソッドの `exec` 属性の値です。`completed` プロパティは、このサービスのデフォルトインスタンス (`first-boot-script-svc:default`) を定義する `instance` 要素に指定されています。

```
<?xml version="1.0" ?>
<!DOCTYPE service_bundle
  SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<!--
  Manifest created by svcbundle (2014-Jan-14 16:39:30-0700)
-->
<service_bundle type="manifest" name="site/first-boot-script-svc">
  <service version="1" type="service" name="site/first-boot-script-svc">
    <!--
      The following dependency keeps us from starting until the
      multi-user milestone is reached.
    -->
    <dependency restart_on="none" type="service"
      name="multi_user_dependency" grouping="require_all">
      <service_fmri value="svc:/milestone/multi-user"/>
    </dependency>
    <exec_method timeout_seconds="60" type="method" name="start"
      exec="/opt/site/first-boot-script.sh"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute to stop the service. See smf_method(7) for more
      details.
    -->
    <exec_method timeout_seconds="60" type="method" name="stop"
      exec=":true"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute when the service is refreshed. Services are
      typically refreshed when their properties are changed in the
      SMF repository. See smf_method(7) for more details. It is
      common to retain the value of :true which means that SMF will
      take no action when the service is refreshed. Alternatively,
      you may wish to provide a method to reread the SMF repository
      and act on any configuration changes.
    -->
```

```
-->
<exec_method timeout_seconds="60" type="method" name="refresh"
  exec=":true"/>
<property_group type="framework" name="startd">
  <propval type="astring" name="duration" value="transient"/>
</property_group>
<instance enabled="true" name="default">
  <property_group type="application" name="config">
    <propval type="boolean" name="completed" value="false"/>
  </property_group>
</instance>
<template>
  <common_name>
    <loctext xml:lang="C">
      <!--
        Replace this comment with a short name for the
        service.
      -->
    </loctext>
  </common_name>
  <description>
    <loctext xml:lang="C">
      <!--
        Replace this comment with a brief description of
        the service
      -->
    </loctext>
  </description>
</template>
</service>
</service_bundle>
```

## ▼ 初回ブートスクリプトが一度しか実行されないようにする方法

次の手順は、スクリプトが新規にインストールされたシステムの初回ブート時にのみ実行され、かつスクリプトが1度だけ実行されるようにする方法を示しています。

1. **スクリプトを実行するサービスを作成します。**  
この単純なサービスを作成するもっとも簡単な方法は、`svcbundle` コマンドを使用することです (236 ページの「[初回ブートスクリプトを含む SMF マニフェストファイルをマニフェスト作成ツールを使って作成する](#)」を参照)。
2. **スクリプトを実行する前に、スクリプト完了フラグを設定します。**  
サービスマニフェストにブール型の完了プロパティを定義し、その値を `false` に設定します。例91「[生成された SMF サーマニフェスト](#)」のマニフェストに含まれる `completed` プロパティを参照してください。
3. **スクリプトの最後でスクリプト完了フラグを設定します。**  
スクリプトの最後で、`svccfg` コマンドを使用して `completed` プロパティを `true` に設定します。`svcadm` コマンドを使用して、新しいプロパティ値でサービスをリフ

リフレッシュします。例89「[テンプレート初回ブートスクリプト](#)」のサンプルスクリプトの末尾を参照してください。

#### 4. スクリプトが完了した場合は、サービスを無効にします。

サービスマニフェストで、デフォルトのサービスインスタンスが作成され、有効化されます。このサービスはスクリプトで無効化されます。初回ブートスクリプトを終了するときに、`SMF_EXIT_TEMP_DISABLE` 終了コードを使用してサービスの `start` メソッドを終了し、サービスを一時的に無効にします。サービスが無効になり、サービスの `stop` メソッドは実行されません。

サービスをより簡単に再有効化できるため、サービスを永続的に無効にするより、サービスを一時的に無効にすることをお勧めします。状況によっては、実行された構成作業 (ゾーンのクローニングや移行) を更新するためにスクリプト (したがってサービス) を再実行する必要があります。サービスが永続的に無効になっている場合は、`svcadm enable` コマンドを実行してサービスを再有効化する必要があります。サービスをオンラインのままにせずに、やはりサービスを一時的に無効にすることをお勧めします。オンラインのサービスは、リブートのたびに作業を実行しているように見える場合があります。この例では、サービスの名前は `site/first-boot-script-svc` です。AI クライアントがブートされたあとで、サービスが `disabled` 状態になっていることがわかります。

```
$ svcs first-boot-script-svc
STATE          STIME          FMRI
disabled       8:24:16       svc:/site/first-boot-script-svc:default
```

## svcbundle によって作成された生成済みマニフェストのカスタマイズ

`svcbundle` コマンドで生成されたサービスマニフェストは、何も変更せずに要件を満たしている可能性があります。次の例は、サービスマニフェストの変更を示しています。

サービスマニフェストを変更した場合は、`svccfg validate` コマンドを使用して、マニフェストが引き続き有効であることを確認します。

**例 92**                    カスタマイズしたサービスマニフェスト: スクリプトの実行が可能な時間の増加

生成されたサービスマニフェストの次のコピーでは、`exec_method` のデフォルトのタイムアウト (60 秒) を `start` メソッドのために増やしています。 `start` メソッドが初回ブートスクリプトを実行するための十分な時間を確保します。

```
<?xml version="1.0" ?>
<!DOCTYPE service_bundle
  SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
```

```

<!--
  Manifest created by svcbundle (2014-Jan-14 16:39:30-0700)
-->
<service_bundle type="manifest" name="site/first-boot-script-svc">
  <service version="1" type="service" name="site/first-boot-script-svc">
    <!--
      The following dependency keeps us from starting until the
      multi-user milestone is reached.
    -->
    <dependency restart_on="none" type="service"
      name="multi_user_dependency" grouping="require_all">
      <service_fmri value="svc:/milestone/multi-user"/>
    </dependency>
    <!--
      Make sure the start method has adequate time to run the script.
    -->
    <exec_method timeout_seconds="360" type="method" name="start"
      exec="/opt/site/first-boot-script.sh"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute to stop the service. See smf_method(7) for more
      details.
    -->
    <exec_method timeout_seconds="60" type="method" name="stop"
      exec=":true"/>
    <!--
      The exec attribute below can be changed to a command that SMF
      should execute when the service is refreshed. Services are
      typically refreshed when their properties are changed in the
      SMF repository. See smf_method(7) for more details. It is
      common to retain the value of :true which means that SMF will
      take no action when the service is refreshed. Alternatively,
      you may wish to provide a method to reread the SMF repository
      and act on any configuration changes.
    -->
    <exec_method timeout_seconds="60" type="method" name="refresh"
      exec=":true"/>
    <property_group type="framework" name="startd">
      <propval type="astring" name="duration" value="transient"/>
    </property_group>
    <instance enabled="true" name="default">
      <property_group type="application" name="config">
        <propval type="boolean" name="completed" value="false"/>
      </property_group>
    </instance>
    <template>
      <common_name>
        <loctext xml:lang="C">
          <!--
            Replace this comment with a short name for the
            service.
          -->
          </loctext>
        </common_name>
        <description>
          <loctext xml:lang="C">
            <!--
              Replace this comment with a brief description of
              the service
            -->
            </loctext>
          </description>
        </template>
      </service>

```

```
</service_bundle>
$ svccfg validate first-boot-script-svc-manifest.xml
```

**例 93** カスタマイズしたサービスマニフェスト: 非大域ゾーンがインストールされたあとでのスクリプトの実行

次のサービスマニフェストの抜粋では、すべての非大域ゾーンがインストールされたあとで初回ブートスクリプトが実行されるようにするために、`svc:/milestone/multi-user` の依存関係は `svc:/system/zones-install` の依存関係に変更されません。

```
<!--
  The following dependency keeps us from starting until all
  non-global zones are installed.
-->
<dependency restart_on="none" type="service"
  name="ngz_dependency" grouping="require_all">
  <service_fmri value="svc:/system/zones-install"/>
</dependency>
```

## ▼ 初回ブートスクリプトの IPS パッケージを作成および発行する方法

次の内容の IPS パッケージを作成します。

- サーマニフェストファイル。
- 初回ブートスクリプト。
- AI 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。

### 1. ディレクトリ階層を作成します。

この例では、サービスマニフェストは `/lib/svc/manifest/site` にインストールされ、初回ブートスクリプトは `/opt/site` にインストールされます。

```
$ mkdir -p proto/lib/svc/manifest/site
$ mkdir -p proto/opt/site
$ cp first-boot-script-svc-manifest.xml proto/lib/svc/manifest/site
$ cp first-boot-script.sh proto/opt/site
```

### 2. パッケージマニフェストを作成します。

`first-boot-script.p5m` という名前の次のファイルを作成します。

```
set name=pkg.fmri value=first-boot-script@1.0,5.11-0
set name=pkg.summary value="AI first-boot script"
set name=pkg.description value="Script that runs at first boot after AI installation"
set name=info.classification value=\
```

```
"org.opensolaris.category.2008:System/Administration and Configuration"
file lib/svc/manifest/site/first-boot-script-svc-manifest.xml \
  path=lib/svc/manifest/site/first-boot-script-svc-manifest.xml owner=root \
  group=sys mode=0444
dir path=opt/site owner=root group=sys mode=0755
file opt/site/first-boot-script.sh path=opt/site/first-boot-script.sh \
  owner=root group=sys mode=0555
```

初回ブートスクリプトの実行内容によっては、依存性を指定することが必要な場合もあります。このマニフェストを変更した場合は、新しいマニフェストが正しいことを検証します。警告は無視できます。パッケージを作成する方法 (pkgdepend、pkgmogrify、および pkglint コマンドに関する情報を含む) については、『Oracle Solaris 11.3 での Image Packaging System を使用したソフトウェアのパッケージ化と配布』の第 2 章、「IPS を使用したソフトウェアのパッケージ化」を参照してください。

### 3. パッケージのリポジトリを作成します。

この例では、パブリッシャーとして 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。

---

**注記** - インストール時に AI クライアントがアクセスできるディレクトリにリポジトリを作成してください。

---

```
$ pkgrepo create firstbootrepo
$ pkgrepo -s firstbootrepo add-publisher firstboot
```

### 4. パッケージを発行します。

```
$ pkgsend publish -d ./proto -s ./firstbootrepo first-boot-script.p5m
pkg://firstboot/first-boot-script@1.0,5.11-0:20140114T022508Z
PUBLISHED
```

AI 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 パブリッシャーは、次のセクションに示すように AI マニフェストに定義されます。

### 5. パッケージが使用可能であることを確認します。

パッケージを一覧表示して、パッケージが使用可能であることを検証します。

```
$ pkg list -g ./firstbootrepo first-boot-script
NAME (PUBLISHER)          VERSION  IFO
first-boot-script (firstboot)  1.0-0   ---
```

### 6. (オプション) パッケージのインストールをテストします。

-n オプションは、パッケージをインストールしないことを示します。

```
# pkg set-publisher -g ./firstbootrepo firstboot
# pkg publisher
PUBLISHER  TYPE    STATUS P LOCATION
solaris    origin online F http://http://pkg.oracle.com/solaris/release/
firstboot  origin online F file:///home/user1/firstboot/firstbootrepo/
# pkg list -af first-boot-script
```

```

NAME (PUBLISHER)          VERSION  IFO
first-boot-script (firstboot)  1.0-0   ---
# pkg install -nv first-boot-script
    Packages to install:      1
    Estimated space available: 50.68 GB
Estimated space to be consumed: 64.66 MB
    Create boot environment:   No
Create backup boot environment: No
    Rebuild boot archive:      No

Changed packages:
firstboot
  first-boot-script
    None -> 1.0,5.11-0:20140114T022508Z
Planning linked: 0/2 done; 1 working: zone:z2
Linked image 'zone:z2' output:
|   Estimated space available: 50.68 GB
| Estimated space to be consumed: 62.07 MB
|   Rebuild boot archive:      No
|
Planning linked: 1/2 done; 1 working: zone:z1
Linked image 'zone:z1' output:
|   Estimated space available: 50.67 GB
| Estimated space to be consumed: 62.07 MB
|   Rebuild boot archive:      No
|

```

次の手順 AI クライアントが 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 11.3 パッケージリポジトリのコピーと作成』](#)を参照してください。

## ▼ 初回ブートスクリプトの IPS パッケージをインストールする方法

### 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「AI クライアントのインストールパラメータの定義」](#)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 AI クライアントからアクセスできる URI であることを確認します。AI 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. AI クライアントをネットワーク経由でブートします。**

AI クライアントをネットワークブートし、AI を使用して Oracle Solaris 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 AI クライアントがブートされると、そのサービスが実行され、初回ブートスクリプトが実行されます。

