Oracle® Solaris 10 JumpStart から Oracle Solaris 11 自動インストーラへの移行

Part No: E25765
2012 年 3 月、E25765-03

ORACLE®

Copyright © 2011, 2012, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT END USERS:

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

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

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

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

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

120411@25097

目次

はじめに 5

1 **JumpStart** と自動インストーラの比較 9
JumpStart と AI の類似点と相違点 9
JumpStart から AI への移行情報 10

2 ルールおよびプロファイルファイルの変換 13
ルールのキーワードと条件の指令の比較 13
プロファイルのキーワードと AI マニフェストの指令の比較 16
`js2ai` の使用による JumpStart ルールおよびプロファイルから AI 条件およびマニフェストへの変換 23
`js2ai` による JumpStart デバイス指定の変換方法 24
ソフトウェアパッケージの変換 26
`js2ai` を使用したルールおよびプロファイルの変換例 29
詳細なプロファイル変換情報の表示 37
出力 AI マニフェストの検証 37

3 構成ファイルの変換 39
`sysidcfg` ファイルキーワードとシステム構成プロファイル指令の比較 39
`js2ai` を使用した `sysidcfg` ファイルからシステム構成プロファイルへの変換 41
`js2ai` を使用した `sysidcfg` の変換例 42
詳細な構成変換情報の表示 44
出力システム構成プロファイルの検証 44

4 **Oracle Solaris 11** サーバー上の **JumpStart** を使用した **Oracle Solaris 10** のインストール 47
Oracle Solaris 10 JumpStart サーバーとしての Oracle Solaris 11 システムの設定 48

既知の問題の回避策 50
rm_install_client スクリプトの問題 50
check スクリプトの問題 51

はじめに

Oracle Solaris 10 JumpStart から Oracle Solaris 11 自動インストーラへの移行は、いずれも Oracle Solaris の自動インストール機能である JumpStart から自動インストーラ (AI) への移行に役立つ情報を提供します。

本書では、次の手順およびツールについて説明します。

- JumpStart と AI の違い
- JumpStart ルールおよびプロファイルキーワードを AI 条件およびマニフェストキーワードに変換する方法
- `sysidcfg` ファイルを AI システム構成プロファイルに変換する方法
- `js2ai` 変換ユーティリティーの使用方法
- Oracle Solaris 10 および Oracle Solaris 11 の両方のインストールに対応するように 1 つのサーバーを設定する方法

対象読者

本書は、Oracle Solaris 11 オペレーティングシステム (OS) の自動インストールを実行するために AI の使用に移行する JumpStart ユーザーを対象としています。

内容の紹介

本書には次の章が含まれています。

- 第 1 章「JumpStart と自動インストーラの比較」には、JumpStart および AI を使用して各タスクを行う方法を説明したインストールタスクの表を示します。
- 第 2 章「ルールおよびプロファイルファイルの変換」には、JumpStart ルールおよびプロファイルキーワードと、同等の AI 条件およびマニフェスト指令が記載されている表を示します。この章では、`js2ai` ユーティリティーを使用した変換の例も示します。
- 第 3 章「構成ファイルの変換」には、`sysidcfg` ファイルキーワードと、同等の AI システム構成プロファイル指定が記載されている表を示し、`js2ai` ユーティリティーを使用した変換の例を示します。

- 第 4 章「Oracle Solaris 11 サーバー上の JumpStart を使用した Oracle Solaris 10 のインストール」では、Oracle Solaris 11 サーバーを JumpStart インストールサーバーと AI インストールサーバーの両方として使用する方法について説明します。

関連情報

『Oracle Solaris 10 9/10 インストールガイド (カスタム JumpStart/上級編)』には、JumpStart の使用方法が記載されています。

『Oracle Solaris 11 システムのインストール』には、Oracle Solaris 11 OS をインストールするためのさまざまな方法 (AI の使用を含む) が説明されています。

『Oracle Solaris の管理: 一般的なタスク』の第 6 章「サービスの管理 (概要)」では、Oracle Solaris サービス管理機構 (SMF) 機能について説明します。SMF プロファイルを使用して、システムを構成できます。

`pkg(5)` マニュアルページでは、インストール用のソフトウェアパッケージの格納や取得を可能にする Oracle Solaris Image Packaging System (IPS) 機能について説明します。`pkg(1)` マニュアルページでは、IPS パッケージのインストール方法について説明します。

Oracle Solaris 11 システムを管理する方法の詳細は、Oracle Solaris 11 システム管理ドキュメントを参照してください。

Oracle サポートへのアクセス

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

表記上の規則

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

表 P–1 表記上の規則

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

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

- C シェル

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

- C シェルのスーパーユーザー

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

- Bash シェル、Korn シェル、および Bourne シェル

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

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

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

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

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

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

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

第 1 章

JumpStart と自動インストーラの比較

この章では、次の内容について説明します。

- JumpStart と自動インストーラ (AI) の類似点と相違点
- JumpStart から AI への移行に役立つ参照情報

JumpStart と AI の類似点と相違点

JumpStart と AI の両方とも、手入力なしでネットワーク上の複数のシステムのインストールを行うことができます。クライアントはネットワークブートされ、クライアントがブートすると、インストーラが引き継ぎます。

JumpStart は、Oracle Solaris 10 OS 以前のバージョンの Oracle Solaris OS をインストールします。AI は Oracle Solaris 11 OS をインストールします。

JumpStart と AI には、次の共通した特徴があります。

- システム構成をインストールサーバーに格納することで、手入力なしで複数のクライアントのネットワークインストールを行います。
- 1 つの自動インストールで、異なるクライアントに対して異なる種類のインストールを行います。
- x86 と SPARC の両方のクライアントをインストールします。

表 1–1 JumpStart と AI のステップの比較

ステップ	JumpStart	AI
インストールサーバーを設定します。	`setup_install_server`(1M) コマンドを使用します。	`installadm create-service` コマンドを使用します。
インストールにクライアントを追加します。	`add_install_client`(1M) コマンドを使用します。	`installadm create-client` コマンドを使用します。

表 1–1 JumpStart と AI のステップの比較 (続き)

ステップ	JumpStart	AI
派生したプロファイルを作成します	開始スクリプトを使用します。	派生したマニフェストメカニズムを使用します。
インストール手順を指定します。	プロファイルファイルを使用します。	AI マニフェストファイルを使用します。
クライアントをプロビジョニングします。	DVD を使用します。	インターネットまたはローカルネットワーク上のパッケージリポジトリを使用します。
クライアントのカスタマイズを指定します。	ルールファイルを使用して、クライアントをプロファイルファイルに関連付けます。	`installadm` コマンドの `create-manifest`、`create-profile`、または `set-criteria` サブコマンドを使用して、クライアントを AI マニフェストおよびシステム構成プロファイルに関連付けます。
クライアント構成を指定します。	終了スクリプトと `sysidcfg`(4) ファイルを使用します。	SMF システム構成プロファイルファイルを使用します。初回ブート時に 1 回実行される SMF サービスによって実行されるスクリプトを使用します。

JumpStart から AI への移行情報

JumpStart ルール、プロファイル、および構成ファイルを **AI** 条件、**AI** マニフェスト、および **SMF** システム構成プロファイルに変換します。

`js2ai` ユーティリティーを取得します。

```
# pkg install install/js2ai
```

JumpStart データを AI データに変換するには、次の手順に従ってください。

- `js2ai`(1M)
- 第 2 章「ルールおよびプロファイルファイルの変換」
- 第 3 章「構成ファイルの変換」

1 つのサーバーを **JumpStart** インストールサーバーおよび **AI** インストールサーバーの両方として使用します。

第 4 章「Oracle Solaris 11 サーバー上の JumpStart を使用した Oracle Solaris 10 のインストール」

AI クライアントプロビジョニングマニフェストを動的に派生させます。

『Oracle Solaris 11 システムのインストール』の「クライアントインストール時の AI マニフェストの作成」を参照してください。JumpStart 開始スクリプトは、インストーラに渡されるインストールパラメータを動的に操作する機能を提供します。AI は、クライアントのインストール時にクライアント属性を照会し、そのク

ライアント向けにカスタマイズされたプロビジョニングマニフェストを動的に派生させることができます。環境変数はクライアントのハードウェア属性を指定し、これらのほとんどは、JumpStart 開始スクリプトで使用される環境変数と同じです。

AI インストール用のソフトウェアパッケージリポジトリにアクセスします。

インターネット上にある Oracle Solaris 11 パッケージリポジトリを使用してください (pkg.oracle.com)

パッケージリポジトリのローカルコピーを行います (『Oracle Solaris 11 パッケージリポジトリのコピーおよび作成』)。

システム構成の指示を行います。

SMF プロファイルの作成については、『Oracle Solaris 11 システムのインストール』の第 11 章「クライアントシステムの構成」を参照してください。

AI クライアントインストールの一部として非大域ゾーンをインストールするために AI マニフェストおよび SMF プロファイルを作成することについては、『Oracle Solaris 11 システムのインストール』の第 12 章「ゾーンのインストールと構成」を参照してください。

初回ブート時に **1** 回実行してユーザー定義スクリプトを実行する **SMF** サービスを作成します。

『Oracle Solaris 11 システムのインストール』の第 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。

あるいは、必要なインストールを **1** つのインストールファイル内に正確に作成します。

カスタムインストールイメージの作成については、『カスタム Oracle Solaris 11 インストールイメージの作成』を参照してください。

第 2 章

ルールおよびプロファイルファイルの変換

この章では、JumpStart ルールおよびプロファイルファイルを AI 条件ファイルおよび AI マニフェストに変換する方法を示します。ほとんどの変換は、`js2ai` ユーティリティーを使用して実行できます。`js2ai` ユーティリティーについての詳細は、`js2ai(1M)` のマニュアルページを参照してください。

