

Oracle® Solaris 11.2 ソフトウェアの追加と更新

ORACLE®

Part No: E53746
2014 年 7 月

Copyright © 2007, 2014, 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場合、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したこと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。

目次

このドキュメントの使用方法	9
1 Image Packaging System の概要	11
Image Packaging System	11
IPS の概念	12
IPS パッケージ	12
障害管理リソース識別子	14
パブリッシャー、リポジトリ、およびパッケージアーカイブ	16
リポジトリの起点とミラー	17
イメージとブート環境	17
パッケージのファセットとバリエーション	18
インストール権限	19
2 ソフトウェアパッケージに関する情報の取得	21
パッケージのインストール状態情報の表示	21
インストール済みのパッケージ	22
インストール可能なパッケージ	22
最新のパッケージ	22
更新が利用可能なパッケージ	23
使用可能なすべてのパッケージ	23
パッケージの名前変更と廃止	24
特定バージョンに凍結されているパッケージ	25
パッケージの説明またはライセンスの表示	25
パッケージの説明、サイズ、完全な FMRI の表示	26
パッケージライセンスの表示	27
パッケージマニフェストからの情報の表示	27
パッケージによってインストールされるファイルの一覧表示	27
パッケージによってインストールされるファイルの属性の表示	28
他のファイルシステムオブジェクトおよび属性の表示	29
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	29

ライセンス要件の表示	30
パッケージの検索	30
pkg search コマンドと pkg contents コマンドの比較	31
検索クエリーの指定	31
指定されたファイルを提供するパッケージの識別	33
指定した SMF サービスを提供するパッケージの識別	33
分類またはカテゴリ別のパッケージの一覧表示	34
依存パッケージの表示	35
グループパッケージ内のすべてのパッケージの一覧表示	35
3 ソフトウェアパッケージのインストールおよび更新	37
操作のプレビュー	38
パッケージのインストールおよび更新	39
一般的なインストールオプション	40
新しいパッケージのインストール	44
新しいブート環境へのパッケージのインストール	47
パッケージの拒否	48
パッケージの更新	49
パッケージのダウングレード	50
インストール済みパッケージの問題の修正	50
pkg fix コマンドと pkg revert コマンドの比較	50
パッケージの検証と検証エラーの修正	51
ファイルの復元	53
パッケージのアンインストール	55
イメージの再インストール	56
非大域ゾーンの操作	57
大域ゾーンと非大域ゾーンの関係	57
システムリポジトリおよびプロキシサービス	59
複数の非大域ゾーンの同時更新	61
4 Oracle Solaris イメージの更新またはアップグレード	63
イメージの更新の概要	63
イメージ更新のベストプラクティス	64
使用可能なバージョンのチェック	64
更新操作のプレビュー	65
新しいブート環境の指定	66
インストールするバージョンの指定	67
更新前のバージョン制約の指定	67
カスタム Incorporation のインストール	68

カスタム Incorporation パッケージの作成	68
アップグレード制御パッケージのインストール	71
アップグレード制御パッケージの更新	72
イメージのアップグレード	73
イメージのダウングレード	75
5 インストールされるイメージの構成	77
パブリッシャーの構成	77
パブリッシャー情報の表示	78
パッケージパブリッシャーの追加、変更、削除	79
プロキシの指定	84
オプションのコンポーネントのインストールの制御	86
バリエーションおよびファセット値がパッケージのインストールに与える影響	86
バリエーション値およびファセット値の例	87
バリエーション値の表示と変更	88
ファセット値の表示と変更	89
指定したバージョンへのパッケージのロック	90
incorporation によって指定されたバージョン制約の緩和	91
デフォルトのアプリケーション実装の指定	94
メディエーション内の参加者の識別	94
優先アプリケーションの変更	96
グループパッケージに含まれる一部のパッケージのインストールの回避	97
イメージとパブリッシャーのプロパティの構成	100
ブート環境ポリシーイメージのプロパティ	100
パッケージの署名のプロパティ	102
追加のイメージのプロパティ	104
イメージのプロパティの設定	106
イメージの作成	107
操作履歴の表示	108
A パッケージのインストールおよび更新のトラブルシューティング	111
トラブルシューティングの初期手順	111
pkg:/entire のインストール済みバージョンの確認	112
構成済みのパブリッシャーの起点についての内容の確認	112
インストールの再試行	115
パブリッシャーまたはリポジトリにアクセスできない	116
パッケージリポジトリにアクセスできない	116
SSL 証明書の問題	117
場所が見つかりません	118

サービスが使用不可	120
使用可能な更新がない	121
パッケージをインストールできない	121
制約を満たすことができない	122
incorporation によって制約されたパッケージの更新	123
適切な依存関係が見つからないときの Incorporation の更新	127
インストール済みの依存関係が許容されないときの Incorporation の更新	130
必要なパッケージが見つからない	130
必要なパッケージが拒否された	131
パッケージが期待どおりに更新されない	132
同期リンクされたパッケージをインストールできない	133
非大域ゾーンをインストールできない	134
イメージを修正できない	135
ファイルが回収された	136
格納されるイメージメタデータの最小化	136
パッケージのインストールパフォーマンスの増大	137
B IPS のグラフィカルユーザーインターフェース	139
パッケージマネージャーの使用	139
パッケージマネージャーのコマンド行のオプション	140
Web インストールの使用	140
更新マネージャーの使用	143
更新マネージャーのコマンド行のオプション	145
索引	147

例目次

例 5-1	新しいパブリッシャーの指定	80
例 5-2	パブリッシャー構成のインポート	80
例 5-3	パブリッシャーの鍵と証明書の指定	82
例 5-4	パブリッシャーの鍵と証明書の取り消し	83
例 5-5	回避リストへのパッケージの追加と回避リストからのパッケージの削除	98
例 A-1	Java Runtime Environment のロック解除と更新	123
例 A-2	依存関係がロック解除されていて別個に更新される場合の pkg:/entire の更新	127

このドキュメントの使用方法

- **概要** – Oracle Solaris Image Packaging System (IPS) 機能のソフトウェアインストール機能について説明します。IPS コマンドによって、ソフトウェアパッケージの一覧表示および検索、ソフトウェアのインストールおよび削除、ならびに新しい Oracle Solaris オペレーティングシステムリリースへのアップグレードが可能になります。
- **対象読者** – ソフトウェアのインストールと管理、およびシステムイメージの管理を行うシステム管理者。
- **前提知識** – Oracle Solaris システムの管理経験。

製品ドキュメントライブラリ

この製品に関する最新情報および既知の問題については、ドキュメントライブラリ (<http://www.oracle.com/pls/topic/lookup?ctx=E56342>) に記載されています。

Oracle サポートへのアクセス

Oracle ユーザーは My Oracle Support から電子サポートにアクセスできます。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。

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。

◆◆◆ 第 1 章

Image Packaging System の概要

Oracle Solaris Image Packaging System (IPS) は、ソフトウェアパッケージの一覧表示および検索、ソフトウェアのインストールおよび削除、ならびに新しい Oracle Solaris オペレーティングシステムリリースへのアップグレードを可能にするフレームワークです。IPS コマンドを使用して、インストールできるパッケージやインストールできるパッケージのバージョンを制限できます。

Image Packaging System

Oracle Solaris 11 ソフトウェアは、IPS パッケージで配布されます。IPS パッケージは、IPS パブリッシャーが提供する IPS パッケージリポジトリに格納されます。IPS パッケージは、Oracle Solaris 11 のイメージにインストールされます。IPS のコマンド行インタフェースから実行できる機能の一部は、パッケージマネージャーのグラフィカルユーザーインタフェースを使用して実行できます。

IPS ツールは次に示す機能を提供します。パブリッシャーやリポジトリなどの用語の定義については、[12 ページの「IPS の概念」](#)を参照してください。

- ソフトウェアパッケージを一覧表示、検索、インストール、インストールの制限、更新、および削除します。
- パッケージパブリッシャーを一覧表示、追加、および削除します。検索の優先順やステイックネスなどの、パブリッシャーの属性を変更します。署名ポリシーなどのパブリッシャーのプロパティを設定します。
- 新しいオペレーティングシステムリリースにイメージをアップグレードします。
- 既存の IPS パッケージリポジトリのコピーを作成します。新しいパッケージリポジトリを作成します。『[Oracle Solaris 11.2 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。
- パッケージを作成および公開します。『[Oracle Solaris 11.2 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](#)』を参照してください。

- ブート環境およびその他のイメージを作成します。

IPS を実行するには、Oracle Solaris 11 OS を実行している必要があります。Oracle Solaris 11 OS をインストールするには、『[Oracle Solaris 11.2 システムのインストール](#)』を参照してください。

IPS の概念

このセクションでは、IPS に関連する用語および概念を定義します。

IPS パッケージ

IPS パッケージはマニフェストというテキストファイルで定義します。パッケージマニフェストには、鍵/値のペアとおそらくデータペイロードの定義された形式でパッケージアクションが記述されます。パッケージアクションには、ファイル、ディレクトリ、リンク、ドライバ、依存関係、グループ、ユーザー、ライセンス情報が含まれます。パッケージアクションは、パッケージのインストール可能なオブジェクトを表します。set アクションと呼ばれるアクションは、分類、サマリー、説明などのパッケージメタデータを定義します。

パッケージアクションおよびアクションキーを指定して、パッケージを検索できます。パッケージアクションの説明については、『[Oracle Solaris 11.2 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「[パッケージの内容: アクション](#)」または [pkg\(5\)](#) マニュアルページを参照してください。

incorporation パッケージおよびグループパッケージは、ファイルなどの内容を提供しません。incorporation パッケージおよびグループパッケージは、関連するパッケージのセットのインストールに役立つ依存関係を指定します。

incorporation パッケージ

incorporation パッケージは、インストール可能なほかのパッケージのバージョンを指定しません。*incorporation* パッケージをインストールしても、ほかのパッケージはインストールされません。*incorporation* パッケージにより、その *incorporation* パッケージの *incorporate* 依存関係パッケージをインストールした場合に、規定されたバージョンの依存パッケージのみがイン

ストールできるようになります。たとえば、インストールされた incorporation パッケージ内で incorporate 依存関係として指定されたパッケージがバージョン値 1.4.3 を持つ場合、1.4.3 未満か 1.4.4 以上のバージョン値を持つそのパッケージのバージョンをインストールすることはできません。たとえば、バージョン値 1.4.3.7 を持つパッケージのバージョンはインストールできません。

incorporation パッケージは通常、互換性を持つパッケージバージョンスペース内にサーフェスを定義するために、多くの incorporate 依存関係を指定します。そのような incorporate 依存関係セットを含むパッケージは、通常 *incorporation* と呼ばれます。incorporation は通常、同時に作成されるソフトウェアパッケージのセットを定義するために使用され、個別にバージョン管理されません。incorporate 依存関係は、ソフトウェアの互換性のあるバージョンがまとめてインストールされるようにするために Oracle Solaris でよく使用されます。

incorporation パッケージ内で incorporate 依存関係として指定されているパッケージ自体が、incorporation パッケージの場合もあります。このように、incorporation パッケージのマニフェスト内でパッケージが指定されていない場合でも、多くのパッケージが incorporation パッケージによって影響を受ける可能性があります。そのインストールが incorporation パッケージによって影響を受けるパッケージは、その incorporation パッケージによって制約されます。A-incorporation によって incorporation 対象となっている incorporation パッケージ B-incorporation を更新すると、A-incorporation と、A-incorporation に制約されているほかのすべてのパッケージも更新されます。

incorporation は、使用中のサポート可能なイメージを容易に維持できるようにするために、制約されたパッケージを同期的に強制アップグレードします。一般的に、incorporation パッケージによって制約されているパッケージをインストールしたり更新したりしてはいけません。代わりに、incorporation パッケージを更新する必要があります。制約されているパッケージはアンインストールできますが、制約されているパッケージをインストールまたは更新する場合、バージョンが制約されます。関連情報については、[91 ページの「incorporation によって指定されたバージョン制約の緩和」](#)を参照してください。

pkg://solaris/entire パッケージは、ほかの多くの incorporation パッケージ上の incorporate 依存関係を指定して、イメージにインストールされているほとんどのシステムソフトウェアのバージョンを制約する特殊な incorporation パッケージです。



注意 - entire という名前のパッケージを削除しないでください。entire パッケージは、システムパッケージのバージョンを制約して、結果として得られるパッケージセットがサポート可能なイメージになるようにします。適切なシステム更新と正しいパッケージ選択は、この incorporation によって変わります。entire 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。

グループパッケージ

グループパッケージは、機能やツールを構成する一連のパッケージを指定します。グループパッケージをインストールすると、そのグループパッケージ内のすべての group 依存関係パッケージがインストールされます。グループパッケージ内で group 依存関係としてに指定されたパッケージは、パッケージのバージョンを指定しません。グループパッケージは、コンテンツ管理ツールであり、バージョン管理ツールではありません。

グループパッケージは、それらの group 依存関係に指定されているパッケージを提供しますが、それらのパッケージがすでにインストールされているか、回避リストに含まれている場合は除きます。回避リストについては、[97 ページの「グループパッケージに含まれる一部のパッケージのインストールの回避」](#)を参照してください。

たとえば、group/feature/storage-server パッケージは、ストレージに関連するドライバ、サービス、ファイルシステム、I/O コンポーネント、ライブラリ、およびユーティリティを提供します (これらがまだインストールされていない場合)。group/system/solaris-minimal-server パッケージは、サポートされる Oracle Solaris 環境に最低限必要なパッケージのセットを提供します。グループパッケージで提供されているすべてのパッケージを一覧表示する方法の例については、[29 ページの「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」](#)を参照してください。

グループパッケージをアンインストールしても、group 依存関係内に指定されているすべてのパッケージが必ずしもアンインストールされるわけではありません。グループパッケージをアンインストールするとき、ほかのソフトウェアによって必要とされているパッケージで依然としてインストールされているものは、アンインストールされません。

障害管理リソース識別子

それぞれのパッケージは障害管理リソース識別子 (FMRI) によって表されます。パッケージの完全な FMRI は、次の形式のスキーム、パブリッシャー、パッケージ名、およびバージョン文字列で構成されます。

```
scheme://publisher/name@version:dateTtimeZ
```

スキーム、パブリッシャー、およびバージョン文字列はオプションです。IPS コマンドオペランドでは、パッケージを一意的に識別するパッケージ名の最小部分を使用することができ、? および * の文字を、1 つ以上のパッケージと一致させるための glob(3C) スタイルのワイルドカードとして使用できます。

次に示すパッケージの FMRI の例は、suri ストレージライブラリについてのものです。

```
pkg://solaris/system/library/storage/suri@0.5.11,5.11-0.175.2.0.0.34.0:20140303T145535Z
```

スキーム

pkg

パブリッシャー

solaris

パブリッシャーを指定する場合、パブリッシャー名の前に `pkg://` または `//` を付ける必要があります。

パッケージ名

system/library/storage/suri

パッケージ名は、スラッシュ (/) 文字で区切られた任意の数のコンポーネントから成る階層形式です。IPS コマンドでは、コマンドで使用するパッケージ名によってそのパッケージが一意に識別される場合、パッケージ名の前のコンポーネントは省略できます。完全なパッケージ名を指定するがパブリッシャーは省略する場合、完全なパッケージ名の前に、`pkg://` または `//` ではなく、`pkg:/` または `/` を付けることができます。短縮したパッケージ名を指定する場合は、パッケージ名の左側にほかの文字を使用しないでください。

バージョン

パッケージのバージョンには 4 つの部分があります。

コンポーネントバージョン: 0.5.11

オペレーティングシステムに緊密に結合されたコンポーネントの場合、コンポーネントバージョンは通常、そのバージョンのオペレーティングシステムでの `uname -r` の値を含みます。独自の開発ライフサイクルを持つコンポーネントの場合、コンポーネントバージョンはドットで区切られたリリース番号 (2.4.10 など) です。

ビルドバージョン: 5.11

ビルドバージョンはコンマ (,) の後に続ける必要があります。ビルドバージョンは、パッケージの内容が構築されたオペレーティングシステムのバージョンを指定します。

ブランチバージョン: 0.175.2.0.0.34.0

ブランチバージョンはハイフン (-) の後に続ける必要があります。ブランチバージョンはベンダー固有の情報を提供します。

Oracle Solaris パッケージでは、パッケージ FMRI のバージョン文字列のブランチバージョン部分に次の情報が示されます。

メジャーリリース番号: 0.175

メジャーまたはマーケティング開発リリースのビルド番号。この例では、0.175 は Oracle Solaris 11 を示します。

更新リリース番号: 2

この Oracle Solaris リリースの更新リリース番号。更新値は、Oracle Solaris リリースの FCS (First Customer Shipment) では 0、そのリリースの最初の更新では 1、そのリリースの 2 回目の更新では 2 というようになります。この例では、1 は Oracle Solaris 11.1 を示します。

SRU 番号: 0

この更新リリースのサポート・リポジトリの更新 (SRU) 番号。SRU にはバグ修正のみが入っており、新しい機能は含まれていません。Oracle サポート・リポジトリはサポート契約下のシステムでのみ使用できます。

予約済み: 0

このフィールドは、Oracle Solaris パッケージで現在使用されていません。

SRU ビルド番号: 34

SRU のビルド番号、またはメジャーリリースの更新番号。

ナイトリービルド番号: 0

個々のナイトリービルドのビルド番号。

タイムスタンプ: 20140303T145535Z

タイムスタンプはコロン (:) の後に続ける必要があります。タイムスタンプは、ISO-8601 基本形式 (YYYYMMDDTHHMMSSz) のパッケージが発行された日時です。

パブリッシャー、リポジトリ、およびパッケージアーカイブ

パブリッシャーは、1 つ以上のパッケージを提供する人または組織を示します。パブリッシャーは、パッケージリポジトリまたはパッケージアーカイブを使用してパッケージを配布できます。パブリッシャーは、好きな検索順序で構成できます。パッケージインストールコマンドを指定し、パッケージ仕様にパブリッシャーの名前が含まれない場合、そのパッケージに対して、検索順序の先頭のパブリッシャーが検索されます。指定されたパッケージ FMRI パターンに一致するものが見つからない場合は、検索順序の 2 番目のパブリッシャーが検索されるというように、パッケージが見つかるか、すべてのパブリッシャーが検索されるまで繰り返されます。

リポジトリは、パッケージが公開される場所であり、またそれらのパッケージが取得される場所です。場所は URI (Universal Resource Identifier) によって指定されます。カタログは、リポジトリ内のすべてのパッケージのリストです。

パッケージアーカイブは、パブリッシャーの情報と、そのパブリッシャーによって提供された 1 つ以上のパッケージを含むファイルです。

リポジトリの起点とミラー

起点は、パッケージのメタデータ (カタログ、マニフェスト、検索インデックスなど) とパッケージの内容 (ファイル) の両方を含むパッケージリポジトリです。イメージ内の特定のパブリッシャーに対して複数の起点が構成されている場合、IPS クライアントは、パッケージデータの取得元として最適な起点を選択しようとします。

ミラーは、パッケージの内容のみを含むパッケージリポジトリです。ミラーリポジトリからパッケージをインストールおよび更新するクライアントは、起点リポジトリからメタデータをダウンロードする必要があります。IPS クライアントがパッケージのコンテンツをミラーからダウンロードする場合でも、クライアントはパブリッシャーのカタログを取得するために起点にアクセスします。パブリッシャーに対してミラーが構成されている場合、IPS クライアントは、パッケージの内容の取得にミラーを優先します。イメージ内の特定のパブリッシャーに対して複数のミラーが構成されている場合、IPS クライアントは、パッケージの内容の取得元として最適なミラーを選択しようとします。すべてのミラーにアクセスできない、必要な内容がない、または遅くなる場合、IPS クライアントは起点から内容を取得します。詳細は、[pkg\(5\)](#) マニュアルページの「パブリッシャーおよびリポジトリに関するセクション」を参照してください。

注記 - ミラーリポジトリとして指定されたリポジトリの内容とメタデータの両方が完全であっても、同じパッケージの同じバージョンが、同じパブリッシャーの起点リポジトリ内にも存在しないかぎり、そのミラーリポジトリの内容にアクセスすることはできません。

イメージとブート環境

イメージは、IPS パッケージをインストールでき、その他の IPS 操作を実行できる場所です。

ブート環境 (BE) は、イメージのブート可能なインスタンスです。物理または仮想システム上に複数の BE を維持することができ、異なるオペレーティングシステムバージョンなどの異なるソフトウェアバージョンを各 BE にインストールできます。システムをブートするとき、システム上の任意の BE にブートすることを選択できます。パッケージ操作の結果として、新しい BE が自動的に作成されることがあります。新しい BE が自動的に作成されるかどうかは、[100 ページの「ブート環境ポリシーイメージのプロパティ」](#)で説明するように、イメージポリシーに依存します。また、[40 ページの「ブート環境オプション」](#)に記載されているオプションを指定することによって、新しい BE を明示的に作成することもできます。新規 BE を作成するための `beadm` コマンドの使用方法については、『[Oracle Solaris 11.2 ブート環境の作成と管理](#)』を参照してください。

パッケージは、BE の一部であるファイルシステムにのみインストールできます。たとえば、デフォルトの Oracle Solaris 11 インストールでは、`rpool/ROOT/BEname/` の下のデータセットのみがパッケージ操作にサポートされます。

もう 1 つのイメージの例として、Oracle Solaris ゾーンがあります。非大域ゾーンは、大域ゾーンと呼ばれる Oracle Solaris オペレーティングシステムのインスタンス内で作成される、仮想化されたオペレーティングシステム環境です。大域ゾーンは親イメージで、その大域ゾーン内の非大域ゾーンはその大域ゾーンの子イメージです。IPS コマンド出力では、非大域ゾーンは親の大域ゾーンイメージにリンクされていることから、しばしばリンクされたイメージと呼ばれます。

57 ページの「非大域ゾーンの操作」に記載されているように、大域ゾーンで実行される IPS コマンドは、非大域ゾーンに影響することがあります。大域ゾーンで実行される IPS コマンドは、カーネルゾーン (`solaris-kz` ブランドゾーン) または Oracle Solaris 10 ゾーン (`solaris10 branded zones`) に影響しません。このガイドで、「非大域ゾーン」とは `solaris` ブランドの Oracle Solaris 11 非大域ゾーンを意味します。ゾーンについては、『[Oracle Solaris ゾーン の紹介](#)』を参照してください。

パッケージのファセットとバリエーション

ソフトウェアには、オプションのコンポーネントや、相互に排他的なコンポーネントが含まれることがあります。オプションのコンポーネントの例には、ロケールやドキュメントがあります。相互に排他的なコンポーネントの例には、SPARC バイナリと x86 バイナリや、デバッグバイナリと非デバッグバイナリなどがあります。IPS では、オプションのコンポーネントをファセットと呼び、相互に排他的なコンポーネントをバリエーションと呼びます。

ファセットとバリエーションは、イメージに設定される特殊プロパティです。ファセットとバリエーションは、パッケージマニフェスト内のアクションに設定されるタグでもあります。アクションのファセットおよびバリエーションタグの値とイメージに設定されたファセットおよびバリエーションの値の比較により、そのパッケージアクションがインストール可能かどうかを判別します。たとえば、イメージ内で特定のロケールファセットを `false` に設定した場合、そのファセットを指定しているファイルアクションはどれもインストールされず、そのファセットを指定している現在インストール済みのファイルアクションはアンインストールされます。

ほとんどのバリエーションには任意の値を設定できます。アクションに設定されるファセットタグは、値 `true` のみを持つことができます。イメージに設定されるファセットプロパティは、値 `true` または `false` のみを持つことができます。

次のアルゴリズムは、イメージに設定されたファセットとバリエントが、特定のアクションがインストールされるかどうかに影響するしくみを示しています。

- ファセットまたはバリエントタグのないアクションは常にインストールされます。
- ファセットタグのあるアクションは、イメージ上のタグに一致するすべてのファセットまたはファセットパターンが `false` に設定されていないかぎり、インストールされます。いずれかのファセットが `true` に設定されているか、明示的に設定されていない (`true` はデフォルト) 場合、アクションがインストールされます。
- バリエントタグのあるアクションは、すべてのバリエントタグの値がイメージに設定されている値と同じ場合にのみインストールされます。
- ファセットタグとバリエントタグの両方があるアクションは、ファセットとバリエントの両方でアクションのインストールが許可されている場合にインストールされます。

イメージに設定されているファセットとバリエントの値を表示または変更するには、[86 ページの「オプションのコンポーネントのインストールの制御」](#)を参照してください。

インストール権限

[第2章「ソフトウェアパッケージに関する情報の取得」](#)で説明しているコマンドには、特別な権限を使用する必要はありません。IPS パッケージのインストールや、オペレーティングシステムのアップグレード、パブリッシャーおよびイメージプロパティの構成などのタスクには、さらに強い権限が必要になります。必要な権限を取得するには、次のいずれかの方法を使用します。必要なプロファイルまたは役割を判別する方法など、プロファイルおよび役割に関する情報の詳細は、『[Oracle Solaris 11.2 でのユーザーとプロセスのセキュリティ保護](#)』を参照してください。

権利プロファイル

`profiles` コマンドを使用して、自分に割り当てられている権利プロファイルを一覧表示します。たとえば、ソフトウェアインストールに関連する権利プロファイルがある場合、`pkg` および `beadm` コマンドを使用して、パッケージのインストールと更新、およびブート環境の管理を行うことができます。

役割

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

sudo コマンド

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

◆◆◆ 第 2 章

ソフトウェアパッケージに関する情報の取得

この章では、パッケージに関する次の種類の情報を得るためのコマンドについて説明します。

- パッケージがインストールされているか、またはインストールや更新ができるかどうか
- パッケージの説明、サイズ、およびバージョン
- グループパッケージに含まれるパッケージ
- 指定したファイルを提供するパッケージ
- 指定した SMF サービスを提供するパッケージ
- 指定したパッケージに依存するパッケージ
- 特定のカテゴリに含まれるパッケージ

構成済みのパブリッシャーに対してパッケージリポジトリの内容が変更された可能性がある場合、セッションの開始時に使用可能なパッケージの一覧を更新して、最新情報を受け取っていることを確認します。パッケージの一覧を更新するには、`pkg refresh` コマンドを実行します。たとえば、リポジトリが新規パッケージによって更新された場合、`pkg refresh` を実行するまでは、これらの新規パッケージが使用可能なパッケージの一覧に表示されない場合があります。

この章で説明されているコマンドのすべてのオプションの完全なリストについては、[pkg\(1\)](#) のマニュアルページを参照してください。

パッケージのインストール状態情報の表示

`pkg list` コマンドは、現在のイメージにパッケージがインストールされているかどうか、および更新が使用可能かどうかを示します。オプションやオペランドを指定しない場合、`pkg list` コマンドによって、現在のイメージにインストールされているすべてのパッケージが一覧表示されます。結果を絞り込むには、パッケージ名を 1 つ以上指定します。パッケージ名にはワイルドカードを使用できます。引数が直接 `pkg` に渡され、シェルで展開されないようにするために、ワイルドカードは引用符で囲みます。

インストール済みのパッケージ

次の例に示すように、`pkg list` コマンドは、一致するパッケージごとに 1 行の情報を表示します。I 列内の「i」は、これらのパッケージがこのイメージにインストールされていることを示します。

```
$ pkg list '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
isvtoolkit (isvpub)      1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.2.0.0.34.0 i--
```

丸括弧で囲まれたパブリッシャー名は、`isvpub` のパブリッシャーが、このイメージのパブリッシャーの検索順序で先頭のパブリッシャーではないことを示します。このイメージにインストールされた `dtrace-toolkit` パッケージは、検索順序の先頭のパブリッシャーによって発行されています。

インストール可能なパッケージ

このイメージにインストールされているパッケージと、インストールされていないが、インストール可能なパッケージの最新バージョンを一覧表示するには、`-a` オプションを使用します。

```
$ pkg list -a '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
image/nvidia/cg-toolkit  3.0.15-0.175.2.0.0.17.0 ---
isvtoolkit (isvpub)      1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.2.0.0.34.0 i--
```

この出力は、`image/nvidia/cg-toolkit` パッケージが使用可能であり、このイメージにインストールできることを示しています。

最新のパッケージ

このイメージにインストールできないパッケージも含めて、一致するすべてのパッケージの最新バージョンを一覧表示するには、`-n` オプションを使用します。

```
$ pkg list -n '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
developer/dtrace/toolkit 0.99-0.173.0.0.0.1.0  --r
image/nvidia/cg-toolkit  3.0.15-0.175.2.0.0.17.0 ---
isvtoolkit (isvpub)      1.0             i--
```

```
system/dtrace/dtrace-toolkit      0.99-0.175.2.0.0.35.0    ---
```

developer/dtrace/toolkit パッケージと system/dtrace/dtrace-toolkit パッケージは、このイメージにインストールできません。このことは、これらのパッケージが `-a` オプションでは一覧表示されなかったことからわかります。developer/dtrace/toolkit パッケージは名前が変更されています。詳細は、[24 ページの「パッケージの名前変更と廃止」](#)を参照してください。

このイメージに現在インストールされているバージョンよりも新しいバージョンの dtrace-toolkit パッケージが、構成済みのパブリッシャーに存在します。次の例の「Reason」行に示されているように、dtrace-toolkit パッケージが依存関係になっているほかのパッケージも更新した場合、新しいバージョンがインストール可能になることもあります。簡潔に示すために、次のコマンド出力ではタイムスタンプが省略されています。pkg update および pkg install コマンドについては、[第3章「ソフトウェアパッケージのインストールおよび更新」](#)で説明されています。

```
$ pkg update -nv dtrace-toolkit
No updates are available for this image.
$ pkg install -nv dtrace-toolkit@0.99-0.175.2.0.0.35
pkg install: No matching version of system/dtrace/dtrace-toolkit can be installed:
  Reject:  pkg://solaris/system/dtrace/dtrace-toolkit@0.99,5.11-0.175.2.0.0.35.0
  Reason:  This version is excluded by installed incorporation
pkg://solaris/consolidation/osnet/osnet-incorporation@0.5.11,5.11-0.175.2.0.0.34.0
```

更新が利用可能なパッケージ

`-u` オプションは、新しいバージョンが使用可能な、インストール済みの一致するすべてのパッケージを一覧表示します。[22 ページの「最新のパッケージ」](#)に例示されているように、より新しいバージョンがあるパッケージの数が、このイメージ内で更新可能なパッケージの数より多い場合があります。パッケージは、イメージの制約で許可されたバージョンにのみ更新できます。この制約は、インストール済みパッケージの依存関係およびパブリッシャーの構成によってイメージに課せられるものです。

```
$ pkg list -u '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
system/dtrace/dtrace-toolkit  0.99-0.175.2.0.0.34.0  i--
```

使用可能なすべてのパッケージ

このイメージにインストールできないパッケージも含めて、一致するすべてのパッケージの使用可能なすべてのバージョンを一覧表示するには、`-af` オプションを使用します。`-f` オプションは、-

a オプションを使用しないと使用できません。これらの結果を絞り込むために、バージョン文字列の一部を指定する場合があります。特殊なバージョン文字列 `@latest` を指定すると、`-n` オプションで表示されるのと同じ結果が表示されます。

```
$ pkg list -af '*toolkit@0.99-0.175.2'
$ pkg list -af '*toolkit@latest'
```

パッケージの名前変更と廃止

次の例で、O 列の「r」はパッケージの名前が変更されていることを示し、O 列の「o」はパッケージが廃止されたことを示します。

```
$ pkg list -n developer/dtrace/toolkit database/mysql-50 web/amp
NAME (PUBLISHER)          VERSION          IFO
database/mysql-50        5.0.91-0.171    --o
developer/dtrace/toolkit 0.99-0.173.0.0.0.1.0 --r
web/amp                  0.5.11-0.174.0.0.0.0.0 --r
```

これらのパッケージはどれもインストール可能ではありません。廃止されたパッケージをインストールしようとする、インストールは失敗し、このイメージに必要な更新が存在しないというメッセージが表示されます。名前が変更されたパッケージをインストールしようとする、システムは、名前が変更されたパッケージの名前が変更された場所に、パッケージのインストールを試行します。

`pkg info` コマンドを使用して、名前が変更されたパッケージの新しい名前を判別します。パッケージはインストールされていないため、`-r` オプションを使用して、構成済みのパッケージリポジトリを照会します。次の例に示されている「Renamed to」行を参照してください。

```
$ pkg info -r web/amp
Name: web/amp
Summary:
State: Not installed (Renamed)
Renamed to: group/feature/amp@0.5.11-0.174.0.0.0.0.0
            consolidation/ips/ips-incorporation
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.174.0.0.0.0.0
Packaging Date: September 21, 2011 07:15:02 PM
Size: 5.45 kB
FMRI: pkg://solaris/web/amp@0.5.11,5.11-0.174.0.0.0.0.0:20110921T191502Z
```