## ▼ 初回ブートスクリプトまたはサービスを更新する方法

スクリプトまたはサービスマニフェストを変更した場合は、この手順を使用して更新をインストールします。

**1. 更新されたファイルをプロトタイプディレクトリにコピーします。**

```
$ cp first-boot-script-svc-manifest.xml proto/lib/svc/manifest/site
$ cp first-boot-script.sh proto/opt/site
```

**2. パッケージのバージョンを増分します。**

パッケージマニフェストで、`pkg.fmri` 属性の値をたとえば次のように変更します。

```
first-boot-script@1.0,5.11-0.1
```

**3. 新しいバージョンを公開します。**

パッケージの新しいバージョンをリポジトリに公開します。

```
$ pkgsend publish -d ./proto -s ./firstbootrepo first-boot-script.p5m
pkg://firstboot/first-boot-script@1.0,5.11-0.1:2013123T231948Z
PUBLISHED
```

**4. パッケージを更新します。**

`pkg list -af` コマンドを使用して、新しいバージョンにアクセスできるようにします。場合によっては、`pkg refresh firstboot` コマンドを使用してパッケージリストをリフレッシュする必要があります。`pkg update` コマンドを使用してパッケージを更新します。

5. テストシステムをリブートします。
6. ブートが完了したら、インストールされたパッケージを確認します。  
セキュリティ上のベストプラクティスでは、次のコマンドを実行して、パッケージ内のファイルシステムオブジェクトがセキュアでない方法で変更されていないことを確認することが推奨されます。詳細は、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[パッケージの検証と検証エラーの修正](#)」を参照してください。

```
# pkg verify -v
```

## 初回ブートサービスのテスト

AI インストールをテストする前にサービスをテストするには、テストシステムにパッケージをインストールし、そのテストシステムをリブートするだけです。

```
# pkg install first-boot-script
   Packages to install: 1
   Create boot environment: No
   Create backup boot environment: No

DOWNLOAD                                PKGS      FILES    XFER (MB)   SPEED
Completed                                1/1       2/2      0.0/0.0     0B/s

PHASE                                     ITEMS
Installing new actions                    7/7
Updating package state database           Done
Updating image state                      Done
Creating fast lookup database             Done
Reading search index                     Done
# pkg list first-boot-script
NAME (PUBLISHER)                          VERSION    IFO
first-boot-script (firstboot)            1.0-0     i--
# pkg info first-boot-script
   Name: first-boot-script
   Summary: AI first-boot script
   Description: Script that runs at first boot after AI installation
   Category: System/Administration and Configuration
   State: Installed
   Publisher: firstboot
   Version: 1.0
   Build Release: 5.11
   Branch: 0
   Packaging Date: Dec 23, 2013 02:50:31 PM
   Size: 3.89 KB
   FMRI: pkg://firstboot/first-boot-script@1.0,5.11-0:20131223T145031Z
```

テストシステムをリブートします。スクリプトが上に示したように新しいブート環境を作成したら、必ずその新しいブート環境からブートしてください。

スクリプトが `/opt/site` ディレクトリ内にあり、スクリプトの効果が正しいことを確認します。

サービスの状態を確認します。スクリプトが完了して正常終了した場合は、サービスが無効な状態になっているはずですが。

```
# svcs first-boot-script-svc
STATE          STIME    FMRI
disabled       8:24:16 svc:/site/first-boot-script-svc:default
```

次のいずれかのコマンドを使用して、`completed` プロパティの値を確認します。

```
# svcprop first-boot-script-svc:default
config/completed boolean true
# svcprop -p config/completed first-boot-script-svc:default
true
```

サービスのログファイルを確認する場合は、次のコマンドを使用してログファイルの場所を見つけます。

```
# svcs -x first-boot-script-svc
svc:/site/first-boot-script-svc:default (?)
State: disabled since Dec 23, 2013 08:24:16 AM PDT
Reason: Temporarily disabled by service method: "Configuration completed."
See: http://support.oracle.com/msg/SMF-8000-1S
See: /var/svc/log/site-first-boot-script-svc:default.log
Impact: This service is not running.
```

ログファイルには次の情報が含まれています。

```
[ Jul 23 08:22:57 Enabled. ]
[ Jul 23 08:24:14 Executing start method ("/opt/site/first-boot-script.sh"). ]
[ Jul 23 08:24:16 Method "start" exited with status 101. ]
[ Jul 23 08:24:16 "start" method requested temporary disable: "Configuration completed" ]
[ Jul 23 08:24:16 Rereading configuration. ]
```

## AI サーバーを使用した AI クライアントのインストール

---

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。

### AI クライアントのインストール方法

AI 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 AI 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 AI 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 AI 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 AI クライアントをネットワーク経由でブートします。
2. AI クライアントは DHCP サーバーに接続し、クライアントのネットワーク構成と AI サーバーの場所を取得します。SPARC クライアントは、必要に応じて、OBP に設定された `network-boot-arguments` 変数を使用してこの情報を取得することもできます。
3. AI クライアントが次のいずれかのソースからネットイメージをロードします。
  - `installadm create-client` コマンドを使ってこの AI 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. AI クライアントが、150 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、AI クライアントがリブートします。または、システム管理者が AI クライアントをリブートします。

6. リポート中に、AI クライアントが次のいずれかの方法で構成されます。
  - 151 ページの「システム構成プロファイルの選択」の説明に従って決められたシステム構成プロファイルを使用する
  - 対話型システム構成ツールでの管理者の応答を使用する
7. リポート後に、AI クライアントに対して設定された任意の初回ブートスクリプトが実行されます。

インストールが完了すると、自動インストールの成功メッセージが画面に表示され、完了メッセージが `/system/volatile/install_log` ファイルに表示され、その AI クライアント上の `svc:/application/auto-installer` SMF サービスが `online` 状態になります。

## SPARC および x86 AI クライアントのシステム要件

自動インストールを実行する AI 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。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 AI クライアントとして使用できます。

ネットワーク経由で AI インストールを実行する SPARC および x86 クライアントは、次の要件を満たしている必要があります。

メモリー	最小 1G バイト
ディスク容量	最小 13G バイト
ネットワークアクセス	AI クライアントは、インストール中に次のリソースにアクセスできる必要があります。 <ul style="list-style-type: none"> <li>■ ネットワーク構成情報を提供する DHCP サーバー</li> <li>■ AI サーバー</li> <li>■ AI クライアント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</li> </ul>

SPARC システム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。

AI では、ネットワーク経由でブートするために、SPARC クライアントに対する WAN ブートサポートが必要です。 `network-boot-arguments` が `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、AI 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 AI 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 AI クライアントをインストールすることはできません。『[Oracle Solaris 11.3 システムのインストール](#)』の第 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
```

## AI クライアントの設定

AI サーバー上で、特定の AI 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。

- AI クライアントの MAC アドレス
- AI クライアントがインストールに使用するインストールサービスの名前

x86 クライアントの場合は、必要に応じて、`installadm create-client` コマンドに `-b` オプションを使用してブートプロパティを指定できます。SPARC クライアントの場合、AI クライアントをブートするために使用するブート引数を、AI クライアントのブート時に入力するブートコマンドに含めます。

## SPARC AI クライアントの設定

次の例では、MAC アドレス `00:14:4f:a7:65:70` を持つ SPARC クライアントを `solaris11_3-sparc` インストールサービスに関連付けています。

```
# installadm create-client -n solaris11_3-sparc -e 00:14:4f:a7:65:70
```

SPARC の `wanboot-cgi` ブートファイルが `create-service` によってすでに構成されているため、DHCP サーバーでの構成は必要ありません。詳細は、[107 ページの「インストールサービスの作成」](#)を参照してください。

この `installadm create-client` コマンドの次の結果が `/etc/netboot` ディレクトリに表示されます。

```
dr-xr-x---  2 webservd webservd   4 Apr  9 08:53 0100144FA76570
```

## x86 AI クライアントの設定

次の例では、MAC アドレス 0:e0:81:5d:bf:e0 を持つ x86 クライアントを solaris11\_3-i386 インストールサービスに関連付けています。このコマンドによって出力される DHCP 構成を DHCP サーバーに追加する必要があります。この DHCP 構成が行われない場合、AI クライアントは solaris11\_3-i386 インストールサービスをブートできません。

```
# installadm create-client -n solaris11_3-i386 -e 0:e0:81:5d:bf:e0
No local DHCP configuration found. If not already configured, the
following should be added to the DHCP configuration:
  Boot server IP       : 10.80.239.5
  Boot file(s)        :
    bios clients (arch 00:00): 0100E0815DBFE0.bios
    uefi clients (arch 00:07): 0100E0815DBFE0.uefi
```

次の例は、installadm がこの AI クライアントの PXE ブートエントリを設定する方法を示しています。

```
host 00E0815DBFE0 {
  hardware ethernet 00:E0:81:5D:BF:E0;
  if option arch = 00:00 {
    filename "0100E0815DBFE0.bios";
  } else if option arch = 00:07 {
    filename "0100E0815DBFE0.uefi";
  }
}
```

この installadm create-client コマンドの次の結果が /etc/netboot ディレクトリに表示されます。

```
lrwxrwxrwx  1 root root   21 May  6 10:32 0100E0815DBFE0 -> ./0100E0815DBFE0.bios
lrwxrwxrwx  1 root root   44 May  6 10:32 0100E0815DBFE0.bios -> ./solaris11_3-i386/boot/
grub/pxegrub2
lrwxrwxrwx  1 root root   51 May  6 10:32 17:49 0100E0815DBFE0.uefi -> ./solaris11_3-i386/
boot/grub/grub2netx64.efi
-rw-r--r--  1 root root  1744 May  6 10:32 17:49 grub.cfg.0100E0815DBFE0
-rw-r--r--  1 root root  1204 May  6 10:32 17:49 menu.conf.0100E0815DBFE0
```

## サービスからの AI クライアントの削除

インストールサービスから AI 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 -e macaddr
```

AI 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。

## AI 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、AI クライアントをブートします。このセクションでは、SPARC または x86 クライアントをブートする方法について説明します。AI クライアントのコンソールでインストールの進行状況をモニターできます。インストール中に発生する可能性のあるエラーもクライアントのコンソールに表示されます。また、インストールの進捗状況をリモートからモニター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モニター

ssh を使用することで、自動 AI クライアントへのネットワークアクセスを有効にできます。このアクセスを使用して、`/system/volatile/install_log` インストールログファイルの進捗状況をモニターすることにより、進行中のインストールをリモートからモニタリングできます。

特定のインストールサービスのすべての AI クライアントでリモートアクセスを有効にするには、インストール構成ファイルでオプション `livessh` を `enable` に設定します。このアクセスが有効になっている場合は、ユーザー名 `jack` とパスワード `jack` を使用して AI クライアントにログインできます。

個々の AI クライアントで、ブートコマンド行にこのオプションを設定することもできます。

### x86 AI クライアントのインストールのモニタリング

x86 システムの場合は、次の例に示すように、`create-service` サブコマンドで `-b` オプションを使用して、そのサービスを使用するすべての AI クライアントのブートプロパティを設定します。

```
# installadm create-service -a i386 -b livessh=enable
```

次の抜粋は、このプロパティが `/etc/netboot/svname/grub.cfg` ファイルにどのように表示されるかを示しています:

```
$multiboot $kern /platform/i86pc/kernel/amd64/unix -B livessh=enable,...
```

1 つの x86 クライアントの ssh を有効にするには、ブートコマンド行に `livessh` を指定します。手順については、『[Oracle Solaris 11.3 システムのブートとシャットダウン](#)』の「[ブート時に GRUB メニューを編集してカーネル引数を追加する](#)」を参照してください。

## SPARC AI クライアントのインストールのモニタリング

SPARC システムの場合は、`/etc/netboot` ディレクトリ下にマウントされたサービスのネットイメージディレクトリ `/etc/netboot/svcname/system.conf` を介して `system.conf` ファイルにアクセスします。

`system.conf` ファイルでは、オプションは名前と値のペアとして定義されます。次の例では、`livessh` オプションは `enable` に設定されています。

```
$ cat /etc/netboot/solaris11_3-sparc/system.conf
...install_service=solaris11_3-sparc
install_svc_address=$serverIP:5555
livessh=enable
...
```

1つの SPARC クライアントの `ssh` を有効にするには、ブートコマンド行に `livessh` を指定します。次の例は、この引数を指定する 2 種類の方法を示しています。

```
ok boot net:dhcp - livessh
ok boot net:dhcp - Livessh=enable
```