AI マニフェストはディスクレイアウトやインストールするソフトウェアパッケージなどのシステムプロビジョニングを指定する XML ファイルです。AI マニフェストについては、『Oracle Solaris 11 システムのインストール』の第 10 章「クライアントシステムのプロビジョニング」および `ai_manifest(4)` のマニュアルページを参照してください。

AI ではクライアント条件を使用して、クライアントシステムがインストールを完了するために使用する AI マニフェストファイルを指定します。AI マニフェストが AI インストールサービスに追加されるとき、コマンド行またはファイルで条件を指定できます。この章では、`js2ai` ツールを使用して JumpStart `rules` ファイルを AI 条件ファイルに変更する方法を示します。AI クライアント条件の指定の詳細は、『Oracle Solaris 11 システムのインストール』の第 9 章「インストールのカスタマイズ」を参照してください。

ルールのキーワードと条件の指令の比較

次の表では、JumpStart ルールのキーワードと AI 条件の指令を比較します。AI はこれらの条件を使用して、正しい AI マニフェストまたはシステム構成プロファイルを特定のクライアントに適用します。

表 2–1 JumpStart ルールファイルのキーワードと AI 条件の指令の比較

JumpStart ルールファイルのキーワード	AI 条件ファイルの指令
any	サポートされていません。どの選択条件にも一致しないクライアントシステムの場合、AI インストールサービスはデフォルトの AI マニフェストを提供します。
arch	cpu コマンドオプション: -c cpu=sparc 条件ファイル: <ai_criteria name="cpu"> <value>sparc</value> </ai_criteria>
disksize	サポートされていません。AI マニフェストで指定されたターゲットのディスクが最低限必要なサイズより大きい場合、AI はターゲット上にインストールします。
domainname	サポートされていません。
hostaddress	ipv4 コマンドオプション: -c ipv4=10.6.68.127 条件ファイル: <ai_criteria name="ipv4"> <value>10.6.68.127</value> </ai_criteria>
hostname	サポートされていません。AI でホストを一意に識別するには、上に示した IP アドレスを使用するか、MAC アドレスを使用します。 コマンドオプション: -c mac=0:3:ba:33:9d:b6 条件ファイル: <ai_criteria name="mac"> <value>0:3:ba:33:9d:b6</value> </ai_criteria>
installed	サポートされていません。AI マニフェストで指定されたターゲットのディスクが最低限必要なサイズより大きい場合、AI はターゲット上にインストールします。

表 2–1 JumpStart ルールファイルのキーワードと AI 条件の指令の比較 (続き)

JumpStart ルールファイルのキーワード	AI 条件ファイルの指令
karch	arch コマンドオプション: -c arch=i86pc 条件ファイル: <ai_criteria name="arch"> <value>i86pc</value> </ai_criteria>
memsize	mem コマンドオプション: -c mem=2048 条件ファイル: <ai_criteria name="mem"> <value>2048</value> </ai_criteria>
model	platform コマンドオプション: -c platform=SUNW,Sun-Fire-T200 条件ファイル: <ai_criteria name="platform"> <value>SUNW,Sun-Fire-T200</value> </ai_criteria>
network	network、範囲を指定した network、または範囲を指定した ipv4 を使用します。 コマンドオプション、単一の network 値: -c network="10.0.0.0" 条件ファイル、単一の network 値: <ai_criteria name="network"> <value>10.0.0.0</value> </ai_criteria> コマンドオプション、ipv4 範囲: -c ipv4=10.0.0.1-10.0.0.64 条件ファイル、ipv4 範囲: <ai_criteria name="ipv4"> <range>10.0.0.1 10.0.0.64</range> </ai_criteria>
osname	サポートされていません。
probe	サポートされていません。

表 2–1 JumpStart ルールファイルのキーワードと AI 条件の指令の比較 (続き)

JumpStart ルールファイルのキーワード	AI 条件ファイルの指令
`totaldisk`	サポートされていません。AI マニフェストで指定されたターゲットのディスクが最低限必要なサイズより大きい場合、AI はターゲット上にインストールします。

プロファイルのキーワードと AI マニフェストの指令の比較

次の表では、JumpStart プロファイルのキーワードと AI マニフェストの指令を比較します。AI は、XML マニフェストファイルを使用してクライアントインストールを定義します。AI マニフェストについては、『Oracle Solaris 11 システムのインストール』の第 10 章「クライアントシステムのプロビジョニング」および `ai_manifest`(4) のマニュアルページを参照してください。

`any` で指定されるデバイスなど、クライアントのインストールプロセスが開始するまで不明な値を指定する場合、派生したマニフェストスクリプトの使用を検討してください。派生したマニフェストスクリプトを使用して、たとえばディスクサイズに基づいてスワップサイズを指定したり、使用可能なディスクに基づいてミラー化を指定したりすることができます。派生したマニフェストスクリプトについては、『Oracle Solaris 11 システムのインストール』の「クライアントインストール時の AI マニフェストの作成」を参照してください。

表 2–2 JumpStart プロファイルファイルのキーワードと AI マニフェストの指令の比較

JumpStart プロファイルファイルのキーワード	AI マニフェストの指令
`archive_location`	サポートされていません。AI はフラッシュアーカイブからインストールしません。
`backup_media`	サポートされていません。`backup_media` キーワードは `install_type` の `upgrade` オプションと一緒にのみ使用します。 AI はアップグレードインストールタイプをサポートせず、初期インストールのみサポートします。

`boot_device` *device*
`boot_device c1t0d0`
	`<target>`
	`  <disk whole_disk="true">`
	`  <disk_name name="c1t0d0" name_type="ctd"/>`
	`  </disk>`
	`  <logical nodump="true" noswap="false"/>`
	`</target>`

表 2–2 JumpStart プロファイルファイルのキーワードと AI マニフェストの指令の比較 (続き)

JumpStart プロファイルファイルのキーワード	AI マニフェストの指令
`boot_device` *device eeprom* `boot_device c0t0d0s0 update`	上記と同じです。このキーワードの 2 番目のトークン値 (SPARC システムの場合は `update`、x86 システムの場合は `preserve`) は、AI ではサポートされていません。AI では、SPARC システムの EEPROM は、インストールしたシステムが指定したターゲットデバイスから自動的にブートするように、常にそのデバイスに更新されます。x86 システムでは、ファームウェアは更新されません。
`bootenv`	サポートされていません。AI はブート環境を作成し、Oracle Solaris 11 OS をそのブート環境にインストールします。
`client_arch`	サポートされていません。インストールされるクライアントアーキテクチャーは AI マニフェストでなく AI インストールサービスによって定義されます。
`client_root`	サポートされていません。ターゲットディスクの種類、ベンダー、およびサイズを指定できます。ターゲットディスクのルート領域の量は指定できません。`ai_manifest`(4) のマニュアルページを参照してください。
`client_swap`	サポートされていません。デフォルトでは、AI はルートプールにスワップボリュームを作成します。スワップスライスを指定できます。`ai_manifest`(4) のマニュアルページを参照してください。
`cluster` *cluster-name*	サポートされていません。`cluster` および `package` 指定の扱い方については、26 ページの「ソフトウェアパッケージの変換」を参照してください。 Oracle Solaris 11 OS は、`pkg`(5) グループパッケージを使用します。グループパッケージは、マニフェストでほかのパッケージを指定するようにして指定します。デフォルトの AI マニフェストには、標準の Oracle Solaris 11 インストールに必要なパッケージが含まれています。このパッケージリストはカスタマイズ可能です。
`cluster` *cluster-name* `delete`	サポートされていません。`delete` スイッチは、`install_type` の `upgrade` オプションと一緒にのみ使用します。AI はアップグレードインストールタイプをサポートせず、初期インストールのみサポートします。
`dontuse`	サポートされていません。

表 2–2 JumpStart プロファイルファイルのキーワードと AI マニフェストの指令の比較 (続き)

JumpStart プロファイルファイルのキーワード	AI マニフェストの指令

`fdisk` *disk_name type size*
`fdisk c0t3d0 solaris all`
	`<target>`
	`  <disk whole_disk="true">`
	`  <disk_name name="c0t3d0" name_type="ctd"/>`
	`  <partition action="create" name="1" part_type="191"/>`
	`  </disk>`
	`  <logical nodump="true" noswap="false"/>`
	`</target>`
	
	AI でサポートされているディスクおよびパーティション属性の完全な一覧については、`ai_manifest`(4) のマニュアルページを参照してください。
	
	`js2ai` による変換:
	
	`js2ai` 変換の場合、*disk_name* の値はデバイスにする必要があります。`all` のデバイスはサポートされていません。`fdisk` タイプは `solaris` にする必要があります。サイズ 0 または `delete` はサポートされていません。パーティション分割が `default` で、`rootdisk` が設定されていない場合、`js2ai` は最初に見つかった `fdisk solaris` パーティションをルートディスクとして設定します。

`filesys`	UFS ファイルシステムはサポートされません。AI は ZFS ファイルシステムをインストールします。 `js2ai` による変換: ルートに使用するデバイスを決定するための方法が他にない場合、`filesys` の行で `/` のマウントポイントが付いたデバイスがルートプールとして使用されます。 指定されたマウントポイントが `/` または `swap` の場合は、ローカルおよびミラー化されたファイルシステムがサポートされます。 マウントポイントが `/` または `swap` でない場合、行はログに記録されて無視されます。JumpStart *fsoptions* はサポートされません。 サイズの検証は実行されません。このマニフェストを使用して正しくインストールを行うには、生成された AI マニフェスト内で指定されるサイズを調整することが必要な場合があります。

表 2–2 JumpStart プロファイルファイルのキーワードと AI マニフェストの指令の比較 (続き)