`web/amp` パッケージをインストールしようとする、`group/feature/amp` パッケージが代わりにインストールされます (まだインストールされておらず、このイメージにインストール可能な場合)。

次の例で、「Renamed to」に示すパッケージはすでにインストールされているため、パッケージシステムから、必要な更新が存在しないという報告が出されます。

```
$ pkg info -r developer/dtrace/toolkit
Name: developer/dtrace/toolkit
Summary:
State: Not installed (Renamed)
Renamed to: pkg:/system/dtrace/dtrace-toolkit@0.99,5.11-0.173.0.0.0.0
consolidation/osnet/osnet-incorporation
Publisher: solaris
Version: 0.99
Build Release: 5.11
Branch: 0.173.0.0.0.1.0
Packaging Date: August 26, 2011 02:55:51 PM
Size: 5.45 kB
FMRI: pkg://solaris/developer/dtrace/
toolkit@0.99,5.11-0.173.0.0.0.1.0:20110826T145551Z
$ pkg list dtrace-toolkit
NAME (PUBLISHER)          VERSION          IFO
system/dtrace/dtrace-toolkit  0.99-0.175.2.0.0.34.0  i--
$ pkg install developer/dtrace/toolkit
No updates necessary for this image.
```

特定バージョンに凍結されているパッケージ

F 列の「f」は、そのパッケージが凍結されていることを示します。パッケージが凍結されている場合は、凍結されたバージョンと一致するパッケージのみをインストールまたは更新できます。パッケージの凍結については、[90 ページの「指定したバージョンへのパッケージのロック」](#)を参照してください。

```
$ pkg list openssl
NAME (PUBLISHER)          VERSION          IFO
library/security/openssl  1.0.1.6-0.175.2.0.0.34.0  if-
```

パッケージの説明またはライセンスの表示

`pkg info` コマンドは、名前、説明、インストール状態、バージョン、パッケージ化の日付、パッケージのサイズ、完全な FMRI などのパッケージに関する情報を表示します。オプションやオペランドを指定しない場合、`pkg info` コマンドによって、現在のイメージにインストールされているすべてのパッケージに関する情報が表示されます。結果を絞り込むには、パッケージ名を 1 つ以上指定します。パッケージ名にはワイルドカードを使用できます。引数が直接 `pkg` に渡され、シェルで展開されないようにするために、ワイルドカードは引用符で囲みます。

info および list サブコマンドはいずれも、パッケージ名、パブリッシャー、およびバージョン情報を表示します。pkg list コマンドは、パッケージの更新が存在するかどうか、このイメージに更新をインストールできるかどうか、およびパッケージが廃止されたり、名前が変更されたり、凍結されたりしていないかどうかを表示します。pkg list コマンドは、パッケージのサマリーと完全な FMRI も表示できます。pkg info コマンドは、パッケージのサマリー、説明、カテゴリ、サイズを表示し、個別にライセンス情報を表示できます。

パッケージの説明、サイズ、完全な FMRI の表示

pkg list -s コマンドを使用して、パッケージのサマリーを表示できます。

```
$ pkg list -s entire
NAME (PUBLISHER)    SUMMARY
entire              Incorporation to lock all system packages to the same build
```

pkg list -v コマンドは、完全なパッケージ FMRI を一覧表示します。

```
$ pkg list -v entire
FMRI                                                    IFO
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z ---
```

pkg info コマンドは、詳細な情報を表示します。

```
$ pkg info entire
Name: entire
Summary: Incorporation to lock all system packages to the same build
Description: This package constrains system package versions to the same
             build. WARNING: Proper system update and correct package
             selection depend on the presence of this incorporation.
             Removing this package will result in an unsupported system.
Category: Meta Packages/Incorporations
State: Installed
Publisher: solaris
Version: 0.5.11
Branch: 0.175.2.0.0.34.0
Packaging Date: March  3, 2014 06:26:43 PM
Size: 5.46 kB
FMRI: pkg://solaris/entire@0.5.11-0.175.2.0.0.34.0:20140303T182643Z
```

24 ページの「[パッケージの名前変更と廃止](#)」に示すように、pkg info コマンドを使用して、名前が変更されたパッケージの新しい名前を見つけることができます。

パッケージライセンスの表示

一致するパッケージのライセンステキストを表示するには、`--license` オプションを使用します。この情報はきわめて長くなることがあります。`--license` オプションが指定されない場合に `pkg info` コマンドで表示されるその他の情報は表示されません。

```
$ pkg info --license osnet-incorporation
You acknowledge that your use of this Oracle Solaris software product
is subject to (i) the license terms that you accepted when you obtained
the right to use Oracle Solaris software; or (ii) the license terms that
you agreed to when you placed your Oracle Solaris software order with
Oracle; or (iii) the Oracle Solaris software license terms included with
the hardware that you acquired from Oracle; or, if (i), (ii) or (iii)
are not applicable, then, (iv) the OTN License Agreement for Oracle
Solaris (which you acknowledge you have read and agree to) available at
http://www.oracle.com/technetwork/licenses/solaris-cluster-express-license-167852.html.
Note: Software downloaded for trial use or downloaded as replacement
media may not be used to update any unsupported software.
```

パッケージマニフェストからの情報の表示

`pkg contents` コマンドは、パッケージのファイルシステムの内容を表示します。オプションやオペランドを指定せずにこのコマンドを使用すると、現在のイメージにインストールされているすべてのパッケージのパス情報が表示されます。表示するパッケージの内容を指定するには、コマンドオプションを使用します。結果を絞り込むには、パッケージ名を 1 つ以上指定します。パッケージ名にはワイルドカードを使用できます。引数が直接 `pkg` に渡され、シェルで展開されないようにするために、ワイルドカードは引用符で囲みます。

`contents` サブコマンドと `search` サブコマンドはどちらも、パッケージの内容をクエリーします。`pkg contents` コマンドは、パッケージのアクションと属性を表示します。`pkg search` コマンドは、クエリーに一致するパッケージを一覧表示します。

パッケージによってインストールされるファイルの一覧表示

次の例では、`pkg contents` コマンドのデフォルト動作、つまり、このイメージにインストール可能なファイルシステムオブジェクトごとの `path` 属性値の表示が示されています。

```
$ pkg contents entire
```

```
pkg: This package delivers no filesystem content, but may contain metadata. Use
the -o option to specify fields other than 'path', or use the -m option to show
the raw package manifests.
```

```
$ pkg contents zip
PATH
usr/bin/zip
usr/bin/zipcloak
usr/bin/zipnote
usr/bin/zipsplit
usr/share/man/man1/zip.1
usr/share/man/man1/zipcloak.1
usr/share/man/man1/zipnote.1
usr/share/man/man1/zipsplit.1
```

pkg contents コマンドは、このイメージにインストール可能な内容のみ表示します。パッケージマニフェストを表示した場合 (-m オプションを使用)、zip パッケージに 12 個のファイルアクションがあることがわかります。この出力に表示されていない 4 つのファイルは、このイメージにインストールできないファイルです。このイメージは x86 アーキテクチャーです。表示されていないファイルは、SPARC アーキテクチャーの 4 つの /usr/bin ファイルです。バリエーションとファセットについては、[86 ページの「オプションのコンポーネントのインストールの制御」](#)を参照してください。

パッケージによってインストールされるファイルの属性の表示

インストールされたファイルについて、パス以外の詳細な情報を一覧表示するには、pkg contents コマンドの -t オプションと -o オプションを使用します。

-t オプションは、表示するアクションのタイプを指定します。コンマ区切りリストで複数のタイプを指定するか、-t オプションを複数回指定することができます。

-o オプションは、表示するアクションの属性を指定します。コンマ区切りリストで複数の属性を指定するか、-o オプションを複数回指定することができます。パッケージのアクションと属性のリストについては、[pkg\(5\)](#) のマニュアルページを参照してください。

```
$ pkg contents -t file -o owner,group,mode,pkg.size,path zip
OWNER GROUP MODE PKG.SIZE PATH
root bin 0555 231260 usr/bin/zip
root bin 0555 110852 usr/bin/zipcloak
root bin 0555 104960 usr/bin/zipnote
root bin 0555 109340 usr/bin/zipsplit
root bin 0444 86192 usr/share/man/man1/zip.1
root bin 0444 2705 usr/share/man/man1/zipcloak.1
root bin 0444 2396 usr/share/man/man1/zipnote.1
```

```
root bin 0444 1837 usr/share/man/man1/zipsplit.1
```

デフォルトで、出力は、パスまたは `-o` オプションで指定された最初の属性によってソートされます。`-s` オプションを使用して、別の属性をソート鍵として指定できます。`-s` オプションは複数回指定できます。

他のファイルシステムオブジェクトおよび属性の表示

次の例は、指定されたパッケージによってインストールされたリンクのパスおよびターゲットを示しています。`pkg(5)` のマニュアルページに示す属性のほかに、いくつかの疑似属性を使用できます。疑似属性の一覧については、[pkg\(1\)](#) のマニュアルページを参照してください。

次の例で、`pkg.name` 疑似属性は、指定されたアクションを提供するパッケージの名前を示しています。この例では、Python 2.6.8 と Python 2.7.3 の両方がインストールされており、`/usr/bin/python` リンクに依存しない場合に特定バージョンにアクセスするために使用するパスがコマンドによって表示されます。複数バージョンの調整については、[94 ページの「デフォルトのアプリケーション実装の指定」](#)を参照してください。

```
$ pkg contents -t link -a path=usr/bin/python -o path,target,pkg.name
PATH          TARGET      PKG.NAME
usr/bin/python python2.6 runtime/python-26
usr/bin/python python2.7 runtime/python-27
```

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

Oracle Solaris には、いくつかのシステムインストールグループパッケージが用意されています。Oracle Solaris 11 GUI インストーラは `solaris-desktop` グループ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および自動インストーラインストールのデフォルトの AI マニフェストは、`solaris-large-server` グループパッケージをインストールします。非大域ゾーン用のデフォルトのインストールマニフェストは、`solaris-small-server` グループパッケージをインストールします。`solaris-minimal-server` グループパッケージは、Oracle Solaris を実行するために必要な、サポートされる最小限のパッケージのセットをインストールします。

次のコマンドを使用して、各グループに含まれる一連のパッケージを表示できます。

```
$ pkg contents -Hro fmri -t depend -a type=group solaris-minimal-server
network/ping
```

```
service/network/ssh
shell/tcsh
shell/zsh
system/network
```

-t オプションは、パッケージの depend アクションに一致します。-a オプションはタイプ group の depend アクションに一致します。-o オプションは、グループ depend アクションの fmri 属性のみを表示します。グループパッケージではファイルなどの内容は指定されないことを思い出してください。グループパッケージは、グループに含まれるほかのパッケージを指定します。グループパッケージの詳細については、[14 ページの「グループパッケージ」](#)を参照してください。

各パッケージのサマリーも表示するには、pkg list -s コマンドを使用します。

```
$ pkg list -Has `pkg contents -Hro fmri -t depend -a type=group solaris-minimal-server`
network/ping                               Ping command
service/network/ssh                         Secure Shell protocol server
shell/tcsh                                  Tenex C-shell (tcsh)
shell/zsh                                    Z Shell (zsh)
system/network                              Core Network Infrastructure
```

ライセンス要件の表示

次の例は、パッケージライセンスに同意する必要があるすべてのパッケージを表示します。

```
$ pkg contents -rt license -a must-accept=true -o license, pkg.name '*'
LICENSE PKG.NAME
BCL     developer/java/jdk-7
BCL     runtime/java/jre-7
LICENSE developer/java/jdk-6
LICENSE library/java/java-demo-6
LICENSE runtime/java/jre-6
lic_OTN consolidation/osnet/osnet-incorporation
lic_OTN install-image/solaris-auto-install
```

ライセンステキストを表示する方法については、[27 ページの「パッケージライセンスの表示」](#)を参照してください。

パッケージの検索

指定したパターンにデータが一致するパッケージを検索するには、pkg search コマンドを使用します。

pkg search コマンドと pkg contents コマンドの比較

pkg search コマンドは、pkg contents コマンドと同じようにパッケージの内容を調べます。pkg contents コマンドは内容を返しますが、pkg search コマンドは検索クエリーに一致するパッケージの名前を返します。次の表に、これら 2 つのコマンドについてのいくつかの類似点と相違点を示します。

pkg contents	<ul style="list-style-type: none"> ■ インストール済みパッケージを調べます。このイメージに対して構成されているすべてのパブリッシャーに関連付けられたリポジトリ内のパッケージを調べる場合は、-r オプションを使用します。 ■ 調べるリポジトリの URI を指定する場合は、-g オプションを使用します。 ■ アクションを指定する場合は、-t オプションを使用します。 ■ 属性および属性値を指定する場合は、-a オプションを使用します。 ■ 結果の列を指定する場合は、-o オプションを使用します。 ■ 結果をソートする場合は、-s オプションを使用します。
pkg search	<ul style="list-style-type: none"> ■ このイメージに対して構成されているすべてのパブリッシャーに関連付けられたリポジトリ内でパッケージを検索します。インストール済みパッケージのみ検索する場合は、-l オプションを使用します。 ■ 検索するリポジトリの URI を指定する場合は、-s オプションを使用します。 ■ アクションを指定する場合は、検索クエリーを使用します。 ■ 属性および属性値を指定する場合は、検索クエリーを使用します。 ■ 結果の列を指定する場合は、-o オプションを使用します。

ヒント - 指定したパッケージの内容を表示するには、pkg contents コマンドを使用し、クエリーに一致するパッケージを表示するには、pkg search コマンドを使用します。関心のある内容を提供するパッケージがわかっている場合は、pkg contents コマンドを使用します。

検索クエリーの指定

デフォルトでは、検索クエリーは、大文字と小文字を除いて完全一致する一連の用語です。大文字と小文字を区別する検索を指定する場合は、-I オプションを使用します。

クエリー用語には、? と * のワイルドカードを使用できます。フレーズを検索するには、単一引用符または二重引用符を使用します。ワイルドカードや引用符を使用するときは、使用しているシェルを必ず考慮してください。

複数のクエリー用語を指定できます。デフォルトで、複数の用語は AND で結合されます。OR で 2 つの用語を明示的に結合できます。

検索クエリーは次の構造化された形式で表現できます。

package:action:index:token

<i>package</i>	検索するパッケージの名前または複数のパッケージに一致するパターン。
<i>action</i>	pkg(5) のマニュアルページのアクションに関するセクションに一覧表示されているアクションの名前。
<i>index</i>	pkg(5) のマニュアルページのアクションに関するセクションに一覧表示されている <i>action</i> の属性の名前。
<i>token</i>	<i>index</i> の値または <i>index</i> の値に一致するパターン。

欠けているフィールドは、暗黙的にワイルドカード化されます。

すべての属性が検索可能なわけではありません。たとえば、*mode* は *file* アクションの属性ですが、*mode* は *index* の有効な値ではありません。

index の一部の値は、ほかの属性から派生した値です。たとえば、*index* に *basename* を指定できますが、これは *file* または *dir* アクションの *path* 属性の最後のコンポーネントです。*index* の便利な値には、*file* および *dir* アクションの *basename* と *path*、*depend* アクションの依存関係タイプ (*require* や *group* など)、*driver* アクションの *driver_name* と *alias* などがあります。

token の値は、*index* で指定された属性の値と比較されます。たとえば、次に示す部分的な *driver* アクションの場合、*index* には属性名 *alias* を指定し、*token* には *pci108e** を指定できます。

```
driver alias=pci108e,1647 alias=pci108e,16a7
```

set アクションの構文は少し違います。*set* アクションの 2 つの属性は、*name* と *value* です。この場合、*index* の値は *name* 属性の値であり、*token* の値は一致する *value* 属性の値と比較されます。次の例は、ドライバパッケージに対する部分的な *set* アクションを示しています。

```
set name=pkg.summary value="Broadcom NetXtreme II 10GbE NIC Driver"
```

次の例は、*action* に *set* を、*index* に *pkg.summary* を、*token* に *Broadcom* を指定します。*search.match* および *pkg.name* 列指定子は疑似属性です。[pkg\(1\)](#) のマニュアルページを参照してください。

```
$ pkg search -o search.match, pkg.name pkg.summary:Broadcom
SEARCH.MATCH                               PKG.NAME
Broadcom NetXtreme II 10GbE NIC Driver    driver/network/ethernet/bnx
Broadcom 57xx 1GbE NIC Driver             driver/network/ethernet/bge
Broadcom NetXtreme II 1GbE NIC Driver     driver/network/ethernet/bnx
Broadcom BCM4401 NIC Driver               driver/network/ethernet/bfe
Broadcom HT1000 SATA driver              driver/storage/bcm_sata
```

set アクションの name 属性の明確に定義された値に

は、pkg.fmri、info.classification、pkg.description、pkg.summary などがあります。[pkg\(5\)](#) のマニュアルページの設定アクションに関する項目を参照してください。

デフォルトでは、現在インストールされているパッケージバージョン以上の一致のみが表示されます。一致したすべてのバージョンを表示するには、-f オプションを使用します。

デフォルトでは、一致するすべてのアクションについて結果が表示されるため、1つのパッケージに複数行の結果が生成されることがあります。一致する各パッケージを1回だけ表示するには、-p オプションを使用します。

指定されたファイルを提供するパッケージの識別

次の例は、libpower ライブラリが system/kernel/power パッケージから取得されたことを示しています。

```
$ pkg search -Hlo pkg.name /lib/libpower.so.1
system/kernel/power
$ pkg search -lo path, pkg.name libpower.so.1
PATH                               PKG.NAME
lib/libpower.so.1 system/kernel/power
$ pkg search -Hlo path, pkg.name basename:libpower.so.1
lib/libpower.so.1 system/kernel/power
$ pkg search -Hlo path, pkg.name 'path:*libpower.so.1'
lib/libpower.so.1 system/kernel/power
```

指定した SMF サービスを提供するパッケージの識別

特定の SMF サービスがどのパッケージで提供されているかを表示するには、サービスの名前を org.opensolaris.smf.fmri 属性の値として検索します。

```
$ pkg search -o pkg.name, search.match 'org.opensolaris.smf.fmri:*network/http*'
PKG.NAME                               SEARCH.MATCH
web/java-servlet/tomcat svc:/network/http
```

```
web/proxy/squid          svc:/network/http
web/proxy/privoxy       svc:/network/http
web/server/lighttpd-14  svc:/network/http
web/server/apache-22    svc:/network/http
web/server/apache-22    svc:/network/http:apache22
web/server/lighttpd-14  svc:/network/http:lighttpd14
web/proxy/privoxy       svc:/network/http:privoxy
web/proxy/squid         svc:/network/http:squid
web/java-servlet/tomcat svc:/network/http:tomcat6
```

この場合、各属性は、インスタンス名が指定されたサービス名とインスタンス名が指定されていないサービス名の、2つの値を持ちます。次の例は、この属性がパッケージマニフェストでどのように指定されるかを示しています。

```
set name=org.opensolaris.smf.fmri value=svc:/network/http value=svc:/network/http:apache22
```

次の例は、これと同じ情報を、各パッケージが1回だけ示されるようにして表示しています。コロン文字をエスケープすることで、コロン文字は別の検索クエリーフィールドでなく *token* の一部として解釈されます。

```
$ pkg search -o pkg.name,search.match 'org.opensolaris.smf.fmri:*network/http\:*'
PKG.NAME          SEARCH.MATCH
web/server/apache-22  svc:/network/http:apache22
web/server/lighttpd-14  svc:/network/http:lighttpd14
web/proxy/privoxy     svc:/network/http:privoxy
web/proxy/squid       svc:/network/http:squid
web/java-servlet/tomcat  svc:/network/http:tomcat6
```

分類またはカテゴリ別のパッケージの一覧表示

次の例は、`info.classification` 属性の値に「Source Code Management」を持つ、すべてのインストール済みパッケージを識別します。

```
$ pkg search -Hlo pkg.shortfmri info.classification:'source code management'
pkg:/developer/versioning/sccs@0.5.11-0.175.2.0.0.8.0
pkg:/developer/versioning/git@1.7.9.2-0.175.2.0.0.34.0
pkg:/developer/versioning/mercurial-27@2.2.1-0.175.2.0.0.34.0
```

次の例は、この一致検索に使用されるパッケージメタデータを示しています。

```
set name=info.classification value="org.opensolaris.category.2008:Development/Source Code Management"
```

この情報は、`pkg info` コマンドからの出力の「Category」行に表示されます。

検索可能なほかの分類については、『[Oracle Solaris 11.2 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「[分類の値](#)」を参照してください。

また、次の例に示すように、パッケージ名のコンポーネントの 1 つを推測するように `pkg list` コマンドを使用することもできます。

```
$ pkg list '*storage*'
$ pkg list -a '*database*'
```

依存パッケージの表示

次の例は、指定したパッケージの依存関係であるパッケージを示しています。

次の例は、`system/kernel/power` パッケージに `require` 依存関係のあるパッケージを示しています。

```
$ pkg search -Hlo pkg.name require:system/kernel/power
system/kernel/dynamic-reconfiguration/i86pc
system/hal
```

次の `pkg contents` コマンドは、検索の結果を確認します。要求された出力 `action.raw` は、パッケージマニフェストに示されるものとまったく同じアクションを表示する疑似属性です。

```
$ pkg contents -rt depend -a fmri='*power*' -o pkg.name,action.raw i86pc system/hal
PKG.NAME                                ACTION.RAW
system/hal                               depend fmri=pkg:/system/kernel/power
@0.5.11-0.175.2.0.0.34.0 type=require variant.opensolaris.zone=global
system/kernel/dynamic-reconfiguration/i86pc depend fmri=pkg:/system/kernel/power
type=require
```

次の例は、多くのパッケージで `pkg:/x11/server/xorg@1.14.99` への `exclude` 依存関係があることを示しています。

```
$ pkg search -lo pkg.name,fmri 'depend:exclude:*xorg*'
PKG.NAME                                FMRI
x11/server/xorg/driver/xorg-video-ati   pkg:/x11/server/xorg@1.14.99
x11/server/xvnc                          pkg:/x11/server/xorg@1.14.99
x11/server/xserver-common                pkg:/x11/server/xorg@1.14.99
x11/server/xorg/driver/xorg-input-vmouse pkg:/x11/server/xorg@1.14.99
x11/server/xephyr                       pkg:/x11/server/xorg@1.14.99
...
```

グループパッケージ内のすべてのパッケージの一覧表示

Oracle Solaris 11 GUI インストーラは `solaris-desktop` グループ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および自動インストーラインストールのデフォルトの AI マニフェスト

は、`solaris-large-server` グループパッケージをインストールします。非大域ゾーン用のデフォルトのインストールマニフェストは、`solaris-small-server` グループパッケージをインストールします。`solaris-minimal-server` グループパッケージは、Oracle Solaris を実行するために必要な、サポートされる最小限のパッケージのセットをインストールします。次の検索書式を使用して、各グループに含まれている一連のパッケージを表示できます。

```
$ pkg search -Ho fmri '*/solaris-minimal-server:depend:group:*'
network/ping
service/network/ssh
shell/tcsh
shell/zsh
system/network
```

この例で、`-o pkg.name` は、クエリーの `package` フィールドに指定されたパッケージの名前のみを返します。

```
group/system/solaris-minimal-server
```

`-o fmri` オプションは、`group` タイプの依存関係として、`solaris-minimal-server` パッケージで指定されたパッケージの `fmri` 属性の値を返します。

`pkg search` コマンドは、指定したパッケージのアクションの属性の値を返しています。この例では、その属性値はパッケージ名になります。このコマンドの結果の数は、[29 ページの「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」](#)で示す、類似の `pkg contents` コマンドの結果の数より多くなることがあり、この理由は、これらの検索結果に、インストール可能なパッケージだけでなく、指定したパッケージの `group` タイプの `depend` アクションで名前が付けられているすべてのパッケージの名前が含まれるためです。たとえば、このイメージにインストールできないパッケージバリエーションおよびファセットが含まれることがあります。この相違を確認するには、`solaris-large-server` パッケージを使用して両方の例を試してみてください。

◆◆◆ 第 3 章

ソフトウェアパッケージのインストールおよび更新

パッケージのインストールと更新は、一部のパッケージを特定のバージョンに制限する、パブリッシャーの検索順序を構成する、パッケージの署名プロパティを設定するなどのイメージの構成によって影響を受けます。イメージの構成については、[第5章「インストールされるイメージの構成」](#)で説明しています。

すでにインストールされているパッケージ、インストールに使用できるパッケージ、および使用可能な更新のあるパッケージを判断する方法については、[第2章「ソフトウェアパッケージに関する情報の取得」](#)で取り上げています。

この章では、次のタスクの実行方法を示します。

- 試験的なインストールを実行して、インストールが成功するかどうか、何がインストールされるかを確認する
- パッケージをインストール、更新、およびアンインストールする
- パッケージを検証する
- インストールされたパッケージの問題を修正する
- インストールされたファイルをその元の内容に復元する
- パッケージをアンインストールする

[57 ページの「非大域ゾーンの操作」](#)では、Oracle Solaris ゾーンに固有のパッケージ操作のさまざまな側面について説明します。

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、[19 ページの「インストール権限」](#)を参照してください。

この章で説明されているコマンドのすべてのオプションの完全なリストについては、[pkg\(1\)](#) のマニュアルページを参照してください。

操作のプレビュー

この章と第5章「インストールされるイメージの構成」で示す多くのコマンドには `-n` オプションがあり、変更を加えることなくコマンドが何を実行するかを確認できます。

ヒント - 使用可能な場合は常に `-n` オプションを使用することをお勧めします。`-n` オプションと 1 つまたは複数の詳細オプション (`-nv`、`-nvv`) を使用して、コマンドの効果を確認してから、`-n` オプションを付けずにコマンドを実行します。

次の例は、実際には実行しないパッケージインストールに関する情報を表示します。

```
$ pkg install -nv group/feature/amp
      Packages to install:      6
      Mediators to change:     1
      Services to change:      2
      Estimated space available: 22.70 GB
Estimated space to be consumed: 751.08 MB
      Create boot environment:  No
Create backup boot environment:  No
      Rebuild boot archive:    No

Changed mediators:
mediator mysql:
      version: None -> 5.1 (system default)

Changed packages:
solaris
database/mysql-51
      None -> 5.1.37,5.11-0.175.2.0.0.34.0:20140303T160611Z
database/mysql-common
      None -> 5.11,5.11-0.175.2.0.0.34.0:20140303T161628Z
group/feature/amp
      None -> 0.5.11,5.11-0.175.2.0.0.33.0:20140217T134747Z
web/server/apache-22/module/apache-dtrace
      None -> 0.3.1,5.11-0.175.2.0.0.34.0:20140303T175456Z
web/server/apache-22/module/apache-fcgid
      None -> 2.3.9,5.11-0.175.2.0.0.34.0:20140303T175502Z
web/server/apache-22/module/apache-php5
      None -> 5.2.17,5.11-0.175.1.0.0.18:20120611T210317Z

Services:
restart_fmri:
      svc:/system/manifest-import:default
      svc:/system/rbac:default
```

この出力は、このインストール操作は新規 BE でなく現在の BE で実行されること、およびこの現在の BE のバックアップは作成されないことを示しています。新規 BE またはバックアップ BE を求めるオプションまたはイメージプロパティを指定することもできます。「Changed packages」セクションには、`amp` グループパッケージがインストールされ、その 5 つのグループ

依存関係がインストールされることが示されています。出力には、インストールされる各パッケージのバージョンが示されています。トークン `None` は、これらのパッケージが現在インストールされておらず、したがって更新されないことを示しています。

次のコマンドは、多数のパッケージが影響を受けるため、大量の出力が生成されます。このファセットを設定すると、すべてのパッケージについてのローカライズされたすべての内容がインストールされます。このプレビューコマンドを実行することで、この操作のスケジュールを決定する方法や、追加する新規ロケールを少なくするかどうかが変更される場合もあります。この出力は、新しい BE はデフォルトで作成されませんが、バックアップ BE が作成されることを示しています。

```
$ pkg change-facet -nv 'facet.locale.*=true'
    Packages to change:      130
    Variants/Facets to change:  1
    Estimated space available: 22.70 GB
    Estimated space to be consumed: 3.45 GB
    Create boot environment:    No
    Create backup boot environment: Yes
    Rebuild boot archive:      No
Changed variants/facets:
    facet.locale.* (local): False -> True
Changed packages:
solaris
...
```

パッケージのインストールおよび更新

次の表は、`pkg install` と `pkg update` コマンドの類似点と相違点を示します。

<code>pkg install</code>	<ul style="list-style-type: none"> ■ オペランドとして 1 つ以上のパッケージ名が必要です。 ■ 現在インストールされていないパッケージをインストールします。 ■ すでにインストールされているパッケージを更新します。 ■ パッケージのダウングレードは行いません。指定したインストール済みパッケージのバージョンの方が低いと、システムはそのパッケージをインストールしません。
<code>pkg update</code>	<ul style="list-style-type: none"> ■ すでにインストールされているパッケージの名前をオペランドとして 0 個以上指定します。 ■ インストール済みパッケージを更新します。 ■ パッケージ名を指定しないか '*' を指定すると、イメージにインストールされているすべてのパッケージが更新されます。 ■ インストール済みパッケージを、FMRI に指定されているバージョンにダウングレードします。

<p>■ まだインストールされていないパッケージはインストールしません。まだインストールされていないパッケージを指定すると、システムはそのパッケージをインストールしません。</p>
--

インストールおよび更新時に、`preserve` 属性と `overlay` 属性を持つファイルがどのように処理されるかについては、[pkg\(5\)](#) のマニュアルページの `file` アクションのこれらの属性を参照してください。

一般的なインストールオプション

このセクションでは、複数のインストール関連のコマンドに共通するオプションについて説明します。メディアータの設定または設定解除、バリエーションまたはファセットの変更、パッケージの修正、およびファイルを元に戻す操作は、パッケージのインストール、更新、またはアンインストールも伴うことがあります。

ブート環境オプション

パッケージをインストール、更新、またはアンインストールした場合に、新しい BE またはバックアップ BE が自動的に作成されることがあります。BE に関するイメージポリシーの制約内で、下に示すオプションを使用して、新しい BE とバックアップ BE の作成を制御できます。新しい BE とバックアップ BE について、および BE に関するイメージポリシーの設定方法については、[100 ページの「ブート環境ポリシーイメージのプロパティ」](#)を参照してください。

次の BE オプションを使用して、新しい BE またはバックアップ BE を作成するかどうかを強制したり、BE にカスタム名を指定したり、新しい BE をアクティブにしないように指定したりします。これらのオプションは、`install`、`exact-install`、`uninstall`、`update`、`revert`、`set-mediator`、`unset-mediator`、`change-variant`、および `change-facet` サブコマンドで使用できます。

`--no-be-activate`

BE が作成される場合に、それを次回ブート時にアクティブ BE として設定しません。

コマンド出力で、新しいブート環境が作成されたことを示すメッセージに注意してください。新しいブート環境を作成してアクティブにすると、`--no-be-activate` オプションを指定しない場合、その BE が次のリブート時にデフォルトでブートされます。

`pkg` コマンドとは別にアクティブ BE を表示および変更するには、[beadm\(1M\)](#) コマンドを使用します。

--no-backup-be

バックアップ BE を作成しません。

--require-backup-be

新しい BE が作成されない場合に、バックアップ BE を作成します。このオプションを指定しないと、イメージポリシーに基づいてバックアップ BE が作成されます。バックアップ BE が自動的に作成される状況の説明については、[100 ページの「ブート環境ポリシーイメージのプロパティ」](#)を参照してください。

--backup-be-name *name*

バックアップ BE が作成される場合に、デフォルトの名前ではなく、*name* という名前を付けます。--backup-be-name を使用すると --require-backup-be が暗黙的に指定されます。

--deny-new-be

新しい BE を作成しません。新しい BE が必要な場合、インストール、更新、アンインストール、または元に戻す操作は実行されません。

--require-new-be

新しい BE を作成します。このオプションを指定しないと、イメージポリシーに基づいて BE が作成されます。BE が自動的に作成される状況の説明については、[100 ページの「ブート環境ポリシーイメージのプロパティ」](#)を参照してください。このオプションを --require-backup-be と組み合わせることはできません。

--be-name *name*

BE が作成される場合に、デフォルトの名前ではなく、*name* という名前を付けます。--be-name を使用すると --require-new-be が暗黙的に指定されます。このオプションを使用することは、操作を行うためのもっとも安全な方法です