ブートコマンド行での `livessh` の指定は、サービスの `system.conf` ファイルに指定されたすべての設定をオーバーライドします。たとえば、`system.conf` ファイルに `livessh=enable` が指定されている場合に、特定の AI クライアントで `livessh` を無効にするには、ブートコマンド行で `livessh=disable` を指定します。

```
ok boot net:dhcp - livessh=disable
```

## SPARC AI クライアントのインストール

OBP プロンプトから SPARC AI クライアントをネットワーク経由でブートします。セキュアなダウンロードを使用するか、DHCP を使用するかを決定します。

### セキュアなダウンロードを使用した SPARC AI クライアントのインストール

資格を使用してセキュリティ保護される SPARC AI クライアントの場合、ネットブートファイルとブートファイルシステムは、セキュリティキーを使用して構成される SPARC OBP ファームウェアを通じてネットワーク経由でセキュアにダウンロードできます。ダウンロードされるブートファイルおよびファイルシステムを検証するために、OBP にファームウェア鍵を指定する必要があります。

ハッシュダイジェスト (HMAC) は SHA1 アルゴリズムを使用して計算され、採用される暗号化メソッドは AES です。

## ハッシュ鍵と暗号化鍵の設定

OBP コマンドプロンプトで HMAC および暗号化鍵を設定できます。

次のコマンド例は、AI で生成された SHA1 値を使用して、SPARC クライアントコンソールで OBP HMAC を設定します。

```
ok set-security-key wanboot-hmac-sha1 767280bd72bca8cef3d679815dfca54638691ec5
```

次のコマンド例は、SPARC クライアントコンソール上で OBP AES 暗号化鍵を設定します。

```
ok set-security-key wanboot-aes 38114ef74dc409a161099775f437e030
```

## ハッシュ鍵と暗号化鍵の再設定

AI クライアントの OBP 鍵が AI サーバーの構成で再生成される場合、影響を受ける SPARC クライアント上では、認証された AI インストールを実行するために鍵を更新する必要があります。既存の OBP 鍵を無効化して新しい OBP 鍵を生成するには、`installadm` コマンドで `-H` および `-E` オプションを使用します。サーバー認証専用、特定 AI クライアント用、特定インストールサービス用、およびデフォルト AI クライアント用の OBP 鍵を生成することについては、[127 ページの「SPARC クライアントの OBP セキュリティキー」](#)を参照してください。

## ハッシュ鍵と暗号化鍵の削除

HMAC 鍵および暗号化鍵を削除すると、その AI クライアントは認証が不要になるか、認証を試行しなくなります。sec プロパティが `require-client-auth` または `require-server-auth` のいずれかに設定されたインストールサービスを使用して、AI を使用してクライアントをインストールすることができなくなります。

OBP コマンドプロンプトで HMAC 鍵および暗号化鍵を削除するには、鍵を設定するときに使用するものと同じコマンドを使用しますが、値を指定しないでください。

```
ok set-security-key wanboot-hmac-sha1
ok set-security-key wanboot-aes
```

## DHCP を使用した SPARC AI クライアントのインストール

DHCP を使用している場合は、次のネットワークブートコマンドを使用します。

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

## DHCP を使用しない SPARC AI クライアントのインストール

DHCP を使用しない場合、次のコマンドを使用して、OBP の `network-boot-arguments` 変数を設定します。この変数は、OBP に持続的に設定されます。

```
ok setenv network-boot-arguments host-ip=client-ip,  
router-ip=router-ip,subnet-mask=subnet-mask,hostname=hostname,  
file=wanboot-cgi-file
```

次に、次のコマンドを使用して AI クライアントをネットワーク経由でブートします。

```
ok boot net - install
```

---

**注記** - `network-boot-arguments` 変数を使用するときは、SPARC クライアントに DNS の構成情報がありません。この AI クライアントで使用されている AI マニフェストに、IPS パッケージリポジトリの場所やマニフェスト内のその他の URI として、ホスト名ではなく IP アドレスが指定されていることを確認してください。

---

## SPARC AI 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. AI クライアントがブートし、そのネットワーク構成と `wanboot-cgi` ファイルの場所を DHCP サーバーまたはその OBP に設定された `network-boot-arguments` 変数から取得します。
2. `wanboot-cgi` プログラムが `wanboot.conf` を読み取り、WAN ブートバイナリの場所を AI 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、AI 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 がブートします。
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、AI クライアントへの Oracle Solaris OS のインストールを実行します。

### ▼ OBP からブートディスクを設定する方法

通常、AI サーバーを使用して AI クライアントをインストールする場合は、AI マニフェスト内でインストール先のディスクを選択しています。インストール先のディス

クが定義されていない場合は、`boot-device` OBP パラメータがチェックされます。このパラメータが設定されていない場合は、大きさが十分な最初のディスクが使用されます。マニフェスト内でディスクが選択されていないときに AI が OS を間違ったディスク上に配置しないようにするには、`boot-device` OBP パラメータを設定します。

1. システムを `ok PROM` プロンプトにします。

```
# init 0
```

2. システム上のデバイスを一覧表示します。

```
ok devalias
...
disk1          /pci@306/pci@1/SUNW,qlc@0/fp@0,0/disk@w202400a0b836a3b9,3
disk0          /pci@306/pci@1/SUNW,qlc@0/fp@0,0/disk@w202400a0b836a3b9,1
disk           /pci@304/pci@2/usb@0/storage@1/disk@0,0
```

3. `boot-device` パラメータを適切なディスクに設定します。

```
ok setenv boot-device /pci@306/pci@1/SUNW,qlc@0/fp@0,0/disk@x202400a0b836a3b39,1
```

4. (オプション) ブートデバイスを確認します。

```
ok printenv boot-device
boot-device = /pci@306/pci@1/SUNW,qlc@0/fp@0,0/disk@x202400a0b836a3b39,1
```

5. システムをブートして AI インストールを開始します。

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

## x86 AI 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 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。

AI クライアントがブートしたら、ブートするネットワークデバイスを選択します。

x86 クライアントの AI ブート中に、次のイベントが発生します。

1. AI クライアントがブートして IP アドレスを取得し、ブートファイルが DHCP サーバーによって提供された場所からダウンロードされます。
2. ブートファイルが読み込まれ、GRUB メニューファイルを読み取ります。
3. ユーザーが GRUB メニューから 2 番目のオプションである「Oracle Solaris 11.3 Automated Install」を選択します。

4. ブートファイルがブートアーカイブファイルを取得し、Oracle Solaris OS が TFTP を使用してブートされます。
5. GRUB メニューによって提供される HTTP を使用して、ネットイメージのアーカイブ solaris.zlib および solarismisc.zlib がダウンロードされます。
6. AI マニフェストとシステム構成プロファイルが、mDNS 検索またはブートされた GRUB メニューエントリで指定された AI インストールサービスからダウンロードされます。
7. AI マニフェストで AI インストールプログラムが呼び出され、インストールが実行されます。

システムの PXE ブートが正常に行われると、次のメッセージが少しの間表示されてから、GRUB メニューが表示されます。

```
Intel(R) Boot Agent PXE Base Code (PXE-2.1 build 0.86)
Copyright(C) 1997-2007, Intel Corporation
```

```
CLIENT MAC ADDR 00 14 4F 29 04 12 GUID FF2000008 FFFF FFFF FFFF 7BDA264F1400
CLIENT IP: 10.6.68.29 MASK: 255.255.255.0 DHCP IP: 10.6.68.49
GATEWAY: 10.6.68.1
```

GRUB メニューに 2 つのメニューエントリが表示されます。自動インストールを開始するには、2 番目のエントリを選択します。

```
Oracle Solaris 11.3 Text Installer and command line
Oracle Solaris 11.3 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、AI クライアントがブートされると、[273 ページの「コマンド行から自動インストールを開始する」](#)に示すような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。

## boot コマンドのインストールオプション

ブートプロンプトから、AI クライアントのブート時に次のオプションを使用できます。

install	自動インストールセッションを開始します。
livessh	ミニルートがブートされるときに ssh を有効にします。例については、 <a href="#">251 ページの「Secure Shell を使用したりリモートからのインストールのモニター」</a> を参照してください。
aimanifest	AI クライアントマニフェストへの URI を指定します。

install-env-profile	ミニルートで (たとえば、ネットワークまたはネームサービスを設定するために) 使用されるプロファイルへの URI を指定します。
profile	AI クライアントプロファイルへの URI を指定します。

## AI 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

AI 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モニターできます。

### 自動インストールの成功メッセージ

次のメッセージが表示された場合は、インストールが成功しました。

```
Automated Installation finished successfully
The system can be rebooted now
Please refer to the /system/volatile/install_log file for details
After reboot it will be located at /var/log/install/install_log
```

Reboot to start the installed system

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、AI クライアントは自動的にリブートしません。



# ◆◆◆ 第 15 章

## 自動インストールのトラブルシューティング

---

この章では、考えられるいくつかの障害と回復の方法について説明します。

### AI 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、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

AI クライアント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 AI 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 AI 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。

AI サーバーの Web サーバーのログを `/var/ai/image-server/logs/*` で確認することもできます。このログは、AI クライアントが予期しない AI マニフェストやシステム構成プロファイルまたは間違った AI マニフェストやシステム構成プロファイルを受け取った場合、または初期ブートフェーズ後に AI クライアントがファイルをダウンロードできない場合に役に立ちます。

#### 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、AI 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 AI クライアントに提供していることを確認します。

```
# /sbin/dhclientd 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 AI クライアントに提供するように設定されていません。DHCP 管理者に問い合わせるこの問題を修正してください。

`/etc/resolv.conf` ファイルが存在していて、正しく構成されている場合は、考えられる次の問題がないか確認して、システム管理者に解決方法を問い合わせてください。

- DNS サーバーが IPS リポジトリのサーバー名を解決していない可能性があります。
- DNS サーバーに到達するデフォルトルートが存在しません。

## AI クライアントブートエラーの確認

AI クライアント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。

- [261 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」](#)
- [265 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」](#)
- [268 ページの「SPARC および x86 のエラーメッセージ」](#)

## ブートディスクが見つからない

自動インストール中にブートディスクが見つからない場合は、ブートディスクを確認して AI マニフェストを変更します。

1. SPARC の OBP または x86 の BIOS でブートデバイスを明示的に選択します。
2. システムをリブートします。
3. インストールされるシステムにログインします。
4. インストール中に使用されるデバイスを特定します。このデバイスは、`format` コマンドで表示される `SYS/HDD*` 受容体名または `CTD` ディスク名によって識別できます。
5. `/system/volatile/ai.xml` マニフェストを変更して、「`boot_disk`」値を置き換えます。次に例を示します。

```
<disk_keyword key="SYS/HDD1" name_type="receptacle"/>
```

```
<disk_keyword key="c0t5000CCA012B2A254d0" name_type="ctd"/>
```

6. インストーラサービスをリフレッシュします。

```
# svcadm clear auto-installer
```

この問題に関連して、UEFI モードでインストールする場合は、AI マニフェストのターゲット `boot_disk` セクションにディスクが指定されていても、ブートディスクが見つかりません。UEFI インストールでのこの制限は、システムにまだ何もインストールされていない場合には、マニフェスト内の `boot_disk` セクションを使用してブートディスクを識別できないことを意味します。回避方法として、派生マニフェストを使用して、代替のインストール先ディスクを指定します。

## インストール後に X86 システムがブートしない

UEFI 仕様では、EFI 保護 MBR パーティションをアクティブとしてマークすることが禁止されていますが、BIOS の実装では、BIOS モードでブートするには、少なくとも 1 台のハードディスクにブート可能/アクティブとしてマークされた MBR パーティションが少なくとも 1 つ存在する必要があります。fdisk ユーティリティを使用して、このパーティションのステータスを「ACTIVE」に設定します。

## SPARC のネットワークブートエラーと考えられるエラー原因

このセクションでは、ネットワーク経由で SPARC クライアントをブートしたときに起こり得るエラーまたは問題と、考えられる原因について説明します。

- [261 ページの「BOOTP/DHCP 応答を待機してタイムアウトになった」](#)
- [262 ページの「ブートの読み込みに失敗」](#)
- [262 ページの「内部サーバーのエラーまたは WAN ブートの警告」](#)
- [263 ページの「「ERROR 403: Forbidden」または「ERROR 404: Not Found」](#)」
- [264 ページの「自動インストーラが開始されない」](#)
- [264 ページの「無効な HMAC 値」](#)