JumpStart プロファイルファイルのキーワード	AI マニフェストの指令

`filesys c1t0d0s0 10000 /`	次の AI マニフェストの一部は x86 プラットフォーム用のものです。SPARC プラットフォームの場合、`partition` 要素は指定されません。
	
	`<target>`
	`  <disk>`
	`  <disk_name name="c1t0d0" name_type="ctd"/>`
	`  <partition action="create" name="1" part_type="191">`
	`  <slice action="create" force="true"`
	`  in_vdev="rpool_vdev" in_zpool="rpool" name="0">`
	`  <size val="10000mb"/>`
	`  </slice>`
	`  </partition>`
	`  </disk>`
	`  <logical nodump="true" noswap="false">`
	`  <zpool is_root="true" name="rpool">`
	`  <vdev name="rpool_vdev" redundancy="none"/>`
	`  </zpool>`
	`  </logical>`
	`</target>`
	
	`js2ai` による変換:
	
	`js2ai` ツールでは、ルートファイルシステム (`/`) および swap の変換のみサポートします。

表 2–2 JumpStart プロファイルファイルのキーワードと AI マニフェストの指令の比較 (続き)

JumpStart プロファイルファイルのキーワード	AI マニフェストの指令

`filesys mirror:rpool c6t0d0s0 c6t1d0s0 60048 /`	次の AI マニフェストの一部は x86 プラットフォーム用のものです。SPARC プラットフォームの場合、`partition` 要素は指定されません。
	`<target>`
	`  <disk>`
	`  <disk_name name="c6t1d0" name_type="ctd"/>`
	`  <partition action="create" name="1" part_type="191">`
	`  <slice action="create" force="true"`
	`  in_vdev="rpool_vdev" in_zpool="rpool" name="0">`
	`  <size val="60048mb"/>`
	`  </slice>`
	`  </partition>`
	`  </disk>`
	`  <disk>`
	`  <disk_name name="c6t0d0" name_type="ctd"/>`
	`  <partition action="create" name="1" part_type="191">`
	`  <slice action="create" force="true"`
	`  in_vdev="rpool_vdev" in_zpool="rpool" name="0">`
	`  <size val="60048mb"/>`
	`  </slice>`
	`  </partition>`
	`  </disk>`
	`  <logical nodump="true" noswap="false">`
	`  <zpool is_root="true" name="rpool">`
	`  <vdev name="rpool_vdev" redundancy="mirror"/>`
	`  </zpool>`
	`  </logical>`
	`</target>`

`filesys` *server*`:`*path*	サポートされていません。AI はリモートファイルシステムをサポートしません。
`forced_deployment`	サポートされていません。AI はフラッシュアーカイブからインストールしません。
`geo`	サポートされていません。AI では、言語サポート用の地理的地域はシステム構成プロファイルで指定されます。『Oracle Solaris 11 システムのインストール』の第 11 章「クライアントシステムの構成」を参照してください。
`install_type`	AI は `initial_install` のみサポートします。
`layout_constraint`	サポートされていません。AI は `initial_install` のみサポートします。
`local_customization`	サポートされていません。AI はフラッシュアーカイブからインストールしません。

表 2–2 JumpStart プロファイルファイルのキーワードと AI マニフェストの指令の比較 (続き)

JumpStart プロファイルファイルのキーワード	AI マニフェストの指令
`locale`	サポートされていません。AI では、ロケールサポートはシステム構成プロファイルで指定されます。『Oracle Solaris 11 システムのインストール』の第 11 章「クライアントシステムの構成」を参照してください。
`metadb`	サポートされていません。
`no_content_check`	サポートされていません。AI はフラッシュアーカイブからインストールしません。
`no_master_check`	サポートされていません。AI はフラッシュアーカイブからインストールしません。
`num_clients`	サポートされていません。

`package SUNWpampkcs11`
`package SUNWpampkcs11 add`
	`<software type="IPS">`
	`  <software_data action="install">`
	`  <name>pkg:/SUNWpampkcs11</name>`
	`  </software_data>`
	`</software>`
	
	パッケージ名が有効であることを確認する方法については、26 ページの「ソフトウェアパッケージの変換」を参照してください。

`package SUNWpampkcs11 add nfs golden:/packages/Solaris_10`	AI マニフェストでは、すべてのソフトウェアは IPS パッケージリポジトリから取得されます。IPS リポジトリの場所は HTTP またはファイルとすることができます。ソフトウェアソースとして複数の IPS リポジトリを指定できます。パブリッシャーの指定方法については `ai_manifest(4)` のマニュアルページを参照してください。 `js2ai` による変換: `js2ai` ツールは *retrieval_type* `location` 引数を無視します。

`package SUNWpampkcs11 delete`	`<software type="IPS">`
	`  <software_data action="uninstall">`
	`  <name>pkg:/SUNWpampkcs11</name>`
	`  </software_data>`
	`</software>`
	
	AI は初期インストールのみサポートするため、パッケージを削除するために AI マニフェストを使用する必要はありません。

表 2–2 JumpStart プロファイルファイルのキーワードと AI マニフェストの指令の比較 (続き)

JumpStart プロファイルファイルのキーワード	AI マニフェストの指令
`partitioning default` `partitioning explicit`	ターゲットのインストールデバイスを指定しない場合、AI はデフォルトのインストールデバイスとデフォルトのパーティション分割を決定します。カスタムのパーティション分割を指定するには、`ai_manifest(4)` のマニュアルページを参照してください。 `js2ai` による変換: `js2ai` では、パーティション分割は `default` または `explicit` とする必要があります。`explicit` の場合、`swap` および `/` のみがサポートされます
`partitioning existing`	サポートされていません。`js2ai` では、パーティション分割は `default` または `explicit` とする必要があります。
`patch`	サポートされていません。AI は `initial_install` のみサポートします。 システムまたは特定のパッケージを更新するには、インストールされたシステム上で `pkg update` コマンドを使用します。

`pool newpool auto auto auto c1t0d0s0`	`<target>`
	`  <disk>`
	`  <disk_name name="c1t0d0" name_type="ctd"/>`
	`  <partition action="create" name="1" part_type="191">`
	`  <slice action="create" force="true"`
	`  in_vdev="rpool_vdev" in_zpool="newpool" name="0"/>`
	`  </partition>`
	`  </disk>`
	`  <logical nodump="false" noswap="false">`
	`  <zpool is_root="true" name="newpool">`
	`  <vdev name="rpool_vdev" redundancy="none"/>`
	`  </zpool>`
	`  </logical>`
	`</target>`
	
	`js2ai` による変換:
	
	プロファイル内でプールが指定された場合、`js2ai` は指定されたデバイスを使用して ZFS ルートプールを作成します。ZFS ルートプールに使用するデバイスを `js2ai` が決定する際、`pool` キーワードが他のすべてのキーワードに優先されます。
	
	`js2ai` はプールサイズ、スワップサイズ、またはダンプサイズのあらゆる検証を実行しません。このマニフェストを使用したインストールに成功するには、生成された AI マニフェスト内のサイズを調整することが必要な場合があります。

表 2–2 JumpStart プロファイルファイルのキーワードと AI マニフェストの指令の比較 (続き)

JumpStart プロファイルファイルのキーワード	AI マニフェストの指令
pool newpool auto auto auto any	js2ai で、物理デバイス名の代わりに any を指定する場合、any パラメータを含む指定よりも前にデバイス情報を提供しておく必要があります。たとえば、この pool 指定の前に root_device または usedisk 指定を行うことができます。例については、34 ページの「mirrorpool.profile エラーの修正」を参照してください。
root_device c1t0d0s0	<target> <disk whole_disk="true"> <disk_name name="c1t0d0" name_type="ctd"/> <partition action="create" name="1" part_type="191"> <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0"/> </partition> </disk> <logical nodump="true" noswap="false"> <zpool is_root="true" name="rpool"> <vdev name="rpool_vdev" redundancy="none"/> </zpool> </logical> </target> js2ai による変換: root_device が指定された場合、js2ai は指定されたデバイスに rootdisk を設定します。
system_type	AI マニフェストはシステムタイプを区別しません。 js2ai による変換: 値 standalone のみがサポートされています。
usedisk	js2ai による変換: js2ai ツールは後続の any または rootdisk 指定を解決するために、指定された 1 つ以上のデバイスを使用する場合があります。指定されたがこの目的で使用されないデバイスは、ZFS ルートプールがミラー化されていない場合、js2ai によってこのプールに追加されます。

js2ai の使用による JumpStart ルールおよびプロファイルから AI 条件およびマニフェストへの変換

js2ai ユーティリティーで -r オプションを使用すると、JumpStart ルールおよび関連付けられたプロファイルが AI 条件およびマニフェストに変換されます。最初は -S オプションを使用して検証をスキップします。

```
/usr/sbin/js2ai -rS [-d sysidcfg_dir] [-D destination_dir]
```

このコマンドは、rules ファイルおよび rules ファイルによって参照されるプロファイルで変換操作を実行します。rules ファイルで参照される各プロファイルは、AI クライアントプロビジョニングマニフェスト (/usr/share/auto_install/manifest/default.xml) に対して処理されます。このステップでは、JumpStart rules ファイルで指定されたプロファイルごとに、AI_*profile_name* という名前のディレクトリを作成します。AI_*profile_name* ディレクトリには、このプロファイルを参照するルールに対応する AI 条件ファイルが criteria-*rule_number*.xml の形式で格納されます。また、AI_*profile_name* ディレクトリには、*profile_name* プロファイルファイルに対応する AI マニフェストファイルが *profile_name*.*arch*.xml の形式で格納されます。

変換が正常に完了したというメッセージが表示されない場合、エラーレポートおよび js2ai.log ファイルを調べます。エラーレポートおよびログファイルは、警告、処理エラー、サポートされない項目、変換エラー、および検証エラーを報告します。エラーレポートは stdout への表出力で、rules およびプロファイルファイルの変換中に発生したタイプごとのエラーの番号を示します。ログファイルには問題が記述されます。