非大域ゾーンで動作するオプション

[57 ページの「非大域ゾーンの操作」](#)で説明されているように、大域ゾーンで実行されるパッケージのインストール、除去、および更新の一部のみが、非大域ゾーンに自動的に影響します。-r オプションは、大域ゾーンに入力した pkg 操作と同じ操作を非大域ゾーンで実行し、-r を使用しなかった場合に影響を及ぼすよりもさらに多くのパッケージに影響を及ぼすこととなります。これらのオプションは、install、uninstall、update、change-variant、および change-facet サブコマンドで使用できます。

-r

この操作を大域ゾーンで実行し、インストール済みの solaris ブランドのすべての非大域ゾーンでも実行します。非大域ゾーンに対する影響は、各非大域ゾーンにログインしてコマンドを直接実行することと似ています。

このオプションを指定せずに `pkg` コマンドを大域ゾーンで実行すると、57 ページの「[非大域ゾーンの操作](#)」に記載されているように、非大域ゾーンは大域ゾーンとの互換性を維持するために必要な範囲で変更されます。このオプションを指定すると、`pkg` 操作は `-z` および `-Z` オプションによって制限される場合を除き、インストール済みのすべての非大域ゾーンに適用されます。`-z` オプションおよび `-Z` オプションによって除外されたゾーンは、大域ゾーンとの同期を維持するための更新が必要であれば、変更されることもあります。

`-z zone`

指定された非大域ゾーンでのみこの操作を実行します。`-z` オプションは複数回指定できます。`-z` オプションは、`-r` オプションと同時に指定した場合のみ使用できます。`-z` オプションは `-Z` オプションと一緒に使用できません。

`-Z zone`

指定されたゾーンを除くすべての非大域ゾーンでこの操作を実行します。`-Z` オプションは複数回指定できます。`-Z` オプションは、`-r` オプションと同時に指定した場合のみ使用できます。`-Z` オプションは `-z` オプションと一緒に使用できません。

次のオプションは、大域ゾーンと同時に更新する非大域ゾーンの数を指定します。このオプションは、`install`、`exact-install`、`uninstall`、`update`、`change-variant`、および `change-facet` サブコマンドで使用できます。

`-c n`

n 個までのインストール済みの `solaris` ブランドの非大域ゾーンを大域ゾーンと並行して更新します。 n が 0 または負の数値の場合、すべての非大域ゾーンが大域ゾーンと同時に更新されます。

環境変数 `PKG_CONCURRENCY` も値 n に設定できます。`-c` オプションは `PKG_CONCURRENCY` 設定をオーバーライドします。`-c` オプションが指定された場合、`PKG_CONCURRENCY` は無視されます。

サービスアクションのオプション

パッケージがインストールまたは更新される時、指定されたサービスの再開またはリフレッシュなどの SMF サービスアクションがパッケージに指定されることがあります。多数のパッケージを操作している場合、すべてのサービスアクションが終了する前に `pkg` 操作が終了することがあります。このとき、関連付けられたサービスがまだ利用できないという理由で、新しくインストールされたソフトウェアを使用できない場合があります。

この問題を回避するには、次のいずれかのオプションを使用して、SMF アクチュエータを `pkg` コマンドと同期させて実行します。これらのオプションは、`install`、`uninstall`、`update`、`change-variant`、および `change-facet` サブコマンドで使用できます。

--sync-actuators

このオプションを指定した場合、`pkg` コマンドは、`pkg` が呼び出されたゾーン (大域ゾーンまたは非大域ゾーン) 内ですべての SMF アクチュエータが完了するまで返されません。

--sync-actuators-timeout *timeout*

このオプションを指定した場合、`pkg` コマンドは、すべての SMF アクチュエータが完了するか、*timeout* 期間が終了するか、いずれか短い方が終了するまで返されません。指定された *timeout* (秒単位) までにアクチュエータが完了しない場合、`pkg` コマンドは動作を続行し、リターンコード 8 で終了します。

ライセンスオプション

パッケージをインストールまたは更新する前に、ライセンスを受諾することが必要な場合があります。必要なライセンスを表示して受諾するには、次のオプションを使用します。これらのオプションは、`install`、`exact-install`、`update`、`fix`、`change-variant`、および `change-facet` サブコマンドで使用できます。

--licenses

この操作の一環としてインストールまたは更新されるパッケージのすべてのライセンスを表示するには、`--licenses` オプションを使用します。この操作を続行するために受諾する必要があるライセンスだけでなく、すべてのパッケージのライセンスが表示されます。続行するために受諾する必要があるライセンスは、`--licenses` オプションを指定しなくても表示されます。ほかの操作を開始せずにパッケージのライセンスを表示するには、[27 ページの「パッケージライセンスの表示」](#)に記載されているように `pkg list` コマンドを使用します。受諾する必要があるライセンスを一覧表示するには、[30 ページの「ライセンス要件の表示」](#)に記載されているように `pkg info` コマンドを使用します。

--accept

更新またはインストールされるパッケージのライセンス条項に同意することを示す場合は、`--accept` オプションを使用します。このオプションを指定せず、パッケージのライセンスに同意が必要になった場合、必要なライセンスが表示されてインストール操作は失敗します。

その他のインストールオプション

--no-index

デフォルトでは、パッケージをインストール、更新、またはアンインストールしたときに検索インデックスが更新されます。これらの操作が正しく終了したあとに検索インデックスを更新しない場合は、`--no-index` オプションを使用します。このオプションを指定すると、多数のパッケージをインストールする場合に、いくらか時間を節約できます。すべてのインストー

ル、更新、およびアンインストール操作を完了したら、`pkg refresh` を使用して、使用可能なパッケージのリストと、指定された各パブリッシャーのパブリッシャーメタデータを更新できます。パブリッシャーを指定しない場合、すべてのパブリッシャーを対象にリフレッシュが実行されます。このオプションは、`install`、`exact-install`、`uninstall`、および `update` サブコマンドで使用できます。

--no-refresh

--no-refresh オプションを指定した場合、使用可能なパッケージやその他のメタデータの最新のリストを取得するために、イメージのパブリッシャーのリポジトリに接続しません。このオプションは、`install`、`exact-install`、および `update` サブコマンドで使用できます。

新しいパッケージのインストール

デフォルトで、パッケージを提供したパブリッシャーの検索順序で先頭のパブリッシャーから、イメージの残りの部分と互換性のあるパッケージの最新バージョンがインストールされます。最新バージョンを明示的に要求するには、パッケージ FMRI のバージョン部分に `latest` を使用します。

パッケージがすでにインストールされている場合は、現在インストールされているバージョンを提供したパブリッシャーから、残りのイメージと互換性のある最新バージョンのパッケージをインストールすることによってパッケージが更新されます。

複数のパッケージが指定され、かつ指定されたパッケージのいずれかをこのイメージにインストールできない場合、指定されたパッケージは一切インストールされません。

パッケージが回避リストにある場合は、インストールすると回避リストから削除されます。回避リストについては、[97 ページの「グループパッケージに含まれる一部のパッケージのインストールの回避」](#)を参照してください。

インストール可能なパッケージの識別と指定

イメージで複数のパブリッシャーが有効になっている場合は、パブリッシャーのステイックネスと検索順序を設定したり、またはパッケージ FMRI でパブリッシャーを指定したりすることによって、パッケージを提供するパブリッシャーを制御できます。また、インストールするバージョンをパッケージ FMRI で指定することもできます。パッケージ FMRI の説明については、[14 ページの「障害管理リソース識別子」](#)を参照してください。パブリッシャーのステイックネスと検索順序の設定については、[77 ページの「パブリッシャーの構成」](#)を参照してください。

パッケージ名がパブリッシャーを指定しない場合、一致するパッケージを提供する最初のパブリッシャーがインストール元として使用されます。該当するパブリッシャーがこのイメージ内にインストール可能なパッケージのバージョンを提供しない場合、インストール操作は失敗します。このイメージにインストール可能なパッケージのバージョンを提供するパブリッシャーを確認するには、`pkg list -a` コマンドを使用します。

次のコマンドは、`atool` のインストール可能なバージョンが構成済みパブリッシャーから利用可能であるが、検索順序で最初のパブリッシャーには、このイメージにインストールできないバージョンがあることを示しています。`pkg list` コマンドのオプションについては、[21 ページの「パッケージのインストール状態情報の表示」](#)を参照してください。

```
$ pkg list -a atool
NAME (PUBLISHER)  VERSION  IFO
atool (isvpub)    2.0      ---
$ pkg list -af atool
NAME (PUBLISHER)  VERSION  IFO
atool              1.1      ---
atool (isvpub)    2.0      ---
```

この場合、次のインストールコマンドは失敗します。パッケージシステムでは、検索順序の最初にあるパブリッシャーからパッケージ名 `atool` に一致するものが検索されますが、そのパッケージはインストールできません。

```
$ pkg install atool
```

このパッケージをインストールするには、次の例に示すように、より具体的なパッケージ名を指定します。

```
$ pkg install //isvpub/atool
$ pkg install atool@2.0
```

実際にインストールを実行する前に、インストールされるものを確認するには、`-nv` オプションを使用します。エラーメッセージを受け取る場合、[付録A パッケージのインストールおよび更新のトラブルシューティング](#)を参照してください。

パッケージのソースの指定

指定したパッケージリポジトリまたはパッケージアーカイブを、パッケージデータの取得元になるイメージ内のソースのリストに一時的に追加するには、`-g` オプションを使用します。クライアント SSL 証明書が必要なリポジトリは、このオプションとともに使用できません。このオプションは、子イメージ (非大域ゾーン) を持つイメージでは使用できません。このイメージに非大域ゾーンが

インストールされている場合は、`pkg set-publisher` コマンドを使用してこのパブリッシャーと起点を追加します。このオプションは複数回指定できます。

-g オプションを指定すると、パッケージの取得時に、イメージで有効になっているパブリッシャーが優先されます。

- 指定されたパッケージ名またはパッケージ名パターンに一致するパッケージが、イメージ内で有効になっているパブリッシャーから利用可能な場合に、その同じパブリッシャーが -g オプションで指定された場所に見つからないとき、パッケージシステムは、イメージ内で有効になっているパブリッシャーからパッケージをインストールしようとします。`install` または `update` のあとに、イメージ内で構成されていないパブリッシャーによって提供されるすべてのパッケージは、起点なしでイメージ構成に追加されます。
- 指定されたパッケージ名またはパッケージ名パターンに一致するパッケージが、イメージ内で有効になっているパブリッシャーから利用可能な場合に、その同じパブリッシャーが -g オプションで指定された場所にパッケージを発行しているとき、パッケージシステムは、-g オプションで指定された場所からパッケージをインストールしようとします。

次の例では、イメージ内に構成されている `solaris` パブリッシャーから `btool` パッケージが利用可能です。`btool` パッケージは、リポジトリの起点 `http://pkg.example1.com/` の `devtool` パブリッシャーからも利用可能ですが、`devtool` パブリッシャーはイメージ内に構成されていません。イメージ内に構成されているパブリッシャーからパッケージが利用可能な場合は、構成済みのパブリッシャーが -g のソースより優先されるため、コマンドは `solaris` パブリッシャーからパッケージをインストールしようとします。

```
$ pkg install -g http://pkg.example1.com/ btool
```

`devtool` パブリッシャーからパッケージをインストールするには、パッケージ名にパブリッシャーの名前を指定します。

```
$ pkg install -g http://pkg.example1.com/ //devtool/btool
```

次の例で、`isvpub` は起点 `/export/IPSpkgrepos/isvrepo` でイメージ内に構成されているパブリッシャーです。`isvpub` パブリッシャーは `http://pkg.example2.com/` にあるリポジトリにもパッケージを発行していますが、イメージ内で構成されているこのパブリッシャーにその起点は指定されていません。同じパブリッシャーが両方の場所にパッケージを提供しているため、次のコマンドは `http://pkg.example2.com/` の場所からパッケージをインストールしようとします。

```
$ pkg install -g http://pkg.example2.com/ atool
```

[79 ページの「パッケージパブリッシャーの追加、変更、削除」](#)にあるパブリッシャーのステイツキネスの説明も参照してください。

新しいブート環境へのパッケージのインストール

ヒント - 新しい BE を明示的に指定することは、インストールまたは更新のもっとも安全な方法です。BE が作成される状況については、[100 ページの「ブート環境ポリシーイメージのプロパティ」](#)を参照してください。

新しい BE は指定したインストール、アンインストール、または更新の変更が適用された現在の BE のクローンになります。現在の BE は変更されません。システムは自動的に再起動されません。システムの次回の再ブート時に、新しい BE がデフォルトのブート選択肢になります。現在の BE も引き続きブートに使用できます。

--no-be-activate オプションを指定した場合、次回のリブート時に、新しい BE がデフォルトのブート選択肢になりません。

新規 BE の作成を強制するか、新規 BE がデフォルトで作成される場合に新規 BE にわかりやすい名前を付けるには、--be-name オプションを使用します。

[38 ページの「操作のプレビュー」](#)の例では、group/feature/amp パッケージをインストールするとき、新規 BE がデフォルトで作成されないことを示しました。次の出力の一部では、--be-name オプションが指定されているため、新規 BE が作成されます。

```
$ pkg install -v --be-name s1lamp group/feature/amp
      Packages to install:      6
      Mediators to change:     1
      Estimated space available: 22.70 GB
      Estimated space to be consumed: 751.08 MB
      Create boot environment:  Yes
      Activate boot environment: Yes
      Create backup boot environment: No
      Rebuild boot archive:     No
```

インストール操作の終わりに次のメッセージが表示されます。

```
A clone of s11 exists and has been updated and activated.
On the next boot the Boot Environment s1lamp will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

pkg list コマンドは、現在の BE に group/feature/amp パッケージがインストールされていないため、group/feature/amp パッケージがインストールされていないことを報告します。group/feature/amp パッケージは新しい s1lamp BE にインストールされます。

```
$ pkg list group/feature/amp
pkg list: no packages matching 'group/feature/amp' installed
```

beadm list コマンドを使用して、システムに s11amp という新しいアクティブな BE があることを確認します。「N」で示された BE は現在ブートされています。「R」で示された BE はリブート時のデフォルトです。リブート時のデフォルトの BE を変更するには、beadm activate コマンドを使用します。

```
$ beadm list
BE          Active Mountpoint Space   Policy Created
--          -
s11         N      /           30.92M static 2014-03-05 08:51
s11amp     R      -           25.75G static 2014-03-26 10:45
```

新しい BE に group/feature/amp パッケージがインストールされていることを確認します。新規 BE をマウントし、-R オプションを使用して、マウントされた BE 上で操作を行います。I 列の「i」は、group/feature/amp パッケージがインストールされていることを示します。

```
$ beadm mount s11amp /mnt
$ beadm list
BE          Active Mountpoint Space   Policy Created
--          -
s11         N      /           30.92M static 2014-03-05 08:51
s11amp     R      /mnt        25.75G static 2014-03-26 10:45
$ pkg -R /mnt list group/feature/amp
NAME (PUBLISHER)   VERSION                IFO
group/feature/amp 0.5.11-0.175.2.0.0.33.0 i--
```

s11amp BE を必ずアンマウントしてください。

```
$ beadm unmount s11amp
```

パッケージの拒否

指定したパッケージがインストールされないようにするには、pkg install コマンドの --reject オプションを使用します。一致するパッケージがすでにインストールされている場合、それらはこの操作の一環として削除されます。

グループ依存関係である拒否されたパッケージは、回避リストに配置されます。回避リストについては、[97 ページの「グループパッケージに含まれる一部のパッケージのインストールの回避」](#)を参照してください。

次のコマンド例は、cvs パッケージを除く、developer-gnu パッケージのすべてのグループ依存関係をインストールします。

```
$ pkg install --reject developer/versioning/cvs group/feature/developer-gnu
```

パッケージの更新

install または update サブコマンドを使用して、インストールされているパッケージを、現在インストールされているバージョンを提供したパブリッシャーから、残りのイメージと互換性のあるパッケージの最新バージョンに更新できます。まだインストールされていないパッケージが意図せずにインストールされることを避けるには、pkg update コマンドを使用して、パッケージを更新します。

イメージで複数のパブリッシャーが有効になっている場合は、パブリッシャーのスティッキネスと検索順序を設定したり、またはパッケージ FMRI でパブリッシャーを指定したりすることによって、パッケージを提供するパブリッシャーを制御できます。また、インストールするバージョンをパッケージ FMRI で指定することもできます。パッケージの最新バージョンを明示的に要求するには、パッケージ名のバージョン部分に latest を使用します。パッケージ FMRI の説明については、[14 ページの「障害管理リソース識別子」](#)を参照してください。パブリッシャーのスティッキネスと検索順序の設定については、[77 ページの「パブリッシャーの構成」](#)を参照してください。

更新されるパッケージの一部となる保持される構成ファイルは、ファイルの preserve 属性の値と、ファイルが変更されたかどうかに応じて、インストール、保存、または名前変更されます。パッケージの更新中にファイルが保持されるしくみについては、[pkg\(5\)](#) のマニュアルページのファイルアクションに関するセクションで preserve 属性を参照してください。

パブリッシャーのスティッキネスと検索順序および -g オプションの使用については、[44 ページの「新しいパッケージのインストール」](#)を参照してください。

現在インストールされていないパッケージの更新を試行すると、pkg update 操作はパッケージを更新せずに終了します。インストールされていないパッケージを無視することで、更新する一部のパッケージが現在インストールされていない場合に pkg update が失敗するのを回避するには、--ignore-missing オプションを使用します。

パッケージ FMRI またはパターンが指定されていないとき、または指定されたパターンがアスタリスク文字 (*) である場合の pkg update コマンドの特殊な動作については、[第4章「Oracle Solaris イメージの更新またはアップグレード」](#)を参照してください。

パッケージのダウングレード

pkg update コマンドを使用して、パッケージのアップグレードだけでなくダウングレードを行うこともできます。パッケージをダウングレードするには、現在インストールされているよりも古いバージョンのパッケージ FMRI を指定します。パッケージ FMRI の説明については、[14 ページの「障害管理リソース識別子」](#)を参照してください。インストールされているパッケージのバージョンと、構成されているパブリッシャーから使用可能なバージョンを表示するには、pkg list コマンドを使用します。

ダウングレードされるパッケージの一部となる保持される構成ファイルは、ファイルの preserve 属性の値と、ファイルが変更されたかどうかに応じてインストールまたは名前変更されます。パッケージのダウングレード中にファイルが保持されるしくみについては、[pkg\(5\)](#) のマニュアルページのファイルアクションに関するセクションで preserve 属性を参照してください。

-g オプションの使用については、[44 ページの「新しいパッケージのインストール」](#)を参照してください。

インストール済みパッケージの問題の修正

IPS では、インストール済みパッケージが正しくインストールされていることを検証したり、検証の問題を修正したり、インストール済みファイルをそれがパッケージされた状態に復元したりする操作が提供されています。

pkg fix コマンドと pkg revert コマンドの比較

pkg fix コマンドと pkg revert コマンドは両方とも、インストール済みパッケージのコンポーネントを再インストールします。次の表に、これら 2 つのコマンドについてのいくつかの類似点と相違点を示します。

pkg fix	<ul style="list-style-type: none"> ■ パッケージに対して操作が行われます。パッケージ名に一致する 1 つ以上のパッケージ名またはパターンをオペランドとして指定します。 ■ pkg verify で失敗したパッケージについてのみ操作が行われます。 ■ pkg verify で報告されたエラーのみ修正します。ほかの内容またはメタデータをパッケージからふたたび提供しません。
---------	--

pkg revert	<ul style="list-style-type: none"> ■ ファイルに対して操作が行われます。1 つ以上のパッケージ名またはタグ名をオペラントとして指定します。 ■ オペラントで識別されたファイルをふたたび提供します。ほかの内容またはメタデータをパッケージからふたたび提供しません。
------------	--

パッケージの検証と検証エラーの修正

イメージ内のパッケージのインストールを検証するには、`pkg verify` コマンドを使用します。関連するパブリッシャーの現在の署名ポリシーが `ignore` でない場合、各パッケージの署名がポリシーに基づいて検証されます。署名ポリシーが適用されるしくみについては、[102 ページの「署名付きパッケージのイメージプロパティ」](#)を参照してください。インストール済みパッケージの内容は、独自の内容解析に基づいて検証されるため、ほかのプログラムの場合とは異なる結果が返されることがあります。

パッケージ名を指定しない場合、すべてのインストール済みパッケージが検査されます。`-v` オプションは、インストール済みパッケージごとに少なくとも 1 行の情報メッセージを提供します。次の例には、出力の少量のサンプルのみが表示されています。`pkg/depot` パッケージのインストールにエラーがあります。

```
$ pkg verify -v
PACKAGE                                STATUS
pkg://solaris/archiver/gnu-tar          OK
pkg://solaris/audio/audio-utilities     OK
pkg://solaris/benchmark/x11perf        OK
...
pkg://solaris/package/pkg/depot         ERROR
  dir: var/cache/pkg/depot
    Group: 'pkg5srv (97)' should be 'bin (2)'
  file: var/log/pkg/depot/access_log
    editable file has been changed
  file: var/log/pkg/depot/error_log
    editable file has been changed
...
pkg://solaris/security/sudo             OK
  file: etc/sudoers
    editable file has been changed
...
pkg://solaris/x11/xlock                  OK
pkg://solaris/x11/xmag                   OK
pkg://solaris/x11/xvidtune               OK
```

`pkg verify` コマンドで報告されたパッケージエラーを修正するには、`pkg fix` コマンドを使用します。

pkg verify の出力には、インストールされた sudo パッケージのコンポーネントはパッケージされたコンポーネントと異なるということが表示されますが、これらの相違点は検証エラーとして報告されません。pkg fix は変更を行いません。/etc/sudoers ファイルは置換されません。

```
$ pkg fix pkg://solaris/security/sudo
No repairs for this image.
```

/etc/sudoers ファイルを削除すると、検証は失敗し、pkg fix によってファイルが置換されません。

```
$ pkg fix pkg://solaris/security/sudo
Verifying: pkg://solaris/security/sudo          ERROR
      file: etc/sudoers
      Missing: regular file does not exist
Created ZFS snapshot: 2014-03-13-22:05:42
Repairing: pkg://solaris/security/sudo
Creating Plan (Evaluating mediators):
```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
Completed	1/1	1/1	0.0/0.0	990B/s

PHASE	ITEMS
Updating modified actions	1/1
Updating package state database	Done
Updating package cache	0/0
Updating image state	Done
Creating fast lookup database	Done

1 つのファイルをダウンロードし、1 つのアクション (file アクション) を変更したという記述が示すように、欠落したファイルのみが置換されます。その他の sudo パッケージ内容は影響を受けていません。修復を実行する前に、操作によって現在のインストールのスナップショットが保存されています。pkg fix 出力の「Created ZFS snapshot」の行を参照してください。修復は現在のイメージ内で実行されました。

```
$ zfs list -r rpool/ROOT/s11
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool/ROOT/s11                      16.3G 22.5G 26.1G /
rpool/ROOT/s11@2014-03-13-23:52:19 249M   -    26.1G -
```

pkg verify の出力は、インストールされている pkg/depot パッケージ内のディレクトリの所有権にエラーがあることを示しています。pkg fix の出力は、「Verifying」セクションにのみエラーを表示しています。パッケージ化されたコンポーネントとのほかの相違点は表示されません。

```
$ ls -ld /var/cache/pkg/depot
drwxr-xr-x  3 pkg5srv  pkg5srv      3 Dec  2 19:47 /var/cache/pkg/depot/
$ pkg fix pkg://solaris/package/pkg/depot
Verifying: pkg://solaris/package/pkg/depot          ERROR
      dir: var/cache/pkg/depot
```

```

Group: 'pkg5srv (97)' should be 'bin (2)'
Created ZFS snapshot: 2014-03-13-22:18:52
Repairing: pkg://solaris/package/pkg/depot
Creating Plan (Evaluating mediators):

```

PHASE	ITEMS
Updating modified actions	1/1
Updating package state database	Done
Updating package cache	0/0
Updating image state	Done
Creating fast lookup database	Done

次の出力は、エラーのみが修正されたことを示します。インストールされたコンポーネントとパッケージ化されたコンポーネントのその他の相違点は、そのまま残っています。

```

$ ls -ld /var/cache/pkg/depot
drwxr-xr-x  3 pkg5srv bin          3 Dec  2 19:47 /var/cache/pkg/depot/
$ pkg verify -v pkg://solaris/package/pkg/depot
PACKAGE                                     STATUS
pkg://solaris/package/pkg/depot            OK
file: var/log/pkg/depot/access_log
      editable file has been changed
file: var/log/pkg/depot/error_log
      editable file has been changed

```

ファイルの復元

ファイルをパッケージ化されたときの状況に復元する場合は、`pkg revert` コマンドを使用します。ファイルの所有権および保護も復元されます。



注意 - 一部の編集可能ファイルを元に戻すと、システムがブート不可になったり、その他の異常動作の原因になったりする可能性があります。

重要な編集可能ファイルを元に戻すには、`--require-backup-be` オプションを使用します。

名前付きファイルを元に戻す

次の例は、パッケージ化されたバージョンとは異なる `pkg/depot` パッケージからの 2 つのインストール済みファイルの 1 つを指定します。

```

$ pkg revert -v /var/log/pkg/depot/access_log
Packages to fix:      1
Estimated space available: 21.08 GB

```

```
Estimated space to be consumed: 460.87 MB
  Create boot environment:      No
Create backup boot environment: No
  Rebuild boot archive:        No
```

Changed packages:

```
solaris
  package/pkg/depot
    0.5.11,5.11-0.175.2.0.0.33.0:20140217T134751Z
DOWNLOAD          PKGS      FILES    XFER (MB)   SPEED
Completed          1/1       1/1      0.0/0.0    50B/s
```

```
PHASE              ITEMS
Updating modified actions  1/1
Updating package state database  Done
Updating package cache      0/0
Updating image state        Done
Creating fast lookup database  Done
```

指定されたファイルはパッケージ化されたバージョンで置き換えられました。pkg.depot パッケージのほかのコンポーネントは変更されていません。

タグ付けされたファイルおよびディレクトリを元に戻す

次の操作を実行するには、`--tagged` オプションを使用します。

- 指定されたタグ名でタグ付けされたすべてのファイルを元に戻します。
- 指定したタグ名を持つディレクトリの下にあり、指定されたパターンに一致する、パッケージ化されていないファイルまたはディレクトリを削除します。

詳細については、『[Oracle Solaris 11.2 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「ファイルアクション」および『[Oracle Solaris 11.2 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「ディレクトリアクション」の `revert-tag` 属性の説明を参照してください。

次の例には、タグ名 `dev-init` を持つタグ付きのファイルを含むパッケージが表示されています。

```
$ pkg contents -o pkg.name,path -s pkg.name -t file -a revert-tag=dev-init '*'
PKG.NAME          PATH
system/device-administration etc/mpxio/devid_path.cache
system/device-administration etc/dev/chassis_aliases
system/device-administration etc/dev/.chassis_aliases
system/device-administration etc/dev/reserved_devnames
system/kernel      etc/path_to_inst
system/network     etc/dladm/datalink.conf
```

次のコマンドは、タグ名 `dev-init` を持つすべてのファイルを元に戻す操作のプレビューを示します。ブートアーカイブが再構築される点に注意してください。新規ブート環境またはブート環境のバックアップを作成するオプションを使用することをお勧めします。

```
$ pkg revert -nv --tagged dev-init
      Packages to fix:          6
      Estimated space available: 22.39 GB
      Estimated space to be consumed: 468.60 MB
      Create boot environment:  No
      Create backup boot environment:  No
      Rebuild boot archive:      Yes

Changed packages:
solaris
system/core-os
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T144208Z
system/device-administration
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T144459Z
system/io/usb
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T145048Z
system/kernel
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T145214Z
system/kernel/platform
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T145112Z
system/network
  0.5.11,5.11-0.175.2.0.0.34.0:20140303T150218Z
```

パッケージのアンインストール

インストールされているパッケージを削除するには、`pkg uninstall` コマンドを使用します。

パッケージのインストール操作を取り消す場合、`pkg history` コマンドを使用して、インストールされた正確な内容を検出することができます。詳細は、[108 ページの「操作履歴の表示」](#)を参照してください。

現在インストールされていないパッケージのアンインストールを試行すると、`pkg uninstall` 操作はパッケージをアンインストールせずに終了します。インストールされていないパッケージを無視することで、アンインストールする一部のパッケージが現在インストールされていない場合に `pkg uninstall` が失敗するのを回避するには、`--ignore-missing` オプションを使用します。

パッケージがグループ依存関係の場合、パッケージをアンインストールするとパッケージは回避リストに配置されます。回避リストについては、[97 ページの「グループパッケージに含まれる一部のパッケージのインストールの回避」](#)を参照してください。

イメージの再インストール

最終的に必要な結果が正確にわかっていて、その結果に到達するために、大量のパッケージをアンインストールするなどの大量のパッケージ変更が必要な場合、`pkg exact-install` コマンドを使用した方がよい場合もあります。`pkg exact-install` コマンドの結果は、指定されたパッケージとそれらの依存関係のみがインストールされたイメージとなります。現在インストール済みのパッケージのうち、`pkg exact-install` コマンド行で指定されず、指定されたパッケージの依存関係でもないものは、削除されます。

`pkg exact-install` コマンドは、回避リストにあるパッケージをインストールしないという制約を無視します。パッケージが回避リストにある場合は、インストールすると回避リストから削除されます。回避リストについては、[97 ページの「グループパッケージに含まれる一部のパッケージのインストールの回避」](#)を参照してください。`pkg exact-install` コマンドは、凍結リストにあるパッケージを更新しないという制約を無視します。凍結されたパッケージについては、[90 ページの「指定したバージョンへのパッケージのロック」](#)を参照してください。

それ以外については、`exact-install` サブコマンドは `install` サブコマンドと同じように動作します。イメージバリエーションとファセットの設定、イメージプロパティ設定、およびパブリッシャー設定は保持されます。いずれかのパッケージをこのイメージにインストールできない場合、指定されたパッケージは一切インストールされません。[57 ページの「非大域ゾーンの操作」](#)に記載されているように、非大域ゾーンはパッケージの更新または削除の結果による影響を受けません。`exact-install` には `-r` オプションを使用できません。

パブリッシャーのスティッキネスと検索順序および `-g` オプションの使用については、[44 ページの「新しいパッケージのインストール」](#)を参照してください。

`pkg exact-install` コマンドを使用する際は、次のプラクティスを推奨します。

- [22 ページの「インストール可能なパッケージ」](#)に記載されているように、`pkg list -a` を使用して、構成済みのパブリッシャーから使用可能なパッケージのバージョンを確認します。`exact-install` を使用して現在のバージョンを再インストールする場合、新しいバージョンがインストール可能であれば、インストールするパッケージの一覧にパッケージ FMRI のバージョン部分を指定する必要があります。
- インストールするパッケージの一覧に、`entire incorporation` パッケージを含めます。
- インストールするパッケージの一覧に、`solaris-minimal-server` パッケージなどのシステムグループパッケージの 1 つを含めます。

- `-nv` または `-nvv` オプションを付けてコマンドを最初に実行し、インストールされるものと削除されるものを正確に調べます。
- わかりやすい名前を持つ新規 BE にインストールするために、`--be-name` オプションを使用します。

次の例では、現在のイメージと同じバージョンの最小インストールによる新規イメージが作成されます。