### 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
```

タイムアウトのメッセージは、AI クライアントが DHCP リクエストを送信し、そのリクエストに対する応答がないことを示しています。このエラーは、DHCP 構成問題によって発生している可能性があります。DHCP サーバーで AI 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、AI クライアントの 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 AI 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 AI クライアントの DHCP 構成を確認してください。構成が正しいと思われる場合は、サブネットに別の DHCP サーバーがあるかどうかを確認してください。

## 内部サーバーのエラーまたは WAN ブートの警告

AI クライアントが、ブートアーカイブのダウンロードを開始するための IP アドレスと始値を取得したあと、クライアントが `boot_archive` を検出またはダウンロードできない場合があります。

- AI 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
Mon Aug 5 20:46:43 wanboot alert: miniinfo: Request returned code 500
Mon 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
Mon Aug 5 20:53:02 wanboot alert: miniroot: Request returned code 403
Mon 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 AI 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。

## 「ERROR 403: Forbidden」または「ERROR 404: Not Found」

「ERROR 403: Forbidden」および「ERROR 404: Not Found」メッセージは、AI クライアントが `boot_archive` を正常にダウンロードして Oracle Solaris カーネルをブートしたあと、イメージアーカイブの 1 つを取得できなかった場合に表示されます。エラーメッセージには、どのファイルが問題を引き起こしているのかが表示されます。たとえば、次の SPARC クライアントでの出力では、`solaris.zlib` ファイルが存在しないか、指定した場所でアクセスできません。

```

<time unavailable> wanboot info: Starting DHCP configuration
<time unavailable> wanboot info: DHCP configuration succeeded
<time unavailable> wanboot progress: wanbootfs: Read 368 of 368 kB (100%)
<time unavailable> wanboot info: wanbootfs: Download complete
Mon May 5 18:57:36 wanboot progress: miniroot: Read 235737 of 235737 kB (100%)
Mon May 5 18:57:36 wanboot info: miniroot: Download complete
SunOS Release 5.11 Version 11.3 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2015-05-05 18:52:30-- http://10.134.125.136:5555/export/auto_install/11_3_sparc/solaris.zlib

```

```
Connecting to 10.134.125.136:5555... connected.  
HTTP request sent, awaiting response... 404 Not Found  
2015-05-05 18:52:30 ERROR 404: Not Found.
```

```
Could not obtain http://10.134.125.136:5555/export/auto_install/11_3_sparc/solaris.zlib  
from install server  
Please verify that the install server is correctly configured and reachable from the AI  
client
```

この問題は、次のいずれかの状態によって引き起こされる場合があります。

- WAN ブートで構成されたイメージパスが正しくない。
- イメージパスが存在しないか不完全である。
- アクセス権の問題により、アクセスが拒否される。

DHCP の構成、または `installadm create-service` を実行したときに指定したネットイメージの内容を確認してください。WAN ブート構成を確認してください。

## 自動インストーラが開始されない

Oracle Solaris OS を AI クライアントにインストールする場合は、インストールを開始するためにブートするときに `install` 引数を含める必要があります。

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

`install` ブート引数なしでブートすると、SPARC クライアントで自動インストーラのブートイメージがブートされますが、インストールは開始されません。この時点から自動インストールを開始する方法については、[273 ページの「コマンド行から自動インストールを開始する」](#)を参照してください。

## 無効な HMAC 値

SPARC の AI クライアントをブートした直後に SPARC のコンソールに「Invalid HMAC value」というメッセージが表示され、システムが `ok` プロンプトに戻った場合は、次のいずれかの状況が問題の原因です。

- AI クライアントは認証によってセキュリティー保護されていますが、OBP 鍵が設定されていません。解決するには、クライアントのファームウェアに OBP 鍵を設定します。認証の詳細は、[118 ページの「自動インストールのセキュリティーの向上」](#)を参照してください。OBP 鍵の設定の詳細については、[252 ページの「セキュアなダウンロードを使用した SPARC AI クライアントのインストール」](#)を参照してください。
- AI クライアントはセキュリティー保護されていませんが、OBP 鍵が設定されています。解決するには、クライアントのファームウェアの OBP 鍵を設定解除します。[253 ページの「ハッシュ鍵と暗号化鍵の再設定」](#)を参照してください。

- AI クライアントの認証を要求するポリシーがクライアントのインストールサービスにあるが、クライアントに適用可能な資格情報が割り当てられていない。ポリシー `require-client-auth` を持つすべての AI クライアントのサービスに、利用可能な資格情報があることを確認します。

次の手順は、問題を特定する方法を示しています。

1. AI サーバーでセキュリティーが無効にされていないことを確認します。 `installadm list -sv` を使用して、セキュリティーが有効にされているかどうかを確認します。
2. AI クライアントのインストールサービスでセキュリティーが無効にされていないことを確認します。 `installadm list -vn svcname` を使用して、セキュリティーが無効になっていないかどうかを確認します。
3. AI クライアントがカスタム資格情報を使用している場合は、 `installadm list -ve macaddr` を使用してファームウェア鍵の値を取得します。
4. AI クライアントがカスタムクライアントではない場合は、 `installadm list -vn default-sparc` を使用して `default-sparc` サービスに定義されているファームウェアの鍵があるかどうかを確認します。
5. `installadm list -vn svcname` を使用して AI クライアントのサービスのポリシーを確認します。
6. `default-sparc` サービスの資格情報がない場合は、 `installadm list -sv` コマンドを使用してデフォルトクライアントの資格情報を探します。デフォルトのクライアントの資格情報がある場合は、デフォルトの AI クライアントにリストされているファームウェアを使用します。
7. デフォルトクライアントの資格情報がない場合は、 `installadm list -vn default-sparc` を使用してサービスのポリシーに `require-server-auth` が設定されているかどうかを確認します。設定されている場合は、 `installadm list -sv` でデフォルトの AI クライアントにリストされるファームウェアの鍵を使用します。

## 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、ネットワーク経由で x86 クライアントをブートしたときに起こり得るエラーまたは問題と、考えられる原因について説明します。

- [266 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」](#)
- [266 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」](#)
- [267 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」](#)
- [267 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」](#)

- [268 ページの「自動インストーラが開始されない」](#)

## 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 Proxy DHCP offers were received
PXE-MOF: Exiting Intel Boot Agent
```

タイムアウトメッセージは、AI 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 AI 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

AI 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。

AI サーバーで、この AI クライアントの `grub.cfg` ファイルまたは `menu.lst` ファイルが有効なブートアーカイブを指しているかどうかを確認してください。AI サーバー上のイメージのブートディレクトリは、`installadm list` で表示されたイメージパスに対する `df -k` のこの抜粋例に示すように、`/etc/netboot` ディレクトリの下にループバックマウントされます。

```
Filesystem      1K-blocks      Used Available Use% Mounted on
/export/auto_install/solaris11_3-i386
  92052473  36629085  55423388  40% /etc/netboot/default-i386
/export/auto_install/solaris11_3-i386
  92052473  36629085  55423388  40% /etc/netboot/solaris11_3-i386
```

## 送信した HTTP 要求の結果が 403 Forbidden または 404 Not Found になる

AI サーバー上で、`/etc/netboot` 下の `grub.cfg` ファイルまたは `menu.lst` ファイルで指定された場所にあるインストールプログラムのいずれかがアクセス不可能であるか、またはその場所に存在しない場合、AI クライアントはブートできますが、そのファイルをダウンロードできません。エラーメッセージには、どのファイルが問題を引き起こしているのかが表示されます。たとえば、次の x86 クライアント上の出力では、`solaris.zlib` ファイルが指定の場所に存在しません。

```
SunOS Release 5.11 Version 11.3 64-bit
Copyright (c) 1983, 2015, Oracle and/or its affiliates. All rights reserved.
Remounting root read/write
Probing for device nodes ...
Preparing network image for use
Downloading solaris.zlib
--2015-05-05 20:02:26-- http://10.134.125.136:5555/export/auto_install/solaris11_3-i386/
solaris.zlib
Connecting to 10.134.125.136:5555... connected.
HTTP request sent, awaiting response... 404 Not Found
2015-05-05 20:02:26 ERROR 404: Not Found.

Could not obtain http://10.134.125.136:5555/export/auto_install/solaris11_3-i386/
solaris.zlib from install server
Please verify that the install server is correctly configured and reachable from the client

Requesting System Maintenance Mode
(See /lib/svc/share/README for more information.)
Console login service(s) cannot run
```

`installadm create-service` コマンドを実行したときに指定したターゲットディレクトリの内容を確認してください。

## 自動インストーラが開始されない

Oracle Solaris OS を x86 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 11.3 Text Installer and command line
Oracle Solaris 11.3 Automated Install
```

最初の GRUB メニューエントリを選択した場合、またはプロンプトがタイムアウトした場合は、システムで自動インストールのブートイメージがブートされますが、インストールは開始されません。この時点から自動インストールを開始する方法については、[273 ページの「コマンド行から自動インストールを開始する」](#)を参照してください。

## SPARC および x86 のエラーメッセージ

次のエラーは、SPARC と x86 の両方のインストールに共通しています。

- [268 ページの「自動インストール失敗メッセージ」](#)
- [268 ページの「IPS サーバーを使用できない」](#)
- [270 ページの「パッケージが見つからない」](#)

### 自動インストール失敗メッセージ

インストール中に障害が発生した場合、次のメッセージが表示されます。

```
21:43:34 Automated Installation Failed. See install log at /system/volatile/install_log
Automated Installation failed
Please refer to the /system/volatile/install_log file for details
Jul 6 21:43:34 solaris svc.startd[9]: application/auto-installer:default failed fatally:
transitioned to maintenance (see 'svcs -xv' for details)
```

### IPS サーバーを使用できない

Oracle Solaris OS をインストールするには、AI マニフェストに定義されている IPS パッケージリポジトリにクライアントがアクセスする必要があります。AI 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 21 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 21 15:54:47 UTC 2013
PLATFORM: SUNW,Sun-Fire-V445, CSN: -, HOSTNAME: line2-v445
SOURCE: software-diagnosis, REV: 0.1
EVENT-ID: c8a5b809-ece4-4399-9646-d8c64d78aac7
DESC: A service failed - a start, stop or refresh method failed.
AUTO-RESPONSE: The service has been placed into the maintenance state.
IMPACT: svc:/application/auto-installer:default is unavailable.
REC-ACTION: Run 'svcs -xv svc:/application/auto-installer:default' to determine the generic
  reason
why the service failed, the location of any logfiles, and a list of other services
  impacted. Please
refer to the associated reference document at http://support.oracle.com/msg/SMF-8000-YX for
  the latest service
procedures and policies regarding this diagnosis.
```

`/system/volatile/install_log` ファイルで次のようなメッセージを確認してください。

```
TransportFailures: Framework error: code: 6 reason: Couldn't resolve host
'pkg.example.com'
URL: 'http://pkg.example.com/solaris/versions/0/'

TransportFailures: Framework error: code: 7 reason: Failed connect to
pkg.example.com:80; Connection refused
URL: 'http://pkg.example.com/solaris/versions/0/'

TransportFailures: http protocol error: code: 404 reason: Not Found
URL: 'http://pkg.oracle.com/mysolaris/versions/0/'
```

表示されるメッセージに応じて、次の考えられる対処法を試してください。

- ping を使用するなどして、失敗した AI クライアント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。259 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 AI 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

11.3 パッケージリポジトリのコピーと作成』の第3章、「リポジトリへのアクセスの提供」を参照してください。

- AI マニフェスト内の URI に誤字がないことを確認します。
- 次のようなコマンドを使用して、パッケージリポジトリが有効かどうかを確認します。

```
$ pkg list -g http://pkg.example.com/solaris/ entire
```

カタログをリフレッシュしたり、索引を再作成したりすることが必要な場合もあります。

## パッケージが見つからない

AI マニフェストで指定されたパッケージのいずれかが、IPS リポジトリで見つからない場合、ディスクにパッケージをインストールする前にインストーラが失敗します。次の例では、インストーラがパッケージ `mypkg` を IPS リポジトリで見つけることができませんでした。次の出力は、コンソールに表示される内容の例です。

```
14:04:02      Failed Checkpoints:
14:04:02      generated-transfer-1230-1
14:04:02      Checkpoint execution error:
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      pkg:/mypkg
14:04:02      Automated Installation Failed. See install log at /system/volatile/install_log
```

次の出力は、`/system/volatile/install_log` ログファイルの一部分の例です。

```
PlanCreationException: The following pattern(s) did not match any allowable packages.
Try using a different matching pattern, or refreshing publisher information:
```

```
pkg:/mypkg
```

問題のパッケージが有効なパッケージかどうかを確認してください。別の IPS リポジトリからこのパッケージを入手できる場合は、`source` 要素に `publisher` 要素をもう1つ追加して、その IPS リポジトリを AI マニフェストに追加します。

## セキュアな AI クライアントでのブートエラー

AI クライアントをブートするときに次のようなメッセージが表示される場合は、TLS 証明書がまだ有効ではないことを意味しています。

```
SSL3_GET_RECORD:wrong version number - secure HTTPS GET REQUEST to unsecured HTTP port
```