1. 処理エラーを修正します。
2. rules ファイルおよびプロファイルファイルから、サポート外項目として一覧表示されたすべての行を削除します。
3. 変換エラーを検査して、可能な場合はエラーを修正します。それ以外の場合は、エラーの原因となっている行を削除します。
4. 警告メッセージを検査して、修正が必要ないことを確認します。

変換が正しく完了したことを示すメッセージを受け取ったら、-S オプションを付けずに js2ai コマンドを実行して、出力された AI マニフェストを検証します。検証エラーは AI マニフェストファイル内で修正する必要があります。

特定の出力 AI マニフェストを適切な AI DTD に対して検証するには、js2ai コマンドに -V オプションを付けて実行します。

```
/usr/sbin/js2ai -V manifest_file
```

AI マニフェストは、AI インストールサービスに追加されるときも検証されます。

js2ai ユーティリティーを使用して、rules ファイルとその rules ファイルに関連付けられたすべてのプロファイルの代わりに、1 つの JumpStart プロファイルファイルを変換するには、-r オプションの代わりに -p オプションを使用します。

```
/usr/sbin/js2ai -p JS_profile_name [-d sysidcfg_dir] [-D destination_dir]
```

js2ai による JumpStart デバイス指定の変換方法

このセクションでは、js2ai が JumpStart プロファイル指定から AI マニフェスト用のターゲット要素を決定する方法について説明します。

システムのルートディスクを決定する方法

プロファイルの変換プロセス中は、js2ai はプロファイルが参照するクライアントシステムにアクセスしないため、js2ai はできる限り JumpStart と一致するプロセスを使用して、変換中にルートディスクを決定しようとします。

js2ai ツールは次のステップを実行して、ルートディスクで使用するデバイスを決定します。

1. プロファイルで root_device キーワードが指定されている場合、js2ai は rootdisk をスライスが存在するデバイスに設定します。
2. rootdisk が設定されておらず、プロファイルで boot_device キーワードが指定されている場合、js2ai は rootdisk をブートデバイスに設定します。
3. rootdisk が設定されておらず、partitioning default が指定され、solaris fdisk エントリが指定されている場合、js2ai は rootdisk を指定された disk_name に設定します。
4. rootdisk が設定されておらず、プロファイルで filesys *cwtxdysz size* / エントリが指定されている場合、js2ai は rootdisk をそのエントリで指定された *cwtxdysz* ディスクに設定します。
5. rootdisk が設定されておらず、プロファイルで usedisk *disk_name* エントリが指定されている場合、js2ai は rootdisk をそのエントリで指定された *disk_name* ディスクに設定します。
6. rootdisk が設定されておらず、プロファイルで次の指定が見つかった場合 (*size* は 0 または delete でなく、*disk_name* は all でない)、rootdisk はこの *disk_name* に設定されます。

   fdisk *disk_name* solaris *size*
7. rootdisk が設定されていない場合、デバイスが rootdisk として指定されたオカレンスで変換エラーが生成されます。

any デバイスの変換方法

js2ai ツールは次のステップを実行して、any キーワードが指定されているときに使用するデバイスを決定します。

1. any デバイスが指定されていて、キーワードアクション (ミラー化されていないプール、または / マウントポイントを使用した filesys) が指定されている場合、any デバイスは rootdisk に設定されます (rootdisk が設定されている場合)。
2. any デバイスが変換されておらず、usedisk 文がプロファイル内に存在する場合、any デバイスが usedisk 文で指定されたデバイスに設定されます。
3. any デバイスが変換されておらず、any デバイスが指定されたアクションによって ZFS ルートプールが作成される場合、AI はそのデバイスを選択します。ミラー化されたプールが指定されている場合、これは適用できません。

ZFS ルートプールを決定する方法

js2ai ツールは次のステップを実行して、ZFS ルートプールで使用するデバイスを決定します。ZFS ルートプールが決定されると、その後に検出される定義がすでに決定されている ZFS ルートプールと競合する場合、エラーとしてフラグが設定されます。

1. プロファイルで pool キーワードが指定されている場合、js2ai は ZFS ルートプールを pool キーワードで指定されたデバイスに設定します。
2. ZFS ルートプールが決定されておらず、プロファイルでマウントポイント / を使用して filesys が指定された場合、ZFS ルートプールは指定されたデバイスを使用して作成されます。
3. ZFS ルートプールが決定されておらず、プロファイル内のすべてのキーワードが処理され、rootdisk が設定されている場合、ZFS ルートプールは rootdisk デバイスを使用して作成されます。
4. ZFS ルートプールが決定されておらず、パーティションタイプが default の場合、AI は ZFS ルートプールで使用するデバイスを選択します。
5. ZFS ルートプールが決定されておらず、処理中にエラーが発生しなかった場合、AI は ZFS ルートプールで使用するデバイスを選択します。
6. ZFS ルートプールがミラー化されたプールではなく、指定された 1 つ以上の usedisk デバイスが rootdisk デバイスまたは any デバイスの変換で使用されていない場合、これらのディスクが ZFS ルートプールに追加されます。

ソフトウェアパッケージの変換

JumpStart プロファイルでは cluster および package キーワードを使用してソフトウェアをシステムにインストールします。cluster キーワードは AI ではサポートされません。IPS では、合体パッケージまたはグループパッケージをインストールする構文は、他のパッケージをインストールする構文と同じです。JumpStart プロファイル内の cluster を単に package に変更すると、js2ai は AI マニフェスト内で正しいパッケージインストール指定を作成します。

ヒント – AI マニフェスト内のパッケージ名を確認してください。AI マニフェスト内でインストール用に指定されたパッケージが、その AI マニフェスト内で指定されたどのパブリッシャー起点からも利用できない場合、クライアントインストールは失敗します。

IPS パッケージ名は SVR4 パッケージ名と異なります。たとえば、SVR4 パッケージ SUNWpampkcs11 の名前は IPS では library/security/pam/module/pam-pkcs11 に変更されます。

一部の SVR4 パッケージ名は IPS 内に存在するため、SVR4 名を使用して IPS パッケージをインストールできます。たとえば、AI マニフェストで SUNWpampkcs11 パッケージのインストールが指定された場合、library/security/pam/module/pam-pkcs11 パッケージが自動的にインストールされます。この場合、パッケージの名前が変更されます。

一部の SVR4 パッケージ名は IPS 内に存在しません。これらの場合、パッケージ名を変更するか、その指定を AI マニフェストから削除する必要があります。たとえば、SUNWCall および SUNWCuser パッケージは IPS 内で名前変更されません。AI マニフェストでこれらのパッケージが指定された場合、インストールは失敗します。

js2ai ユーティリティーは、JumpStart プロファイルファイルからの指定を含む新規 AI マニフェストを構築するための基礎として、/usr/share/auto_install/manifest/default.xml AI マニフェストを使用します。このデフォルト AI マニフェストは、基本オペレーティングシステムをインストールする 2 つのパッケージ (entire および solaris-large-server) のインストールを指定します。これら 2 つのパッケージの他は、おそらく追加のツールおよびアプリケーションのインストールを指定するだけで済みます。

AI マニフェスト内で特定のパッケージ名が使用できるかどうかを判断するには、Oracle Solaris 11 システムで pkg list コマンドを使用します。AI マニフェスト内で指定された IPS パッケージリポジトリ起点からのパッケージを一覧表示するための -g オプションを必ず使用してください。この例では、AI マニフェストは http://pkg.oracle.com/solaris11/release リポジトリ起点を指定します。

```
$ pkg list -af -g http://pkg.oracle.com/solaris11/release SUNWCall SUNWCuser
pkg list: no packages matching ‘SUNWCuser, SUNWCall’ known
```

このメッセージは、これら 2 つのパッケージがこの AI マニフェスト内で使用できないことを示しています。

```
$ pkg list -af -g http://pkg.oracle.com/solaris11/release SUNWpampkcs11
NAME (PUBLISHER)                                        VERSION              IFO
SUNWpampkcs11                                           0.6.0-0.133          --r
```

最も右側の列の「r」は、このパッケージの名前が変更されていることを示します。この名前を AI マニフェスト内で使用できますが、pkg info コマンドを使用してパッケージの新しい名前を決定した方がよい場合もあります。

次の出力の「Renamed to」行を参照してください。SUNWpampkcs11 パッケージの名前は library/security/pam/module/pam-pkcs11 に変更されています。今後の Oracle Solaris 更新との互換性を高めるために、AI マニフェスト内で library/security/pam/module/pam-pkcs11 を指定した方がよい場合もあります。

```
$ pkg info -r SUNWpampkcs11
          Name: SUNWpampkcs11
       Summary:
         State: Not installed (Renamed)
```

```
    Renamed to: library/security/pam/module/pam-pkcs11@0.6.0-0.133
                consolidation/sfw/sfw-incorporation
     Publisher: solaris
       Version: 0.6.0
 Build Release: 5.11
        Branch: 0.133
Packaging Date: Wed Oct 27 18:50:11 2010
          Size: 0.00 B
          FMRI: pkg://solaris/SUNWpampkcs11@0.6.0,5.11-0.133:20101027T185011Z
$ pkg info -r pam-pkcs11
          Name: library/security/pam/module/pam-pkcs11
       Summary: The OpenSC PKCS#11 PAM Login Tools
      Category: System/Security
         State: Not installed
     Publisher: solaris
       Version: 0.6.0
 Build Release: 5.11
        Branch: 0.173.0.0.0.0.487
Packaging Date: Sun Aug 28 00:16:46 2011
          Size: 1.92 MB
          FMRI: pkg://solaris/library/security/pam/module/pam-pkcs11@...

$ pkg list -af -g http://pkg.oracle.com/solaris/release SUNWmysql
NAME (PUBLISHER)                                        VERSION              IFO
SUNWmysql                                               4.0.24-0.142         --o
```