```
$ pkg list -Hv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z
$ pkg exact-install --be-name s11.2 entire@0.5.11,5.11-0.175.2.0.0.34 solaris-minimal-server
```

非大域ゾーンの操作

ほとんどの IPS コマンドは、コマンドを大域ゾーンで使用する場合と同じ方法で非大域ゾーンで使用できます。ゾーンの概要情報は、[17 ページの「イメージとブート環境」](#)を参照してください。

パッケージインストールについては、[57 ページの「大域ゾーンと非大域ゾーンの関係」](#)および [61 ページの「複数の非大域ゾーンの同時更新」](#)に記載されているように、大域ゾーンと非大域ゾーンは親子関係を持ちます。

[59 ページの「システムリポジトリおよびプロキシサービス」](#)に記載されているように、大域ゾーンと非大域ゾーンの重要な違いは、パッケージパブリッシャーの使用方法です。

大域ゾーンと非大域ゾーンの関係

インストールされた solaris ブランドの非大域ゾーンは、大域ゾーンでの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の影響を受けることがあります。

ファセットとバリエーションを変更すると、パッケージのインストールおよび削除が行われ、非大域ゾーンに影響することがあります。

大域ゾーンから非大域ゾーンを更新する場合、非大域ゾーンをブートする必要はありません。非大域ゾーンをただインストールするだけで、大域ゾーンのパッケージ変更による影響を受けることとなります。

インストールおよび更新コマンドを大域ゾーンで実行すると、デフォルトでは、大域ゾーンとインストール済みの各非大域ゾーンは順に更新され、非大域ゾーンの変更は、非大域ゾーンが大域ゾーンとの互換性を維持するために必要な範囲で実行されます。

- 非大域ゾーンで必要最小限の更新を実行する代わりに、大域ゾーンで実行するのと同じ操作を非大域ゾーン内で実行するには、41 ページの「非大域ゾーンで動作するオプション」に記載されている `-r` オプションを使用します。
- 複数の非大域ゾーンを大域ゾーンと同時に更新するには、41 ページの「非大域ゾーンで動作するオプション」に記載され、61 ページの「複数の非大域ゾーンの同時更新」に示されているように、`-c` オプションを使用します。

ヒント - 大域ゾーンとともに非大域ゾーンに加えられる変更を確認するには、`-nv` オプションを使用します。

非大域ゾーンにログインしているときにパッケージコマンドを実行すると、その非大域ゾーンだけが影響を受けます。次に例を示すように、非大域ゾーンはその親となる大域ゾーンと異なってもかまいません。

- 異なるパッケージをインストールできます。
- 結果として大域ゾーンとの互換性が保たれる場合は、同じパッケージの異なるバージョンをインストールできます。
- 異なるパッケージを回避リストに載せることができます。
- 異なるパッケージを凍結したり、異なるバージョンで凍結したりできます。
- 異なるデフォルト実装を選択するようにメディアータを設定できます。
- 異なるファセットを設定できます。

大域ゾーンにインストールされるバージョンにより、非大域ゾーンにインストールされるパッケージのバージョンが制限されることがあります。非大域ゾーンの一部のパッケージは、非大域ゾーンと大域ゾーンで同じバージョンでなければならないため、更新やダウングレードができません。たとえば、`entire` という名前のパッケージは、各非大域ゾーンと大域ゾーンで同じバージョンでなければなりません。

非大域ゾーンにインストールされるパッケージが `parent` 依存関係を持つ場合、大域ゾーン内でそのパッケージを更新すると、非大域ゾーンでそのパッケージが更新されます。`parent` 依存関係を持つパッケージに依存しているパッケージも影響を受けます。

`parent` 依存関係による影響を受けないパッケージは、大域ゾーンにインストールされているバージョンと異なるバージョンで非大域ゾーンにインストールできます。非大域ゾーンで異なるバー

ジョンをインストールするには、`pkg install` コマンドでバージョンを指定するか、必要なバージョンでバージョンを凍結します。

非大域ゾーンにパッケージをインストールする際に役立つ関連情報については、[133 ページの「同期リンクされたパッケージをインストールできない」](#)および[134 ページの「非大域ゾーンをインストールできない」](#)を参照してください。

システムリポジトリおよびプロキシサービス

非大域ゾーンでは、「システムリポジトリ」は大域ゾーン内に構成されているパッケージリポジトリへのアクセスを提供します。大域ゾーンに加えられたパブリッシャー構成の変更は、システムリポジトリを介してただちにすべての非大域ゾーンで認識されます。

非大域ゾーンに構成されたパブリッシャーの起点またはミラーは、その場所が大域ゾーンパブリッシャーリストに構成されていなくても、大域ゾーンからアクセス可能でなければなりません。たとえば、`localsw` パブリッシャーが非大域ゾーンに構成されているが、大域ゾーンには構成されていない場合、`localsw` パブリッシャーのすべての起点およびミラーは大域ゾーンからアクセス可能でなければなりません。

システムリポジトリは `http`、`https`、および `v4` ファイルリポジトリと `.p5p` アーカイブリポジトリをプロキシ設定できます。

ゾーンプロキシは、ゾーン内で実行されている `pkg` コマンドが、大域ゾーン内で実行されているシステムリポジトリと通信できるようにするサービスです。ゾーンプロキシには 2 つの部分があります。次のサービスは大域ゾーンで実行されます。

```
svc:/application/pkg/zones-proxy:default
```

次のサービスは非大域ゾーンで実行されます。

```
svc:/application/pkg/zones-proxy-client:default
```

システムリポジトリとゾーンプロキシサービスの詳細については、[pkg.sysrepo\(1M\)](#) のマニュアルページを参照してください。

次の例は、大域ゾーンのパブリッシャーを示しています。

```
global:~$ pkg publisher
PUBLISHER      TYPE      STATUS P LOCATION
solaris        origin   online F http://pkg.oracle.com/solaris/release/
solaris        origin   online F file:///export/IPSpkgrepos/Solaris/
```

```
devtool (disabled)  origin  online F http://pkg.example1.com/
isvpub             origin  online F http://pkg.example2.com/
```

次の例は、非大域ゾーンにログインしているときに、これらの同じパブリッシャーがどのように見えるかを示しています。

```
z1:~$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris (syspub)   origin   online T <system-repository>
isvpub (syspub)   origin   online T <system-repository>
z1:~$ pkg publisher -F tsv
PUBLISHER  STICKY  SYSPUB  ENABLED  TYPE      STATUS  URI
PROXY
solaris    true    true    true     origin   online  http://pkg.oracle.com/solaris/release/
http:
//localhost:1008
isvpub     true    true    true     origin   online  http://pkg.example2.com/
http:
//localhost:1008
```

非大域ゾーンでは、システムリポジトリは常にプロキシとして表示されます。これは、非大域ゾーンが大域ゾーンのシステムリポジトリと通信するために使用するプロキシです。

非大域ゾーン内からシステムリポジトリを再構成することはできません。たとえば、パブリッシャーの場所が <system-repository> の場合、パブリッシャーの起点やプロパティ、およびパブリッシャーの検索順序を変更することはできません。パブリッシャーが大域ゾーンに追加または再構成された場合、これらの変更は非大域ゾーンによってただちに認識されます。パブリッシャーが大域ゾーン内で設定解除される場合、そのパブリッシャーからパッケージが非大域ゾーンにインストールされないかぎり、パブリッシャーは非大域ゾーン内で設定解除されます。

ヒント - 大域ゾーン内のパブリッシャーを設定解除する前に、非大域ゾーン内でそのパブリッシャーからのパッケージをアンインストールしてください。

パブリッシャーにアクセスできない場合、[84 ページの「プロキシの指定」](#)に記載されているように、大域ゾーンにプロキシを設定することができます。http_proxy および https_proxy 環境変数を使用する場合と使用方法についての説明を含む、非大域ゾーンを持つ場合のプロキシ設定の詳細については、『[Oracle Solaris ゾーンの作成と使用](#)』の「[インストール済みのゾーンがあるシステムでのプロキシ構成](#)」を参照してください。

大域ゾーンですでに構成されているパブリッシャーについて、pkg list コマンドは、大域ゾーンと非大域ゾーンの両方で同じ結果を生成します。

```
z1:~$ pkg list -a isvtool
NAME (PUBLISHER)  VERSION  IFO
isvtool (isvpub)  2.0     ---
isvtool (isvpub)  1.0     ---
```

リポジトリが大域ゾーンで構成されていなくても、リポジトリは非大域ゾーンにアクセスできるネットワークまたはファイルシステムとすることができます。非大域ゾーンのパブリッシャー構成は、大域ゾーンのパブリッシャー構成と一致するか、大域ゾーンのパブリッシャー構成のスーパーセットでなければなりません。たとえば、`localsw` パブリッシャーは起点 `file:///export/IPSpkgrepos/localrepo` を使用して非大域ゾーン内で構成できますが、これは、`localsw` パブリッシャーが大域ゾーン内で構成されていなくても、この場所が大域ゾーン内でアクセス可能であるためです。

複数の非大域ゾーンの同時更新

デフォルトでは、大域ゾーンで `pkg update` コマンドを使用すると、パッケージシステムは大域ゾーンと各非大域ゾーンを順に更新します。複数の非大域ゾーンを同時に更新するには、`-c` オプションを使用するか、大域ゾーンで `PKG_CONCURRENCY` 環境変数を設定します。詳細は、[41 ページの「非大域ゾーンで動作するオプション」](#)を参照してください。

次の例では、両方の非大域ゾーンが大域ゾーンと同時に更新されます。非大域ゾーンは親の大域ゾーンイメージにリンクされているため、この出力では非大域ゾーンをリンクされたイメージとして示しています。

```
global:~$ pkg update -C 0 --be-name s11.2
Startup: Linked image publisher check ... Done
Startup: Refreshing catalog 'solaris' ... Done
Startup: Refreshing catalog 'isvpub' ... Done
Startup: Checking that pkg(5) is up to date ... Done
Planning: Solver setup ... Done
Planning: Running solver ... Done
Planning: Finding local manifests ... Done
Planning: Package planning ... Done
Planning: Merging actions ... Done
Planning: Checking for conflicting actions ... Done
Planning: Consolidating action changes ... Done
Planning: Evaluating mediators ... Done
Planning: Planning completed in 39.00 seconds
      Packages to remove:  2
      Packages to install:  1
      Packages to update: 640
      Create boot environment: Yes
      Create backup boot environment: No

Planning: Linked images: 0/2 done; 2 working: zone:z1 zone:z2
Planning: Linked image 'zone:z1' output:
| Packages to install:  1
| Packages to update: 161
| Services to change:  2
```

```
\
Planning: Linked images: 1/2 done; 1 working: zone:z2
Planning: Linked image 'zone:z2' output:
| Packages to install: 1
| Packages to update: 161
| Services to change: 2
\
Planning: Finished processing linked images.
Download: 0/12068 items 0.0/350.9MB 0% complete
...
Download: 11664/12068 items 336.1/350.9MB 95% complete
Download: Completed 350.91 MB in 187.08 seconds (0B/s)
Download: Linked images: 0/2 done; 2 working: zone:z1 zone:z2
Download: Linked images: 1/2 done; 1 working: zone:z1
Download: Finished processing linked images.
Actions: 1/23382 actions (Removing old actions)
Actions: 3867/23382 actions (Installing new actions)
Actions: 8192/23382 actions (Updating modified actions)
...
Actions: 23266/23382 actions (Updating modified actions)
Actions: Completed 23382 actions in 96.16 seconds.
Finalize: Updating package state database ... Done
Finalize: Updating package cache ... Done
Finalize: Updating image state ... Done
Finalize: Creating fast lookup database ... Done
Finalize: Reading search index ... Done
Finalize: Building new search index ... Done
Finalize: Linked images: 0/2 done; 2 working: zone:z1 zone:z2
Finalize: Linked images: 1/2 done; 1 working: zone:z2
Finalize: Finished processing linked images.

A clone of s11 exists and has been updated and activated.
On the next boot the Boot Environment s11u1 will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

◆◆◆ 第 4 章

Oracle Solaris イメージの更新またはアップグレード

第3章「ソフトウェアパッケージのインストールおよび更新」では、コマンド行で指定された 1 つまたは複数のパッケージのインストール、更新、修正、およびアンインストールについて説明しました。この章では、Oracle Solaris イメージを、次にサポートされる更新または次のリリースにアップグレードする方法について説明します。システムをアップグレードすることは、ブート可能なイメージを更新することを意味します。17 ページの「イメージとブート環境」に記載されているように、物理マシンまたは仮想マシンは複数のブート可能なイメージを持つことができます。

更新には強力な権限が必要です。詳細は、19 ページの「インストール権限」を参照してください。

pkg update コマンドのすべてのオプションの完全な一覧については、pkg(1) のマニュアルページを参照してください。

イメージの更新の概要

パッケージ FMRI またはパターンを指定せずに、あるいはパターンとしてアスタリスク文字 (*) を使用して pkg update コマンドを使用すると、使用可能な更新があるすべてのインストール済みパッケージが、インストール済みパッケージの依存関係およびパブリッシャーの構成によってシステムに課せられた制約で許可される最新バージョンに更新されます。

- 更新されたインストール済みパッケージによって必要となる新しい依存関係でないかぎり、新しいパッケージはインストールされません。
- インストール済みパッケージが更新されるのは、構成済みパブリッシャーによって提供される更新バージョンが、更新されるイメージの制約に適合する場合に限られます。制約は、パッケージ依存関係のほかに、第5章「インストールされるイメージの構成」に記載されている、ユーザーが制御可能な次のような構成によって課されます。

- 特定のバージョンにロックされたパッケージ
- イメージに設定されたファセットとバリエント
- 構成されたパッケージの署名プロパティ
- パブリッシャーの検索順序およびスティッキネス
- 必要ないずれかのパッケージをインストールできない場合、パッケージは一切更新またはインストールされません。[付録A パッケージのインストールおよび更新のトラブルシューティング](#)を参照してください。

現在のイメージに非大域ゾーンがインストールされている場合、これらのゾーンも更新されます。[57 ページの「非大域ゾーンの操作」](#)を参照してください。

イメージ更新のベストプラクティス

更新する前に、次の手順を実行してください。

- リリースノートを読みます。
- 構成済みのパブリッシャーの起点から使用可能なパッケージバージョンを確認します。パブリッシャーに対して `pkg refresh` コマンドを実行するか、起点の場所で `pkgrepo refresh` コマンドを実行することが必要な場合もあります。
- `-nv` オプションを指定して `pkg update` コマンドを使用することで、実際には更新を実行せずに、更新されるパッケージのリストを表示します。

更新を行うとき、`--be-name` または `--require-new-be` オプションを使用して、現在のブート環境でなく新規ブート環境で変更を行います。詳細は、[40 ページの「ブート環境オプション」](#)および[100 ページの「ブート環境ポリシーイメージのプロパティ」](#)を参照してください。

使用可能なバージョンのチェック

オペレーティングシステムのリリースを更新する場合は、`pkg:/entire incorporation` パッケージの入手可能なバージョンを確認します。次のコマンドは、Oracle Solaris 11 11/11 SRU 10 がインストールされており、Oracle Solaris 11 11/11 SRU 11、12、および 13 が入手可能であり、さらに現在構成されている `solaris` パブリッシャーから Oracle Solaris 11.1 が入手可能であることを示しています。FMRI 内のフィールドについては、[14 ページの「障害管理リソース識別子」](#)を参照してください。

```
$ pkg list -af entire
```

NAME (PUBLISHER)	VERSION	IFO
entire	0.5.11,5.11-0.175.1.0.0.24.2	---
entire	0.5.11,5.11-0.175.0.13.0.4.0	---
entire	0.5.11,5.11-0.175.0.12.0.4.0	---
entire	0.5.11,5.11-0.175.0.11.0.4.1	---
entire	0.5.11,5.11-0.175.0.10.0.5.0	i--

これらの中に必要なバージョンがない場合は、solaris パブリッシャー起点を別のパッケージリポジトリの場所に設定する必要があります。

デフォルトでは、各パッケージは、現在インストールされているバージョンを提供したパブリッシャーから更新されます。パブリッシャーのステイツキネスと検索順を指定することにより、パッケージを提供するパブリッシャーを制御できます。[79 ページの「パッケージパブリッシャーの追加、変更、削除」](#)を参照してください。

更新操作のプレビュー

次のコマンドは、更新によって実際にどのパッケージがインストールされるか (存在する場合) を示しています。-v オプションが指定されているため、このコマンドは、完全な FMRI (複数のバージョンを含む) の 627 のすべてのパッケージが更新され、3 つのパッケージが削除され、1 つの新しいパッケージがインストールされることを示しています。この例では、その出力のほとんどが省略され、entire パッケージのみが示されています。-n オプションが指定されているため、実際に更新は実行されません。-n オプションなしで更新を実行する前に、この出力を確認してください。

```
$ pkg update -nv
    Packages to remove:      3
    Packages to install:    1
    Packages to update:     627
    Estimated space available: 48.43 GB
    Estimated space to be consumed: 3.14 GB
    Create boot environment: Yes
    Activate boot environment: Yes
    Create backup boot environment: No
    Rebuild boot archive:    Yes

Changed packages:
solaris
...
  entire
    0.5.11,5.11-0.175.0.10.0.5.0:20120803T182627Z ->
    0.5.11,5.11-0.175.1.0.0.24.2:20120919T190135Z
...
```

前の例は、Oracle Solaris 11.1 の pkg:/entire incorporation パッケージがインストールされることを示しています。

- それに応じて、`entire incorporation` によって制約されているすべてのインストール済みパッケージが更新されます。
- パッケージ `FMRI` が指定されなかったため、`entire incorporation` によって制約されていないインストール済みパッケージもすべて更新されます。すべてのインストール済みパッケージが、システムの制約で許可された最新バージョンに更新されます。この制約は、インストール済みパッケージの依存関係およびパブリッシャーの構成によってシステムに課せられるものです。
- 更新されたインストール済みパッケージで新しい依存関係が指定されている場合は、インストール済みパッケージが削除され、新しいパッケージがインストールされる可能性があります。

新しいブート環境の指定

65 ページの「更新操作のプレビュー」の例では、`-n` オプションなしでこのコマンドを実行した場合、この更新に対して新規 BE が作成されることを示しています。`-n` オプションなしでこのコマンドを実行した場合は、更新出力の最後に次のメッセージが表示されます。

```
A clone of currentBE exists and has been updated and activated.
On the next boot the Boot Environment newBE will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

現在の BE は変更されません。すべての変更は新しい BE で行われます。

新しい BE を明示的に指定することは、インストールまたは更新のもっとも安全な方法です。BE が作成される状況については、100 ページの「ブート環境ポリシーイメージのプロパティ」を参照してください。`--be-name` オプションを使用すると、新しい BE にわかりやすい名前を付けることもできます。新しい BE がアクティブになるため、次回システムをブートしたときにはこの新しい環境がデフォルトでブートされます。次回のリポートで新規 BE をデフォルトにしないようにする場合、`pkg update` コマンドで `--no-be-activate` オプションを使用します。`beadm activate` コマンドを使用して、デフォルトブート BE をいつでも変更できます。

新規 BE に問題がなければ、古い BE を破棄してかまいません。

ヒント - オペレーティングシステムのリリースごとに、以前の BE を保持しておいてください。必要に応じて以前の古い BE でブートし、古い BE を使用して、そのバージョンと、インストールした新しい次のバージョンの間のバージョンに更新することができます。

インストールするバージョンの指定

許可される最新バージョンに更新したくない場合は、`pkg update` コマンドでパッケージ名 (バージョン文字列の部分を含む) を指定できます。次の例は、新しいバージョンが許可される場合でも、Oracle Solaris 11 11/11 SRU 13 に更新するために `entire incorporation` のバージョンを指定する方法を示しています。必ず再度 `-nv` オプションを使用し、再度出力を確認してください。

```
$ pkg update -nv entire@0.5.11,5.11-0.175.0.13
  Packages to remove:      2
  Packages to install:     1
  Packages to update:     486
  Estimated space available: 48.39 GB
  Estimated space to be consumed: 2.50 GB
  Create boot environment: Yes
  Activate boot environment: Yes
  Create backup boot environment: No
  Rebuild boot archive:    Yes

Changed packages:
solaris
...
  entire
    0.5.11,5.11-0.175.0.10.0.5.0:20120803T182627Z ->
    0.5.11,5.11-0.175.0.13.0.4.0:20121106T194623Z
...
```

一部のインストール済みパッケージは、`entire incorporation` によって制約されたパッケージに依存しない場合もあります。これらのパッケージは、`entire incorporation` だけを更新しても更新されません。これらのパッケージを、その名前によって、同じ `pkg update` コマンドに追加できます。

更新前のバージョン制約の指定

任意の Oracle Solaris 11 11/11 バージョンへの更新は許可するが、Oracle Solaris 11.1 への更新を許可しない場合は、次のコマンドに示すように `entire incorporation` を凍結できます。`0.175.0` を指定すると、`entire` パッケージを `0.175.1` ではなく、たとえば `0.175.0.13` に更新できることを示します。

```
$ pkg freeze -c "Keep this image at 11 11/11." entire@0.5.11,5.11-0.175.0
entire was frozen at 0.5.11,5.11-0.175.0
$ pkg freeze
NAME      VERSION      DATE      COMMENT
```

```
entire 0.5.11,5.11-0.175.0 30 Jan 2013 15:50:01 PST Keep this image at 11 11/11.  
$ pkg list entire  
NAME (PUBLISHER)  VERSION                                IFO  
entire            0.5.11,5.11-0.175.0.10.0.5.0  if-
```

パッケージの凍結の詳細は、[90 ページの「指定したバージョンへのパッケージのロック」](#)を参照してください。

カスタム Incorporation のインストール

前のセクションで示した `pkg freeze` コマンドを使用するのと同じように、独自のカスタム incorporation パッケージを使用して、必要な制約を指定することができます。incorporation パッケージと、Oracle Solaris でのその使用についての詳細は、[12 ページの「incorporation パッケージ」](#)を参照してください。

incorporation パッケージを作成し、ローカルの IPS パッケージリポジトリまたはパッケージアーカイブファイルからパッケージをインストールします。制約を変更するには、カスタム incorporation パッケージを変更および再提供し、`pkg update` を使用して、新規 incorporation をインストールします。

カスタム incorporation を使用して、インストール可能なソフトウェアのバージョンを制御することにより、複数のパッケージリポジトリを維持しなくても、異なるマシン上でさまざまなバージョンの Oracle Solaris を容易に維持することができます。それぞれのイメージにより、異なるバージョンのカスタム更新制御 incorporation パッケージをインストールできます。それぞれのシステムで必要なすべてのソフトウェアバージョンを格納する同一のパッケージリポジトリが、すべてのシステムで共有されます。

カスタム Incorporation パッケージの作成

イメージにインストール可能なコアのオペレーティングシステムパッケージのバージョンは、`pkg:/entire incorporation` パッケージによって制御されます。システムのアップグレードを制御するには、特定バージョンの `pkg:/entire` パッケージを `incorporate` 依存関係として指定するパッケージを作成します。

カスタム Incorporation パッケージマニフェストの作成

次の例は、インストール可能な pkg:/entire パッケージのバージョンを制御するカスタム incorporation パッケージに対する upgradectl.p5m という名前のマニフェストを示します。このマニフェストの設定の一部を以下に示します。

```
set name=pkg.fmri value=upgradectl@1.0
set name=pkg.summary value="Incorporation to constrain the version of the OS"
set name=pkg.description value="This package controls the version of \
pkg://solaris/entire that can be installed."
set name=info.classification value="org.opensolaris.category.2008:Meta Packages/
Incorporations"
set name=pkg.depend.install-hold value=core-os
set name=variant.opensolaris.zone value=global value=nonglobal
set name=variant.arch value=sparc value=i386
depend fmri=feature/package/dependency/self type=parent variant.opensolaris.zone=nonglobal
depend fmri=pkg://solaris/entire type=require
depend fmri=pkg://solaris/entire@0.5.11,5.11-0.175.1.0 type=incorporate
```

pkg.depend.install-hold

ユーザーが pkg update upgradectl コマンドを入力すると、pkg:/entire パッケージも自動的に更新されます。

variant.opensolaris.zone

このパッケージは、大域ゾーンと非大域ゾーンの両方にインストールできます。parent 依存関係の説明も参照してください。

variant.arch

このパッケージは、SPARC と x86 の両方のシステムにインストールできます。

parent 依存関係

このパッケージは、大域ゾーンにすでにインストール済みの場合に限り、非大域ゾーンにインストールできます。

require 依存関係

upgradectl パッケージは、pkg://solaris/entire パッケージがすでにインストールされているか、これと同じ操作でインストール可能である場合に限りインストールできます。

incorporate 依存関係

pkg://solaris/entire パッケージを指定のバージョンでインストールする必要があります。指定された精度の桁数によっては、複数のバージョンが incorporate 依存関係を満たす場合があります。この例では、0.175.1.0 は Oracle Solaris 11.1 SRU 0 を指定します。このアップグレード制御パッケージは、サポート更新のない Oracle Solaris 11.1 にシステムを維持します。ただし、このアップグレード制御パッケージは、pkg:/entire incorporation による制約を受けないパッケージの更新を許可します。

アップグレード制御パッケージの公開

upgradectl パッケージをファイルベースのローカルリポジトリに公開します。このリポジトリは、この新しいパッケージを開発およびテストするためのものです。一般的な用途のリポジトリを作成する場合、リポジトリ用の別個のファイルシステムを作成するなどの追加のステップを含める必要があります。一般的な用途のパッケージリポジトリを作成することについては、『[Oracle Solaris 11.2 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。

システム上にパッケージ開発リポジトリを作成します。pkgrepo コマンドについては、[pkgrepo\(1\)](#) のマニュアルページを参照してください。

```
$ pkgrepo create myrepo
```

このリポジトリのデフォルトパブリッシャーを設定します。デフォルトパブリッシャーは、リポジトリの publisher/prefix プロパティの値です。

```
$ pkgrepo -s myrepo set publisher/prefix=site
```

upgradectl パッケージを開発リポジトリに公開します。

```
$ pkgsend -s myrepo publish upgradectl.p5m
pkg://site/upgradectl@1.0,5.11:20131104T072336Z
PUBLISHED
```

リポジトリのデフォルトパブリッシャーが、パッケージ FMRI に適用されていることに注意してください。

リポジトリを調べて、パッケージが公開されたことを確認します。

```
$ pkgrepo -s myrepo list
PUBLISHER NAME          0 VERSION
site      upgradectl    1.0,5.11:20131104T072336Z
$ pkg list -vg myrepo
FMRI                                IFO
pkg://site/upgradectl@1.0,5.11:20131104T072336Z    ---
```

パッケージを、共有されている場所にある別個の ZFS ファイルシステム内のローカルリポジトリに提供します。

```
$ pkgrecv -s myrepo -d /export/IPSpkgrepos/Solaris upgradectl
Processing packages for publisher site ...
Retrieving and evaluating 1 package(s) ...
PROCESS   ITEMS   GET (MB)   SEND (MB)
Completed    1/1    0.0/0.0    0.0/0.0
```

リポジトリ内のパッケージと、パッケージが incorporation で指定する pkg:/entire のバージョンを確認します。

```

$ pkg info -g /export/IPSpkgrepos/Solaris upgradectl
    Name: upgradectl
    Summary: Incorporation to constrain the version of the OS
    Description: This package controls the version of pkg://solaris/entire that
                can be installed.
    Category: Meta Packages/Incorporations
    State: Not installed
    Publisher: site
    Version: 1.0
    Build Release: 5.11
    Branch: None
    Packaging Date: November 20, 2013 01:01:05 AM
    Size: 0.00 B
    FMRI: pkg://site/upgradectl@1.0,5.11:20131120T010105Z
$ pkg contents -Hro fmri -t depend -a type=incorporate upgradectl
pkg://solaris/entire@0.5.11,5.11-0.175.1.0

```

IPS パッケージの作成と提供についての詳細は、『[Oracle Solaris 11.2 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「[パッケージの作成および発行](#)」を参照してください。

パブリッシャーの起点の設定

site パブリッシャーの起点を設定します。システムリポジトリはこの情報で自動的に更新されるため、非大域ゾーンでは site パブリッシャーからパッケージにアクセスできるようになります。

```

$ pkg set-publisher -g /export/IPSpkgrepos/Solaris site
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F https://pkg.oracle.com/solaris/support/
site               origin   online F file:///export/IPSpkgrepos/Solaris/

```

アップグレード制御パッケージのインストール

パッケージをインストールします。このケースでは、インストール済みの pkg://entire のバージョンが、アップグレード制御パッケージによって incorporation 指定されたバージョンと同じであるため、変更はほとんどありません。パッケージは非大域ゾーンにもインストールされることに注意してください。

```

$ pkg list -v entire
FMRI                                                    IFO
pkg://solaris/entire@0.5.11,5.11-0.175.1.0.0.24.2:20120919T190135Z  i--
$ zoneadm list
global
z1

```

```

$ pkg install upgradectl
    Packages to install: 1
    Create boot environment: No
    Create backup boot environment: No

Planning linked: 0/1 done; 1 working: zone:z1
Planning linked: 1/1 done
Downloading linked: 0/1 done; 1 working: zone:z1
Downloading linked: 1/1 done
PHASE                                ITEMS
Installing new actions                9/9
Updating package state database       Done
Updating image state                  Done
Creating fast lookup database         Done
Reading search index                  Done
Updating search index                  1/1
Executing linked: 0/1 done; 1 working: zone:z1
Executing linked: 1/1 done

```

次のコマンドは、インストール済みのバージョンよりも新しいバージョンの pkg:/entire が、構成済みの solaris パブリッシャーから入手できますが、アップグレードの試行は新しくインストールされたアップグレード制御パッケージによって制御されていることを示しています。

```

$ pkg list -af entire
NAME (PUBLISHER)                VERSION                                IFO
entire                          0.5.11-0.175.1.13.0.6.0             ---
entire                          0.5.11-0.175.1.12.0.5.0             ---
entire                          0.5.11-0.175.1.11.0.4.0             ---
entire                          0.5.11-0.175.1.10.0.6.0             ---
entire                          0.5.11-0.175.1.10.0.5.0             ---
...
$ pkg update
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to latest available versions.
This may indicate an overly constrained set of packages are installed.
latest incorporations:
...
Try specifying expected results to obtain more detailed error messages.
$ pkg update -nv entire@0.5.11-0.175.1.13.0.6.0
pkg update: No matching version of entire can be installed:
  Reject: pkg://solaris/entire@0.5.11,5.11-0.175.1.13.0.6.0:20131108T211557Z
  Reason: This version is excluded by installed incorporation pkg://site/
  upgradectl@1.0,5.11:20131120T010105Z

```

アップグレード制御パッケージの更新

システムを新しいバージョンに更新することをユーザーに許可する準備が整ったら、upgradectl.p5m マニフェストを更新し、新しい更新制御パッケージを再発行して再提供します。次のマニフェストでは、更新制御パッケージのバージョンと、entire incorporation の

バージョンが更新されています。ユーザーへの支援として、アップグレード制御パッケージのバージョン 1.10 は、pkg:/entire パッケージの更新済みバージョン 0.175.1.10 に対応するように設定されます。

```
set name=pkg.fmri value=upgradectl@1.10
set name=pkg.summary value="Incorporation to constrain the version of the OS"
set name=pkg.description value="This package controls the version of \
pkg://solaris/entire that can be installed."
set name=info.classification value="org.opensolaris.category.2008:Meta Packages/
Incorporations"
set name=pkg.depend.install-hold value=core-os
set name=variant.opensolaris.zone value=global value=nonglobal
set name=variant.arch value=sparc value=i386
depend fmri=feature/package/dependency/self type=parent variant.opensolaris.zone=nonglobal
depend fmri=pkg://solaris/entire type=require
depend fmri=pkg://solaris/entire@0.5.11,5.11-0.175.1.10 type=incorporate
```

次のコマンドにより、更新制御パッケージが再発行および再提供されます。

```
$ pkgsend -s myrepo publish upgradectl.p5m
pkg://site/upgradectl@1.10,5.11:20131120T021902Z
PUBLISHED
$ pkgrepo -s myrepo list
PUBLISHER NAME                                O VERSION
site      upgradectl                          1.10,5.11:20131120T021902Z
site      upgradectl                          1.0,5.11:20131120T010105Z
$ pkgrecv -s myrepo -d /export/IPSpkgrepos/Solaris upgradectl
Processing packages for publisher site ...
Retrieving and evaluating 1 package(s)...
PROCESS                ITEMS      GET (MB)   SEND (MB)
Completed              1/1        0.0/0.0    0.0/0.0
$ pkg refresh site
$ pkg list -af pkg://site/upgradectl
NAME (PUBLISHER)                VERSION                IFO
upgradectl (site)              1.10                  ---
upgradectl (site)              1.0                   i--
```

イメージのアップグレード

次の pkg update コマンドではパッケージが指定されていないため、すべてのパッケージは入手可能で許可されている最新バージョンに更新されます。コマンドにより、アップグレード制御パッケージは入手可能な最新バージョンに更新され、これによってイメージがアップグレードされますが、この理由は、upgradectl パッケージの pkg.depend.install-hold 設定により、upgradectl パッケージが更新されるときに pkg:/entire パッケージが更新されるためです。イメージは、新規 upgradectl incorporation で指定されている pkg:/entire incorporation のバージョンにアップグレードされます。

```

$ pkg update --be-name s11u1_10
    Packages to remove: 1
    Packages to update: 186
    Mediators to change: 1
    Create boot environment: Yes
    Create backup boot environment: No

Planning linked: 0/1 done; 1 working: zone:z1
Linked image 'zone:z1' output:
| Packages to remove: 1
| Packages to install: 3
| Packages to update: 73
| Mediators to change: 1
| Services to change: 3
`

Planning linked: 1/1 done
DOWNLOAD                                PKGS      FILES    XFER (MB)  SPEED
Completed                                187/187   16139/16139  507.9/507.9  562k/s

Downloading linked: 0/1 done; 1 working: zone:z1
Downloading linked: 1/1 done
PHASE                                     ITEMS
Removing old actions                      1473/1473
Installing new actions                    3451/3451
Updating modified actions                 16378/16378
Updating package state database           Done
Updating package cache                   187/187
Updating image state                      Done
Creating fast lookup database             Done
Reading search index                    Done
Building new search index                 851/851
Executing linked: 0/1 done; 1 working: zone:z1
Executing linked: 1/1 done