この問題の原因は、AI クライアントのシステム時間が証明書が生成された時間より前であるためである可能性があります。AI クライアントのシステム時間を確認してください。セキュリティー資格情報を生成して割り当てる方法の詳細は、[118 ページの「自動インストールのセキュリティーの向上」](#)を参照してください。

## セキュリティー関連の AI の障害

[118 ページの「自動インストールのセキュリティーの向上」](#)に説明されているように AI サーバーと AI クライアントをセキュリティー保護し、それらの AI クライアントのブートまたはインストールで問題が発生した場合は、次の手順を試して認証エラーをチェックしてください。

- AI クライアントの `/var/ai/image-server/logs/` にある Apache の `access_log` および `error_log` を確認します。
- AI クライアントのコンソールにログオンします。`/tmp/install_log` ファイルおよび `/system/volatile/` にある SMF サービスのログを検査します。
- ブートアーカイブを AI クライアントにロードしたあとに、イメージファイル、AI マニフェスト、またはシステム構成プロファイルを取得しようとして認証が失敗した場合は、ネットワークが一時的に中断している可能性があります。AI サーバーが正常に機能していることを確認し、インストールを再開してください。
- `openssl s_client` コマンドを使用して、接続をテストしてください。

```
$ openssl s_client -key client-key -cert client-certificate \
-Cacert server-CA-certificate -connect AI-server-address:port
```

- `installadm list -s -v` コマンドを使用して、AI サーバーのセキュリティーの状態が有効または無効のいずれであるかを表示します。[例48「AI サーバーのセキュリティー情報の一覧表示」](#)を参照してください。
- `installadm list -v -n svcnameS` コマンドを使用して AI クライアントのサービスポリシーを確認します。
- CA 証明書に対して割り当てられている資格情報を確認します。`installadm list` サブコマンドに `-k` および `-c` オプションを指定して使用し、割り当てられている鍵および証明書を一覧表示します。`diff` などの文字比較ユーティリティーを使用して、それらの鍵および証明書を予期されている鍵および証明書と比較します。
- パスフレーズが AI クライアントの `/var/ai/ai-webserver/tls.key/server.key` から削除されていることを確認します。X.509 非公開鍵ファイル内のパスフレーズは削除されている必要があります。
- 次の例のように適切な鍵、証明書、および CA 証明書を使用し、`wget` コマンドを使用して AI イメージからファイルを取得してください。

```
$ wget --private-key=client-key --certificate=client-certificate \
--ca-certificate=server-CA-certificate \
http://AI-server-address:5555/path-to-file-in-image
```

## 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。

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 11.3 Text Installer and command line
Oracle Solaris 11.3 Automated Install
```

デフォルトのエントリ「テキストインストーラおよびコマンド行」では、イメージはブートされますが、手入力なしの自動インストールは開始されません。

そのエントリのカーネル行に `install=true` ブートプロパティが指定されていないことを確認してください。

メディアからの x86 クライアントのブート

メディアから x86 システムをブートしたあと、インストールを開始しない場合は、GRUB メニューを編集して、ブートするエントリのカーネル行から `install=true` ブートプロパティを削除します。

一般に、x86 インストールでは、ブート元の GRUB エントリのカーネル行に `install=true` ブートプロパティを指定した場合は、インストールが自動的に開始されます。自動インストールを開始せずに x86 ベースのシステムをブートする場合は、GRUB ブートエントリに `install=true` ブートプロパティが指定されていないことを確認します。このプロパティが指定されている場合は、『[Oracle Solaris 11.3 システムのブートとシャットダウン](#)』の「[ブート時に GRUB メニューを編集してカーネル引数を追加する](#)」の説明に従ってブートエントリを編集し、このプロパティを削除します。

AI クライアントがブートされると、273 ページの「[コマンド行から自動インストールを開始する](#)」に示されている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。

## コマンド行から自動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 xterm)
5 Reboot
```

Please enter a number [1]:

オプション 3 を選択してシェルを開きます。

シェルが実行されたら、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 svc:/application/manifest-locator:default
$ svcadm enable svc:/application/auto-installer:default
```



## パート IV

### 関連タスクの実行

このセクションでは、インストールプロセスに関連したトピックについて説明します。この章の内容は次のとおりです。

- [付録A Oracle Configuration Manager の操作](#)
- [付録B デバイスドライバユーティリティの使用](#)



## Oracle Configuration Manager の操作

---

この章では、Oracle Configuration Manager の概要に加えて、Oracle Solaris リリースを実行しているシステム上でこのサービスを使用する手順について説明します。

### Oracle Configuration Manager の概要

Oracle Configuration Manager は、システムの構成情報を収集し、それを Oracle リポジトリにアップロードするために使用します。この情報のコレクタは、OCM サーバー上のすべての製品に関する情報を収集する中央コレクタとして構成することも、個々の収集サイト内の情報を収集するように構成することもできます。詳細は、[278 ページの「Oracle Configuration Manager の中央コレクタについて」](#)を参照してください。

顧客サポート担当者はこの情報を利用して、より適切なサービスを提供できます。Oracle Configuration Manager を使用する利点のいくつかを次に示します。

- サポート問題の解決時間を短縮します
- 予防的な問題回避を実現します
- ベストプラクティスや Oracle ナレッジベースへのアクセスが向上します
- 顧客のビジネスニーズへの理解を高め、一貫した対応とサービスを提供します

Oracle Configuration Manager は、接続または切断のいずれかのモードで実行できます。切断モードは、システムがインターネットに接続しておらず、Oracle Support Hub を構成できない場合にのみ必要です。このモードでは、構成情報を手動で収集し、サービスリクエストのために Oracle に情報をアップロードできます。

接続モードでは、次のような複数のネットワーク構成で Oracle Configuration Manager を実行できます。

- システムをインターネットに直接接続できます。
- プロキシサーバー経由でシステムをインターネットに接続できます。
- システムはインターネットに直接アクセスできませんが、Oracle Support Hub 経由でインターネットに接続するイントラネットプロキシサーバーにアクセスできます。

- システムはインターネットに直接アクセスできませんが、プロキシサーバー経由でインターネットに接続する Oracle Support Hub にアクセスできます。

Oracle Configuration Manager の設定および構成の詳細は、『[Oracle Configuration Manager インストールおよび管理ガイド](#)』を参照してください。このドキュメントの残りでは、Oracle Configuration Manager に関連する Oracle Solaris タスクに焦点を当てます。

---

**注記** - プロキシまたは Oracle Support Hub を使用するように Oracle Configuration Manager を構成するには、`configCCR` コマンドを対話型モードで実行する必要があります。詳細は、『[Oracle Support Hub](#)』を参照してください。

---

Oracle Solaris 11 のインストール中、ソフトウェアは Oracle リポジトリへの匿名接続を設定しようとしています。成功した場合は、情報の提供を一切求められることなく、この接続によりインストールプロセスを進めることができます。可能であれば、登録またはネットワーク構成の変更は、システムが完全にインストールされたあとで行うようにしてください。匿名で読み込まれたデータはどの組織にも結び付けられません。ソフトウェアが Oracle リポジトリに接続できなかった場合は、システムを手動で登録してから、Oracle Configuration Manager サービスを有効化できます。

## Oracle Configuration Manager の中央コレクタについて

Oracle Solaris オペレーティングシステムの一部としてインストールされた Oracle Configuration Manager コレクタは、中央コレクタとして構成され、指定されます。Oracle Configuration Manager コレクタの利点 (パーソナライズされたサポートエクスペリエンス、サポート問題の迅速な解決、予防的な問題回避など) を得るには、個々の Oracle インストールの構成データを収集してアップロードする必要があります。これは、通常、Oracle ホームにインストールされたコレクタのタスクです。しかし、Oracle ホームのコレクタが構成されていない場合や、切断されたままになっている場合もあります。中央コレクタの目的は、これらの Oracle ホームを収集し、それらを独自の My Oracle Support (MOS) 資格でアップロードすることです。中央コレクタには次のような特徴があります。

- 中央コレクタは次を収集します。
  - 自身が配置されている Oracle ホーム
  - 構成済みのコレクタが存在しないシステム上の Oracle ホーム
  - コレクタが切断モードになっている Oracle ホーム
  - コレクタが登録を認証している Oracle ホーム

Oracle ホームのコレクタが `ORACLE_CONFIG_HOME` の指定を使用して構成されている場合、中央コレクタはそのホームを収集しません。

- コレクタインストールを中央コレクタとして指定するには、`root` の役割を使用して、`setupCCR` および `configCCR` コマンドに `-c` オプションを指定します。その

後、`-c` オプションを指定せずに `configCCR` コマンドを実行すると、コレクタから中央コレクタの指定が破棄されます。`-c` オプションを指定して `setupCCR` および `configCCR` コマンドを実行すると、コレクタが中央コレクタとして指定されます。Oracle Solaris オペレーティングシステムの一部としてインストールされたコレクタは、`root` アクセス権を使用してインストールされるため、システムの中央コレクタとして動作します。

- Oracle Universal Installer の中央インベントリは、中央コレクタが収集する Oracle ホームの候補のセットを取得するソースです。中央インベントリは、ドキュメントで説明されているように、インストーラによって検索されます。Oracle Solaris オペレーティングシステムのインストーラ中央インベントリポイントのデフォルトの場所は、`/var/opt/oracle/oraInst.loc` です。Oracle インストールのインベントリを別の場所に配置するように選択すると、中央インベントリはそれを検索および収集できません。
- このリリースでは、Oracle Solaris OS の構成情報のほかに、Oracle WebLogic を使用する Oracle Database および Oracle Fusion Middleware ベースの製品のみが中央コレクタによって収集されます。
- 中央コレクタが Oracle ホームから収集した構成データはすべて、中央コレクタの My Oracle Support 資格を使用してアップロードされます。

## Oracle Configuration Manager の管理

次のタスクマップには、Oracle Solaris システム上での Oracle Configuration Manager の使用に関連する複数の手順が含まれています。

タスク	説明	手順の参照先
Oracle Configuration Manager サービスを有効にします。	構成変更を行なったあとで、Oracle Configuration Manager サービスを有効にします。	<a href="#">280 ページの「Oracle Configuration Manager サービスを有効にする方法」</a>
Oracle Configuration Manager サービスを無効にします。	重大な構成変更を行う前に、Oracle Configuration Manager サービスを無効にします。	<a href="#">280 ページの「Oracle Configuration Manager サービスを無効にする方法」</a>
システムを Oracle リポジトリに手動で登録します。	登録資格を変更します。	<a href="#">280 ページの「システムを Oracle リポジトリに手動で登録する方法」</a>
データ収集時間を変更します。	データ収集の頻度と時間を再設定します。	<a href="#">281 ページの「Oracle Configuration Manager のデータ収集の時間と頻度を変更する方法」</a>

## ▼ Oracle Configuration Manager サービスを有効にする方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. Oracle Configuration Manager サービスを有効にします。

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

## ▼ Oracle Configuration Manager サービスを無効にする方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. Oracle Configuration Manager サービスを無効にします。

```
# svcadm disable system/ocm
```



---

注意 - Oracle Solaris システム上で `emCCR stop` コマンドを実行しないでください。サービスに対するすべての変更は、Oracle Solaris のサービス管理機能 (SMF) を使用して行う必要があります。

---

## ▼ システムを Oracle リポジトリに手動で登録する方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. ユーザー登録を変更します。

```
# configCCR
```

Oracle Configuration Manager ソフトウェアにより、電子メールアカウントとパスワードの指定を求められます。お使いの My Oracle サポート ID に関連付けられた電子メールアカウントを使用することをお勧めします。

可能な場合、システムは登録サーバーと直接通信します。そうでない場合は、Oracle Support Hub の URL の指定を求められます。サイトで URL が使用可能な場合は、それをここに指定します。Oracle Support Hub の URL を指定しなかった場合、または登録サーバーとまだ通信できない場合は、ネットワークプロキシの指定を求められます。

登録が完了すると、データ収集が開始されます。

参照 詳細は、[configCCR\(1M\)](#) のマニュアルページまたは [Oracle Configuration Manager Installation and Administration Guide](#) を参照してください。configCCR コマンドを使用した対話型セッションの完全な例については、「[configCCR](#)」のページを参照してください。

## ▼ Oracle Configuration Manager のデータ収集の時間と頻度を変更する方法

### 1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

### 2. データ収集の頻度を再設定します。

この例では、データ収集時間を毎週月曜日の午前 6:00 に再設定します。