最も右側の列の「o」は、このパッケージが廃止されていることを示します。このパッケージ名は AI マニフェスト内で使用できません。pkg list コマンドにワイルドカードを指定して使用するか、pkg search コマンドを使用して、使用可能な別の名前でパッケージが使用できるかどうかを判断します。

```
$ pkg list -af SUNWmysql*
NAME (PUBLISHER)                                        VERSION              IFO
SUNWmysql                                               4.0.24-0.142         --o
SUNWmysql-base                                          0.5.11-0.133         --r
SUNWmysql-python                                        0.5.11-0.162         --o
SUNWmysql-python26                                      0.5.11-0.133         --r
SUNWmysql5                                              5.0.86-0.171         --o
SUNWmysql5                                              5.0.86-0.133         --r
SUNWmysql51                                             5.1.37-0.133         --r
SUNWmysql51lib                                          5.1.37-0.133         --r
SUNWmysql51test                                         5.1.37-0.133         --r
SUNWmysql5jdbc                                          5.1.5-0.171          --o
SUNWmysql5jdbc                                          5.1.5-0.133          --r
SUNWmysql5test                                          5.0.86-0.171         --o
SUNWmysql5test                                          5.0.86-0.133         --r
SUNWmysqlt                                              4.0.24-0.142         --o
$ pkg info -r SUNWmysql51
          Name: SUNWmysql51
       Summary:
         State: Not installed (Renamed)
    Renamed to: database/mysql-51@5.1.37-0.133
                consolidation/sfw/sfw-incorporation
     Publisher: solaris
       Version: 5.1.37
 Build Release: 5.11
        Branch: 0.133
```

```
Packaging Date: Wed Oct 27 18:49:18 2010
          Size: 0.00 B
          FMRI: pkg://solaris/SUNWmysql51@5.1.37,5.11-0.133:20101027T184918Z
```

AI マニフェスト内で SUNWmysql を database/mysql-51 に置換した方がよい場合もあります。

js2ai を使用したルールおよびプロファイルの変換例

このセクションでは、単一の js2ai コマンドの使用による、JumpStart rules ファイルおよびその rules ファイルによって参照されるすべてのプロファイルファイルの変換を示します。デフォルトでは、変換された各プロファイルは、AI_*profile-filename* という名前のディレクトリに出力されます。そのプロファイルを選択するルールは、AI 基準ファイルと同じディレクトリに出力されます。

JumpStart ルールファイルとプロファイルファイルの例

この例では次の rules ファイルを使用します。

```
# The following rule matches only one system:

hostname sample_host    - fdisk.profile         -

# The following rule matches only one system:

hostaddress 10.6.68.127 - mirrorfilesys.profile -

# The following rule matches any system that is on the 924.222.43.0 network:

network 924.222.43.0    - rootdisk.profile      -

# The following rule matches all x86 systems:

arch i386               - mirrorpool.profile    -
```

fdisk.profile ファイルの内容は次のとおりです。

```
install_type initial_install
system_type server
root_device c1t0d0s0
usedisk c1t0d0
fdisk rootdisk solaris all
partitioning explicit
filesys rootdisk.s1 5000 swap
filesys rootdisk.s0 10000 /
cluster SUNWCall
```

mirrorfilesys.profile ファイルの内容は次のとおりです。

```
install_type initial_install
partitioning default
filesys mirror c6t0d0s0 c6t1d0s0 60048
cluster SUNWCuser
```

mirrorpool.profile ファイルの内容は次のとおりです。

```
install_type initial_install
partitioning default
pool newpool auto auto auto mirror any any
cluster SUNWCuser
```

rootdisk.profile ファイルの内容は次のとおりです。

```
install_type initial_install
partitioning explicit
filesys rootdisk.s0 15000 /
filesys rootdisk.s1 1000 swap
cluster SUNWCall
```

ルールファイルオプションを指定した js2ai の使用

次のコマンドを使用して、この rules ファイルを処理します。検証が実行されなかったため、エラーレポートでは検証エラーはハイフン文字として表示されます。検証は -S オプションによって抑制されます。

```
# js2ai -rS
                                  Process  Unsupported  Conversion  Validation
Name                   Warnings   Errors   Items        Errors      Errors
-------------------    --------   -------  -----------  ----------  ----------
rules                         0         0            1           0           -
fdisk.profile                 0         0            2           0           -
mirrorfilesys.profile         0         0            2           0           -
mirrorpool.profile            0         0            1           1           -
rootdisk.profile              0         0            1           2           -

Conversion completed. One or more failures occurred.
For errors see js2ai.log
# cat js2ai.log
rules:line 3:UNSUPPORTED: unsupported keyword: hostname
fdisk.profile:line 2:UNSUPPORTED: unsupported value for 'system_type' specified: server
fdisk.profile:line 9:UNSUPPORTED: unsupported keyword: cluster
mirrorfilesys.profile:line 3:UNSUPPORTED: unsupported mount point of 'unnamed' specified,
mount points other than '/' and 'swap' are not supported
mirrorfilesys.profile:line 4:UNSUPPORTED: unsupported keyword: cluster
mirrorpool.profile:line 3:CONVERSION: unable to convert 'any' device to physical device. Replace 'any'
with actual device name
mirrorpool.profile:line 4:UNSUPPORTED: unsupported keyword: cluster
rootdisk.profile:line 3:CONVERSION: unable to convert 'rootdisk.s0'.  Replace'rootdisk.' with actual
device name
rootdisk.profile:line 4:CONVERSION: unable to convert 'rootdisk.s1'.  Replace'rootdisk.' with actual
device name
rootdisk.profile:line 5:UNSUPPORTED: unsupported keyword: cluster
```

出力は、AI_*profile-filename* という名前のディレクトリに格納されます。JumpStart ルールから作成されたすべての AI 条件ファイルは、rules ファイル内のルールの位

置から名前が付けられます。AI マニフェストには *profile-filename*. *arch*.xml という名前が付けられ、*arch* は generic、x86、または sparc です。

```
# ls AI_*
AI_fdisk.profile:
fdisk.profile.x86.xml

AI_mirrorfilesys.profile:
criteria-2.xml      mirrorfilesys.profile.generic.xml

AI_mirrorpool.profile:
criteria-2.xml      mirrorpool.profile.generic.xml

AI_rootdisk.profile:
criteria-3.xml      rootdisk.profile.generic.xml
```

同等の AI 条件ファイル

ログファイルでは、JumpStart ルールキーワード hostname が、AI マニフェストを選択するための AI 条件キーワードとしてサポートされないと報告されています。hostname キーワードは、fdisk.profile プロファイルを使用するクライアントを指定するために使用されます。hostname は AI マニフェストを選択するためにサポートされるキーワードではないため、新しい AI_fdisk.profile ディレクトリには AI 条件ファイルが含まれません。

この問題は、rules ファイル内で hostname を hostaddress に変更することによって修正できます。

この問題は、hostname システムを識別する MAC アドレスまたは IP アドレスを指定する AI 条件ファイルを作成することによっても修正できます。たとえば、sample_host の MAC アドレスが 0:14:4F:20:53:97 の場合、次の条件ファイルは JumpStart ルール hostname sample_host と同等です。

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="mac">
    <value>0:14:4F:20:53:97</value>
  </ai_criteria>
</ai_criteria_manifest>
```

システムの MAC アドレスを見つけるには、『Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化』および dladm(1M) のマニュアルページに記載されているように、dladm コマンドを使用してください。

JumpStart ルール hostaddress 10.6.68.127 について、js2ai ユーティリティーは AI 条件ファイル AI_mirrorfilesys.profile/criteria-2.xml を自動的に作成し、JumpStart の hostaddress キーワードを AI の ipv4 キーワードに置換します。

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="ipv4">
```

```
    <value>
      10.6.68.127
    </value>
  </ai_criteria>
</ai_criteria_manifest>
```

JumpStart ルール network 924.222.43.0 について、js2ai ユーティリティーは AI 条件ファイル AI_rootdisk.profile/criteria-3.xml を自動的に作成し、指定されたネットワークアドレスに基づいて IP アドレスの範囲を指定します。

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="ipv4">
    <range>
      924.222.43.0 924.222.43.255
    </range>
  </ai_criteria>
</ai_criteria_manifest>
```

JumpStart ルール arch i386 について、js2ai ユーティリティーは AI 条件ファイル AI_mirrorpool.profile/criteria-4.xml を自動的に作成し、JumpStart の arch キーワードを AI の cpu キーワードに置換します。

```
<?xml version="1.0" encoding="utf-8"?>
<ai_criteria_manifest>
  <ai_criteria name="cpu">
    <value>
      i386
    </value>
  </ai_criteria>
</ai_criteria_manifest>
```

同等の AI マニフェストファイル

js2ai ユーティリティーは通常、エラーが報告される場合であっても、JumpStart プロファイルごとに AI マニフェストを作成します。このセクションでは、出力される AI マニフェストがさらに完全になるようにするために、一部の一般的なエラーに対処する方法について説明します。

fdisk.profile エラーの修正

js2ai ユーティリティーは fdisk.profile JumpStart プロファイルについて次のエラーを表示しました。

```
fdisk.profile:line 2:UNSUPPORTED: unsupported value for ’system_type’ specified: server
fdisk.profile:line 9:UNSUPPORTED: unsupported keyword: cluster
```

これら 2 つの行は無視され、出力 AI マニフェストに影響しません変換でエラーメッセージを出さないようにする場合は、これら 2 つの行を削除できます。fdisk.profile ファイルの内容は次のとおりです。