A clone of s11u1_0 exists and has been updated and activated.
On the next boot the Boot Environment s11u1_10 will be
mounted on '/'. Reboot when ready to switch to this updated BE.

```

現在の BE が変更されていないこと、および新しい BE に更新済みパッケージが含まれていることを確認します。

```

$ pkg list entire upgradectl
NAME (PUBLISHER)                                VERSION                                IFO
entire                                             0.5.11-0.175.1.0.0.24.2             i--
upgradectl (site)                               1.0                                  i--
$ beadm mount s11u1_10 /mnt
$ pkg -R /mnt list entire upgradectl
NAME (PUBLISHER)                                VERSION                                IFO
entire                                             0.5.11-0.175.1.10.0.6.0             i--
upgradectl (site)                               1.10                                 i--
$ beadm unmount s11u1_10

```

イメージのダウングレード

オペレーティングシステムのリリースをダウングレードするには、ダウングレードするバージョンより古い BE にブートし、そこからアップグレードします。たとえば、Oracle Solaris 11 11/11 SRU 10 から Oracle Solaris 11 11/11 SRU 13 に更新したあとで、SRU 12 イメージが必要なことがわかった場合は、SRU 10 BE にリブートし、そこから SRU 12 に更新します。

◆◆◆ 第 5 章

インストールされるイメージの構成

この章では、パッケージパブリッシャーの構成、インストール可能なパッケージの制限、パッケージ署名ポリシーの設定、ブート環境 (BE) ポリシーの構成などのイメージ全体に適用する特性を構成する方法について説明します。

- 起点、検索順序、鍵と証明書、およびプロキシの設定などのパブリッシャーの構成
- バリエーションおよびファセットの設定によるオプションコンポーネントのインストールの制御
- 指定したバージョンへのパッケージのロック
- incorporation によって指定されたバージョン制約の緩和
- メディエーションの使用によるアプリケーションのデフォルト実装の指定
- グループパッケージに含まれる一部のパッケージのインストールの回避
- BE 作成ポリシーやパッケージ署名ポリシーなどのイメージプロパティおよびパブリッシャープロパティの構成
- イメージの作成
- パッケージ操作履歴の表示

これらの操作の多くでは、強力な権限が必要です。詳細は、[19 ページの「インストール権限」](#)を参照してください。

この章で説明されているコマンドのすべてのオプションの完全なリストについては、[pkg\(1\)](#) のマニュアルページを参照してください。

パブリッシャーの構成

ソフトウェアをインストールして更新するには、pkg クライアントがパッケージリポジトリに接続できる必要があります。

パブリッシャー情報の表示

このイメージに対して構成されているパッケージパブリッシャーに関する情報を表示するには、`pkg publisher` コマンドを使用します。パブリッシャーは、パッケージ FMRI にパブリッシャーが指定されていない場合に、パッケージを見つけるために検索される順番で一覧表示されます。

デフォルトで、`solaris` パブリッシャーは、新しくインストールされた Oracle Solaris 11 システムに構成されます。パブリッシャーの起点を確認するには、`pkg publisher` コマンドを使用します。

```
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F http://pkg.oracle.com/solaris/release/
isvpub            (non-sticky) origin   online F file:///export/IPSpkgrepos/isvrepo/
devtool           (disabled) origin   online F http://pkg.example1.com/
```

TYPE 列は、LOCATION 値が起点かミラーかを示します。詳細は、[17 ページの「リポジトリの起点とミラー」](#)を参照してください。

STATUS 列と LOCATION 列の間にある P 列は、その場所がプロキシ設定されているかどうかを指定します。この列の値は true (T) または false (F) です。ファイルリポジトリはプロキシ設定されません。値が T になっている HTTP リポジトリは、`pkg set-publisher` コマンドで起点が追加されたときに `--proxy` オプションで指定されたプロキシを使用してプロキシ設定されています。`pkg publisher` に `-F tsv` オプションを指定すると、その場所に設定されているプロキシが PROXY 列に表示されます。

```
$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB ENABLED TYPE      STATUS URI                                     PROXY
solaris   true  false true   origin online http://pkg.oracle.com/solaris/release/ -
isvpub    false false true   origin online file:///export/IPSpkgrepos/isvrepo/ -
devtool   true  false false   origin online http://pkg.example1.com/ -
```

P 列の F あるいは PROXY 列の - は、その場所が `pkg set-publisher` コマンドを使用してプロキシ設定されなかったことを示します。その場所が、`http_proxy` 環境変数を設定することによってプロキシ設定されている場合、`pkg publisher` からの出力は、F または - を表示したままになります。プロキシを設定するためのさまざまな方法については、[84 ページの「プロキシの指定」](#)を参照してください。

それらのパブリッシャーの詳細構成を表示するには、名前でパブリッシャーを指定します。

```
$ pkg publisher solaris
```

```

Publisher: solaris
Alias:
Origin URI: http://pkg.oracle.com/solaris/release/
SSL Key: None
SSL Cert: None
Client UUID: e15e3228-eada-11df-80ab-8023183d954b
Catalog Updated: March 4, 2014 11:48:02 PM
Enabled: Yes
Properties:
    proxied-urls = []

```

パブリッシャーの検索順序の先頭のパブリッシャーのみを表示する場合は、`-P` オプションを使用します。

```

$ pkg publisher -P
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/

```

有効になっているパブリッシャーのみを表示する場合は、`-n` オプションを使用します。

```

$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
isvpub             (non-sticky) origin  online F file:///export/IPSpkgrepos/isvrepo/

```

パッケージパブリッシャーの追加、変更、削除

次の操作を実行するには、`pkg set-publisher` コマンドを使用します。

- 新しいパブリッシャーを構成する
- パブリッシャーの起点とミラーを設定する
- パブリッシャーのステイックネスを設定する
- パブリッシャーの検索順序を設定する
- パブリッシャープロパティを設定または設定解除したり、パブリッシャープロパティ値を追加または削除する
- パブリッシャーの SSL 鍵および証明書を指定する
- パブリッシャーのプロキシを設定する
- パブリッシャーを有効または無効にする
- パブリッシャーを削除する

`pkg set-publisher` コマンドには 2 つの形式があります。詳細は、[pkg\(1\)](#) のマニュアルページを参照してください。

- 1つの形式では、パブリッシャーの名前が必須オペランドになります。
- 別の形式では、リポジトリ URI が `-p` オプションの引数として指定され、パブリッシャー情報はその指定されたリポジトリから取得されます。パブリッシャー名がオプションのオペランドになるため、複数のパブリッシャーがそのリポジトリにパッケージを公開する場合は、名前が付いたパブリッシャーのみ構成することができます。

パブリッシャーの追加

以下の例は、パブリッシャーを追加するための2つの方法を示します。

例 5-1 新しいパブリッシャーの指定

次のコマンドは、`-g` オプションで指定した起点 URI を持つ `devtool` という新しいパブリッシャーを追加し、このパブリッシャーが検索順序の先頭になるように設定します。指定したパブリッシャーを検索順序の先頭に設定するには、`-P` オプションまたは `--search-first` オプションを使用します。

```
$ pkg set-publisher -P -g http://pkg.example1.com/release/ devtool
```

例 5-2 パブリッシャー構成のインポート

指定したリポジトリ URI からパブリッシャーの構成情報を取得するには、`-p` オプションを使用します。パブリッシャーを指定した場合、一致するパブリッシャーのみが追加または更新されます。パブリッシャーを指定しない場合、すべてのパブリッシャーが必要に応じて追加または更新されます。

```
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris             origin   online F http://pkg.oracle.com/solaris/release/
$ pkg set-publisher -p /export/IPSpkgrepos/myrepo
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris             origin   online F http://pkg.oracle.com/solaris/release/
site                origin   online F file:///export/IPSpkgrepos/myrepo/
```

パブリッシャーの起点の追加と変更

次のコマンドは、`solaris` パブリッシャーに起点を追加する方法を示しています。イメージ内の特定のパブリッシャーに対して複数の起点が構成されている場合、IPS クライアントは、パッケージデータの取得元として最適な起点を選択しようとします。

```
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
$ pkg set-publisher -g /export/IPSpkgrepos/Solaris solaris
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
solaris            origin  online F file:///export/IPSpkgrepos/Solaris/
```

指定したパブリッシャーの起点として URI を削除するには、-G オプションを使用します。

パブリッシャーの起点 URI を変更するには、新しい URI を追加し、古い URI を削除します。

```
$ pkg set-publisher -G '*' -g file:///export/IPSpkgrepos/isvrepo/ isvpub
```

パブリッシャーのミラーの追加と変更

指定したパブリッシャーのミラーとして URI を追加するには、-m オプションを使用します。起点とミラーの違いについては、[17 ページの「リポジトリの起点とミラー」](#)を参照してください。同じパッケージの同じバージョンが、同じパブリッシャーの起点リポジトリ内にも存在しないかぎり、ミラーリポジトリの内容にアクセスすることはできません。

```
$ pkg set-publisher -m http://pkg.example3.com/ devtool
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
devtool            origin  online F http://pkg.example1.com/
devtool            mirror  online F http://pkg.example3.com/
```

指定したパブリッシャーのミラーとして URI を削除するには、-M オプションを使用します。

パブリッシャーのミラー URI を変更するには、新しい URI を追加し、古い URI を削除します。

パブリッシャーの検索順序およびスティッキネスの設定

新しく追加されたパブリッシャーは、デフォルトで固定です。パブリッシャーが非固定の場合は、このパブリッシャーからインストールされたパッケージを別のパブリッシャーから更新できます。パブリッシャーのスティッキネスを設定するには、--sticky および --non-sticky オプションを使用します。

新しく追加されたパブリッシャーは、デフォルトで検索順序の最後になります。パブリッシャーの検索順序は、インストールするパッケージを検索するために使用されます。パッケージが最初にインストールされたときのパブリッシャーが非固定の場合は、更新するパッケージを検索する

ためにパブリッシャーの検索順序が使用されます。パブリッシャーの検索を変更するには、`--search-before`、`--search-after`、および `--search-first` オプションを使用します。`-P` オプションは、`--search-first` オプションの同義語です。

一致するパッケージを提供する最初のパブリッシャーがインストール元として使用されます。該当するパブリッシャーがこのイメージ内にインストール可能なパッケージのバージョンを提供しない場合、インストール操作は失敗します。この検索順序でさらに下にあるパブリッシャーからインストールするには、パブリッシャーの名前やパッケージのバージョン文字列などの詳細情報をパッケージ FMRI に指定します。

パブリッシャープロパティの構成

パブリッシャープロパティを設定および設定解除したり、パブリッシャープロパティの値を追加および削除したりするには、次のオプションを使用します。

- `--set-property property=value`
- `--add-property-value property=value`
- `--remove-property-value property=value`
- `--unset-property property`

`publisher-search-order` および `signature-required-names` プロパティは、複数の値を取ることができます。

103 ページの「[パッケージの署名プロパティの構成](#)」の `pkg set-publisher` の例を参照してください。

パブリッシャーの鍵と証明書の構成

例 5-3 パブリッシャーの鍵と証明書の指定

クライアントの SSL 鍵を指定するには、`-k` オプションを使用します。クライアントの SSL 証明書を指定するには、`-c` オプションを使用します。指定した証明書を信頼できる CA 証明書として追加するには、`--approve-ca-cert` オプションを使用します。ユーザーが承認した CA 証明書のハッシュが、このパブリッシャーに対する `pkg publisher` コマンドの出力に一覧表示されます。78 ページの「[パブリッシャー情報の表示](#)」を参照してください。

```
$ pkg set-publisher -k /root/creds/example.key -c /root/creds/example.cert \
```

```
--approve-ca-cert /tmp/example_file.pem isvpub
```

例 5-4 パブリッシャーの鍵と証明書の取り消し

指定した証明書を取り消された証明書として扱うには、`--revoke-ca-cert` オプションを使用します。ユーザーが取り消した CA 証明書のハッシュが、このパブリッシャーに対する `pkg publisher` コマンドの出力に一覧表示されます。

承認された証明書の一覧および取り消された証明書の一覧から指定した証明書を削除するには、`--unset-ca-cert` オプションを使用します。

パブリッシャープロキシの構成

指定した起点またはミラーの内容を取得する永続的なプロキシ URI を指定するには、`--proxy` オプションを使用します。プロキシの値は `protocol://host[:port]` で、`protocol` は `http` または `https` で、`:port` はオプションです。プロキシを設定するためのさまざまな方法については、[84 ページの「プロキシの指定」](#)を参照してください。

パブリッシャーの有効化および無効化

新しく追加されたパブリッシャーは、デフォルトで有効になります。無効なパブリッシャーは、パッケージリストの作成時や `install`、`uninstall`、または `update` 更新パッケージ操作で使用されません。無効なパブリッシャーのプロパティを設定または表示することはできません。1 つだけのパブリッシャーが有効な場合、そのパブリッシャーは無効にできません。

次のコマンドは、`isvpub` パブリッシャーを有効にし、それを検索順序で `devtool` パブリッシャーの前に設定します。

```
$ pkg set-publisher --enable --search-before devtool isvpub
```

パブリッシャーを無効にするには `--disable` オプションを使用します。パブリッシャーの起点が一時的に使用できないなどの場合にパブリッシャーを無効にする場合があります。すべてのパブリッシャーが使用できない場合、パッケージのインストールおよび更新操作は失敗します。

パブリッシャーの削除

パブリッシャーを削除するには、`pkg unset-publisher` コマンドを使用します。

```
$ pkg unset-publisher devtool
```

プロキシの指定

プロキシを設定する方式により、効果と利点は異なります。たとえば、`pkg set-publisher` コマンドは、パブリッシャー構成の一部としてプロキシを格納し、`http_proxy` 環境変数を使用すると、認証済みプロキシを設定できます。

プロキシを設定するための `pkg set-publisher` コマンドの設定

`pkg set-publisher` コマンドの `--proxy` オプションは、指定されたパブリッシャーの起点とミラー URI の永続的プロキシ URI を設定します。プロキシ値は、パブリッシャーの構成の一部として格納されます。パブリッシャーの構成の一部としてプロキシ値を格納することで、子イメージによって使用されるシステムリポジトリが自動的に更新されます。パブリッシャーの構成の一部としてプロキシ値を格納することは、異なるパブリッシャーに対して異なるプロキシを使用できるということも意味します。

```
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F file:///export/IPSpkgrepos/Solaris/
$ pkg publisher -F tsv
PUBLISHER STICKY SYSPUB ENABLED TYPE      STATUS URI                                PROXY
solaris   true   false true   origin online file:///export/IPSpkgrepos/Solaris/ -
$ pkg set-publisher -g http://pkg.oracle.com/solaris/release/ --proxy proxyURI solaris
$ pkg publisher solaris
  Publisher: solaris
  Alias:
  Origin URI: file:///export/IPSpkgrepos/Solaris/
  SSL Key: None
  SSL Cert: None
  Origin URI: http://pkg.oracle.com/solaris/release/
  Proxy: proxyURI
  SSL Key: None
  SSL Cert: None
  Client UUID: e15e3228-eada-11df-80ab-8023183d954b
  Catalog Updated: July 11, 2013 11:32:46 PM
  Enabled: Yes
  Properties:
    proxied-urls = []
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F file:///export/IPSpkgrepos/Solaris/
solaris            origin   online T http://pkg.oracle.com/solaris/release/
$ pkg publisher -F tsv
```

```
PUBLISHER STICKY SYSPUB ENABLED TYPE STATUS URI PROXY
solaris true false true origin online file:///export/IPSpkgrepos/Solaris/ -
solaris true false true origin online http://pkg.oracle.com/solaris/release/ proxyURI
```

このイメージに非大域ゾーンがある場合、システムリポジトリはこのプロキシ情報を使用して自動的に更新されます。system-repository サービスにプロパティを設定する必要はありません。非大域ゾーン内のパブリッシャープロキシを調べた場合、大域ゾーンに表示されるものと同じプロキシ URI は表示されません。大域ゾーンでは、システムリポジトリはプロキシ URI を使用します。非大域ゾーンでは、システムリポジトリはプロキシそのものとして機能し、非大域ゾーンが大域ゾーンのシステムリポジトリと通信できるようになります。57 ページの「大域ゾーンと非大域ゾーンの関係」に、非大域ゾーン内でシステムリポジトリパブリッシャーが表示された場合の例が示されています。

pkg set-publisher コマンドの --proxy オプションを使用して、認証済みポリシーを設定することはできません。--proxy オプションの値を、protocol://user:password@host の形式にすることはできません。

プロキシを設定するための環境変数の使用

プロキシ環境変数値は、そのプロトコルのすべての URI に適用されます。実行時は、http_proxy 環境変数の値が、pkg set-publisher コマンドの --proxy オプションで設定された値をオーバーライドします。プロキシ環境変数の詳細については、curl(1) のマニュアルページの「環境」セクションを参照してください。

非大域ゾーンを持つイメージ内で http_proxy 環境変数を設定した場合、svc:/application/pkg/system-repository SMF サービス内のプロキシプロパティをこれらの同じ値に設定して、サービスをリフレッシュします。

```
$ svccfg -s system-repository:default setprop config/http_proxy = astring: proxyURI
$ svccfg -s system-repository:default listprop config/*proxy
config/https_proxy astring
config/http_proxy astring proxyURI
$ svcprop system-repository:default | grep proxy
config/https_proxy astring ""
config/http_proxy astring ""
$ svcadm refresh system-repository:default
$ svcprop system-repository:default | grep proxy
config/https_proxy astring ""
config/http_proxy astring proxyURI
```

pkg publisher コマンドは、環境変数または SMF サービスプロパティを設定することによって設定されたプロキシを表示しません。

`http_proxy` 環境変数の値を変更した場合、`system-repository` サービスプロパティを更新してサービスをリフレッシュするようにしてください。

オプションのコンポーネントのインストールの制御

ソフトウェアには、オプションのコンポーネントや、相互に排他的なコンポーネントが含まれることがあります。オプションのコンポーネントの例には、ロケールやドキュメントがあります。相互に排他的なコンポーネントの例には、SPARC バイナリと x86 バイナリや、デバッグバイナリと非デバッグバイナリなどがあります。IPS では、オプションのコンポーネントをファセット、相互に排他的なコンポーネントをバリエーションと呼びます。

現在のイメージに設定されているバリエーションとファセットの値を表示し、現在のイメージのバリエーションとファセットを変更できます。イメージに設定されているファセットとバリエーションの現在の値を表示するには、`pkg facet` コマンドと `pkg variant` コマンドを使用します。イメージに設定されているファセットとバリエーションの現在の値を変更するには、`pkg change-facet` コマンドと `pkg change-variant` コマンドを使用します。[pkg\(1\)](#) のマニュアルページと下の例を参照してください。バリエーションとファセットを変更すると、多数のパッケージが更新され、新しい BE が必要になることがあります。変更する前に、どのような変更が行われるかを確認するには、`-nv` を使用します。

バリエーションおよびファセット値がパッケージのインストールに与える影響

ファセットとバリエーションはイメージに設定される特殊なプロパティであり、パッケージマニフェスト内のアクションに設定されるタグです。アクションのファセットおよびバリエーションタグの値とイメージに設定されたファセットおよびバリエーションタグの値の比較により、そのパッケージアクションがインストール可能かどうかを判別します。

各ファセットタグおよびバリエーションタグには名前と値があります。1 つのアクションに複数のファセットタグおよびバリエーションタグを付けることができます。複数のファセットタグやバリエーションタグが付くコンポーネントの例として、開発者によって使用されるアーキテクチャー固有のヘッダーファイルや SPARC 大域ゾーン専用のコンポーネントがあります。

ほとんどのバリエーションタグは、さまざまな値を持つことができます。バリエーション値を指定するパッケージをインストールするために、バリエーションタグの値をイメージに設定する必要があります。`arch`

および zone バリエントは、イメージを作成し、その初期コンテンツをインストールするプログラムによって設定されます。イメージ内の `debug.*` バリエントはデフォルトで `false` です。

アクションに設定されるファセットタグは、値 `true` のみを持つことができます。イメージ内で特定のファセット値を `false` に設定した場合、そのファセットを指定しているファイルまたはその他のアクションはどれもインストールされず、そのファセットを指定している現在インストール済みのファイルアクションはアンインストールされます。

次のアルゴリズムは、イメージに設定されたファセットとバリエントが、特定のアクションがインストールされるかどうかに影響するしくみを示しています。

- ファセットまたはバリエントタグのないアクションは常にインストールされます。
- ファセットタグのあるアクションは、イメージ上のタグに一致するすべてのファセットまたはファセットパターンが `false` に設定されていないかぎり、インストールされます。いずれかのファセットが `true` に設定されているか、明示的に設定されていない (`true` はデフォルト) 場合、アクションがインストールされます。
- バリエントタグのあるアクションは、すべてのバリエントタグの値がイメージに設定されている値と同じ場合にのみインストールされます。
- ファセットタグとバリエントタグの両方があるアクションは、ファセットとバリエントの両方でアクションのインストールが許可されている場合にインストールされます。

バリエント値およびファセット値の例

ほとんどのバリエントは任意の数の値を設定できます。たとえば、`arch` バリエントには、`i386`、`sparc`、`ppc`、`arm`、またはディストリビューションでサポートされているどんなアーキテクチャーでも設定できます。(Oracle Solaris では `i386` と `sparc` のみが使用されます。)例外は `debug` バリエントです。`debug` バリエントには、`true` または `false` のみを設定でき、ほかの値では動作が不確定になります。`file` アクションに非デバッグバージョンとデバッグバージョンの両方がある場合、次の例に示すように、両方のバージョンに該当する `debug` バリエントが明示的に設定されている必要があります。

```
file group=sys mode=0644 overlay=allow owner=root \
  path=etc/motd pkg.csize=115 pkg.size=103 preserve=true \
  variant.debug.osnet=true
```

```
file group=sys mode=0644 overlay=allow owner=root \
  path=etc/motd pkg.csize=68 pkg.size=48 preserve=true \
  variant.debug.osnet=false
```

Oracle Solaris では、次のバリエントタグが一般に使用されます。

バリエーション名	使用可能な値
variant.arch	sparc, i386
variant.opensolaris.zone	global, nonglobal
variant.debug.*	true, false

ファセットはブール型です。それらには true (有効) または false (無効) のみ設定できます。デフォルトで、イメージ内のすべてのファセットは、true に設定されていると見なされます。アクションのファセットタグの値には true のみを指定すべきであり、それ以外の値では動作が不確定になります。

イメージに設定されるファセットは、doc.man などの完全なファセットか、locale.* などのパターンになります。この柔軟性は、ファセット名前空間の一部を無効にし、その中の個々のファセットのみを有効にする場合に役立ちます。たとえば、次の例に示すように、すべてのロケールを無効にしてから、1 つか 2 つの特定のロケールのみを有効にすることができます。

```
$ pkg change-facet 'locale.*=false'
[output about packages being updated]
$ pkg change-facet locale.en_US=true
[output about packages being updated]
```

次のリストに、Oracle Solaris で使用される小さなファセットタグの例を示します。

```
facet.devel          facet.doc
facet.doc.html       facet.doc.info
facet.doc.man        facet.doc.pdf
facet.locale.de      facet.locale.en_GB
facet.locale.en_US   facet.locale.fr
facet.locale.ja_JP   facet.locale.zh_CN
```

バリエーション値の表示と変更

設定されているバリエーションの値を表示するには、pkg variant コマンドを使用します。

```
$ pkg variant
VARIANT              VALUE
variant.arch         i386
variant.opensolaris.zone global
```

-v オプションは、インストール済みパッケージに設定できるバリエーション値を示します。

```
$ pkg variant -v
VARIANT              VALUE
variant.arch         i386
variant.arch         sparc
```

```
variant.debug.osnet      false
variant.debug.osnet      true
variant.opensolaris.zone global
variant.opensolaris.zone nonglobal
```

バリエーションの値を変更するには、`pkg change-variant` コマンドを使用します。設定する値を選択するには、`pkg variant -v` コマンドを使用します。

次のコマンドは、多数のパッケージが影響を受けるため、大量の出力が生成されます。新しい BE はデフォルトで作成されませんが、バックアップ BE が作成される可能性があります。BE が作成される状況については、[100 ページの「ブート環境ポリシーイメージのプロパティ」](#)を参照してください。-n オプションを使用すると、この操作を -n なしで実行した場合に何が変更されるかが表示されますが、コマンドによって実際の変更は行われません。

```
$ pkg change-variant -nv --accept 'variant.debug.*=true'
  Packages to update:      851
  Variants/Facets to change:  3
  Estimated space available: 49.88 GB
  Estimated space to be consumed: 270.57 MB
  Create boot environment:  No
  Create backup boot environment:  Yes
  Rebuild boot archive:    No

Changed variants/facets:
  variant variant.debug.*: true
  facet facet.locale.en_US: None
  facet facet.locale.*: None
Changed packages:
solaris
...
```

ファセット値の表示と変更

`pkg change-facet` コマンドを使用してこのイメージにローカルで設定されたか親イメージから継承された、すべてのファセットの現在の値とソースを表示するには、`pkg facet` コマンドを使用します。たとえば、非大域ゾーンは大域ゾーンからのファセット設定を継承します。

```
$ pkg facet
FACETS          VALUE
facet.locale.en_US  True
facet.locale.en    True
facet.locale.*     False
```

ファセットの値を変更するには、`pkg change-facet` コマンドを使用します。

ファセットの値を `None` に設定すると、ファセットの仕様が現在のイメージから削除されます。

次のコマンドは、多数のパッケージが影響を受けるため、大量の出力が生成されます。新しい BE はデフォルトで作成されませんが、バックアップ BE が作成される可能性があります。BE が作成される状況については、100 ページの「ブート環境ポリシーイメージのプロパティ」を参照してください。-n オプションを使用すると、この操作を -n なしで実行した場合に何が変更されるかが表示されますが、コマンドによって実際の変更は行われません。

```
$ pkg change-facet -nv 'facet.locale.*=true'
    Packages to update:      851
    Variants/Facets to change:  1
    Estimated space available: 49.88 GB
    Estimated space to be consumed: 3.13 GB
    Create boot environment:    No
    Create backup boot environment: Yes
    Rebuild boot archive:      No

Changed variants/facets:
    facet facet.locale.*: True
Changed packages:
solaris
...
```

指定したバージョンへのパッケージのロック

パッケージのバージョンを制限する場合は、pkg freeze コマンドを使用します。

パッケージオペランドにバージョンを指定しない場合、名前付きのパッケージをインストールする必要があり、システムにインストールされているバージョンに制限されます。パッケージオペランドにバージョンを指定した場合、この制約（つまり、凍結）は、fmri 属性が指定されたパッケージバージョンの値を持った状態で incorporate 依存関係がインストールされているかのように機能します。

凍結されているパッケージをインストールまたは更新するときは、凍結された時点のバージョンと一致するバージョンである必要があります。たとえば、パッケージが 1.2 で凍結された場合、1.2.1、1.2.9、1.2.0.0.1 などのバージョンに更新することはできません。そのパッケージは 1.3 または 1.1 で終了することはできません。

パッケージオペランドに指定されたパブリッシャーを使用して、一致するパッケージが検索されます。ただし、パブリッシャーの情報は凍結の一環として記録されません。パッケージはパブリッシャーではなくバージョンのみに関して凍結されます。

すでに凍結されているパッケージを凍結すると、新しく指定されたバージョンによって凍結バージョンが置き換えられます。

パッケージを指定しない場合、現在凍結されているパッケージについての情報 (パッケージ名、凍結バージョン、パッケージがいつ凍結されたか、パッケージが凍結された理由) が表示されます。

パッケージを凍結しても、そのパッケージを削除できなくなるわけではありません。パッケージが削除される場合に警告は表示されません。

次の例で、パッケージは現在インストールされているバージョンで凍結されます。-c オプション引数は、パッケージが凍結される理由です。凍結が原因でインストールまたは更新に失敗する場合、その理由が示されます。パッケージのリストの「f」は、そのパッケージが凍結されていることを示します。

```
$ pkg freeze -c "Downgrade to avoid bug" library/security/openssl
library/security/openssl was frozen at 1.0.0.10-0.175.1.0.0.18.0:20120611T201116Z
$ pkg freeze
NAME                                VERSION                                DATE
COMMENT
library/security/openssl 1.0.0.10-0.175.1.0.0.19.0:20120625T171753Z 29 Jul 2012 17:45:44 PDT
Downgrade to
avoid bug
$ pkg list library/security/openssl
NAME (PUBLISHER)                VERSION                                IFO
library/security/openssl        1.0.0.10-0.175.1.0.0.18.0  if-
```

凍結されたパッケージの別のバージョンをインストールしようとすると、凍結に関するメッセージが表示されます。

```
$ pkg update library/security/openssl@1.0.0.10-0.175.1.0.0.20.0
Creating Plan (Solver setup): -
pkg update: No matching version of library/security/openssl can be installed:
  Reject: pkg://solaris/library/security/
openssl@1.0.0.10,5.11-0.175.1.0.0.20.0:20120709T180243Z
  Reason: This version is excluded by a freeze on library/security/openssl at version
  1.0.0.10,5.11-0.175.1.0.0.18.0:20120611T201116Z.
  The reason for the freeze is: Downgrade to avoid bug
```

凍結がパッケージシステムによって自動的に解除されることはありません。凍結によって適用される制約を、指定されたパッケージから削除するには、pkg unfreeze コマンドを使用します。バージョンを提供しても無視されます。

incorporation によって指定されたバージョン制約の緩和

incorporation パッケージは、インストール可能なパッケージのバージョンを指定します。これらのバージョン制約は、更新後もシステムをサポート可能な状態に保つのに役立ちま

す。incorporation パッケージと制約についての詳細は、[12 ページの「incorporation パッケージ」](#)を参照してください。

incorporation 対象のパッケージの一部は、incorporation で指定されたバージョンとは異なるバージョンでダウングレードやアップグレードを安全に行うことができる場合があります。バージョン制約は、incorporation パッケージで指定される `version-lock.package` ファセットによって表現されます。`version-lock.package` ファセットのデフォルト値は `true` です。パッケージのバージョン制約を緩和するには、その `version-lock.` ファセットの値を `false` に設定します。

次の例では、パッケージを以前のバージョンにダウングレードしようとしています。`pkg update` コマンドは、パッケージのアップグレードだけでなくダウングレードも行います。

```
$ pkg list -af library/security/openssl
NAME (PUBLISHER)          VERSION          IFO
library/security/openssl 1.0.1.5-0.175.2.0.0.24.0 i--
library/security/openssl 1.0.1.5-0.175.2.0.0.23.0 ---
$ pkg update library/security/openssl@1.0.1.5-0.175.2.0.0.23.0
Creating Plan (Solver setup): |
pkg update: No matching version of library/security/openssl can be installed:
  Reject: pkg://solaris/library/security/
openssl@1.0.1.5,5.11-0.175.2.0.0.23.0:20130916T191702Z
  Reason: This version is excluded by installed incorporation
  pkg://solaris/consolidation/userland/userland-
incorporation@0.5.11,5.11-0.175.2.0.0.24.0:20131001T160408Z
```

`pkg contents` コマンドは、このバージョン制約が設定されている方法を示します。このパッケージのバージョン制約を緩和するには、その `version-lock` ファセットを `false` に設定します。その後、ダウングレードをもう一度試みます。新しい BE は作成されませんが、バックアップ BE が作成されます。BE が作成される状況については、[100 ページの「ブート環境ポリシーイメージのプロパティ」](#)を参照してください。

```
$ pkg contents -m userland-incorporation | grep 'library/security/openssl'
depend facet.version-lock.library/security/openssl=true
fmri=pkg:/library/security/openssl@1.0.1.5-0.175.2.0.0.24.0 type=incorporate
$ pkg change-facet facet.version-lock.library/security/openssl=false
  Packages to update: 850
  Variants/Facets to change: 1
  Create boot environment: No
  Create backup boot environment: Yes

PHASE                                ITEMS
Removing old actions                  1/1
Updating image state                   Done
Creating fast lookup database          Done
Reading search index                  Done
Building new search index              850/850
```

```

$ pkg update library/security/openssl@1.0.1.5-0.175.2.0.0.23.0
    Packages to update: 1
    Create boot environment: No
    Create backup boot environment: Yes

DOWNLOAD                                PKGS      FILES    XFER (MB)   SPEED
Completed                                1/1       10/10     1.6/1.6     0B/s