```
# emCCR set collection_interval=FREQ=WEEKLY\; BYDAY=MON\; BYHOUR=6
```

参照 詳細は、[emCCR\(1M\)](#) のマニュアルページまたは [Oracle Configuration Manager Installation and Administration Guide](#) を参照してください。



## デバイスドライバユーティリティの使用

---

Oracle デバイスドライバユーティリティ (DDU) は、インストール済みのシステムで検出されたデバイスが最新のリリースによってサポートされているかどうかを報告します。

### デバイスドライバユーティリティの概要

デバイスドライバユーティリティは、インストール済みのシステム上のデバイスとそれらのデバイスを管理するドライバに関する情報を提供します。DDU は、システムで検出されたすべてのデバイスに対するドライバが現在ブートしているオペレーティングシステムに存在するかどうかを報告します。デバイスにドライバが関連付けられていない場合、デバイスドライバユーティリティは、インストールするドライバパッケージを推奨します。

デバイスドライバユーティリティを使用すると、使用しているシステムの情報を HCL (<http://www.oracle.com/webfolder/technetwork/hcl/index.html>) に送信することもできます。情報を送信すると、使用しているシステムとそのコンポーネントが「Reported to Work」として HCL に掲載されます。

このセクションでは、次のタスクについて説明します。

- 283 ページの「デバイスドライバユーティリティを起動する方法」
- 284 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」
- 286 ページの「HCL にシステムを一覧表示させる方法」

### ▼ デバイスドライバユーティリティを起動する方法

デバイスドライバユーティリティは、インストール済みのシステムをブートすると自動的に実行されます。Oracle Solaris OS をインストールしたあとで、手動でデバイスドライバユーティリティを起動することもできます。

- 次のいずれかの方法でデバイスドライバユーティリティを起動します。

- Oracle Solaris テキストインストーライメージをブートします。

テキストインストーラからデバイスドライバユーティリティを起動するには、初期メニューから「Install Additional Drivers」を選択します。

---

**注記** - テキストインストーラがブートすると、デフォルトで自動ネットワーク接続が設定されます。DHCP を使用している場合、デバイスドライバユーティリティを使用するための追加のネットワーク設定は不要です。DHCP を使用していない場合は、初期メニューで「シェル」オプションを選択し、デバイスドライバユーティリティを使用する前に、適切なコマンドを使ってネットワーク設定を手動で構成します。

---

- インストール先システムでデバイスドライバユーティリティを起動します。

インストール先システムのデスクトップからデバイスドライバユーティリティを起動するには、メインメニューから「アプリケーション」→「システムツール」→「デバイスドライバユーティリティ」を選択します。

デバイスドライバユーティリティがシステムをスキャンし、検出されたデバイスのリストを表示します。このリストでは、デバイスを現在管理しているドライバの製造元、モデル、名前などの情報が、検出されたデバイスごとに表示されます。

次の手順    ドライバが関連付けられていないデバイスをユーティリティが検出した場合、デバイスリストでそのデバイスが選択されます。そのデバイスの詳細情報を表示したり、不足しているドライバをインストールしたりできます。[284 ページの「不足しているドライバをインストールする方法」](#)を参照してください。

## ▼ 不足しているドライバをインストールする方法

ドライバが関連付けられていないデバイスをユーティリティが検出した場合、デバイスリストでそのデバイスが選択されます。そのデバイスの詳細情報を表示したり、不足しているドライバをインストールしたりできます。

1. デバイスドライバユーティリティのリストで、デバイス名を右クリックし、次にポップアップメニューから「詳細を表示」を選択します。  
「Device and Driver Details」ウィンドウが表示されます。デバイス名、ベンダー名、ノード名、ドライバ名など、そのデバイスに関する詳細情報が表示されます。
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。正しいパブリッシャーが指定されます。

- 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、未構成のパブリッシャーが提供する IPS パッケージが表示された場合は、次の手順に従います。

- i 「リポジトリ」メニューから「リポジトリを追加」を選択します。

「リポジトリマネージャー」ウィンドウが表示されます。

- ii 新しいリポジトリの名前と URI を追加し、「追加」をクリックします。

- 「パッケージ」フィールドに情報が入力されない場合、「情報」リンクから IPS パッケージの名前を入力して、「インストール」をクリックします。

#### ■ SVR4 または DU ドライバの場合:

- パッケージの URL がわかっている場合、「ファイル/URL」フィールドに URL を入力して「インストール」をクリックします。

- パッケージのコピーがシステム上にある場合、「参照」ボタンをクリックしてパッケージを選択し、「インストール」をクリックします。
- ドライバステータスが「UNK」と表示される場合:
  - a. このドライバで管理するデバイスの名前を選択します。
  - b. 関連するパッケージ情報を「パッケージ」フィールドまたは「ファイル/URL」フィールドに入力し、「インストール」をクリックします。
  - c. (オプション) 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。286 ページの「[HCL にシステムを一覧表示させる方法](#)」を参照してください。

## ▼ HCL にシステムを一覧表示させる方法

特定のデバイスに対して機能することがわかっているドライバに関する情報をほかのユーザーと共有できます。

1. デバイスドライバユーティリティーを起動します。  
283 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 に一覧表示するには、「送信」ボタンをクリックします。  
「ハードウェア互換リストに情報を送信」ウィンドウが開きます。このウィンドウには、システムについて収集されたすべての情報が表示されます。
  - a. 「システムタイプ」を選択します。
  - b. 情報が自動入力されなかったフィールドに、適切な情報を入力します。
    - 製造元名 - システムのメーカーの名称 (たとえば、Toshiba、Hewlett-Packard、Dell など)。
    - 完全なモデル番号。  
「BIOS/ファームウェアのメーカー」は、通常、システムのブート時に表示される BIOS 設定画面の情報です。

- CPU タイプ - CPU のメーカーの名称。
- c. 氏名と電子メールアドレスを入力します。
  - d. 追加のコメントがあれば「一般的なメモ」フィールドに入力し、「保存」をクリックします。保存されたファイルを `device-detect-feedback_ww@oracle.com` に送信します。



# 索引

---

## 数字・記号

- 16 進 MAC アドレス  
条件キーワード, 152

## あ

- 暗号化されたパスワード
  - etc/shadow ファイルからのコピー, 195
- イメージアーカイブ
  - SPARC のインストールのトラブルシューティング, 263
- インスタンス
  - 構成解除, 72
  - 再構成, 73
- インスタンスのインストール
  - AI サーバー上, 89
- インストール
  - インストール前のデフォルトの root のパスワード, 47
  - オプションの概要, 21
  - 自動インストーラ (AI) の使用 参照 AI クライアントインストール
  - テキストインストーラ, 49
  - テキストインストーラおよび USB イメージを使用した, 51
  - ネットワーク経由でのテキストインストーラの使用, 59
  - の追加のオプション, 24
  - 複数のオペレーティングシステム, 29
- インストール構成ファイルディレクトリ変数
  - 派生マニフェスト内, 161
- インストールサーバー 参照 AI サーバー
- インストールサービス
  - AI クライアントの関連付け, 114
  - AI クライアントの削除, 115

- default-arch サービス別名の変更, 110
- DHCP 構成, 105, 108
- grub.cfg ファイル, 111
- GRUB メニュー, 111
- system.conf ファイル, 112
- wanboot.cgi ファイル, 112
- wanboot.conf ファイル, 111
- Web サーバーのポート番号, 104
- Web サーバーファイル, 106
- インストールサービスについての情報の表示, 134
- インストールサービスの更新, 142
- インストールサービスの作成, 107
- インストール手順 参照 AI マニフェスト概要, 84
- クライアントインストール手順 参照 AI マニフェスト
- クライアント構成手順 参照 システム構成プロファイル
- セキュア Web サーバーのポート番号, 105
- セキュリティーの構成, 118
- タスクのリスト, 107
- ネットイメージ
  - IPS パッケージ, 109
  - ISO ファイル, 108
  - デフォルトソース, 111
  - デフォルトの宛先, 111
  - デフォルトベースディレクトリ, 105
- ブートファイル
  - SPARC の wanboot.cgi ファイル, 112
  - x86 grub2netx64.efi ファイル, 112
  - x86 pxegrub2 ファイル, 112
- プロパティーの変更
  - default-manifest プロパティー, 141
- 変更

- セキュリティポリシー, 124
- インストールサービスネットイメージ
  - デフォルトベースディレクトリ, 105
- インストールサービスのネットイメージ
  - IPS パッケージ, 109
  - デフォルトソース, 111
  - デフォルトの宛先, 111
- インストールサービス変数
  - 派生マニフェスト内, 161
- インストールディスクの選択
  - GUI インストール, 42
  - テキストインストーラでの, 53
- インストールの準備
  - テキストインストーラを使用した, 51
- インストールの要件, 29
- インストールプログラム
  - x86 のインストールのトラブルシューティング, 267
- インストールログ
  - AI の障害のトラブルシューティング, 271

## か

- カーネルアーキテクチャー変数
  - 派生マニフェスト内, 161
- 開始
  - コマンド行で自動インストールを, 273
- ガイドライン
  - システムのパーティション分割の, 31
- 拡張パーティション
  - を使用した Oracle Solaris のインストール, 29
- 環境変数
  - AIM\_LOGFILE, 163
- 管理者権限
  - AI サーバー, 101
- 管理レイヤー
  - から情報を削除, 72
- キーボードの選択
  - GUI インストール, 42
  - テキストインストーラでの, 53
- キーボードレイアウト
  - システム構成プロファイル, 201
- 機能グループ
  - 概要, 71
- クライアントアーキテクチャー変数

- 派生マニフェスト内, 160
- クライアント属性環境変数
  - 派生マニフェスト内, 160, 160
- 計画
  - AI サーバーの, 89
- 権限
  - sudo コマンド, 101
  - 権利プロファイル, 101
  - 役割, 101
- 言語の選択
  - GUI インストール, 42
  - テキストインストーラでの, 53
- 更新
  - AI マニフェスト, 178
- 構成 参照 システム構成
  - AI
    - 概要, 86
    - AI マニフェストウィザード, 175
    - Oracle Configuration Manager, 35
    - 複数の IP インタフェース
      - 初回ブートスクリプト内, 235
- 構成解除
  - インスタンス, 72
  - 後に停止, 73
- 構成データ
  - すべてを削除, 72
  - 選択されたデータの削除, 72
- 構成プロファイル 参照 システム構成プロファイル
- コマンド行
  - 自動インストールを開始する, 273
- コンソールモード
  - Live Media インストール, 47

## さ

- サービス管理機能 (SMF) プロファイル
  - AI クライアント構成, 191
- 再構成
  - SCI ツールを使用した, 73
  - インスタンス, 73
  - システム構成ファイルを使用した, 73
- 削除
  - 1 つの CA 証明書, 130
  - AI サーバー資格, 130

- すべての構成データ, 72
- セキュリティー資格, 129
  - 1つの AI クライアント, 129
  - 選択された構成データ, 72
- 作成
  - AI マニフェスト, 178
  - 初回ブートスクリプト, 233
  - 派生マニフェスト, 158
- 資格
  - AI サーバーについての削除, 130
- 時間
  - GUI インストール中の設定, 42
  - テキストインストール中の設定, 53
- システム構成, 191
  - Oracle Auto Service Request, 206
  - Oracle Configuration Manager, 206
  - sysconfig create-profile コマンド, 192
  - インストールサービスへのプロファイルの追加, 193
  - インストール時, 206
  - カスタム IPS パッケージ, 241
  - キーボードレイアウト, 201
  - 構成プロファイルの検証, 192
  - システム構成プロファイルの作成, 191
  - システム識別情報, 199
  - システムのロケール, 200
  - 初回ブートスクリプト, 231, 236
    - 作成, 233
    - 複数の IP インタフェースの構成, 235
  - ゾーンシステム構成プロファイル, 228
  - ゾーンのデフォルトの AI マニフェスト, 227
  - タイムゾーン, 200
  - 端末タイプ, 201
  - ネームサービス, 204
  - ネットワークインフェース, 202
  - ノード名のマッピング, 199
  - プロファイルの例, 208
    - Infiniband リンクプロファイル, 213
    - LDAP プロファイル, 218
    - LDAP を DNS とともに使用するプロファイル, 220
    - NIS サービスのプロファイル, 215
    - NIS と DNS の両方のプロファイル, 216
    - 検索リストを使用した DNS のプロファイル, 217
    - 静的なネットワークプロファイル, 210
    - セキュリティー保護された LDAP プロファイル, 219
    - ネームサービスのプロファイル, 215
- ホスト名, 199
- ユーザー
  - /etc/auto\_home ファイル, 196
  - /etc/passwd ファイル, 196
  - autohome プロパティ, 196
  - root ユーザーアカウント, 195
  - 暗号化されたパスワード, 195
  - オートマウントへの依存, 196
  - 初期ユーザーアカウント, 195
  - 複数のユーザーアカウント, 198
- システム構成ファイルディレクトリ変数
  - 派生マニフェスト内, 160
- システム構成プロファイル, 117
  - 参照 インストールサービスへの追加
  - /usr/share/auto\_install/sc\_profiles プロファイル, 208
  - インストールサービスからの削除, 146
  - 概要, 85
  - 選択アルゴリズム, 151
  - プロファイルの検証, 145
  - プロファイルの更新, 145