```
install_type initial_install
root_device c1t0d0s0
usedisk c1t0d0
fdisk rootdisk solaris all
partitioning explicit
filesys rootdisk.s1 5000 swap
filesys rootdisk.s0 10000 /
```

js2ai ユーティリティーの -p オプションを使用して、このプロファイルのみ処理します。

```
# js2ai -p fdisk.profile
Successfully completed conversion
```

出力 AI マニフェスト AI_fdisk.profile/fdisk.profile.x86.xml の target スタンザには次の内容があります。

```
<target>
  <disk whole_disk="true">
    <disk_name name="c1t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" is_swap="true" name="1">
        <size val="5000mb"/>
      </slice>
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0">
        <size val="10000mb"/>
      </slice>
     </partition>
  </disk>
  <logical nodump="true" noswap="false">
    <zpool is_root="true" name="rpool">
      <vdev name="rpool_vdev" redundancy="none"/>
    </zpool>
  </logical>
</target>
```

software スタンザは /usr/share/auto_install/manifest/default.xml 内のものと同じです。これは、このプロファイルにはパッケージ指定が含まれていないためです。

mirrorfilesys.profile エラーの修正

js2ai ユーティリティーは mirrorfilesys.profile JumpStart プロファイルについて次のエラーを表示しました。

```
mirrorfilesys.profile:line 3:UNSUPPORTED: unsupported mount point of ’unnamed’ specified,
mount points other than ’/’ and ’swap’ are not supported
mirrorfilesys.profile:line 4:UNSUPPORTED: unsupported keyword: cluster
```

JumpStart プロファイルファイルでは、filesys mirror 指定の *file_system* パラメータは省略可能です。js2ai 変換では、*file_system* パラメータは省略できず、/ または swap の 2 つ値のいずれかを持つ必要があります。

mirrorfilesys.profile ファイルを編集して、filesys 指定の末尾に「/」を追加し、cluster 行を削除します。すると、mirrorfilesys.profile ファイルの内容は次のようになります。

```
install_type initial_install
partitioning default
filesys mirror c6t0d0s0 c6t1d0s0 60048 /
```

AI_mirrorfilesys.profile/criteria-2.xml ファイルを別の場所に保存します。次に、js2ai ユーティリティーの -p オプションを使用して、このプロファイルのみ処理します。

```
# js2ai -p mirrorfilesys.profile
Successfully completed conversion
```

AI_mirrorfilesys.profile ディレクトリには、mirrorfilesys.profile.sparc.xml および mirrorfilesys.profile.x86.xml の 2 つの出力 AI マニフェストが格納されています。これら 2 つの AI マニフェストの違いは、mirrorfilesys.profile.x86.xml には 2 つの partition 指定が含まれていることだけです。mirrorfilesys.profile.x86.xml マニフェストの target スタンザの内容は次のとおりです。

```
<target>
  <disk>
    <disk_name name="c6t1d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0">
        <size val="60048mb"/>
      </slice>
    </partition>
  </disk>
  <disk>
    <disk_name name="c6t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0">
        <size val="60048mb"/>
      </slice>
    </partition>
  </disk>
  <logical nodump="true" noswap="false">
    <zpool is_root="true" name="rpool">
      <vdev name="rpool_vdev" redundancy="mirror"/>
    </zpool>
  </logical>
</target>
```

software スタンザは /usr/share/auto_install/manifest/default.xml 内のものと同じです。これは、このプロファイルにはパッケージ指定が含まれていないためです。

mirrorpool.profile エラーの修正

js2ai ユーティリティーは mirrorpool.profile JumpStart プロファイルについて次のエラーを表示しました。

```
mirrorpool.profile:line 4:CONVERSION: unable to convert 'any' device to physical device.
Replace 'any' with actual device name
mirrorpool.profile:line 8:UNSUPPORTED: unsupported keyword: cluster
```

これらのエラーを修正するには、次の 2 つの方法のいずれかを使用します。これらの修正は、両方ともまったく同じ AI マニフェスト出力を生成します。25 ページの「any デバイスの変換方法」も参照してください。

- mirrorpool.profile プロファイルを編集し、any を物理デバイス名に置換します。また、cluster 行を削除します。すると、mirrorpool.profile ファイルの内容は、たとえば次のようになります。

  ```
  install_type initial_install
  partitioning default
  pool newpool auto auto auto mirror c6t0d0s0 c6t1d0s0
  ```

- pool 指定の *vdevlist* 一覧に、物理デバイス名でなく any を指定した場合、any パラメータを含む指定よりも前にデバイス情報を提供しておく必要があります。mirrorpool.profile プロファイルを編集して、プール指定の前に usedisk 指定を追加します。また、cluster 行を削除します。すると、mirrorpool.profile ファイルの内容は、たとえば次のようになります。

  ```
  install_type initial_install
  partitioning default
  usedisk c6t0d0 c6t1d0
  pool newpool auto auto auto mirror any any
  ```

AI_mirrorpool.profile/criteria-4.xml ファイルを別の場所に保存します。次に、js2ai ユーティリティーの -p オプションを使用して、このプロファイルのみ処理します。

```
# js2ai -p mirrorpool.profile
Successfully completed conversion
```

AI_mirrorpool.profile ディレクトリには、mirrorpool.profile.sparc.xml および mirrorpool.profile.x86.xml の 2 つの出力 AI マニフェストが格納されています。これら 2 つの AI マニフェストの違いは、mirrorpool.profile.x86.xml には 2 つの partition 指定が含まれていることだけです。mirrorpool.profile.x86.xml マニフェストの target スタンザの内容は次のとおりです。

```
<target>
  <disk>
    <disk_name name="c6t1d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="newpool" name="0"/>
    </partition>
  </disk>
  <disk>
    <disk_name name="c6t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="newpool" name="0"/>
    </partition>
  </disk>
```

```
  <logical nodump="false" noswap="false">
    <zpool is_root="true" name="newpool">
      <vdev name="rpool_vdev" redundancy="mirror"/>
    </zpool>
  </logical>
</target>
```

software スタンザは /usr/share/auto_install/manifest/default.xml 内のものと同じです。これは、このプロファイルにはパッケージ指定が含まれていないためです。

rootdisk.profile エラーの修正

js2ai ユーティリティーは rootdisk.profile JumpStart プロファイルについて次のエラーを表示しました。

```
rootdisk.profile:line 3:CONVERSION: unable to convert ’rootdisk.s0’.
Replace’rootdisk.’ with actual device name
rootdisk.profile:line 4:CONVERSION: unable to convert ’rootdisk.s1’.
Replace’rootdisk.’ with actual device name
rootdisk.profile:line 5:UNSUPPORTED: unsupported keyword: cluster
```

これらのエラーを修正するには、次の 2 つの方法のいずれかを使用します。これらの修正は、両方ともまったく同じ AI マニフェスト出力を生成します。

- rootdisk.profile プロファイルを編集し、rootdisk. を c0t0d0 に置換します。また、cluster 行を削除します。すると、rootdisk.profile ファイルの内容は次のようになります。

  ```
  install_type initial_install
  partitioning explicit
  filesys c0t0d0s0 15000 /
  filesys c0t0d0s1 1000 swap
  ```

- rootdisk.profile プロファイルを編集して root_device 指定を追加し、cluster 行を削除します。すると、rootdisk.profile ファイルの内容は次のようになります。

  ```
  install_type initial_install
  partitioning explicit
  root_device c0t0d0s0
  filesys rootdisk.s0 15000 /
  filesys rootdisk.s1 1000 swap
  ```

AI_rootdisk.profile/criteria-3.xml ファイルを別の場所に保存します。次に、js2ai ユーティリティーの -p オプションを使用して、このプロファイルのみ処理します。

```
# js2ai -p rootdisk.profile
Successfully completed conversion
```

AI_rootdisk.profile ディレクトリには、rootdisk.profile.sparc.xml および rootdisk.profile.x86.xml の 2 つの出力 AI マニフェストが格納されています。これ

ら 2 つの AI マニフェストの違いは、rootdisk.profile.x86.xml に partition 指定が含まれていることだけです。rootdisk.profile.x86.xml マニフェストの target スタンザの内容は次のとおりです。

```
<target>
  <disk>
    <disk_name name="c0t0d0" name_type="ctd"/>
    <partition action="create" name="1" part_type="191">
      <slice action="create" force="true" in_vdev="rpool_vdev" in_zpool="rpool" name="0">
        <size val="15000mb"/>
      </slice>
      <slice action="create" force="true" is_swap="true" name="1">
        <size val="1000mb"/>
      </slice>
    </partition>
  </disk>
  <logical nodump="true" noswap="false">
    <zpool is_root="true" name="rpool">
      <vdev name="rpool_vdev" redundancy="none"/>
    </zpool>
  </logical>
</target>
```

software スタンザは /usr/share/auto_install/manifest/default.xml 内のものと同じです。これは、このプロファイルにはパッケージ指定が含まれていないためです。

詳細なプロファイル変換情報の表示

ルールまたはプロファイル変換あるいは AI マニフェスト検証に関する詳細な情報を表示するには、js2ai コマンドに -v オプションを指定します。-v オプションを指定すると、処理ステップが表示され、エラーレポートには rules ファイルまたはプロファイルをレポートから省略するのではなく、ゼロが表示されます。

```
# js2ai -v -p rootdisk.profile
Processing profile: rootdisk.profile
Performing conversion on: rootdisk.profile
Generating x86 manifest for: rootdisk.profile
Validating rootdisk.profile.x86.xml
Generating sparc manifest for: rootdisk.profile
Validating rootdisk.profile.sparc.xml

                                  Process  Unsupported  Conversion  Validation
Name                     Warnings Errors   Items        Errors      Errors
----------------------   -------- -------  -----------  ----------  ----------
rootdisk.profile                0        0            0           0           0
Successfully completed conversion
```

出力 AI マニフェストの検証

生成された AI マニフェストを検証するには -V オプションを使用します。