PHASE                                     ITEMS
Removing old actions                      3/3
Installing new actions                    3/3
Updating modified actions                 14/14
Updating package state database           Done
Updating package cache                   1/1
Updating image state                      Done
Creating fast lookup database             Done
Reading search index                     Done
Updating search index                     1/1
$ pkg list library/security/openssl
NAME (PUBLISHER)          VERSION          IFO
library/security/openssl 1.0.1.5-0.175.2.0.0.23.0 i--

```

このパッケージのダウングレードやアップグレードを防止するには、パッケージを現在のバージョンで凍結します。パッケージのリストの「f」は、そのパッケージが凍結されていることを示します。

```

$ pkg freeze -c "Downgrade to avoid bug" library/security/openssl
library/security/openssl was frozen at 1.0.1.5,5.11-0.175.2.0.0.23.0:20130916T191702Z
$ pkg list library/security/openssl
NAME (PUBLISHER)          VERSION          IFO
library/security/openssl 1.0.1.5-0.175.2.0.0.23.0 if-

```

アップグレードやダウングレードを再度有効にするには、`pkg unfreeze` コマンドを使用してバージョンの凍結を解除します。incorporation パッケージで指定されているバージョンより低いバージョンでパッケージがインストールされている場合、このパッケージの `version-lock` フォアセットを `true` に設定すると、incorporation パッケージで指定されているバージョンがインストールされます。

ダウングレードまたはアップグレードしようとしているパッケージに対して、ほかのインストール済みパッケージが `require` 依存関係を持っている場合、そのような関連パッケージのバージョン制約の緩和も必要になることがあります。次の例では、`hexedit` パッケージのバージョン制約は解除されていますが、`system/library` パッケージのバージョン制約のためインストールは拒否されます。

```

$ pkg install editor/hexedit@1.2.12-0.175.2.0.0.25.0
Creating Plan (Solver setup): -
pkg install: No matching version of editor/hexedit can be installed:
  Reject: pkg://solaris/editor/hexedit@1.2.12-0.175.2.0.0.25.0:20131014T170634Z
  Reason: All versions matching 'require' dependency
  pkg://system/library@0.5.11,5.11-0.175.2.0.0.24.0 are rejected

```

```
Reject: pkg://solaris/system/library@0.5.11,5.11-0.175.2.0.0.24.0:20131001T152820Z

pkg://solaris/system/library@0.5.11,5.11-0.175.2.0.0.25.0:20131014T161136Z
Reason: This version is excluded by installed incorporation
pkg://solaris/consolidation/osnet/osnet-
incorporation@0.5.11,5.11-0.175.2.0.0.24.0:20131001T150429Z
```

個々のコンポーネントパッケージのほかに、incorporation のバージョン制約も緩和できます。この場合、version-lock フォアセットを false に設定すると、incorporation をシステムの残りの部分からロック解除できます。incorporation パッケージはロック解除されますが、incorporation に組み込まれているパッケージの同期は保たれます。

デフォルトのアプリケーション実装の指定

アプリケーションまたはツールの複数のバージョンが同じイメージから利用できる場合があります。ユーザーはフルパスを指定することによって、アプリケーションの各バージョンを利用できます。使いやすくするために、優先バージョンと呼ばれる 1 つのバージョンが、/usr/bin などの共通ディレクトリから利用できます。以下に示すように、すべてのバージョンが同じメディアエーションに入っている場合、優先バージョンであるバージョンを簡単に設定し直すことができます。この管理上の選択は、パッケージの更新後も保持されます。

メディアエーションとは、セット内のすべてのリンクでリンクパスが同じで、各リンクのターゲットが異なるリンクのセットです。たとえば、リンクパスが /usr/bin/myapp で、リンクのターゲットが /usr/myapp/myapp1/bin/myapp および /usr/myapp/myapp2/bin/myapp を含む場合があります。メディアエーションの各リンクは、メディアエーションの参加者と呼ばれます。/usr/bin/myapp が現在 myapp1 を呼び出す場合、/usr/bin/myapp が myapp2 を呼び出すように選択を簡単に変更することができます。現在リンクのターゲットであるソフトウェアのバージョンは、優先バージョンです。

メディアエーション内の参加者の識別

イメージ内のすべてのメディアエーション対象リンクの優先バージョンを表示するには、pkg mediator コマンドを使用します。

次の出力で、MEDIATOR は、同じ優先リンクパスを共有するリンクのセットの名前です。VER. SRC. および IMPL. SRC. は、優先バージョンが、システムによって選択されたか、割り当て済み

の優先度に応じて選択されたか (vendor または site)、管理者によって設定されたか (local) を表示します。VERSION は、選択されたメディエーション参加者のバージョンで、リンクが表すソフトウェアのバージョンに類似しているはずですが、VERSION はパッケージ開発者によって設定されます。IMPLEMENTATION は、バージョンストリングに加えて、あるいはバージョンストリングの代わりにパッケージ開発者によって設定できるストリングです。

```
$ pkg mediator
MEDIATOR  VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
gcc-runtime system  4.7   system
java      system  1.7   system
php       system  5.2   system
python    vendor  2.6   vendor
ruby      system  1.9   system
```

-a オプションは、すべてのメディエーション参加者を表します。異なる優先バージョンを選択する場合、このオプションを使用して、選択内容を表示します。次の例は、java メディエーションのすべての参加者を示しています。system キーワードは、このメディエーションの優先バージョンがパッケージの優先設定で指定されておらず、管理者によって設定されなかったことを示します。パッケージシステムは、高い VERSION 値を持つバージョンを優先バージョンとして選択しました。

```
$ pkg mediator -a java
MEDIATOR  VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      system  1.7   system
java      system  1.6   system
```

次の出力では、2 つの異なるバージョンの Java Runtime Environment がこのイメージにインストールされ、バージョン 1.7.0_51 が現在選択されている優先バージョンであることを確認できます。

```
$ pkg list -s '*jre*'
NAME (PUBLISHER)      SUMMARY
runtime/java/jre-6   Java(TM) Platform Standard Edition Runtime Environment (1.6.0_71-b12)
runtime/java/jre-7   Java Platform Standard Edition Runtime Environment (1.7.0_51-b13)
$ java -version
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b13)
Java HotSpot(TM) Server VM (build 24.51-b03, mixed mode)
```

jre-6 パッケージと jre-7 パッケージの両方に、パスが /usr/bin/java であるシンボリックリンクが定義されています。jre-6 パッケージでは、リンクのターゲットは jdk1.6.0 です。jre-7 パッケージでは、リンクのターゲットは jdk1.7.0 です。上記の pkg mediator コマンドおよび java -version コマンドによって、現在の優先バージョンはバージョン 1.7 で、/usr/bin/java リンクのターゲットであることが表示されています。

優先アプリケーションの変更

指定したメディエーションのデフォルトまたは優先のバージョンを設定し直すには、`pkg set-mediator` コマンドを使用します。

`pkg mediator -a` からの出力を使用して、`-v` 引数に対するバージョンか、`-I` 引数に対する実装を選択します。タイプミスをしたか、現在入手できないメディエータのバージョンまたは実装を指定した場合、指定されたメディエータを使用するリンクは削除されます。

バックアップ BE が作成されるかどうかを表示するには、`set-mediator` サブコマンドに `-n` オプションを使用します。バックアップ BE が作成されない場合、`set-mediator` サブコマンドに `--require-backup-be` オプションを指定することができます。メディエータの変更が現在の BE で行われます。メディエータの変更後に現在の BE に問題がないことが確認できたら、`beadm destroy` を使用してバックアップ BE を破棄できます。

前回の出力では、現在選択されている `java` メディエーションの優先バージョンがバージョン 1.7 であると表示されました。次のコマンドは、バージョン 1.6 を優先バージョンとして設定することを示し、つまり `/usr/bin/java` を呼び出すと JRE バージョン 1.6 が呼び出されることを意味します。ユーザーが JRE version 1.7 のフルパスを指定すれば、このバージョンも引き続きシステムで使用可能です。2 つの `pkg mediator` コマンドの出力を比較してください。メディエーションの優先バージョンを変更すると、`VER.` `SRC.` も `local` に変更され、選択が管理者によって指定されたことを示しています。この選択は、リポートおよびパッケージ更新後も持続します。

```
$ pkg mediator java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java     system  1.7    system
$ pkg mediator -a java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java     system  1.7    system
java     system  1.6    system
$ pkg set-mediator -v 1.6 java
    Packages to update: 3
    Mediators to change: 1
    Create boot environment: No
    Create backup boot environment: No

PHASE                                ITEMS
Removing old actions                  2/2
Updating modified actions              3/3
Updating image state                   Done
Creating fast lookup database          Done
Reading search index                  Done
Updating search index                  3/3
```

```
$ pkg mediator java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      local   1.6      system
```

この管理上の選択は、選択された実装がインストールされていなくても、パッケージの更新後も保持されます。選択された実装がインストールされていない場合、メディアエーションされたリンクのターゲットは存在しません。優先される実装をリセットするには、次のいずれかの方法を使用します。

- `pkg set-mediator` コマンドをふたたび使用して、`pkg mediator -a` によって表示された更新済みリストから別の実装を選択します。
- `pkg unset-mediator` コマンドを使用して、システムが新しい実装を選択できるようにします。

```
$ pkg unset-mediator java
```

グループパッケージに含まれる一部のパッケージのインストールの回避

指定したパッケージが `group` 依存関係のターゲットである場合に、それらのインストールを回避するには、`pkg avoid` コマンドを使用します。回避リストに含まれているパッケージでも、イメージとの互換性があるパッケージはいつでも明示的にインストールできます。回避リストに含まれているパッケージをインストールすると、そのパッケージは回避リストから削除されます。`pkg avoid` コマンドを使用すると、グループパッケージをインストールするときに、そのグループパッケージに含まれている指定したパッケージのインストールを回避できます。

引数を使用しない場合、`pkg avoid` コマンドは、回避される各パッケージとそのパッケージへのグループ依存関係のあるすべてのパッケージを表示します。

パッケージを指定した場合、`pkg avoid` コマンドは、指定されたパターンに現在一致するパッケージ名を回避リストに配置します。現在インストールされていないパッケージのみを回避できます。パッケージが現在グループ依存関係のターゲットである場合、パッケージをアンインストールするとそのパッケージは回避リストに登録されます。

回避リストに登録されているパッケージは、`require` 依存関係を満たすために必要であればインストールされ、したがって回避リストから削除されます。その `require` 依存関係が削除されると、パッケージはアンインストールされて回避リストにふたたび配置されます。

指定したパッケージを回避リストから明示的に削除する場合は、`pkg unavoid` コマンドを使用します。

回避リストに登録されており、インストール済みパッケージのグループ依存関係に一致するパッケージは、`unavoid` サブコマンドを使用して回避リストから削除できません。グループ依存性に一致するパッケージを回避リストから削除するには、パッケージをインストールします。

例 5-5 回避リストへのパッケージの追加と回避リストからのパッケージの削除

次のコマンド出力は、`group/feature/amp` グループパッケージがインストールされていないことを示しています。グループパッケージの一部であるパッケージの一部は、明示的に、またはほかのパッケージの `require` 依存関係としてインストールされたため、インストールされています。指定されたパッケージはインストールされていないため、`pkg contents` コマンドに `-r` オプションを使用します。

```
$ pkg list -a group/feature/amp
NAME (PUBLISHER)                VERSION                IFO
group/feature/amp                0.5.11-0.175.2.0.0.33.0  ---
$ pkg list -a `pkg contents -o fmri -Hrt depend -a type=group group/feature/amp`
NAME (PUBLISHER)                VERSION                IFO
database/mysql-51                5.1.37-0.175.2.0.0.34.0  ---
web/php-52                        5.2.17-0.175.2.0.0.34.0  i--
web/php-52/extension/php-apc      3.0.19-0.175.2.0.0.34.0  i--
web/php-52/extension/php-mysql    5.2.17-0.175.2.0.0.34.0  i--
web/server/apache-22             2.2.26-0.175.2.0.0.34.0  i--
web/server/apache-22/module/apache-dtrace  0.3.1-0.175.2.0.0.34.0  ---
web/server/apache-22/module/apache-fcgid  2.3.9-0.175.2.0.0.34.0  ---
web/server/apache-22/module/apache-php5   5.2.17-0.175.1.0.0.18   --r
```

次のコマンドは、まだインストールされておらずこのグループパッケージに属するパッケージの 1 つを回避リストに配置します。このグループパッケージはインストールされていないため、回避リストにグループパッケージは示されません。

```
$ pkg avoid apache-fcgid
$ pkg avoid
    web/server/apache-22/module/apache-fcgid
```

次のコマンドは、回避されたパッケージがグループパッケージのインストール時にインストールされないことを示しています。指定されたパッケージがインストール済みであるため、`pkg contents` コマンドで `-r` オプションは使用されません。

```
$ pkg install group/feature/amp
$ pkg list -a `pkg contents -o fmri -Ht depend -a type=group group/feature/amp`
NAME (PUBLISHER)                VERSION                IFO
database/mysql-51                5.1.37-0.175.2.0.0.34.0  i--
web/php-52                        5.2.17-0.175.2.0.0.34.0  i--
```

```
web/php-52/extension/php-apc          3.0.19-0.175.2.0.0.34.0  i--
web/php-52/extension/php-mysql       5.2.17-0.175.2.0.0.34.0  i--
web/server/apache-22                 2.2.26-0.175.2.0.0.34.0  i--
web/server/apache-22/module/apache-dtrace 0.3.1-0.175.2.0.0.34.0  i--
web/server/apache-22/module/apache-fcgid 2.3.9-0.175.2.0.0.34.0  ---
web/server/apache-22/module/apache-php5 5.2.17-0.175.1.0.0.18   i-r
```

グループパッケージのインストール後は、回避リストにグループパッケージが示されます。

```
$ pkg avoid
  web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

インストール済みのグループパッケージに含まれているパッケージは、`pkg unavoid` コマンドによって回避リストから削除されることはありません。そのようなパッケージを回避リストから削除するには、そのパッケージをインストールします。

```
$ pkg unavoid apache-fcgid
pkg unavaild: The following packages are a target of group dependencies; use install to unavaild these:
  web/server/apache-22/module/apache-fcgid
$ pkg install apache-fcgid
$ pkg avoid
$
```

パッケージがすでにインストールされている場合、そのパッケージを回避リストに登録することはできません。パッケージをアンインストールすると、パッケージが回避リストに登録されます。

```
$ pkg avoid apache-fcgid
pkg avoid: The following packages are already installed in this image; use uninstall to avoid these:
  web/server/apache-22/module/apache-fcgid
$ pkg uninstall apache-fcgid
$ pkg avoid
  web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

グループパッケージに含まれているパッケージをアンインストールすると、そのパッケージは自動的に回避リストに登録されます。

```
$ pkg uninstall database/mysql-51
$ pkg avoid
  database/mysql-51 (group dependency of 'group/feature/amp')
  web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

グループパッケージをアンインストールすると、回避されたパッケージは回避リストに残りますが、グループパッケージとの関連は回避リストに示されなくなります。

```
$ pkg uninstall group/feature/amp
$ pkg avoid
  database/mysql-51
  web/server/apache-22/module/apache-fcgid
$ pkg unavaild database/mysql-51 apache-fcgid
$ pkg avoid
```

\$

イメージとパブリッシャーのプロパティの構成

イメージポリシーを実装するには、イメージのプロパティを設定します。このセクションでは、イメージとパブリッシャーのプロパティと、これらのプロパティの設定方法について説明します。イメージプロパティの説明については、[pkg\(1\)](#) のマニュアルページのイメージプロパティに関するセクションも参照してください。

ブート環境ポリシーイメージのプロパティ

イメージは、IPS パッケージをインストールでき、その他の IPS 操作を実行できる場所です。ブート環境 (BE) は、イメージのブート可能なインスタンスです。システム上に複数の BE を維持することができ、各 BE にそれぞれ異なるソフトウェアバージョンをインストールすることもできます。システムをブートするとき、システム上の任意の BE にブートすることを選択できます。

パッケージ操作の結果として、新しい BE が自動的に作成されることがあります。明示的に新しい BE を作成することもできます。新しい BE が作成されるかどうかは、このセクションで説明するようにイメージポリシーに依存します。

デフォルトで、次のいずれかの操作を実行すると、新しい BE が自動的に作成されます。

- 一部のドライバやその他のカーネルコンポーネントなど、特定のキーシステムパッケージをインストールまたは更新する。重要なシステムコンポーネントは、バリエーションやファセットを変更したり、パッケージをインストール、アンインストール、および更新するときに更新できます。
- `--be-name`、`--require-new-be`、`--backup-be-name`、`--require-backup-be` のいずれかのオプションを指定する。
- `be-policy` イメージポリシーを `always-new` に設定する。このポリシーでは、次のブート時にアクティブに設定されている新しい BE で、すべてのパッケージ操作が実行されます。

新しい BE が作成される場合、システムは次の処理を実行します。

1. 現在の BE のクローンを作成します。

クローンの BE には、元の BE のメインのルートデータセットより下の階層にあるすべての要素が格納されます。共有ファイルシステムは、ルートデータセットの下にはなく、複製されません。代わりに、新しい BE は元の共有ファイルシステムにアクセスします。

2. クローン BE 内でパッケージを更新します。現在の BE 内でパッケージを更新しないでください。

現在の BE で非大域ゾーンが構成されている場合、これらの既存のゾーンが新しい BE で構成されます。

3. `--no-be-activate` が指定されていないかぎり、システムの次回ブート時に、新しい BE をデフォルトのブート選択肢に設定します。現在の BE は代替のブート選択肢として残ります。

バックアップ BE が作成される場合、システムは次の処理を実行します。

1. 現在の BE のクローンを作成します。
2. 現在の BE 内でパッケージを更新します。クローン BE 内でパッケージを更新しないでください。

新しい BE が必要でも、それを作成するための十分な空き領域がない場合、既存の不要な BE を削除できる場合があります。BE の詳細は、『[Oracle Solaris 11.2 ブート環境の作成と管理](#)』を参照してください。

次のイメージプロパティの設定方法については、[106 ページの「イメージのプロパティの設定」](#)を参照してください。

be-policy

パッケージ操作中にいつ BE が作成されるかを指定します。次の値が許可されます。

default

デフォルトの BE 作成ポリシー `create-backup` を適用します。

always-new

次のブート時にアクティブに設定されている新しい BE でパッケージ操作を実行するため、すべてのパッケージ操作に対してリブートを必要とします。明示的に要求されないかぎり、バックアップ BE は作成されません。

このポリシーはもっとも安全ですが、リポートしないとパッケージを追加できないため、ほとんどのサイトの要求よりも厳格です。

create-backup

リブートを必要とするパッケージ操作の場合、このポリシーにより、次のブート時にアクティブに設定される新しい BE が作成されます。パッケージが変更されるか、カーネルに影響する可能性のあるコンテンツがインストールされて操作がライブ BE に影響する場合、バックアップ BE は作成されますがアクティブには設定されません。バックアップ BE を明示的に要求することもできます。

このポリシーは、新しくインストールされたソフトウェアによりシステムが不安定になっている場合にのみ潜在的に危険です。この可能性はありますが、比較的まれです。

when-required

リブートを必要とするパッケージ操作の場合、このポリシーにより、次のブート時にアクティブに設定される新しい BE が作成されます。明示的に要求されないかぎり、バックアップ BE は作成されません。

ライブ BE へのパッケージングの変更によって、それ以上の変更が不可能になる場合、最新の代替 BE が存在しない可能性があるため、このポリシーには最大のリスクがあります。

パッケージの署名のプロパティ

署名付きパッケージをインストールする場合は、このセクションで説明するイメージプロパティとパブリッシャープロパティを、パッケージの署名を検証するように設定します。

署名付きパッケージのイメージプロパティ

次のイメージプロパティを、署名付きパッケージを使用するように構成します。

signature-policy

このプロパティの値により、イメージ内のパッケージのインストール、更新、変更、または検証時に、マニフェストに実行されるチェックが決まります。パッケージに適用される最終的なポリシーは、イメージポリシーとパブリッシャーポリシーの組み合わせに依存します。この組み合わせの厳格さは、少なくとも、この 2 つのポリシーが個別に適用された場合の厳格な方と同じです。デフォルトでは、パッケージクライアントは証明書が失効済みかどうかをチェックしません。そのようなチェック (クライアントから外部 Web サイトへのアクセスが必要な場合がある) を有効にするには、`check-certificate-revocation` イメージプロパティを `true` に設定します。次の値が許可されます。

ignore

すべてのマニフェストの署名を無視します。

verify

署名が含まれているすべてのマニフェストが有効に署名されていることを確認しますが、インストール済みパッケージがすべて署名されている必要はありません。

これがデフォルト値です。

require-signatures

新しくインストールされたすべてのパッケージに、有効な署名が少なくとも 1 つ含まれている必要があります。インストール済みパッケージに有効な署名が含まれていない場合は、`pkg fix` および `pkg verify` コマンドでも警告が表示されます。

require-names

require-signatures と同じ要件に従いますが、signature-required-names イメージプロパティで一覧表示される文字列が、署名の信頼のチェーンを検証するために使用される証明書の共通名としても表示される必要があります。

signature-required-names

このプロパティの値は、パッケージの署名の検証中に、証明書の共通名として表示される必要のある名前の一覧です。

署名付きパッケージのパブリッシャープロパティ

次のパブリッシャープロパティを、特定のパブリッシャーからの署名付きパッケージを使用するように構成します。

signature-policy

このプロパティの機能は、このプロパティが指定したパブリッシャーからのパッケージにのみ適用される点を除き、signature-policy イメージプロパティの機能と同じです。

signature-required-names

このプロパティの機能は、このプロパティが指定したパブリッシャーからのパッケージにのみ適用される点を除き、signature-required-names イメージプロパティの機能と同じです。

パッケージの署名プロパティの構成

パッケージの署名プロパティを構成するには、set-property、add-property-value、remove-property-value、および unset-property サブコマンドを使用します。

特定のパブリッシャーの署名ポリシーと必要な名前を指定するには、set-publisher サブコマンドの --set-property、--add-property-value、--remove-property-value、および --unset-property オプションを使用します。

次の例は、このイメージで、すべてのパッケージが署名されることを必須にするように構成します。また、この例では、信頼のチェーン内のいずれかの証明書の共通名として文字列「oracle.com」が表示されることも必須にします。

```
$ pkg set-property signature-policy require-names oracle.com
```

次の例は、このイメージで、署名されたすべてのパッケージが検証されることを必須にするように構成します。

```
$ pkg set-property signature-policy verify
```

次の例は、このイメージで、パブリッシャー `example.com` からインストールされたすべてのパッケージが署名されることを必須にするように構成します。

```
$ pkg set-publisher --set-property signature-policy=require-signatures example.com
```

次の例は、必要な署名を追加します。この例では、有効と見なされるには、署名の信頼のチェーンに表示される必要のあるイメージの共通名の一覧に文字列 `trustedname` を追加します。

```
$ pkg add-property-value signature-required-names trustedname
```

次の例は、必要な署名を削除します。この例では、有効と見なされるには、署名の信頼のチェーンに表示される必要のあるイメージの共通名の一覧から文字列 `trustedname` を削除します。

```
$ pkg remove-property-value signature-required-names trustedname
```

次の例は、指定したパブリッシャーの必要な署名を追加します。この例では、有効と見なされるには、署名の信頼のチェーンに表示される必要のあるパブリッシャー `example.com` の共通名の一覧に文字列 `trustedname` を追加します。

```
$ pkg set-publisher --add-property-value \
signature-required-names=trustedname example.com
```

追加のイメージのプロパティ

ca-path

SSL 操作の CA 証明書が格納されたディレクトリを指すパス名を指定します。このディレクトリの形式は、ベースとなる SSL 実装に固有です。信頼できる CA 証明書のために別の場所を使用するには、別のディレクトリを指すようにこの値を変更します。CA ディレクトリの要件については、`SSL_CTX_load_verify_locations(3openssl)` の `CPath` に関する項目を参照してください。

デフォルト値は `/etc/openssl/certs` です。

check-certificate-revocation

`true` に設定すると、パッケージクライアントは、署名検証のために使用される証明書の CRL 配布ポイントへのアクセスを試み、発行時よりもあとに証明書が失効していないかどうかを調べます。

デフォルト値は `False` です。

content-update-policy

パッケージシステムがパッケージ操作中に編集不能なファイルを更新する時期を指定します。次の値が許可されます。

default

デフォルトのコンテンツ更新ポリシーを常に適用します。

always

変更された編集不能なファイルを常にダウンロードおよび更新します。

when-required

更新が必要だということがパッケージシステムで決定された場合に限り、変更された編集不能なファイルをダウンロードして更新します。

デフォルト値は `always` です。

flush-content-cache-on-success

これが `true` に設定されている場合、パッケージクライアントはイメージ変更操作が正常に完了したときに内容キャッシュ内のファイルを削除します。BE を作成する操作の場合は、ソースと出力先の両方の BE から内容が削除されます。

このプロパティを使用して、ディスク容量の限られたシステムで内容キャッシュを小さく保つことができます。このプロパティを使用すると、操作が完了するまでの時間が長くなる可能性があります。

デフォルト値は `True` です。

mirror-discovery

このプロパティは、mDNS および DNS-SD を使用してリンクローカル内容ミラーを検出するようにクライアントに命令します。このプロパティを `true` に設定すると、クライアントはミラーを動的に検出し、そのミラーからパッケージ内容のダウンロードを試みます。mDNS を介してその内容を通知するミラーを実行するには、[pkg.depotd\(1M\)](#) を参照してください。

デフォルト値は `False` です。

send-uuid

ネットワーク操作の実行時にイメージの汎用一意識別子 (UUID) を送信します。ユーザーはこのオプションを無効にできますが、一部のネットワークリポジトリは UUID を供給しないクライアントとのやり取りを拒否する場合があります。

デフォルト値は `True` です。

trust-anchor-directory

このプロパティの値は、イメージのトラストアンカーを含むディレクトリのパス名です。このパスはイメージに対して相対的です。

デフォルト値は `ignore` です。

use-system-repo

このプロパティではシステムリポジトリを、イメージおよびパブリッシャーの構成のソースとして、および提供されたパブリッシャーと通信するためのプロキシとしてイメージで使用する

べきかどうかを指定します。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。

デフォルト値は ignore です。

イメージのプロパティの設定

イメージプロパティ設定を表示するには `pkg property` コマンドを使用します。イメージプロパティを構成するには、`set-property`、`add-property-value`、`remove-property-value`、および `unset-property` サブコマンドを使用します。

イメージプロパティの値の表示

イメージのプロパティを表示するには、`pkg property` コマンドを使用します。

```
$ pkg property
PROPERTY                                VALUE
be-policy                                 default
ca-path                                   /etc/openssl/certs
check-certificate-revocation             False
flush-content-cache-on-success           False
mirror-discovery                          False
preferred-authority                       solaris
publisher-search-order                   ['solaris', 'isvpub']
send-uuid                                  True
signature-policy                          verify
signature-required-names                  []
trust-anchor-directory                    etc/certs/CA
use-system-repo                            False
```

`publisher-search-order` プロパティを設定するために、`pkg set-publisher` コマンドの検索順序オプションを使用する場合があります。[81 ページの「パブリッシャーの検索順序およびスティッキネスの設定」](#)を参照してください。

イメージプロパティの値の設定

イメージプロパティの値を設定したり、プロパティを追加して設定したりするには、`pkg set-property` コマンドを使用します。

次の例では、`mirror-discovery` プロパティの値を設定します。

```
$ pkg set-property mirror-discovery true
```

```
$ pkg property -H mirror-discovery
mirror-discovery True
```

イメージプロパティの値のリセット

指定したプロパティの値をそのデフォルト値にリセットするには、`pkg unset-property` コマンドを使用します。

```
$ pkg unset-property mirror-discovery
$ pkg property -H mirror-discovery
mirror-discovery False
```

イメージの作成

イメージは、IPS パッケージとそれらの関連ファイル、ディレクトリ、リンク、および依存関係をインストールでき、その他の IPS 操作を実行できる場所です。

`pkg image-create` コマンドを使用して作成されたイメージは、ブート可能ではありません。ブート可能なイメージを作成するには、`pkg` コマンドを使用して `--be-name` または `--require-new-be` オプションを付けたり、`beadm` または `zonecfg` および `zoneadm` コマンドを使用したりします。`pkg image-create` コマンドは、パッケージおよびオペレーティングシステムのディストリビューションを維持するといったタスク向けに使用されています。

`pkg image-create` コマンドには、イメージが作成されるディレクトリとなるオペランドが必要です。作成されるイメージのデフォルトタイプはユーザーイメージです。次のいずれかのイメージタイプを指定できます。

フル フルイメージは完全なシステムを提供できます。フルイメージでは、イメージ自体の中ですべての依存関係が解決され、IPS が一貫した方法で依存関係を維持します。Oracle Solaris OS のインストールを完了した時点で、ルートファイルシステムとその内容がフルイメージに含まれています。フルイメージを指定するには、`-F` または `--full` オプションを使用します。

部分 部分イメージは、所定の `dir` パスを囲むフルイメージ（親イメージ）にリンクされます。部分イメージは、それ自体では完全なシステムを提供しません。部分イメージを指定するには、`-P` または `--partial` オプションを指定します。

非大域ゾーンは部分イメージです。非大域ゾーンコンテキストでイメージを使用するには、`-z` または `--zone` オプションを指定して、適切なバリア

ントを設定します。ゾーンイメージでは、IPS は、パッケージ内の依存関係によって定義されているとおりに、非大域ゾーンと大域ゾーンの一貫性を維持します。ゾーンについての詳細は、[17 ページの「イメージとブート環境」](#)を参照してください。

ユーザー ユーザーイメージは、再配置可能なパッケージのみを含みます。これはイメージの型を指定しない場合に作成されるデフォルト型のイメージです。ユーザーイメージを指定するには、`-U` または `--user` オプションを指定します。

パッケージリポジトリ URI を指定するには、`-p` または `--publisher` オプションを使用します。パブリッシャーの名前も提供した場合、イメージの作成時にそのパブリッシャーのみが追加されます。パブリッシャーの名前を提供しない場合、指定されたりリポジトリによって認識されているすべてのパブリッシャーがイメージに追加されます。このパブリッシャーに関連付けられたカタログは、初期作成操作に続いて取得が試みられます。

クライアント SSL 認証を使用するパブリッシャーの場合、`-c` オプションおよび `-k` オプションを使用して、クライアント鍵とクライアント証明書を登録します。この鍵と証明書は、イメージ作成中に追加されるすべてのパブリッシャーのために使用されます。

バリエーション値、ファセット値、およびイメージプロパティ値を設定するには、`--variant`、`--facet`、および `--set-property` オプションを使用します。

操作履歴の表示

`pkg` コマンドの履歴を表示するには、`pkg history` コマンドを使用します。デフォルトでは次の情報が表示されます。

- 操作の開始時間
- `install` などの操作の名前
- `pkg` などのクライアント
- 操作の結果: Succeeded または Failed

追加情報やさらに詳細な情報を表示するにはオプションを使用します。

`-l`

デフォルトの情報に加えて次の情報を表示します。

- クライアントのバージョン

- 操作を実行したユーザーの名前
- 新しい BE が作成されたかどうか
- 操作が終了した時間
- 発行された完全なコマンド
- コマンドの実行中に発生したエラー
- update などの操作について、変更されたパッケージの完全な FMRI

-n *number*

最新のものから順に指定された数の操作のみを表示します。

```
$ pkg history -n4
START                OPERATION            CLIENT              OUTCOME
2013-08-06T16:32:03  fix                  pkg                 Succeeded
2013-08-06T16:41:47  revert              pkg                 Succeeded
2013-08-06T17:56:22  set-property        pkg                 Succeeded
2013-08-06T17:56:53  unset-property      pkg                 Succeeded
```

-o *column[,column]...*

列名を指定するコンマ区切りリストを使用して出力を表示します。[pkg\(1\)](#) のマニュアルページの列名のリストを参照してください。

```
$ pkg history -o start,time,operation,outcome -n4
START                TIME                OPERATION            OUTCOME
2013-08-06T16:32:03  0:00:27            fix                  Succeeded
2013-08-06T16:41:47  0:00:43            revert              Succeeded
2013-08-06T17:56:22  0:00:00            set-property        Succeeded
2013-08-06T17:56:53  0:00:00            unset-property      Succeeded
```

-t *time | time-time[,time | time-time]...*

%Y-%m-%dT%H:%M:%S 形式のタイムスタンプのコンマ区切りリストのログレコード (`strftime(3C)` のマニュアルページを参照)。日時の範囲を指定するには、開始と終了のタイムスタンプの間にハイフン (-) を使用します。キーワード `now` は、現在の時間の別名です。指定されたタイムスタンプに重複したタイムスタンプや重なり合う日付範囲が含まれている場合、重複する履歴イベントの単一インスタンスのみが表示されます。