  - プロファイルのコピー, 146
  - プロファイルの選択基準, 117
  - 例, 208
  - を使用した再構成, 73
- システム構成プロファイルテンプレート, 206
  - 変数, 208
- システムのパーティション分割
  - GPT, 39
  - GRUB 2, 39
  - GUI インストーラまたは Live Media ISO イメージ, 39
  - VTOC スライス, 34
  - インストール中の選択および変更, 31
  - ガイドライン, 31
- システムのハングアップ
  - x86 のインストールのトラブルシューティング, 267
- システム要件
  - AI クライアント, 248
  - インストール用, 29

- 自動インストーラ, 64
- 自動インストーラ (AI) 参照 AI
- 自動インストーラ
  - インストーラの概要, 65
  - 概要, 63
  - カスタムマニフェスト, 使用, 66
  - システム要件, 64
- 自動インストーラが失敗する
  - トラブルシューティング, 268
- 手動登録
  - Oracle Configuration Manager, 280
- 条件キーワード
  - AI クライアントの選択用, 152
- 使用事例
  - AI の, 89
- 初回ブートスクリプト
  - 概要, 85
  - 作成, 233
  - テンプレート, 234
  - 複数の IP インタフェースの構成, 235
- スクリプト
  - 初回ブート
    - 作成, 233
- セキュアな AI クライアントでのブートエラー
  - トラブルシューティング, 270
- セキュアな IPS リポジトリ
  - AI マニフェスト内, 188
- セキュリティ資格
  - AI サーバーについての削除, 130
  - 削除, 129
- 選択
  - インストール中のパーティション, 31
- 選択アルゴリズム
  - マニフェストおよびプロファイル用, 149
- ゾーン
  - AI および, 86
  - AI クライアントのインストール, 223
  - AI マニフェスト, 224, 226
  - configuration 要素, 223
  - デフォルト, 227
  - config ファイル, 224
  - インストールサービスへのプロファイルの追加, 226
  - インストールサービスへのマニフェストの追加, 225

- システム構成プロファイル, 228
- ゾーン名 条件キーワード用, 152

## た

- タイムゾーンの選択
  - GUI インストール, 42
  - テキストインストーラでの, 53
- 対話型インストーラ
  - システム (x86) のパーティション分割, 33
- 端末タイプ
  - システム構成プロファイル, 201
- 中央コレクタ
  - Oracle Configuration Manager, 278
- 追加
  - GUI インストール後にほかのパッケージを追加, 48
  - テキストインストール後の追加のパッケージ, 61
  - ツールを使用したデバイスドライバの特定, 283
- 定義
  - Oracle Solaris インスタンス, 71
- 停止
  - 構成解除後, 73
- ディスクサイズ変数
  - 派生マニフェスト内, 160
- ディスク数変数
  - 派生マニフェスト内, 161
- ディスクスライスの再利用
  - AI マニフェスト内, 186
- ディスク名変数
  - 派生マニフェスト内, 160
- ディスク容量の要件
  - インストール用, 29
- ディストリビューションコンストラクタ
  - 説明, 24
- データ収集
  - Oracle Configuration Manager, 281
- テキストインストーラ
  - GUI インストーラに優る利点, 22
  - iSCSI の使用, 50
  - USB イメージの取得, 51
  - インストール後のソフトウェアの追加, 61
  - インストールの準備, 51
  - デフォルトのパッケージセット, 23

- ネットワーク経由でのインストールの開始実行, 59
- を使用したインストール, 49
- テキストインストール
  - VTOC スライスの変更, 34
  - システム要件, 29
- 適用
  - 派生マニフェスト, 158
- テスト
  - 派生マニフェストスクリプト, 172
- デバイスドライバ
  - 情報の特定, 283
  - デバイスドライバユーティリティの使用, 283
  - に関する情報を見つける, 35
- デフォルトのパスワード
  - Live Media または GUI インストール, 42
- デフォルトパスワード
  - テキストインストーラでの, 53
- デフォルトルート
  - 構成
    - 初回ブートスクリプト内, 235
- テンプレート
  - 初回ブートスクリプト, 234
- 統合アーカイブ
  - HTTP を使用したアクセス
    - AI マニフェスト内, 187
  - SSL を使用したアクセス
    - AI マニフェスト内, 187
- ドライバ
  - 特定, 283
  - 見つける, 35
- トラブルシューティング
  - AI インストール, 259

## な

- 認証の失敗
  - トラブルシューティング, 271
- ネイティブ ISA 変数
  - 派生マニフェスト内, 161
- ネームサービスの選択
  - テキストインストール中, 53
- ネットワークアドレス
  - 条件キーワード, 152
- ネットワークインタフェース

- AI サーバー上, 89
- 構成, 202
- ネットワーク構成
  - テキストインストーラおよび, 51
- ネットワークのグルーピング
  - SMF プロパティ, 202
- ネットワーク番号
  - 条件キーワード用, 152
- ネットワーク番号変数
  - 派生マニフェスト内, 161
- ノード名のマッピング
  - システム構成プロファイル, 199

## は

- パーティション分割
  - テキストインストーラを使用した, 49
- 派生マニフェスト
  - AIM\_LOGFILE 環境変数, 163
  - AIM\_MANIFEST 環境変数, 163
  - aimanifest コマンド, 161
  - aiuser 役割, 160
  - インストールサービスへの追加, 174
  - クライアント属性環境変数, 160
  - 作成および適用, 158
  - スクリプトの検証, 174
  - スクリプト例, 163
  - テストスクリプト, 172
  - 変更する初期のマニフェスト, 161
- パッケージが見つからない
  - トラブルシューティング, 270
- パッケージ変数
  - 派生マニフェスト内, 161
- 非大域ゾーン 参照 ゾーン
- 日付
  - GUI インストール中の設定, 42
  - テキストインストール中の設定, 53
- ブート
  - AI クライアント
    - 概要, 88
    - 自動インストールを開始しない, 272
- ブートアーカイブ
  - x86 のインストールのトラブルシューティング, 267
- ブートエラー

- AI クライアントのインストールのトラブルシューティング, 260
- ブートディスクが見つからない
  - AI クライアントのインストールのトラブルシューティング, 260
- ブートの読み込みに失敗
  - SPARC のインストールのトラブルシューティング, 262
- ブートファイル
  - SPARC の `wanboot-cgi` ファイル, 112, 249
  - x86 `grub2netx64.efi` ファイル, 112
  - x86 `pxegrub2` ファイル, 112
  - x86 クライアント `.bios` ファイル, 250
  - x86 クライアント `.uefi` ファイル, 250
- ブートプール
  - AI マニフェスト内, 183
- 複数のオペレーティングシステム
  - インストールの要件, 29
- 複数の IP インタフェースの構成
  - 構成
    - 初回ブートスクリプト内, 235
- 複数の SVR4 パッケージのインストール
  - AI マニフェスト内, 186
- 物理メモリー変数
  - 派生マニフェスト内, 161
- プラットフォーム変数
  - 派生マニフェスト内, 161
- プロセッサタイプ変数
  - 派生マニフェスト内, 160
- プロファイル 参照 システム構成プロファイル
- ベースディレクトリ
  - AI ネットイメージ用, 105
- 変更
  - AI マニフェスト
    - 手動, 156
    - インストール中のパーティション, 31
- ホスト名変数
  - 派生マニフェスト内, 161

## ま

- マニフェスト 参照 AI マニフェスト
- マニュアルページ
  - と `solaris-minimal-server` パッケージ, 189
- マルチキャスト DNS (mDNS), 103

- マルチホームの
  - AI サーバー, 89
- マルチホームの AI サーバー, 104
- 無効化
  - AI マニフェストウィザード, 175
  - Oracle Configuration Manager, 280
- 無効な HMAC 値
  - SPARC のインストールのトラブルシューティング, 264
- メモリーの要件
  - インストール用, 29
- メモリーの M バイト
  - 条件キーワード用, 152
- メモリー変数
  - 派生マニフェスト内, 161

## や

- 有効化
  - Oracle Configuration Manager, 280
- ユーザーの構成
  - GUI インストール, 42
  - テキストインストーラでの, 53

## ら

- ルートプール
  - AI マニフェスト内, 183

## 例

- AI マニフェスト, 182
- 初回ブートスクリプトテンプレート, 234
- ログファイル
  - 自動インストール, 70
- ロケール
  - と `solaris-minimal-server` パッケージ, 188

## A

- AI
  - インストールのカスタマイズ, 149
  - 概要, 81
  - 使用事例, 89
- AI インストールサービス
  - AI クライアントの関連付け, 249

- インストールサービスからの AI クライアントの削除, 250
- AI が開始されない
  - SPARC のインストールのトラブルシューティング, 264
  - x86 のインストールのトラブルシューティング, 268
- AI クライアント
  - /system/volatile/install\_log ファイル, 259
  - CA 証明書の削除, 130
  - SPARC のインストールのトラブルシューティング, 261
  - x86 のインストールのトラブルシューティング, 265
  - インストールのトラブルシューティング, 259
  - 概要, 82
  - コマンド行でインストールを開始する, 273
  - 自動インストールを開始しないでブートする, 272
  - セキュリティー資格の削除, 129
  - 選択条件, 152
  - ブートエラー
    - トラブルシューティング, 260
    - ホスト名条件キーワード, 152
- AI クライアントインストール
  - /system/volatile/install\_log ファイル, 248
  - ssh コマンドを使用したモニタリング, 251
  - インストールメッセージ
    - インストールの開始, 257
    - インストールの成功, 257
  - 概要, 247
  - セキュア, 125
  - ネットワークブート
    - network-boot-arguments OBP 変数, 254
    - SPARC クライアント, 252
    - SPARC の WAN ブートサポート, 248
    - x86 クライアント, 255
- AI クライアント構成 参照 システム構成
- AI クライアントのインストール
  - システム要件, 248
- AI サーバー
  - 管理者権限, 101
  - 計画, 89
  - 構成, 101
    - Web サーバーのポート番号, 104
    - Web サーバーファイルディレクトリ, 106
    - セキュア Web サーバーのポート番号, 105
    - 認証, 118
    - マルチキャスト DNS, 103
    - マルチホーム, 104
  - 資格の削除, 130
  - セキュリティー資格および鍵, 119
  - 設定, 99
  - 設定タスクのサマリー, 99
  - 認証, 119
    - のコンポーネント, 82
  - 要件, 100
- AI システム構成プロファイル
  - インストールサービスへの追加, 193
  - プロファイルの検証, 192
  - プロファイルの作成, 191
  - プロファイルを選択するための条件, 152
- AI のカスタマイズ, 149
- AI のセキュリティー保護
  - 概要, 85
- AI マニフェスト
  - AI マニフェストウィザードを使用した作成, 175
  - installadm コマンドを使用した作成, 178
  - インストール開始前の変更, 155
  - インストールサービスからの削除, 144
  - インストールサービスへの追加, 116
  - インストール時の作成 参照 派生マニフェスト
  - 概要, 84
  - カスタム IPS パッケージのインストール, 243
  - 既存のマニフェストの変更, 155
  - 選択アルゴリズム, 150
  - ゾーンの configuration 要素, 223
  - デフォルトの AI マニフェスト, 189
  - マニフェストの検証, 144
  - マニフェストの更新, 143
  - マニフェストのコピー, 146
  - マニフェストを選択するための条件, 152
  - 例, 182
- AI マニフェストウィザード
  - AI サーバーへのファイルの保存, 176
  - AI マニフェストの作成, 175
  - 無効化, 175

aimanifest コマンド  
  add サブコマンド, 162  
  load サブコマンド, 162  
  set サブコマンド, 162  
  validate サブコマンド, 174  
aiuser 役割, 160  
all\_services プロパティグループ  
  exclude\_networks プロパティ, 104  
  manage\_dhcp プロパティ, 108  
  networks プロパティ, 104  
  webserver\_files\_dir プロパティ, 106  
  webserver\_secure\_files\_dir プロパティ, 106  
Apache のログファイル  
  AI の障害のトラブルシューティング, 271  
arch 条件キーワード, 152  
ASR 参照 Oracle Auto Service Request

## B

boot\_archive ファイル  
  SPARC のインストールのトラブルシューティング, 262

## C

-c オプション  
  sysconfig unconfigure コマンド, 73  
CA 証明書  
  AI クライアントのための削除, 130  
configCCR コマンド  
  -c オプション, 278  
  手動登録と, 280  
configure サブコマンド  
  sysconfig コマンド, 73  
cpu 条件キーワード, 152

## D

-d オプション  
  installadm set-server コマンド, 105  
date\_time 機能グループ  
  説明, 71

## DHCP

  /etc/inet/dhcpd4.conf 構成ファイル, 112, 113, 114  
  dhcpinfo コマンド, 259  
  svc:/network/dhcp/server SMF サービス, 112  
  構成ファイル, 112  
  自動構成, 105, 108  
DHCP サーバー  
  AI および, 83  
DHCP サーバーが応答しない  
  AI クライアントのインストールのトラブルシューティング, 261  
  x86 のインストールのトラブルシューティング, 266  
dhcpinfo コマンド, 259  
DNS  
  AI クライアントインストールのトラブルシューティング, 259  
DNS の選択  
  テキストインストール中, 53  
doctools パッケージ  
  AI マニフェスト内, 189

## E

  /etc/auto\_home ファイル, 196  