```
# js2ai -V ./AI_rootdisk.profile/rootdisk.profile.sparc.xml
Successfully completed conversion
# js2ai -v -V ./AI_rootdisk.profile/rootdisk.profile.sparc.xml
Validating rootdisk.profile.sparc.xml

                                   Process  Unsupported  Conversion  Validation
Name                     Warnings  Errors   Items        Errors      Errors
----------------------   --------  -------  -----------  ----------  ----------
rootdisk.profile.sparc          -        -            -           -           0
Successfully completed conversion
```

第 3 章

構成ファイルの変換

この章では、sysidcfg ファイルを AI システム構成プロファイルに変換する方法を示します。多くの変換は、js2ai ユーティリティーを使用して実行できます。js2ai ユーティリティーについての詳細は、js2ai(1M) のマニュアルページを参照してください。

AI システム構成プロファイルは、システム構成を指定する SMF XML プロファイルファイルです。AI システム構成プロファイルについては、『Oracle Solaris 11 システムのインストール』の第 11 章「クライアントシステムの構成」を参照してください。

sysidcfg ファイルキーワードとシステム構成プロファイル指令の比較

次の表では、sysidcfg ファイルのキーワードと AI システム構成プロファイル指定の例を比較します。異なる場合では、異なる SMF プロパティー値が適用されることもあります。42 ページの「js2ai を使用した sysidcfg の変換例」および『Oracle Solaris 11 システムのインストール』の「システム構成プロファイルの例」の例を参照してください。

表 3–1 sysidcfg ファイルキーワードと構成プロファイル指令の比較

sysidcfg ファイルのキーワード	システム構成プロファイルの指令
keyboard	js2ai ツールは変換を実行しません。sysidcfg ファイルで指定されたキーボードが Oracle Solaris 11 でサポートされていることを確認してください。 サービス名: system/keymap プロパティーグループ名: keymap プロパティー値名: layout

表 3–1 sysidcfg ファイルキーワードと構成プロファイル指令の比較 (続き)

sysidcfg ファイルのキーワード	システム構成プロファイルの指令
name_service	AI は DNS、NIS、および LDAP をサポートします。js2ai ツールは NONE、DNS、NIS、および LDAP の値をサポートします。NIS+ ネームサービスは NIS として変換されます。ネームサービスが構成されている場合は、ネットワークを DefaultFixed として構成する必要があります。 サービス名: system/name-service/switch プロパティーグループ名: config プロパティー値名: netgroup
network_interface	AI では、システムインストールの一部として単一インタフェースの構成のみがサポートされます。この制限のため、js2ai ツールは PRIMARY という名前のインタフェースまたは sysidcfg ファイル内で見つかった最初のインタフェースのみを処理します。name_service が指定されている場合、js2ai ツールはネットワークを DefaultFixed に設定します。適切に構成された DefaultFixed ネットワークは、ホスト名、IP アドレス、ネットマスク、およびゲートウェイを提供する必要があります。自動ネットワーク構成は、ネームサービスが指定されていない場合にのみサポートされます。 サービス名: network/install プロパティーグループ名: install_ipv4_interface プロパティー値名: static_address プロパティーグループ名: install_ipv6_interface
nfs4_domain	サポートされていません。
root_password	js2ai ツールは変換を実行しません。 サービス名: system/config-user プロパティーグループ名: root_account プロパティー値名: password
security_policy	js2ai ツールは値 NONE をサポートします。
service_profile	js2ai ツールは値 limited_net をサポートします。
system_locale	js2ai ツールは変換を実行しません。sysidcfg ファイルで指定されたロケールが Oracle Solaris 11 でサポートされていることを確認してください。 サービス名: system/environment プロパティーグループ名: environment プロパティー値名: LC_ALL
terminal	js2ai ツールは変換を実行しません。sysidcfg ファイルで指定された端末タイプが Oracle Solaris 11 でサポートされていることを確認してください。 サービス名: system/console-login プロパティーグループ名: ttymon プロパティー値名: terminal_type

表 3–1 sysidcfg ファイルキーワードと構成プロファイル指令の比較 (続き)

sysidcfg ファイルのキーワード	システム構成プロファイルの指令
timeserver	js2ai ツールは値 localhost をサポートします。
timezone	js2ai ツールは変換を実行しません。 サービス名: system/timezone プロパティーグループ名: timezone プロパティー値名: localtime

js2ai を使用した sysidcfg ファイルからシステム構成プロファイルへの変換

js2ai ユーティリティーに -s オプションを指定して使用して、この JumpStart 構成に関連付けられた sysidcfg ファイルをシステム構成プロファイルファイルに変換します。最初は -S オプションを使用して検証をスキップします。

```
/usr/sbin/js2ai -sS [-d sysidcfg_dir] [-D destination_dir]
```

js2ai は、処理された sysidcfg ファイルごとに、js2ai コマンドが起動されたディレクトリ内に sc_profile.xml という名前の AI システム構成プロファイルファイルを作成します。sc_profile.xml ファイルに別のディレクトリを指定するには、-D オプションを使用します。

変換が正常に完了したというメッセージが表示されない場合、エラーレポートおよび js2ai.log ファイルを調べます。エラーレポートおよびログファイルは、警告、処理エラー、サポートされない項目、変換エラー、および検証エラーを報告します。エラーレポートは stdout への表出力で、sysidcfg ファイルの変換中に発生したタイプごとのエラーの番号を示します。ログファイルには問題が記述されます。

1. 処理エラーを修正します。
2. sysidcfg ファイルから、サポート外項目として一覧表示されたすべての行を削除します。
3. 変換エラーを検査して、可能な場合はエラーを修正します。それ以外の場合は、エラーの原因となっている行を削除します。
4. 警告メッセージを検査して、修正が必要ないことを確認します。

変換が正しく完了したことを示すメッセージを受け取ったら、-S オプションを付けずに js2ai コマンドを実行して、出力された sc_profile.xml ファイルを検証します。検証エラーは sc_profile.xml ファイル内で修正する必要があります。

特定の出力システム構成プロファイルを検証するには、-V オプションを付けて js2ai コマンドを実行します。

```
# js2ai -V path/sc_profile.xml
```

システム構成プロファイルは、プロファイルを AI インストールサービスに追加したとき、または AI インストールサーバー上で次のコマンドを実行したときにも検証されます。

```
# installadm validate -n install_service_name sc_profile.xml
```

installadm validate コマンドについては、『Oracle Solaris 11 システムのインストール』の「システム構成プロファイルを検証する」を参照してください。

js2ai を使用した sysidcfg の変換例

このセクションでは、js2ai の使用による sysidcfg ファイルから AI システム構成プロファイルファイルへの変換を示します。js2ai は、処理された sysidcfg ファイルごとに、js2ai コマンドが起動されたディレクトリ内に sc_profile.xml という名前の AI システム構成プロファイルファイルを作成します。sc_profile.xml ファイルに別のディレクトリを指定するには、-D オプションを使用します。

サンプル sysidcfg ファイル

この例では、次の sysidcfg ファイルを使用します。

```
timezone=US/Pacific
terminal=xterms
timeserver=localhost
network_interface=primary {
                        hostname=host1
                        ip_address=10.80.127.35
                        netmask=255.255.255.224
                        protocol_ipv6=no
                        default_route=10.80.127.33}
root_password=rJmv5LUXM1OcU
security_policy=none
```

sysidcfg ファイルオプションを指定した js2ai の使用

次のコマンドを使用して、この sysidcfg ファイルを処理します。検証が実行されなかったため、エラーレポートでは検証エラーはハイフン文字として表示されます。検証は -S オプションによって抑制されます。

```
# js2ai -sS
                                Process  Unsupported  Conversion  Validation
Name                  Warnings  Errors   Items        Errors      Errors
-------------------   --------  -------  -----------  ----------  ----------
sysidcfg                     0        0            0           1           -

Conversion completed. One or more failures occurred.
For errors see js2ai.log
```

```
# cat js2ai.log
sysidcfg:line 4:CONVERSION: when the PRIMARY interface is specified, by default the
system will be configured for both IPv4 and IPv6 via automatic network configuration.
The options specified will be ignored. If you wish to configure the interface with
the specified options replace PRIMARY with the name of the interface that should be
configured.
```

同等のシステム構成プロファイルファイル

次に示す変更された sysidcfg ファイルは、js2ai.log ファイルで報告されたエラーに対処します。PRIMARY インタフェース指定はインタフェース名 e1000g で置換されています。

```
timezone=US/Pacific
terminal=xterms
timeserver=localhost
network_interface=e1000g {
                        hostname=host1
                        ip_address=10.80.127.35
                        netmask=255.255.255.224
                        protocol_ipv6=no
                        default_route=10.80.127.33}
root_password=rJmv5LUXM1OcU
security_policy=none

# js2ai -sS
Successfully completed conversion
```

今回はエラーレポートが出力されず、js2ai.log ファイルは空です。

次の sc_profile.xml ファイルが生成されます。

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE service_bundle
  SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle name="system configuration" type="profile">
  <service name="system/timezone" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="timezone" type="application">
        <propval name="localtime" type="astring" value="US/Pacific"/>
      </property_group>
    </instance>
  </service>
  <service name="system/console-login" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="ttymon" type="application">
        <propval name="terminal_type" type="astring" value="xterms"/>
      </property_group>
    </instance>
  </service>
  <service name="system/config-user" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="root_account" type="application">
        <propval name="password" type="astring" value="rJmv5LUXM1OcU"/>
      </property_group>
```

```
    </instance>
  </service>
  <service name="system/identity" type="service" version="1">
    <instance enabled="true" name="node">
      <property_group name="config" type="application">
        <propval name="nodename" type="astring" value="host1"/>
      </property_group>
    </instance>
  </service>
  <service name="network/install" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="install_ipv4_interface" type="application">
        <propval name="name" type="astring" value="e1000g/v4"/>
        <propval name="address_type" type="astring" value="static"/>
        <propval name="static_address" type="net_address_v4" value="10.80.127.35/27"/>
        <propval name="default_route" type="net_address_v4" value="10.80.127.33"/>
      </property_group>
    </instance>
  </service>
  <service name="network/physical" type="service" version="1">
    <instance enabled="true" name="default">
      <property_group name="netcfg" type="application">
        <propval name="active_ncp" type="astring" value="DefaultFixed"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

詳細な構成変換情報の表示

sysidcfg ファイル変換またはシステム構成プロファイル検証に関する詳細な情報を表示するには、js2ai コマンドに -v オプションを指定します。-v オプションを指定すると、処理ステップが表示され、エラーが発生しなかった場合にエラーレポートが省略されるのではなく、エラーレポートにゼロが表示されます。