-N

操作に関するリリースノートテキストを表示するには、`-N` オプションを使用します。`-N` オプションを `-o` オプションとともに使用することはできません。インストールされる一部のパッケージがリリースノートを持つ場合のインストールまたは更新の操作で `-v` オプションを指定した場合、操作の出力にリリースノートが表示されます。新しい BE へのインストール操作の場合、操作の出力では現在の BE の `/tmp` 内に、リリースノートファイルへのパスが指定されます。新しい BE でブートすると、リリースノートは `/usr/share/doc/release-notes` にありますが、あるいは次のコマンドに示すように `-N` オプションを使用して、リリースノートを表示することができます。

```
$ pkg history -N -n 1
```

リリースノートをインストールした操作が、この BE で実行した最後の pkg 操作でない場合、-n 引数で使用する数値を大きくするか、次のコマンドに示すように -t オプションを使用して、リリースノートをインストールした pkg 操作を識別します。

```
$ pkg history -N -t 2013-07-17T08:31:23
```

すべてのコマンド履歴情報を削除するには、pkg purge-history コマンドを使用します。

```
$ pkg purge-history
```



パッケージのインストールおよび更新のトラブルシューティング

この付録では、パッケージをインストールまたは更新する際に見られることがある一部のエラーを処理する方法を示します。この付録では、パフォーマンスを向上し、格納されるメタデータを最小化するためのヒントも示します。

トラブルシューティングの初期手順

インストールするパッケージが構成済みのパブリッシャーから入手可能で、このイメージにインストールできるかどうかを確認します。インストール済みバージョンの `pkg:/entire` と、このイメージに構成されたパブリッシャーの起点の情報が必要です。`pkg:/entire` パッケージの更新が必要な場合もあります。パブリッシャーの起点の変更が必要な場合もあります。

パッケージのインストールにおけるほとんどすべての問題について、次の 2 つの確認を最初に行う必要があります。

- インストールされている `pkg:/entire incorporation` パッケージのバージョンを確認します。
- パッケージパブリッシャーの起点を確認します。

必要なパッケージが構成済みのパブリッシャーから入手可能であることが判別できたら、インストールの進捗に応じて次のステップを使用します。

- インストールまたは更新を行うときは常に `-nv` オプションを使用して、行われる変更内容 (どのパッケージのどのバージョンがインストールまたは更新されるのかや、新しい BE が作成されるかどうかなど) を確認してください。`-v` オプションは、この特定のインストールまたは更新操作に適用されるリリースノートも示します。
- さらに詳細なエラーメッセージを受け取るには、インストールするパッケージの FMRI を、バージョンおよびパブリッシャーも含めて詳しく指定します。

pkg:/entire のインストール済みバージョンの確認

現在インストールされている pkg:/entire incorporation パッケージのバージョンを確認するには、pkg list コマンドを使用します。

```
$ pkg list entire
NAME (PUBLISHER)                VERSION                IFO
entire                          0.5.11-0.175.2.0.0.34.0  i--
$ pkg list -Hv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z  i--
```

pkg:/entire incorporation パッケージは、使用中のサポート可能なイメージを容易に維持できるようにするために、ほかの多くのパッケージのバージョンを制約します。incorporation パッケージによって制約されるパッケージの詳細は、[12 ページの「incorporation パッケージ」](#)を参照してください。

pkg:/entire incorporation パッケージによって制約されているパッケージを直接インストールまたは更新することはできません。pkg:/entire によって制約されているパッケージをインストールまたは更新するには、pkg:/entire パッケージを更新する必要があります。詳細は、[122 ページの「制約を満たすことができない」](#)を参照してください。場合によっては、[123 ページの「incorporation によって制約されたパッケージの更新」](#)に示すように制約を削除できます。

構成済みのパブリッシャーの起点についての内容の確認

パッケージパブリッシャーの起点を確認するには、pkg publisher コマンドを使用します。

```
$ pkg publisher
PUBLISHER  TYPE    STATUS P LOCATION
solaris    origin online F http://pkg.oracle.com/solaris/release/
```

セキュアな URI の場合、必要な鍵および証明書が正しくインストールされていることを確認し、パブリッシャーを構成するときに -k および -c オプションを使用します。

外部の場所へのプロキシがサイトで必要な場合、pkg set-publisher コマンドの --proxy オプションを使用して、そのプロキシを設定します。手順については、[84 ページの「プロキシの指定」](#)を参照してください。

鍵、証明書、プロキシなどのパブリッシャーに関する詳細な情報を表示するには、pkg publisher publisher コマンドを使用します。

有効化されたパブリッシャーのいずれかの起点 URI にアクセスできない場合、必要な場所がアクセス可能であったとしても、インストールまたは更新操作は失敗します。場所にアクセスできない原因となっている問題を修復できない場合、`pkg set-publisher -G`を使用してアクセスできない起点を削除するか、`pkg set-publisher --disable`を使用して、そのパブリッシャーを無効にすることができます。このパブリッシャーが不要になった場合、`pkg unset-publisher`を使用してパブリッシャーを削除します。

パッケージパブリッシャーの起点に、必要なパッケージが含まれていることを確認します。たとえば、`solaris` パブリッシャーの起点が公開リリースリポジトリに設定されている場合、サポートリポジトリからのみ利用可能なバージョンにパッケージを更新することはできません。

必須のインストール済みパッケージが使用可能かどうかの確認

インストール済みのパッケージを更新したり、インストール済みパッケージに依存するパッケージをインストールしたり、または非大域ゾーンをインストールしたりするには、パブリッシャーの起点として設定しているリポジトリが、イメージに現在インストールされているものと同じソフトウェアを最低限含んでいる必要があります。リポジトリには古いソフトウェアまたは新しいソフトウェアを含めることもできますが、イメージにインストールされているものと同じソフトウェアを含む必要があります。

インストール済みパッケージを確認するときは、`pkgrepo list` コマンドを使用し、`pkg list` コマンドを使用しないでください。`pkg list` コマンドは、インストール済みパッケージが構成済みのパブリッシャーの起点から使用可能でない場合でも、パッケージを常に表示します。

次のコマンドは、`pkg:/entire` のインストール済みバージョンがパブリッシャーの起点から使用できないため、指定されたリポジトリはこのイメージについて適切なパブリッシャーの起点ではないことを示しています。

```
$ pkg list entire
NAME (PUBLISHER)                                VERSION                                IFO
entire                                                0.5.11-0.175.2.0.0.34.0            i--
$ pkg publisher
PUBLISHER  TYPE    STATUS P LOCATION
solaris    origin online F http://pkg.oracle.com/solaris/release/
$ pkgrepo list -Hs http://pkg.oracle.com/solaris/release entire@0.5.11-0.175.2.0.0.34.0
pkgrepo list: The following pattern(s) did not match any packages:
entire@0.5.11-0.175.2.0.0.34.0
```

必要なパッケージが一覧表示されない場合、`pkgrepo refresh` コマンドをふたたび実行して、`pkgrepo list` コマンドを再試行してください。

次のコマンドは、`pkg:/entire` のインストール済みバージョンが、指定されたりレジストリから使用できることを示しています。

```
$ pkgrepo list -Hs /export/IPSpkgrepos/Solaris entire@0.5.11-0.175.2.0.0.34.0
solaris   entire                               0.5.11,5.11-0.175.2.0.0.34.0:20140303T182643Z
```

必要なパッケージが構成済みのパブリッシャーから使用できないが、別のレジストリの起点から使用できる場合、次のいずれかのアクションを実行します。

- `solaris` パブリッシャーについてこの起点を追加するには、`pkg set-publisher` コマンドの `-g` オプションを使用します。
- `solaris` パブリッシャーについての起点を変更するには、`pkg set-publisher` コマンドの `-g` および `-G` オプションを使用します。
- 必要なパッケージが別のパブリッシャーから提供される場合、`pkg set-publisher` コマンドを使用して、そのパブリッシャーを追加します。
- 検索するレジストリのリストの最後にレジストリを一時的に追加するには、インストールコマンド (`install`、`uninstall`、`update`、`change-variant`、および `change-facet`) の `-g` オプションを使用します。

インストールするパッケージが使用可能かどうかの確認

インストールするパッケージが構成済みのパブリッシャーから入手可能かどうかを確認するには、次のコマンドを使用します。パッケージを更新する場合、現在インストールされているパッケージのバージョンと、更新後のバージョンの両方が使用可能である必要があります。

```
$ pkg list -af package
```

必要なパッケージが一覧表示されない場合、`pkg refresh` コマンドをふたたび実行して、`pkg list` コマンドを再実行してください。

引き続き必要なパッケージがリストされていない場合は、新しいパブリッシャーまたは新しいパブリッシャー起点を追加します。

インストールするパッケージがこのイメージにインストール可能かどうかの確認

`-af` オプションを使用したときに、必要なパッケージバージョンが一覧表示された場合、`-f` オプションを使用せずに同じコマンドをふたたび使用します。

```
$ pkg list -a package
```

必要なバージョンが依然として一覧表示される場合、このパッケージは制約を受けておらず、ほかのパッケージをインストールまたは更新しなくても、このパッケージをインストールできるはずで
す。

必要なバージョンが一覧表示されない場合、このバージョンは構成済みパブリッシャーから使用
できますが、このイメージにインストール可能ではありません。パッケージがインストールできない
理由として、次のことが考えられます。

- パッケージがバリエーションまたはファセットの設定によって制約されている。
- パッケージのバージョンが、incorporation パッケージによって制約されている。制
約するパッケージを更新したり、場合によっては制約を緩和したりできます。詳細
は、[122 ページの「制約を満たすことができない」](#)を参照してください。
- パッケージのバージョンが、凍結操作によって制約されている。pkg freeze コマンドを実行
します。インストールする異なるバージョンのパッケージに対して require 依存関係を持つ
パッケージに凍結が適用されている可能性があり、両方のバージョンを同時にインストール
することはできません。

インストールの再試行

インストールまたは更新するパッケージを指定し、パブリッシャーを指定しない場合、パッケージ
FMRI またはパターンに一致するパッケージを提供する、パブリッシャーの検索順序で最初の
パブリッシャーがインストールソースとして使用されます。該当するパブリッシャーがこのイメージ
内にインストール可能なパッケージのバージョンを提供しない場合、使用可能な別のパブリッ
シャーがこのイメージ内にインストール可能なパッケージのバージョンを提供したとしても、インス
トール操作は失敗します。この問題に対処するには次のいずれかのアクションを実行します。

- パッケージ FMRI でパブリッシャーを指定します。たとえば、フルパッケージ名の前に
pkg://solaris/ を指定します。
- 必要なパッケージバージョンを提供するパブリッシャーを検索順序の最初のパブリッシャーに
設定するには、pkg set-publisher コマンドの -P オプションを使用します。

インストールまたは更新を行うときは常に -nv オプションを使用して、行われる変更内容 (どの
パッケージのどのバージョンがインストールまたは更新されるのかや、新しい BE が作成される
かどうかなど) を確認してください。-v オプションは、この特定のインストールまたは更新操作
に適用されるリリースノートも示します。

- -nv オプションを使用してもエラーメッセージを受け取らない場合は、-n オプションなしでコ
マンドを再度実行して、インストールまたは更新を実際に行います。新しい BE 内でインス

ツールを実行するか、新しい BE またはバックアップ BE がデフォルトで作成されない場合にバックアップ BE を作成するためのオプションを指定するかどうかを検討します。

- エラーメッセージを受け取る場合は、次のアクションを行います。
 - 問題の診断および修正に役立つ多くの情報を取得するために、パッケージ FMRI に多くのバージョンを指定します。
 - 詳細オプション `-v` を指定します (`-nvv` など)。
 - `pkg history` コマンドを使用します。 `-l` オプションによって、変更されたパッケージの完全な FMRI が表示されます。108 ページの「[操作履歴の表示](#)」を参照してください。

インストールまたは更新する複数のパッケージを指定するか、更新操作でパッケージ指定を省略したとき、いずれかのパッケージをこのイメージにインストールできない場合はインストールまたは更新操作が失敗します。1 つのパッケージをインストールできない場合、パッケージは一切インストールされません。詳細は、インストールできないパッケージのみ指定し、そのパッケージの完全な FMRI を指定し、1 つ以上の `-v` オプションを指定して、コマンドをふたたび呼び出してください。

パブリッシャーまたはリポジトリにアクセスできない

このセクションで説明するエラーは、パブリッシャーの URI にアクセスできないことに関連しています。

パッケージリポジトリにアクセスできない

エラーメッセージ:

- `Couldn't resolve host`
- `Unable to contact any configured publishers`
- `Unable to contact valid package repository`
- `Origin URIs do not appear to point to a valid pkg repository`

パブリッシャーの URI を表示するには、`pkg publisher` コマンドを使用します。次の例の LOCATION 列または URI 行を参照してください。

```
$ pkg publisher
PUBLISHER          TYPE    STATUS P LOCATION
solaris            origin  online F http://pkg.oracle.com/solaris/release/
$ pkg publisher solaris
```

```
Publisher: solaris
Alias:
Origin URI: http://pkg.oracle.com/solaris/release/
```

使用しなくなったパブリッシャーが一覧表示される場合、これらのパブリッシャーを無効にするか、削除します。

```
$ pkg set-publisher --disable publisher
$ pkg unset-publisher publisher
```

使用しているパブリッシャーについては、URI が正しいことを確認してください。

- 各パブリッシャーの起点の場所をブラウザで表示するか、その場所に ping を実行してみてください。
- `pkgrepo list` コマンドを使用して、その起点のパッケージを一覧表示してみてください。

起点の場所が正しくない場合、`pkg set-publisher` コマンドの `-G` および `-g` オプションと一緒に使用して、URI を変更します。

パブリッシャーに複数の起点がある場合は、すべての起点の場所にアクセスできることが必要です。いずれかの起点の場所にアクセスできない場合、`pkg set-publisher` コマンドの `-G` オプションを使用して、その起点を削除します。

パブリッシャーが非大域ゾーン内で構成された場合、そのパブリッシャーが大域ゾーン内で構成されていなくても、そのパブリッシャーについてのすべての場所が大域ゾーンからアクセス可能である必要があります。

SSL 証明書の問題

エラーメッセージ: SSL certificate problem, verify that the CA cert is OK

`pkg` コマンドによって、SSL 証明書の問題に関するメッセージが表示される場合、次の 1 つ以上のアクションを実行してください。

- 必要な鍵および証明書がインストールされていることを確認します。鍵および証明書を取得する方法についての情報は、セキュアリポジトリに関するほかの情報に含まれています。
- `pkg set-publisher` コマンドで `-k` および `-c` オプションを使用して、鍵および証明書がインストールされる場所を指定します。パブリッシャーに対する `pkg publisher` コマンドは、パブリッシャーによって鍵および証明書が検出されることが予想される場所を表示します。鍵および証明書がほかの場所にインストールされている場合、`-k` および `-c` オプションに正しい値を指定することによって、パブリッシャーを再構成します。

次の例の出力には、鍵および証明書のファイルパスが表示され、abc パブリッシャーについて構成された起点の証明書が期限切れであることが表示されています。

```
$ pkg publisher abc
Publisher: abc
Origin URI: https://pkg.oracle.com/abc/release/
  SSL Key: /var/pkg/ssl/keyfile
  SSL Cert: /var/pkg/ssl/certfile
Certificate '/var/pkg/ssl/certfile' has expired.
Please install a valid certificate.
```

- システムの時間と日付が正しいことを確認してください。
- ca-certificates SMF サービスをリフレッシュします。

```
$ svcadm refresh svc:/system/ca-certificates:default
```

- 起点の URI に oc-mgmt が含まれている場合、Ops Center サポート担当者にお問い合わせください。

注記 - クライアント SSL 証明書が必要なリポジトリは、pkg インストールコマンドに -g オプションを使用して指定することができません。

場所が見つかりません

エラーメッセージ: http protocol error: code: 404 reason: Not Found

116 ページの「パッケージリポジトリにアクセスできない」の説明に従って、パブリッシャー URI を確認します。場所を表示したり ping を実行したりできる場合、pkgrepo list コマンドを使用して、リポジトリ内のいずれかのパッケージを表示してみてください。

URI がファイルベースのディレクトリの場合、ファイルおよびディレクトリが pkg5srv ユーザーによって読み取り可能であることを確認してください。pkgrepo verify コマンドを使用して、リポジトリが pkg5srv ユーザーによって読み取り可能かどうかを確認できます。

Web サーバー構成を確認します。詳細は、『Oracle Solaris 11.2 パッケージリポジトリのコピーと作成』の第 5 章「Web サーバーの背後での集積サーバーの実行」を参照してください。

- Apache Web サーバーインスタンスの背後にパッケージデポサーバーを実行している場合、エンコードされたスラッシュをデコードしないように、次の設定を httpd.conf ファイルに含めます。

```
AllowEncodedSlashes NoDecode
```

- デポサーバー pkg/proxy_base を、Apache サーバー上のリポジトリの URL に設定します。

```
$ svccfg -s pkg/server:repo setprop pkg/proxy_base = astring: http://pkg.example.com/
myrepo
$ svcadm refresh pkg/server:repo
```

非大域ゾーンで問題が発生する場合、次のトラブルシューティングステップを実行します。非大域ゾーンはシステムリポジトリと呼ばれる特殊なパッケージリポジトリを使用します。システムリポジトリの詳細は、[pkg.sysrepo\(1M\)](#) のマニュアルページを参照してください。

- 非大域ゾーンを持つイメージ内で `-g` オプションを使用することはできません。代わりに、`pkg set-publisher` コマンドを使用して、そのパブリッシャーおよび起点を明示的に追加します。
- ファイルベースリポジトリ内のファイルおよびディレクトリが `pkg5srv` ユーザーによって読み取り可能であることを確認してください。`pkg5srv` ユーザーは `system-repository` Apache インスタンスを実行します。システムリポジトリの場所を見つける方法を示す例については、[57 ページの「大域ゾーンと非大域ゾーンの関係」](#)を参照してください。
- 外部の場所にアクセスするためにサイトでプロキシが必要な場合、大域ゾーンのパブリッシャーに対してプロキシが正しく指定されていることを確認します。プロキシを指定するには、`pkg set-publisher` コマンドの `--proxy` オプションを使用します。手順については、[84 ページの「プロキシの指定」](#)を参照してください。プロキシの確認方法の 1 つとして、`pkg refresh --full` コマンドからアクセスエラーメッセージを受け取らないことを確認する方法があります。
- サービス `svc:/application/pkg/system-repository:default` が大域ゾーンでオンラインであることを確認します。
- サービス `svc:/application/pkg/zones-proxyd:default` が大域ゾーンでオンラインであること、およびサービス `svc:/application/pkg/zones-proxy-client:default` が非大域ゾーンでオンラインであることを確認します。
- 大域ゾーンで、`/var/log/pkg/sysrepo/*` 内のログファイルを調べ、ファイルの読み取りを試行するときに権限エラーが報告されていないかどうかを確認します。404 または 503 エラーが `/var/log/pkg/sysrepo/access_log` に報告されていないか確認します。`/var/log/pkg/sysrepo/error_log` にエラーが報告されていないか確認します。
- 大域ゾーンにおいて、`/etc/hosts` ファイルで `localhost` が `127.0.0.1` に設定されていることを確認します。`/system/volatile/pkg/sysrepo/sysrepo_httpd.conf` ファイルで、`Listen` が `127.0.0.1:1008` に設定され、`ServerName` が `127.0.0.1` に設定されていることを確認します。

- 大域ゾーンで、ファイル `/system/volatile/pkg/sysrepo/sysrepo_httpd.conf` が次の形式の `Alias` 行を含むかどうかを確認します。

```
$ grep Alias /system/volatile/pkg/sysrepo/sysrepo_httpd.conf
WSGIScriptAlias /wsgi_p5p /etc/pkg/sysrepo/sysrepo_p5p.py
```

`sysrepo_httpd.conf` ファイルに `Alias` 行がない場合、`sysrepo` サービスを再起動します。

```
$ svcadm restart svc:/application/pkg/system-repository:default
```

サービスが使用不可

エラーメッセージ: `http protocol error: code: 503 reason: Service Unavailable`

`pkg publisher` コマンドを使用して、使用しようとしているパッケージリポジトリの場所を検出し、そのシステム上の SMF サービスを調べます。次のコマンドを使用して、使用可能であるが実行中でないパッケージリポジトリ SMF サービスインスタンスと、使用可能な別のインスタンスの実行を妨げているインスタンスを識別します。

```
$ svcs -xv pkg/server
svc:/application/pkg/server: default (image packaging repository)
  State: online since July 25, 2013 07:53:50 AM PDT
    See: /var/svc/log/application-pkg-server:default.log
  Impact: None.
```

いずれかのサービスが問題を報告している場合、`svcs` 出力に一覧表示されるログファイルを確認して、特定の問題を判別します。

`inst_root` プロパティ、`port` プロパティ、およびその他のプロパティが正しく設定されていることを確認します。

```
$ svcprop -p pkg pkg/server:default
$ svcprop -p pkg/inst_root -p pkg/port pkg/server:default
/export/IPSpkgrepos/Solaris
80
```

必要な場合、次の例に示すように、`svccfg` コマンドを使用してプロパティ値をリセットします。

```
$ svccfg -s pkg/server:default setprop pkg/port=1008
```

必要に応じてサービスインスタンスをクリア、リフレッシュ、再起動、および使用可能にするには、`svcadm` コマンドを使用します。

使用可能な更新がない

エラーメッセージ: No updates available for this image

特定のパッケージを更新する場合、次のコマンドを使用して、このイメージに現在インストールされているパッケージのバージョンを表示します。すべてのインストール済みパッケージを更新する場合 (pkg update でパッケージを指定しないか、パッケージ名として '*' を指定)、これらのコマンドの *package* に `pkg:/entire` を使用します。

```
$ pkg list -v package
```

現在インストールされているバージョンが存在しない場合、pkg update ではなく pkg install コマンドを使用してください。

あるバージョンの *package* が現在インストールされている場合、次のコマンドを使用して、構成済みのパブリッシャーから使用できる *package* のバージョンを表示します。

```
$ pkg list -afv package
```

最上位のバージョン番号を持つパッケージがすでにインストールされている場合、それより新しいバージョンはおそらく存在しません。

新しいバージョンが存在する場合、新しいバージョンが使用可能なパッケージリポジトリの場所を判別し、pkg set-publisher コマンドを使用して起点 URI をリセットするか、適切なパブリッシャーの起点 URI を追加します。必要な場合、必要な鍵および証明書をインストールし、-k および -c オプションを使用して、これらを指定します。pkgrepo list コマンドを使用して、現在インストールされているバージョンのパッケージが、構成済みのパブリッシャーからも利用できることを確認します。

次の例に示すように、インストールするパッケージの FMRI にバージョン (またはキーワード latest) を含めて -nv オプションを指定して、pkg update コマンドを再実行します。パッケージ名に指定する情報が多くなれば、エラー出力の情報も多くなります。

```
$ pkg update -nv package@latest
```

パッケージをインストールできない

エラーメッセージ: No matching version of *package* can be installed

次のコマンドを使用して、構成済みのパッケージパブリッシャーから使用できる *package* のバージョンを表示します。

```
$ pkg list -afv package
```

インストールするパッケージの FMRI を詳細に指定します。最初に検出された一致はこのイメージにインストール可能でないかもしれませんが、インストールする特定のバージョンはインストール可能な場合があります。FMRI をさらに特定しても依然としてインストール可能でない場合、FMRI を詳しく指定すると、パッケージをインストールできない理由が詳しく表示されます。

インストールまたは更新するパッケージのバージョンが凍結されていないことを確認してください。バージョンが凍結されたすべてのパッケージを一覧表示するには、引数を指定せずに `pkg freeze` コマンドを使用します。

インストールされている `pkg:/entire` パッケージのバージョンを表示するには、次のコマンドを使用します。

```
$ pkg list -v entire
```

`pkg:/entire incorporation` パッケージまたはほかの `incorporation` パッケージによって制約されているパッケージをインストールまたは更新することはできません。`incorporation` パッケージを更新する必要があります。詳細は、[122 ページの「制約を満たすことができない」](#)を参照してください。

制約を満たすことができない

エラーメッセージ: No solution was found to satisfy constraints

このメッセージは、あるバージョンのパッケージをインストールしようとしたが、そのパッケージのバージョンが、`incorporation` パッケージによる制約で指定されたパッケージのバージョンと一致しないことを示しています。`incorporation` パッケージと制約については、[12 ページの「incorporation パッケージ」](#)を参照してください。

`Incorporation` によって、パッケージのセットは一緒に機能するバージョンに制約されるため、サポート可能なイメージを保持するのに役立ちます。このため、`incorporation` によって制約される 1 つのパッケージを更新しないでください。代わりに、`incorporation` パッケージを更新する必要があり、この結果、制約されたすべてのパッケージは、一緒にテストされた新しいバージョンのセットに更新されます。

パッケージが変更される時 (たとえば更新される時)、pkg クライアントは関連するパッケージとそれらの依存関係を調べます。いずれかの依存パッケージをインストールまたは更新できない場合、インストールまたは更新できないパッケージに依存するパッケージごとに、個別のエラーメッセージが生成されます。大量のエラーメッセージを処理するもっとも効果的な方法は、もっともインデントされたエラーメッセージを最初に調べるやり方です。

incorporation によって制約されたパッケージの更新

エラーメッセージ:

- No suitable version of installed package *package* found
- All versions matching 'incorporate' dependency *package* are rejected
- This version excluded by specified installation version
- This version is excluded by installed incorporation

incorporation によって制約されたパッケージの場合、incorporation パッケージを更新し、制約されたすべてのパッケージを、一緒にテストされたセットとして保持することがベストプラクティスです。

incorporation からの 1 つのパッケージのみ更新する場合、パッケージの `version-lock` フォアセットが `true` に設定されているかどうかを確認します。パッケージに `version-lock` フォアセットが関連付けられている場合、そのパッケージを incorporation からロック解除できません。`version-lock` フォアセットを `false` に設定して制約を除去し、その後パッケージのインストールまたは更新を再試行します。`-nv` オプションと、パッケージ名の FMRI に必要なパッケージのバージョンを指定します。[91 ページの「incorporation によって指定されたバージョン制約の緩和」](#)も参照してください。

例 A-1 Java Runtime Environment のロック解除と更新

次の例は、`runtime/java/jre-7` パッケージを更新する方法を示しています。`jre-7` パッケージは `consolidation/java/java-incorporation` パッケージによって制約され、`java-incorporation` パッケージは同様に `pkg:/entire incorporation` パッケージによって制約されます。

次のコマンドは、`0.175.2.0.0.9.0` バージョンの `jre-7` が現在インストールされており、構成済みのパッケージリポジトリから新しいバージョンが使用可能であることを示しています。

```
$ pkg list -af runtime/java/jre-7
NAME (PUBLISHER)                VERSION                IFO
runtime/java/jre-7              1.7.0.21-0.175.2.0.0.13.0  ---
runtime/java/jre-7              1.7.0.17-0.175.2.0.0.9.0   i--
```

-f オプションを削除すると、可能な更新後のバージョンが表示されます。次の pkg list 出力には、このイメージにインストール可能な新規バージョンがないことが示され、pkg update コマンド出力でこの状態が確認できます。-n オプションは、実行される変更を示しますが、変更を一切加えません。

```
$ pkg list -a runtime/java/jre-7
NAME (PUBLISHER)                VERSION                IFO
runtime/java/jre-7              1.7.0.17-0.175.2.0.0.9.0   i--
$ pkg update -nv runtime/java/jre-7
No updates available for this image.
```

このパッケージを更新できない理由を詳細に示すには、更新後のバージョンを指定します。次の例に示す出力では、インストール済みの java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0 パッケージは、jre-7@1.7.0.21-0.175.2.0.0.13.0 パッケージのインストールを許可しないことを示しています。java-incorporation@0.5.11,5.11-0.175.2.0.0.13.0 パッケージは、jre-7@1.7.0.21-0.175.2.0.0.13.0 パッケージのインストールを許可しますが、インストール済みの entire@0.5.11,5.11-0.175.2.0.0.12.0 incorporation パッケージは jre-7@1.7.0.21-0.175.2.0.0.13.0 パッケージのインストールを許可しません。

```
$ pkg update -nv runtime/java/jre-7@1.7.0.21-0.175.2.0.0.13.0
pkg update: No solution was found to satisfy constraints

maintained incorporations:
  [output omitted]
  pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z

Plan Creation: dependency error(s) in proposed packages:
  [output omitted]
  No suitable version of required package pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z found:
    Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z
    Reason: All versions matching 'incorporate' dependency pkg://runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0 are rejected
    Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z
    Reason: This version excluded by specified installation version
    Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z
    Reason: This version excluded by specified installation version
    Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145534Z
    Reason: This version is excluded by installed incorporation pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z
```

```

Plan Creation: Errors in installed packages due to proposed changes:
  [output omitted]
  No suitable version of installed package pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z found
    Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z
    Reason: All versions matching 'incorporate' dependency pkg://runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0 are rejected
    Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z
    Reason: This version excluded by specified installation version
    Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145534Z
    Reason: This version is excluded by installed incorporation pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z

```

ベストプラクティスは、entire パッケージを更新することです。entire パッケージを更新すると、java-incorporation パッケージが更新され、その結果として jre-7 パッケージが更新されます。この例では、Java パッケージを更新する必要があり、イメージを entire の更新済みバージョンに変更することはできません。

インストール可能な Java ソフトウェアのバージョンは、Java incorporation パッケージの version-lock ファセットを設定することによって制約されます。ほかのソフトウェアを更新せずに Java ソフトウェアを更新するには、Java incorporation パッケージの version-lock ファセットをロック解除し、そのあとで Java incorporation パッケージを更新します。version-lock ファセットの詳細は、[91 ページの「incorporation によって指定されたバージョン制約の緩和」](#)を参照してください。

次のコマンドによって、インストール済みの java-incorporation パッケージの version-lock ファセットの値が false に変更されます。イメージ内の各パッケージがこのファセットに対して検査されるため、更新するパッケージ数は、このイメージにインストールされているパッケージの数です。

```

$ pkg change-facet \
facet.version-lock.consolidation/java/java-incorporation=false
  Packages to update: 856
  Variants/Facets to change: 1
  Create boot environment: No
  Create backup boot environment: Yes

Planning linked: 1/1 done
PHASE                                ITEMS
Removing old actions                  1/1
Updating image state                  Done
Creating fast lookup database         Done
Reading search index                  Done

```

Building new search index

856/856

次のコマンドは、ファセット値が変更されたことを示しています。

```
$ pkg facet
FACETS                                     VALUE
facet.version-lock.consolidation/java/java-incorporation False
```

次のコマンドでは -n オプションが指定されているため、このコマンドによって変更される内容が示されますが、イメージを実際に変更するわけではありません。

```
$ pkg update -nv java-incorporation
Packages to update:      2
Estimated space available: 80.91 GB
Estimated space to be consumed: 687.28 MB
Create boot environment: No
Create backup boot environment: Yes
Rebuild boot archive:   No
```

Changed packages:

```
solaris
consolidation/java/java-incorporation
  0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z -> 0.5.11,5.11-0.175.2.0.0.13.0
:20130429T145534Z
runtime/java/jre-7
  1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z -> 1.7.0.21,5.11-0.175.2.0.0.
13.0:20130429T145626Z
```

次のコマンドは実際の更新を実行します。このコマンドは、現在のイメージ内で更新を実行します。新しいブート環境で更新を実行するために、--be-name オプションを使用する場合もあります。

```
$ pkg update -v java-incorporation
Packages to update:      2
Estimated space available: 80.91 GB
Estimated space to be consumed: 687.28 MB
Create boot environment: No
Create backup boot environment: Yes
Rebuild boot archive:   No
```

Changed packages:

```
solaris
consolidation/java/java-incorporation
  0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z -> 0.5.11,5.11-0.175.2.0.0.13.0
:20130429T145534Z
runtime/java/jre-7
  1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z -> 1.7.0.21,5.11-0.175.2.0.0.
13.0:20130429T145626Z
```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
Completed	2/2	171/171	61.9/61.9	0B/s

PHASE

ITEMS

```

Removing old actions          7/7
Installing new actions        6/6
Updating modified actions     170/170
Updating package state database Done
Updating package cache        2/2
Updating image state          Done
Creating fast lookup database Done
Reading search index          Done
Updating search index         2/2

```

次のコマンドは、このイメージ内で jre-7 パッケージが更新されたことを確認しています。新しいブート環境で更新を実行する場合、beadm mount および pkg -R を使用して、新しいブート環境で次の検査を行います。

```

$ pkg list jre-7
NAME (PUBLISHER)          VERSION          IFO
runtime/java/jre-7       1.7.0.21-0.175.2.0.0.13.0  i--

```

適切な依存関係が見つからないときの Incorporation の更新

エラーメッセージ: A version for 'incorporate' dependency cannot be found

incorporation パッケージとそれらの incorporate 依存関係については、[12 ページの「incorporation パッケージ」](#)を参照してください。

incorporation パッケージの更新に失敗することがある理由の例として、incorporation パッケージの incorporate 依存関係のインストール状態が、次のようになっていることがあります。