  /etc/passwd ファイル, 196  
  /etc/resolv.conf 構成ファイル, 259  
  /etc/svc/profile/site  
  から情報を削除, 72  
  /etc/svc/profile/sysconfig/site-profile.tar ファイル, 72  
emCCR コマンド  
  データ収集の変更, 281  
entire パッケージ  
  AI マニフェストと, 182  
ERROR 403: Forbidden  
  SPARC のインストールのトラブルシューティング, 263  
  x86 のインストールのトラブルシューティング, 267  
ERROR 404: Not Found  
  SPARC のインストールのトラブルシューティング, 263

x86 のインストールのトラブルシューティング, 267

## F

Forbidden エラー

SPARC のインストールのトラブルシューティング, 263

x86 のインストールのトラブルシューティング, 267

## G

-g オプション

sysconfig unconfigure コマンド, 72, 72

GPT パーティション

インストール中の選択および変更, 31

GPT フォーマット

テキストインストーラおよび, 49

grub.cfg ファイル, 111

x86 のインストールのトラブルシューティング, 267

GRUB 2

システムのパーティション分割, 31

テキストインストーラおよび, 49

grub2netx64.efi ファイル, 112

GRUB メニュー, 111

GUI インストーラ

GUI インストールの準備, 40

インストール後にほかのパッケージを追加, 48

インストールに使用されるデフォルトのネットワーク設定とセキュリティー設定, 40

インストールの際に、サポートされていない

グラフィックスカードを使用して、またはグラフィックスカードなしでインストールする, 46

サポートされるプラットフォーム, 39

パーティション分割のガイドライン, 39

GUI インストール

システム要件, 29

例, 42

## H

HMAC ハッシュ鍵, 127, 252

SPARC のインストールのトラブルシューティング, 264

hostname 条件キーワード, 152

HTTP 認証

統合アーカイブへのアクセス用

AI マニフェスト内, 187

## I

i86pc 値

条件キーワード, 152

i86 値

cpu 条件キーワード用, 152

identity 機能グループ

説明, 71

--include-site-profile オプション

sysconfig unconfigure コマンド, 72

installadm コマンド

AI マニフェストの作成, 178

create-client サブコマンド, 114, 249

create-manifest サブコマンド, 116

create-profile サブコマンド, 117

create-service サブコマンド, 107

delete-client サブコマンド, 115, 250

delete-manifest サブコマンド, 144

delete-profile サブコマンド, 146

export サブコマンド, 146

list サブコマンド, 134

set-client サブコマンド

--hash オプション, 130

-x オプション, 129

set-server サブコマンド

-x オプション, 130

set-service サブコマンド, 123, 140

update-manifest サブコマンド, 143

update-profile サブコマンド, 145

update-service サブコマンド, 142

validate サブコマンド, 144, 145, 192

サブコマンドの例, 104, 105

セキュリティー資格の削除, 129

セキュリティーの強化, 119

IP アドレス変数

派生マニフェスト内, 160

IPS サーバーを使用できない

トラブルシューティング, 268

**IPS** リポジトリ

AI および, 83

ipv4 条件キーワード, 152

IPv4 プロパティグループ

SMF プロパティ, 202

IPv6 プロパティグループ

SMF プロパティ, 203

**ISA** 変数

派生マニフェスト内, 160

**iSCSI**

テキストインストーラで使用, 50

**iSCSI** ターゲット

AI マニフェスト内, 183

**iSCSI** ディスク検出

GUI インストール, 42

**iSCSI** ディスクの検出

テキストインストーラでの, 53

**K****keyboard** 機能グループ

説明, 71

**L****LDAP** の選択

テキストインストール中, 53

Linux-スワップパーティション, 31

**Live Media**, 39

参照 GUI インストーラ

GUI インストールの準備, 40

インストール後にほかのパッケージを追加, 48

インストールに使用されるデフォルトのネット

ワーク設定とセキュリティー設定, 40

インストールの際に、サポートされていない

グラフィックスカードを使用して、またはグラ

フィックスカードなしでインストールする, 46

コンソールモードでのインストール, 47

サポートされるプラットフォーム, 39

システム要件, 29

パーティション分割のガイドライン, 39

例, 42

**location** 機能グループ

説明, 71

**M****-M** オプション

installadm set-server コマンド, 105

mac 条件キーワード, 152

man コマンド

と solaris-minimal-server パッケージ, 189

mem 条件キーワード, 152

menu.lst ファイル

x86 のインストールのトラブルシューティング,  
267**Microsoft Windows**

を使用した Oracle Solaris のインストール, 29

**MOS** 参照 My Oracle Support**My Oracle Support**

AI インストール, 206

資格

Oracle Configuration Manager, 278

**N****naming\_services** 機能グループ

説明, 71

network-boot-arguments OBP 変数, 254

**network** 機能グループ

説明, 71

network 条件キーワード, 152

**Not Found** エラーSPARC のインストールのトラブルシューティ  
ング, 263x86 のインストールのトラブルシューティング,  
267**O****OBP** セキュリティーキー

暗号化鍵, 127, 252

ハッシュ鍵 (HMAC), 127, 252, 264

**OCM** 参照 Oracle Configuration Manager**openssl** コマンド

AI の障害のトラブルシューティング, 271

**Oracle Auto Service Request**

AI インストール用の構成, 206

GUI インストール, 42

構成, 35  
 テキストインストーラでの, 53

Oracle Configuration Manager  
 AI インストール用の構成, 206  
 GUI インストール, 42  
 Oracle Universal Installer と, 279  
 概要, 277  
 構成, 35  
 手動登録, 280  
 中央コレクタ, 278  
 データ収集, 281  
 テキストインストーラでの, 53  
 無効化, 280  
 有効化, 280

Oracle Solaris インスタンス  
 構成解除, 72  
 再構成, 73  
 定義されている, 71

Oracle Solaris のインストール  
 システム要件, 29

Oracle Universal Installer  
 Oracle Configuration Manager と, 279

ORCL, SPARC-T4-2 値  
 platform 条件キーワード用, 152

OS パーティション  
 インストール中の選択および変更, 31

OUI 参照 Oracle Universal Installer

**P**

-p オプション  
 installadm set-server コマンド, 104

-P オプション  
 installadm set-server コマンド, 105

PEM フォーマットの X.509 証明書および鍵, 119

pkg コマンド  
 Live Media または GUI インストール後に適用,  
 48  
 既存のインストールをアップデートするために  
 使用, 24  
 テキストインストール後のソフトウェアの追  
 加, 61

platform 条件キーワード, 152

pxegrub2 ファイル, 112

**R**

RAID 構成  
 AI マニフェスト内, 184  
 reconfigure サブコマンドと  
 sysconfig コマンド, 73

**S**

/system/volatile/install\_log ファイル  
 インストール完了メッセージ, 248  
 概要, 259  
 自動インストール, 70

/system/volatile  
 AI の障害のトラブルシューティング, 271

-s オプション  
 sysconfig unconfigure コマンド, 73

SCI ツール  
 を使用した再構成, 73

setupCCR コマンド  
 -c オプション, 278

SI\_\* 変数  
 派生マニフェスト内の, 160

SMF サービス  
 svc:/application/auto-installer, 248  
 svc:/network/dhcp/server, 112  
 svc:/network/dns/client, 204  
 svc:/network/dns/multicast, 103  
 svc:/network/install, 202  
 svc:/system/config-user, 195  
 svc:/system/console-login, 201  
 svc:/system/environment:init, 200  
 svc:/system/identity, 199  
 svc:/system/install/server, 101, 118  
 svc:/system/keymap, 201  
 svc:/system/name-service/switch, 220  
 svc:/system/timezone, 200  
 svc:/system/zones-install, 223  
 初回ブート時に一度実行, 236  
 初回ブート時に一度だけ実行する, 231

SMF サービスのログ  
 AI の障害のトラブルシューティング, 271

SMF サービスマニフェスト  
 svcbundle コマンド, 236  
 カスタマイズ  
 start メソッドのタイムアウト, 239

- 依存関係, 241
  - 初回ブート時に 1 度だけ実行するサービスの例, 237
  - マニフェスト作成ツール 参照 `svcbundle` コマンド
  - SMF 情報
    - 削除, 72
  - SMF プロパティ
    - AI クライアント構成, 191
    - `all_services` プロパティグループ, 101
    - `config` プロパティグループ, 199, 199, 205
    - `enable_mapping` プロパティ, 199
    - `environment` プロパティグループ, 200
    - IPv4 プロパティグループ, 202
    - IPv6 インタフェースのプロパティグループ, 203
    - `root_account` プロパティグループ, 196
    - `timezone` プロパティグループ, 200
    - `user_account` プロパティグループ, 197
    - 表示, 194
  - `solaris-minimal-server` パッケージ
    - `man` コマンドの追加
      - AI マニフェスト内, 189
    - マニュアルページと
      - AI マニフェスト内, 189
    - ロケールと
      - AI マニフェスト内, 188
  - `sparc` 値
    - `cpu` 条件キーワード用, 152
  - `ssh` コマンド
    - AI クライアントインストールのモニタリング, 251
  - SSL クライアント認証
    - 統合アーカイブへのアクセス用
      - AI マニフェスト内, 187
  - `sun4u` 値
    - `arch` 条件キーワード用, 152
  - `sun4v` 値
    - `arch` 条件キーワード用, 152
  - SUNw, SPARC-Enterprise 値
    - `platform` 条件キーワード用, 152
  - `support` 機能グループ
    - 説明, 71
  - `svcbundle` コマンド, 236
  - `svccfg` コマンド
    - プロパティ情報の表示, 194
  - SVR4 パッケージのインストール
    - AI マニフェスト内, 185
  - `sysconfig create-profile` コマンド, 192
  - `sysconfig` コマンド
    - `configure` サブコマンド, 73
    - `reconfigure` サブコマンド, 73
    - `unconfigure` サブコマンド, 72
  - `system.conf` ファイル, 112
  - `system/locale` パッケージ
    - ロケールと
      - AI マニフェスト内, 188, 189
  - `system` 機能グループ
    - すべての構成データの削除, 72
    - 説明, 71
- T**
- `/tmp/install_log`
    - AI の障害のトラブルシューティング, 271
  - TFTP エラー
    - x86 のインストールのトラブルシューティング, 266
  - TLS 証明書が有効ではない
    - トラブルシューティング, 270
  - Transport Layer Security (TLS) プロトコル, 119
- U**
- `/usr/bin/ai-wizard`, 175
  - `/usr/sbin/configCCR` コマンド
    - 手動登録と, 280
  - `/usr/sbin/emCCR` コマンド
    - データ収集の変更, 281
  - `unbounded` キーワード, 152
  - `unconfigure` サブコマンド
    - `-c` オプション, 73
    - `-g` オプション, 72, 72
    - `--include-site-profile` オプション, 72
    - `sysconfig` コマンド, 72
    - `-s` オプション, 73
  - USB イメージ
    - テキストインストーラ用に取得, 51
  - USB コピー用の `dd` コマンド, 52

usbcopy コマンド、dd の使用, 52

users 機能グループ

説明, 71

## V

/var/ai/image-server/logs ディレクトリ

AI の障害のトラブルシューティング, 271

概要, 259

/var/log/install/install\_log ファイル

自動インストール, 70

VTOC スライス

rpool と ZFS ルートプール, 34

インストール中の選択および変更, 31

変更のオプション, 34

## W

WAN ブートサポート, 248

Wan ブートの警告

SPARC のインストールのトラブルシューティング, 262

wanboot.conf ファイル, 111

Web サーバー

セキュアなポート番号, 105

ファイルディレクトリ, 106

ポート番号, 104

webservd ユーザーおよびグループ, 106

wget コマンド

AI の障害のトラブルシューティング, 271

## X

X.509 証明書および鍵, 119

x86

システムのパーティション分割, 33

x86 システムのパーティション分割

のオプション, 33, 34