```
# js2ai -sv
Processing: sysidcfg
Performing conversion on: sysidcfg
Generating SC Profile
Validating sc_profile.xml

                                Process  Unsupported  Conversion  Validation
Name                  Warnings  Errors   Items        Errors      Errors
-------------------   --------  -------  -----------  ----------  ----------
sysidcfg                     0        0            0           0           0
Successfully completed conversion
```

出力システム構成プロファイルの検証

生成されたシステム構成プロファイルを検証するには、-V オプションを使用します。

```
# js2ai -v -V ./sc_profile.xml
Validating sc_profile.xml

                                Process  Unsupported  Conversion  Validation
Name                  Warnings  Errors   Items        Errors      Errors
-------------------   --------  -------  -----------  ----------  ----------
sc_profile                   -        -            -           -           0
Successfully completed conversion
```

第 4 章

Oracle Solaris 11 サーバー上の JumpStart を使用した Oracle Solaris 10 のインストール

JumpStart を使用して、ネットワーク上の SPARC および x86 プラットフォーム上に Oracle Solaris 10 オペレーティングシステムをインストールすることを十分に理解しているユーザーは、JumpStart は Oracle Solaris 11 OS でなく Oracle Solaris 10 OS をインストールするためにのみ使用できるということをおそらくご存じでしょう。ただし、JumpStart インストールサーバーは Oracle Solaris 11 システムとすることができます。

Oracle Solaris 11 サーバーは次の 2 種類の作業を実行できます。

- 自動インストーラを使用して Oracle Solaris 11 OS のインストールを行います。詳細は、『Oracle Solaris 11 システムのインストール』のパート III「インストールサーバーを使用したインストール」を参照してください。
- JumpStart を使用して Oracle Solaris 10 OS のインストールを行います。この章では、JumpStart インストールサーバーを Oracle Solaris 11 システム上に設定する方法について説明します。JumpStart の詳細は、『Oracle Solaris 10 9/10 インストールガイド (カスタム JumpStart/上級編)』を参照してください。

Oracle Solaris 10 JumpStart サーバーとしての Oracle Solaris 11 システムの設定

次のステップでは、Oracle Solaris 11 JumpStart インストールサーバーを作成して Oracle Solaris 10 OS をクライアントシステムにインストールする方法について説明します。

1. JumpStart インストールサーバーになるマシンに Oracle Solaris 11 OS をインストールします。
2. JumpStart インストールサーバーになる Oracle Solaris 11 マシンに静的 IP アドレスを設定します。

   静的 IP アドレスの設定方法については、『Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化』の「IP インタフェースを構成する方法」および `ipadm(1M)` のマニュアルページの手順を参照してください。
3. `system/boot/network` パッケージを `solaris` パブリッシャーからインストールします。

```
# pkg publisher
PUBLISHER                TYPE     STATUS   URI
solaris                  origin   online   http://pkg.oracle.com/solaris11/release/
# pkg install pkg:/system/boot/network
           Packages to install:  1
     Variants/Facets to change:  3
       Create boot environment: No
Create backup boot environment: No
            Services to change:  1

DOWNLOAD                                  PKGS       FILES    XFER (MB)
Completed                                  1/1       13/13      0.0/0.0

PHASE                                        ACTIONS
Install Phase                                  34/34

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

PHASE                                          ITEMS
Reading Existing Index                           8/8
Indexing Packages                                1/1
```

4. 最新の Oracle Solaris 10 DVD イメージをダウンロードします。

   このイメージは、インストールイメージおよび JumpStart ツールのソースです。JumpStart は Oracle Solaris 11 OS でなく Oracle Solaris 10 OS をインストールするためだけに使用できることを思い出してください。
5. Oracle Solaris 10 DVD イメージと `setup_install_server(1M)` コマンドを使用して、JumpStart インストールサーバーを設定します。

   次の例では SPARC 用の Oracle Solaris 10 9/10 DVD イメージを使用しています。

```
# /media/SOL_10_0910_SPARC/Solaris_10/Tools/setup_install_server /export/s10u9_sparc
Verifying target directory...
Calculating the required disk space for the Solaris_10 product
Calculating space required for the installation boot image
Copying the CD image to disk...
Copying Install Boot Image hierarchy...
Copying /boot netboot hierarchy...
Install Server setup complete
```

6. jumpstart_sample ファイルを DVD イメージから JumpStart プロファイルの場所にコピーします。

```
# cd /media/SOL_10_0910_SPARC/Solaris_10/Misc/jumpstart_sample
# mkdir -p /export/profiles/s10profile
# cp -pr * /export/profiles/s10profile
```

7. 50 ページの「rm_install_client スクリプトの問題」 に記載されている回避策を実行します。
8. 51 ページの「check スクリプトの問題」 に記載されている回避策を実行します。
9. check(1M) コマンドに -p *s10-image* オプションを付けて実行し、JumpStart プロファイルを検証します。

   次の例では、簡潔に示すために多くの出力行が省略されています。-p オプションはすべての Oracle Solaris 10 Update リリースで機能するとは限らないことに注意してください。Oracle Solaris 10 Update 7 以降を使用していることを確認してください。(参照 CR 6728067。)

```
# cd /export/profile/s10profile
./check -p /export/s10u9_sparc
45 blocks
Validating rules...
Validating profile host_class...
Validating profile zfsrootsimple...
Validating profile net924_sun4c...
Validating profile upgrade...
Validating profile x86-class...
Validating profile any_machine...
The custom JumpStart configuration is ok.
```

10. udp6 が使用可能でオンラインであることを確認します。

```
# svcs udp6
STATE          STIME   FMRI
disabled       8:36:55 svc:/network/tftp/udp6:default
# svcadm enable network/tftp/udp6
# svcs udp6
STATE          STIME   FMRI
online         8:38:28 svc:/network/tftp/udp6:default
```

11. /tftpboot ディレクトリを /etc/netboot ディレクトリとリンクします。/etc/netboot ディレクトリが存在しない場合は作成します。

```
# ls /etc/netboot
/etc/netboot: No such file or directory
# mkdir -m 755 /etc/netboot
# ln -s /etc/netboot /tftpboot
# ls -l /tftpboot
lrwxrwxrwx   1 root     root          12 Sep 14 8:46:51 /tftpboot -> /etc/netboot
```

JumpStart は /tftpboot をネットワークブート用に使用します。AI は /etc/netboot をネットワークブート用に使用します。この Oracle Solaris 11 サーバーが JumpStart インストールサーバーと AI インストールサーバーの両方の役割を果たすことができる場合は、/tftpboot を /etc/netboot に対するシンボリックリンクにすることで 2 つのツールが共存できるようにする必要があります。

ヒント – このサーバーを JumpStart インストールサーバーまたは AI インストールサーバーのいずれかとしてのみ使用することを計画している場合、このステップは不要です。ただし、後で計画を変更する場合にネットブートエラーの失敗を回避するためには、いずれにしてもこのステップを実行する必要があります。

---

12. add_install_client(1M) を実行してクライアントをインストールします。

```
/export/s10u9_sparc/Solaris_10/Tools/add_install_client -e 8:0:20:fd:f2:18 \
-c line2-x4100:/export/profiles/s10profile \
-p line2-x4100:/export/profiles/s10profile line2-t1 sun4u
Adding Ethernet number for line2-t1.sfbay.sun.com to /etc/ethers
making /tftpboot
enabling network/rarp service
enabling network/rpc/bootparams service
updating /etc/bootparams
copying boot file to /tftpboot/inetboot.SUN4U.Solaris_10-1
```

既知の問題の回避策

このセクションでは、特定の問題を回避するための手順を示します。

rm_install_client スクリプトの問題

CR 6646677: The rm_install_client script does not work with Solaris 10 dig based nslookup

この問題を回避するには、Oracle Solaris 10 ネットイメージの rm_install_client スクリプトに次の変更を行います。たとえば、Oracle Solaris 10 ネットイメージを /export/s10u9/dvds/latest に設定した場合、/export/s10u9/dvds/latest/Solaris_10/Tools/rm_install_client の rm_install_client スクリプトに次の変更を行います。

次に示す行 168 を変更します。

```
ANS=‘nslookup ${K} 2>&1‘
```

次のようになりました。

```
ANS=‘nslookup ${K} | /bin/sed ’/^;;/d’ 2>&1‘
```

次に示す行 273 を変更します。

```
ANS=‘echo $ANS | sed -e ’s/#.*$//’‘
```

次のようになりました。

```
ANS=‘echo $ANS | sed -e ’s/ #.*$//’‘
```

check スクリプトの問題

```
CR 6838095: opensolaris jumpstart server doesn’t run the check script correctly
```

この問題を回避するには、JumpStart プロファイルディレクトリ `/export/profile/s10profile` の `check` スクリプトを変更して `/usr/sunos/bin/sh` を使用するようにします。`check` スクリプトの先頭行を次の行に置換します。

```
#!/usr/sunos/bin/sh
```