- 依存パッケージが異なるバージョンで凍結されている。
- 依存パッケージが高いバージョンですでにインストールされている。
- 依存パッケージが別のパブリッシャーからインストールされており、そのパブリッシャーが固定である。

例 A-2 依存関係がロック解除されていて別個に更新される場合の pkg:/entire の更新

次の例ではパッケージ名が指定されていないため、インストール済みのすべてのパッケージの更新が試行されます。この操作で更新を試行するインストール済みパッケージの 1 つが、pkg:/entire incorporation パッケージです。この例は、すでに高いバージョンでインストールされている pkg:/entire の incorporate 依存関係を示しています。

```
$ pkg update --be-name s11.2
```

```
Creating Plan (Solver setup): /
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to latest available versions.
This may indicate an overly constrained set of packages are installed.
```

```
latest incorporations:
[output omitted]
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
```

The following indicates why the system cannot update to the latest version:

```
No suitable version of required package pkg://solaris/entire@0.5.11,5.11-0.175.2
.0.0.10.0:20130318T181506Z found:
Reject: pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
Reason: A version for 'incorporate' dependency on pkg:/consolidation/ub_javavm
/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0 cannot be found
```

このメッセージでは、システムが更新対象としていた pkg:/entire incorporation パッケージのバージョンが、インストールできない ub_javavm-incorporation パッケージのバージョンを指定しています。1 つのパッケージをインストールできないため、パッケージは一切インストールされず、更新は失敗します。

次の方法を使用すれば、ub_javavm-incorporation パッケージをインストールできない理由の詳細を提供できます。

- -v オプションを使用します。たとえば、詳細な出力を受け取るために、-v または -vv を使用します。
- 更新するパッケージを指定します。一般的に、詳細な情報を入力すると、詳細なメッセージが表示されます。たとえば、パッケージ FMRI に、パッケージ名に加えてバージョンを指定します。

次のコマンドでは -v オプションを指定し、上記のメッセージからコピーされた entire@0.5.11,5.11-0.175.2.0.0.10.0 パッケージへの更新が指定されています。このコマンドでは、--be-name オプションの代わりに -n オプションが指定されています。-n オプションは、実行される内容を表示しますが、このイメージを実際に変更するわけではありません。

```
$ pkg update -nv entire@0.5.11,5.11-0.175.2.0.0.10.0
Creating Plan (Solver setup): /
pkg update: No matching version of entire can be installed:
Reject: pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
Reason: All versions matching 'require' dependency pkg:/consolidation/ub_javavm/ub_javavm-incorporation are rejected
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.151.0.1:20101105T053418Z
pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0.0.0.2.0:20111019T144756Z
pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.17
```

```

5.0.10.1.0.0:20120920T143020Z
Reason: Excluded by proposed incorporation 'entire'
Newer version pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.
5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z is already installed
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,
5.11-0.175.2.0.0.9.0:20130304T213739Z
Reason: Newer version pkg://solaris/consolidation/ub_javavm/ub_javavm-incorpo
ration@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z is already installed
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5
.11-0.175.2.0.0.13.0:20130429T145201Z
Reason: Excluded by proposed incorporation 'entire'

```

これらのメッセージは、このイメージに現在インストールされている `ub_javavm-incorporation` パッケージのバージョンが、更新操作によってインストールされる `pkg:/entire incorporation` パッケージによって指定されたバージョンよりも新しいことを示しています。

次のコマンドは、現在インストールされている `ub_javavm-incorporation` パッケージのバージョンを表示します。

```

$ pkg list ub_javavm-incorporation
NAME (PUBLISHER)                                VERSION                                IFO
consolidation/ub_javavm/ub_javavm-incorporation 0.5.11-0.175.2.0.0.13.0             i--

```

次のコマンドは、システムが更新対象として指定しようとした `pkg:/entire incorporation` パッケージによって指定されている `ub_javavm-incorporation` パッケージのバージョンを示しています。システムが更新対象として指定しようとした `pkg:/entire incorporation` パッケージのバージョンは、上記の出力の先頭にある「Reject」メッセージからのコピーです。-r オプションは、インストール済みイメージではなく構成済みのパッケージリポジトリからこのパッケージを検索します。

```

$ pkg contents -Hrt depend \
-a facet.version-lock.consolidation/ub_javavm/ub_javavm-incorporation=true \
-o fmri entire@0.5.11,5.11-0.175.2.0.0.10.0
consolidation/ub_javavm/ub_javavm-incorporation@0.5.11-0.175.2.0.0.9.0

```

この問題を修正するために、指定された特定のパッケージを除くすべてのインストール済みパッケージを更新するように更新操作に指示できます。`pkg update` コマンドで 1 つ以上の `--reject` オプションを使用して、`--reject` オプションで指定されたパッケージの更新を試行せずに更新を実行します。`--reject` 引数にはワイルドカードを使用できます。次のコマンドで、拒否するパッケージは上記の「Reason: Newer version is already installed」メッセージの前にある、「Reject」メッセージからコピーされたものです。

```

$ pkg update -v --be-name s11.2 \
--reject 'consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.151*' \
--reject 'consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0*' \
--reject 'consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0*'

```

ヒント ---reject 引数を指定するときは注意が必要です。すでにインストール済みで、--reject 引数に一致するすべてのパッケージはアンインストールされます。

インストール済みの依存関係が許容されないときの Incorporation の更新

エラーメッセージ:

- The installed package *package* is not permissible
- Excluded by proposed incorporation

これらのエラーメッセージを受け取った場合、*package* はおそらく、ほかのコアのオペレーティングシステムパッケージと同期して保持する必要があるコアのオペレーティングシステムパッケージです。pkg facet コマンドを使用して、facet.version-lock.*package* ファセットの値を確認してください。この version-lock ファセットの値が false の場合、pkg change-facet コマンドを使用して、このファセットの値を true に変更し、更新操作を再試行してください。

必要なパッケージが見つからない

エラーメッセージ: A version for 'require' dependency cannot be found

必要なパッケージが見つからないという次のメッセージのようなメッセージを受け取る場合、構成済みのパブリッシャーからパッケージが使用可能であるかどうかを確認します。

```
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to
latest available versions.
This may indicate an overly constrained set of packages are installed.
[output omitted]
No suitable version of required package package1 found:
Reject: package1
Reason: A version for 'require' dependency package2 cannot be found
```

package2 が構成済みのパブリッシャーから使用可能であるかどうかを表示するには、次のコマンドを使用します。

```
$ pkg list -afv package2
```

パブリッシャーの起点の場所を確認するには、pkg publisher コマンドを使用します。このパッケージのパブリッシャーは、完全な FMRI の pkg:// の後に示されます。パブリッシャーの起点

の場所を変更することが必要な場合もあります。場所がローカルパッケージリポジトリのとき、リポジトリの更新が必要な場合もあります。

必要なパッケージが拒否された

エラーメッセージ:

- No solution was found to satisfy constraints
- All versions matching 'require' dependency *package* are rejected

pkg update コマンドからの次の出力で、最初のエラーメッセージでは、必要なパッケージ desktop-incorporation の適切なバージョンが見つからないことが示されています。desktop-incorporation パッケージが適切でないのは、そのいずれかの依存パッケージが見つからないためです。適切なバージョンが見つからないため、desktop-incorporation パッケージは拒否されます。desktop-incorporation パッケージが pkg:/entire incorporation によって必要とされているため、pkg update は失敗します。次のコマンドは、desktop-incorporation パッケージが pkg:/entire incorporation によって必要とされていることが示されています。

```
$ pkg search -Hlo pkg.name require:consolidation/desktop/desktop-incorporation
```

その次のエラーメッセージでは、必要なパッケージ python-extra-26 の適切なバージョンが見つからないことが示されています。python-extra-26 パッケージが適切でない理由は、python-extra-26 パッケージには desktop-incorporation パッケージが必要ですが、desktop-incorporation の適切なバージョンが見つからなかったためです。

この情報により、この更新の失敗の解決策は、desktop-incorporation パッケージの必要なバージョンをインストールする方法を見つけることであるということがわかります。

```
pkg update: No solution was found to satisfy constraints
```

```
maintained incorporations:  
[output omitted]
```

```
Plan Creation: dependency error(s) in proposed packages:  
[output omitted]
```

```
No suitable version of required package pkg://solaris/consolidation/desktop  
/desktop-incorporation@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z found:  
  Reject: pkg://solaris/consolidation/desktop/desktop-incorporation  
@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z  
  Reason: A version for 'incorporate' dependency on pkg:/library/python-2  
/python-sexy-26@0.1.9-0.175.0.0.0.1.0 cannot be found  
No suitable version of required package pkg://solaris/library/python-2
```

```
/python-extra-26@2.6.4-0.175.1.0.0.15.0:201205014T200156Z found:
  Reject: pkg://solaris/library/python-2/python-extra-26@
2.6.4-0.175.1.0.0.15.0:201205014T200156Z
  Reason: All versions matching 'require' dependency pkg:/consolidation
/desktop/desktop-incorporation are rejected
  Reject: pkg://solaris/consolidation/desktop/desktop-incorporation
@0.5.11,5.11-0.175.0.0.0.2.0:20111019T132128Z
  [output omitted]
pkg://solaris/consolidation/desktop/desktop-incorporation
@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z
```

次のコマンドは、必要な `desktop-incorporation` パッケージについて見つからないパッケージが、必要ではないことを示しています。`python-sexy-26` パッケージは `desktop-incorporation` パッケージによる `incorporation` 対象となっていますが、必要ではありません。

```
$ pkg search -Hlo pkg.name require:library/python-2/python-sexy-26
$ pkg search -Hlo pkg.name incorporate:library/python-2/python-sexy-26
consolidation/desktop/desktop-incorporation
```

`incorporate` 依存関係は、`python-sexy-26` パッケージがインストールされる場合、指定されたバージョンでインストールされなければならないことを示しています。ただし、`python-sexy-26` パッケージを必要とするパッケージが存在しないため、`python-sexy-26` パッケージをインストールする必要はありません。したがって、この更新の失敗の 1 つの解決策は、`python-sexy-26` パッケージをアンインストールすることです。このパッケージの別のバージョンが現在インストールされており、更新では `desktop-incorporation` `incorporation` によって指定されたバージョンが見つかりませんでした。パッケージをアンインストールすると、更新処理では更新されたパッケージを見つける必要がなくなります。

`python-sexy-26` パッケージが必要であってアンインストールすることを望まない場合、`pkg:/library/python-2/python-sexy-26@0.1.9-0.175.0.0.0.1.0` を提供するパッケージリポジトリを検索します。`pkg set-publisher` コマンドを使用して、そのリポジトリをパブリッシャーの起点の場所に追加するか、`pkgrecv` コマンドを使用して、そのパッケージを現在設定されているパブリッシャーの起点に追加します。

パッケージが期待どおりに更新されない

エラーメッセージ: `pkg update: The installed package package is not permissible.`

更新操作でワイルドカードを使用するとき、更新されるはずの一部のパッケージが更新されない場合にエラーメッセージがまったく表示されないことがあります。ワイルドカードを使用せずにパッケージ名を指定すると、エラーメッセージが表示される場合があります。

たとえば、新しいパッケージが構成済みのパブリッシャーから使用可能であることが確認されたにもかかわらず、次の操作がエラーを出さずに完了し、すべてのパッケージを更新しないことがあります。

```
$ pkg update '*'
```

詳細な情報を取得するには、ワイルドカードを使用する代わりに、更新後の `pkg:/entire` のバージョンを指定します。

```
$ pkg list -Hafv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.9.0:20130304T214506Z ---
$ pkg update -nv pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.9.0:20130304T214506Z
Creating Plan (Solver setup): -
pkg update: The installed package compress/zip is not permissible.
  Reject:  pkg://solaris/compress/zip@3.0,5.11-0.175.2.0.0.7.0:20121119T070339Z
  Reason:  Excluded by proposed incorporation 'consolidation/userland/userland-incorporation'
```

この例で、パッケージ `compress/zip` は `userland-incorporation incorporation` パッケージの更新を妨げ、後者のパッケージは `pkg:/entire incorporation` の更新を妨げています。次のコマンドは、`compress/zip` パッケージの詳細情報を表示します。

```
$ pkg list compress/zip
NAME (PUBLISHER)  VERSION                IFO
compress/zip     3.0-5.11-0.175.2.0.0.7.0  if-
```

この出力は、`compress/zip` パッケージが凍結されていることを示します。パッケージは凍結されているため更新できません。

```
$ pkg unfreeze compress/zip
compress/zip was unfrozen.
```

`compress/zip` パッケージの凍結を解除した場合、元の `pkg update '*'` 操作によって、更新が使用可能なイメージ内のすべてのパッケージが更新されるはずですが、`pkg freeze` および `pkg unfreeze` コマンドについては、[90 ページの「指定したバージョンへのパッケージのロック」](#)を参照してください。

[133 ページの「同期リンクされたパッケージをインストールできない」](#)に示す `pkg sync-linked` エラーも類似していますが、非大域ゾーン内のパッケージが更新を妨げていることを示しています。

同期リンクされたパッケージをインストールできない

エラーメッセージ: `pkg sync-linked: The installed package package is not permissible.`

sync-linked パッケージは、非大域ゾーンのパッケージです。親イメージとのバージョンの不一致という理由でパッケージの拒否が表示される場合があります。

```
Linked progress: -pkg: update failed (linked image exception(s)):
```

```
A 'sync-linked' operation failed for child 'zone:z1' with an unexpected return value of 1 and generated the following output:
```

```
pkg sync-linked: The installed package package is not permissible.  
Reject: package  
Reason: Parent image has a incompatible newer version: package
```

次の理由により、この非互換性メッセージを受け取ることがあります。

- コアオペレーティングシステムパッケージは、非大域ゾーンと大域ゾーンとで同じバージョンにする必要があります。これらのパッケージを非大域ゾーンで個別に更新することはできません。同様に、1つの非大域ゾーン内でこれらのいずれかのパッケージを凍結すると、大域ゾーンとすべての非大域ゾーンで更新が失敗します。

次のコマンドは、大域ゾーンと非大域ゾーンの間で同期を維持する必要があるパッケージを一覧表示します。

```
$ pkg search -o pkg.name :depend:parent:
```

- ゾーンが構成されている BE を更新するとき、その BE をマウントし、`pkg -R` コマンドを使用して代替 BE (ABE) を更新するという操作は、ABE のパブリッシャー構成が、現在ブートされている BE のパブリッシャー構成と異なる場合は実行できません。ABE の非大域ゾーンでは、現在アクティブな BE からのパブリッシャー構成が使用されます。

非大域ゾーンをインストールできない

エラーメッセージ: The following pattern(s) did not match any allowable packages.
Try using a different matching pattern, or refreshing publisher information

コアオペレーティングシステムパッケージは、非大域ゾーンと大域ゾーンとで同じバージョンにする必要があります。このイメージ内の `solaris` パブリッシャーの起点が、大域ゾーンにインストールされたシステムパッケージと同じバージョンを含まないパッケージリポジトリに設定された場合、非大域ゾーンをインストールしようとすると、次のエラーになります。

```
$ zoneadm -z myzone install  
The following ZFS file system(s) have been created:  
  rpool/export/zones/myzone  
Progress being logged to /var/log/zones/zoneadm.20131104T181301Z.myzone.install  
Image: Preparing at /export/zones/myzone/root.
```

```

AI Manifest: /tmp/manifest.xml.9daq.i
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
  Zonename: myzone
Installation: Starting ...

      Creating IPS image
Startup linked: 1/1 done
  Installing packages from:
    solaris
      origin: http://pkg.oracle.com/solaris/release/
Error occurred during execution of 'generated-transfer-4606-1' checkpoint.
Failed Checkpoints:

Checkpoint execution error:

      The following pattern(s) did not match any allowable packages. Try
      using a different matching pattern, or refreshing publisher information:

Installation: Failed. See install log at /system/volatile/install.4606/install_log
ERROR: auto-install failed.

```

非大域ゾーンをインストールするには、solaris パブリッシャーの起点として設定するリポジトリが、非大域ゾーンのインストール場所となる大域ゾーン内にインストールされているものと同じシステムソフトウェアを最低限含んでいる必要があります。リポジトリには古いソフトウェアまたは新しいソフトウェアを含めることもできますが、大域ゾーンにインストールされているものと同じソフトウェアを含む必要があります。次のコマンドは、大域ゾーンにインストールされている pkg:/entire パッケージと同じバージョンが /export/IPSpkgrepos/Solaris リポジトリに含まれているため、このリポジトリはこの大域ゾーンの適切なパブリッシャーの起点であることを示しています。

```

$ pkg list entire
NAME (PUBLISHER)      VERSION                IFO
entire                0.5.11-0.175.2.0.0.26.0  i--
$ pkgrepo list -H -s /export/IPSpkgrepos/Solaris \
entire@0.5.11-0.175.2.0.0.26.0
solaris     entire     0.5.11-0.175.2.0.0.26.0:20131028T190148Z
$ pkg set-publisher -G '*' -M '*' -g /export/IPSpkgrepos/Solaris/ solaris

```

イメージを修正できない

エラーメッセージ: pkg: The image cannot be modified as it is currently in use by another package client

次の例に示すように、エラーメッセージには、イメージがロックされているクライアントの名前と pid が含まれているはずです。

```
pkg: The image cannot be modified as it is currently in use by another package
client: pkg on cbusl0406l, pid 26604.
```

現在のパッケージ処理 (この例では処理 26604) が終了したあとで `pkg` コマンドを再試行してください。処理がすぐに終了しない場合、たとえば `ptree -a` を使用して、処理の調査を開始します。

このイメージについて現在動作中のパッケージ処理は、更新が使用可能かどうかを検査する更新マネージャーである場合があります。更新マネージャーの cron ジョブと `pkg/update` サービスの詳細については、[143 ページの「更新マネージャーの使用」](#)を参照してください。

ファイルが回収された

次の情報メッセージの後に、改修されたファイルのパスと、ファイルの一時的な移動先が続きます。

```
The following unexpected or editable files and directories were
salvaged while executing the requested package operation; they
have been moved to the displayed location in the image:
```

これは情報メッセージのため無視してもかまいませんが、必要に応じて対応することもできます。

ディレクトリは、IPS で参照カウントされます。あるディレクトリを明示的または暗黙的に参照している、イメージ内にインストールされている最後のパッケージがそのディレクトリを参照しなくなると、そのディレクトリは削除されます。そのディレクトリにパッケージ解除されたファイルシステムオブジェクトが含まれている場合、それらの項目は `$IMAGE_META/lost+found` に移動されます。パッケージ解除されたファイルシステムオブジェクトは、IPS パッケージで配信されなかったファイルおよびディレクトリです。IMAGE_META の値は通常 `/var/pkg` です。IMAGE_META ディレクトリについての詳細は、`pkg(5)` のマニュアルページのファイルに関するセクションを参照してください。

格納されるイメージメタデータの最小化

`/var/pkg` ディレクトリはイメージのメタデータを保持します。このディレクトリは非常に大きくなる可能性があります。`/var/pkg` ディレクトリから手動で削除しないでください。

`flush-content-cache-on-success` イメージプロパティの値が `true` に設定されていることを確認します。`flush-content-cache-on-success` プロパティの値はデフォルトで `true`

です。flush-content-cache-on-success の値が true の場合は、pkg install および pkg update の操作が正常に完了したとき、キャッシュ済みファイルが削除されます。flush-content-cache-on-success プロパティが false に設定されている場合、次の例に示すコマンドを使用して、値を true にリセットできます。

```
$ pkg property flush-content-cache-on-success
PROPERTY          VALUE
flush-content-cache-on-success False
$ pkg set-property flush-content-cache-on-success true
$ pkg property -H flush-content-cache-on-success
flush-content-cache-on-success True
```

内容キャッシュのフラッシュ (flush-content-cache-on-success を true に設定すること) によって、一部の pkg 操作が完了するまでの時間が長くなることがあります。

非大域ゾーンには別のキャッシュがあり、/usr/lib/pkg.sysrepo の -c オプションで設定できます。-s オプションで、このキャッシュの最大サイズを設定できます。[pkg.sysrepo\(1M\)](#)のマニュアルページを参照してください。

パッケージのインストールパフォーマンスの増大

次のステップは、パッケージのインストールおよび更新のパフォーマンスの増大に役立ちます。

- ZFS ストレージプールの容量が 80% より低いことを確認します。

```
$ zpool list
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool    186G  75.2G  111G  40%  1.00x  ONLINE  -
```

- ローカルパッケージリポジトリを使用します。『[Oracle Solaris 11.2 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。
- http_proxy が設定されている場合、プロキシのパフォーマンスを確認します。

◆◆◆ 付録 B

IPS のグラフィカルユーザーインターフェース

IPS には、2 つのグラフィカルユーザーインターフェース (GUI) ツールが含まれます。

- パッケージマネージャーでは、パッケージおよびパブリッシャーに対するほとんどの操作と、ブート環境 (BE) に対する一部の操作を実行できます。Oracle Solaris OS および IPS のテクノロジーに不慣れな場合、パッケージマネージャーを使用すると、パッケージをすばやく識別してインストールできます。
- 更新マネージャーは、利用可能な更新があるイメージ内のすべてのパッケージを更新します。

パッケージマネージャーの使用

パッケージマネージャーは、コマンド行から実行できるタスクのサブセットを提供します。

- パッケージの一覧表示、検索、インストール、更新、削除
- パッケージソースの追加と構成
- BE のアクティブ化、名前変更、削除

次のいずれかの方法でパッケージマネージャーを開始します。

ツールバー ツールバーの「パッケージマネージャー」アイコンをクリックします。「パッケージマネージャー」アイコンは回転する矢印の付いたボックスです。

デスクトップアイコン デスクトップの「パッケージマネージャー」アイコンをダブルクリックします。

メニューバー 「システム」->「システム管理」->「パッケージマネージャー」を選択します。

コマンド行 **\$ packagemanager &**

パッケージマネージャーの完全なドキュメントについては、「パッケージマネージャー」メニューバーの「ヘルプ」->「内容」を選択してください。

パッケージマネージャーのコマンド行のオプション

`packagemanager(1)` コマンドでは、次のオプションがサポートされています。

表 B-1 パッケージマネージャーコマンドのオプション

オプション	説明
<code>--image-dir</code> または <code>-R dir</code>	<code>dir</code> をルートとするイメージを操作します。デフォルトの動作では、現在のイメージを操作します。 次のコマンドは、 <code>/aux0/example_root</code> に格納されたイメージを操作します。 \$ packagemanager -R /aux0/example_root
<code>--update-all</code> または <code>-U</code>	利用可能な更新があるすべてのインストール済みパッケージを更新します。このオプションを指定することは、パッケージマネージャーの GUI で「更新」オプションを選択することと同じです。すべてのパッケージの更新については、 143 ページの「更新マネージャーの使用」 を参照してください。
<code>--info-install</code> または <code>-i file.p5i</code>	パッケージマネージャーを Web インストールモードで実行するための <code>.p5i</code> ファイルを指定します。指定するファイルの拡張子が <code>.p5i</code> である必要があります。詳細は、 140 ページの「Web インストールの使用」 を参照してください。
<code>--help</code> または <code>-h</code>	コマンドの使用方法に関する情報を表示します。

Web インストールの使用

Web インストール処理の詳細は、パッケージマネージャーのヘルプを参照してください。

パッケージマネージャーでは、クリック 1 つで簡単に実行できる Web インストール処理を使用したパッケージのインストールがサポートされています。Web インストール処理では `.p5i` ファイルが使用されます。`.p5i` ファイルには、パブリッシャーと、それらのパブリッシャーからインストール可能なパッケージを追加するための情報が格納されます。`.p5i` ファイルの情報は、Web インストール処理によって読み取られて使用されます。

▼ Web インストールファイルを作成する方法

システムにインストールしたパッケージをほかのユーザーがインストールできるようにするために、Web インストール処理を使用して、それらのパッケージファイルのインストール命令をエクスポートすることができます。Web インストール処理が作成する `.p5i` ファイルは、インストールするパッケージとパブリッシャーのインストール命令で構成されます。

1. **パブリッシャーを選択します。**
パッケージマネージャーの「パブリッシャー」ドロップダウンメニューから、.p5i ファイルに含めるパッケージのパブリッシャーを選択します。
2. **パッケージを選択します。**
パッケージマネージャーのパッケージリストペインで、インストール命令を配布するパッケージを選択します。
3. **選択内容をエクスポートします。**
「ファイル」->「選択項目をエクスポート」を選択して、「エクスポート選択項目の確認」ウィンドウを表示します。
4. **選択内容を確認します。**
「OK」ボタンをクリックして選択項目を確認します。「選択項目をエクスポート」ウィンドウが表示されます。
5. **(オプション) ファイル名を変更します。**
.p5i ファイルのデフォルトの名前が表示されます。このファイル名は変更できます。
.p5i 拡張子を変更しないでください。
6. **(オプション) ファイルの場所を変更します。**
.p5i ファイルのデフォルトの場所が表示されます。この場所は変更できます。
7. **Web インストールファイルを保存します。**
「保存」ボタンをクリックして、ファイルの名前と場所を保存します。

▼ Web インストールを使用したパブリッシャーの追加とパッケージをインストールする方法

Web インストール処理を使用すると、.p5i ファイルからパッケージをインストールできます。デスクトップ上のファイルまたは Web サイト上のファイルを使用できます。

1. **パッケージマネージャーを Web インストールモードで起動します。**
 - **デスクトップ上の .p5i ファイルを選択します。**

- パッケージマネージャーをコマンド行から起動し、.p5i ファイルを指定します。

```
$ packagemanager ./wifile.p5i
```

- .p5i ファイルへのリンクを含む URL の場所にアクセスします。

- .p5i ファイルが置かれている Web サーバーでこの MIME タイプが登録済みの場合、.p5i ファイルへのリンクをクリックするだけです。

- .p5i ファイルが置かれている Web サーバーでこの MIME タイプが未登録の場合、.p5i ファイルをデスクトップに保存してそのファイルを選択します。

「インストール/更新」ウィンドウが表示されます。ウィンドウ上部のラベルは「パッケージマネージャー Web インストーラ/次のものがシステムに追加されます:」です。インストールされるパブリッシャーとパッケージの一覧が表示されます。

2. インストールを続けるには「続行」ボタンをクリックします。

- 必要に応じて、パブリッシャーを追加します。

- 必要に応じて、

3. 必要に応じて、パブリッシャーを追加します。

指定されたパッケージパブリッシャーがシステムでまだ構成されていない場合、「パブリッシャーを追加」ウィンドウが表示されます。パブリッシャーの名前と URI はすでに入力されています。

追加するパブリッシャーがセキュアなパブリッシャーである場合、SSL キーと証明書が必要です。システム上の「SSL キー」と「SSL 証明書」を参照して指定します。

パブリッシャーが正常に追加されると、「パブリッシャーを追加しました」ダイアログが表示されます。

4. インストールを続けるには「OK」ボタンをクリックします。

5. 無効化されたパブリッシャーを有効にします。

無効なパブリッシャーからのパッケージが .p5i ファイルに含まれている場合、Web インストールでは「パブリッシャーの有効化」ダイアログが開きます。このダイアログを使用してパブリッシャーを有効にし、パッケージをインストールできるようにします。

「インストール/更新」ウィンドウは、パッケージマネージャーの「インストール/更新」オプションを選択した場合と同じ表示になります。

すべてのパッケージがインストールされると、アプリケーションは閉じます。

更新マネージャーの使用

更新マネージャーは、すべてのインストール済みパッケージを、システムの制約で許可された最新バージョンに更新します。この制約は、インストール済みパッケージの依存関係およびパブリッシャーの構成によってシステムに課せられるものです。この機能は、次の方法で実行される機能と同じ働きをします。

- パッケージマネージャー GUI で、「更新」ボタンを選択するか、「パッケージ」-「更新」メニューオプションを選択します。

- `packagemanager` コマンドを使用します。

```
$ packagemanager --update-all
```

- `pkg` コマンドを使用します。

```
$ pkg update
```

次のいずれかの方法で更新マネージャーを開始します。

ステータスバー 更新が利用可能な場合、ステータスバーに通知が表示されます。通知で指示されている場所をクリックします。更新マネージャーアイコンは、3 つの箱が積み重なったものです。

メニューバー 「システム」->「システム管理」->「更新マネージャー」を選択します。

コマンド行 `$ pm-updatemanager`

自動 更新マネージャーパッケージ `package/pkg/update-manager` は、cron ジョブ `/usr/lib/update-manager/update-refresh.sh` を生成します。

```
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
```

SMF サービス `svc:/application/pkg/update` がオンラインのとき、この cron ジョブは、構成されたパブリッシャーから利用可能な更新パッケージを定期的に確認します (次に示すプロセスの最初の 2 手順)。更新パッケージが利用可能な場合、デスクトップのツールバーに通知が表示されます。通知アイコンを選択すると、更新マネージャー GUI が開きます。

「更新」ウィンドウが表示され、更新処理が始まります。

1. システムはすべてのカタログをリフレッシュします。
2. システムはすべてのインストール済みパッケージを評価し、パッケージの更新が利用可能かどうかを調べます。
 - 更新が利用可能なパッケージがない場合、「有効な更新はありません」というメッセージが表示され、処理は停止します。
 - パッケージの更新が利用可能な場合、更新されるパッケージの一覧が確認のために表示されます。これは、「キャンセル」ボタンをクリックして更新を中止する最後の機会です。

更新を続けるには「続行」ボタンをクリックします。

3. システムはすべてのパッケージの更新をダウンロードし、インストールします。

次のパッケージの更新が利用可能な場合、これらのパッケージが最初に更新されます。その後、ほかのパッケージが更新されます。

```
package/pkg
package/pkg/package-manager
package/pkg/update-manager
```

デフォルトでは、各パッケージは、最初にインストールされたときの取得元であるパブリッシャーから更新されます。元のパブリッシャーが非固定の場合、パッケージの新しいバージョンがこのイメージと互換性があるものであれば、別のパブリッシャーからのパッケージをインストールできます。パブリッシャーを固定または非固定として設定するには、パッケージマネージャーの「パブリッシャーの管理」ウインドウまたは `pkg set-publisher` コマンドを使用します。

更新処理中にエラーが発生した場合、「詳細」パネルが展開され、エラーの詳細が表示されます。問題が発生した段階の隣に、エラーステータスのインジケータが表示されます。

4. 更新されるパッケージとイメージポリシーに応じて、新しい BE が作成されることがあります。

更新に伴ってシステムが新しい BE を作成した場合、デフォルトの BE の名前を編集できます。

再ブートして新しい BE にブートする必要があります。新しい BE はブート時のデフォルトの選択肢になります。現在の BE は、代替のブート選択肢になります。

- システムをすぐに再起動するには「今すぐ再起動」ボタンをクリックします。
- システムをあとから再起動するには「あとで再起動」ボタンをクリックします。

更新マネージャーのコマンド行のオプション

`pm-updatemanager(1)` コマンドでは、次のオプションがサポートされています。

表 B-2 更新マネージャーコマンドのオプション

オプション	説明
<code>--image-dir</code> または <code>-R dir</code>	<code>dir</code> をルートとするイメージを操作します。デフォルトの動作では、現在のイメージを操作します。 次のコマンドは、 <code>/aux0/example_root</code> にあるイメージを更新します。 \$ pm-updatemanager -R /aux0/example_root
<code>--help</code> または <code>-h</code>	コマンドの使用方法に関する情報を表示します。

索引

あ

- アプリケーションバージョンメディエーション, 94
- イメージ, 17
 - アップグレード, 63
 - 再インストール, 56
 - 作成, 107
 - プロパティ, 100
 - BE 作成ポリシー, 100
 - 署名付きパッケージの使用, 102
 - 追加と削除, 103
 - ポリシー, 100
- インストールの制約, 12, 122

か

- カーネルゾーン, 17
- 回避リスト, 97
 - 削除, 44, 56
 - 追加, 48, 55
- 起点リポジトリ, 17
- グループ, 14, 29, 35, 97
- グループパッケージ, 14, 29, 35, 97
- 検索インデックス, 21, 43
- 子イメージ, 17
- 更新の制約, 122
 - entire incorporation パッケージ, 12
- 更新マネージャー, 139

さ

- サーフェス, 12, 122
- システムリポジトリ, 59
- 障害管理リソース識別子 (FMRI), 14
- 親イメージ, 17
- 制約, 12, 122
 - 緩和, 91

- ゾーン, 41, 57
 - Oracle Solaris 10 ゾーン, 17
 - カーネルゾーン, 17
 - 大域ゾーン, 17
 - 非大域ゾーン, 17
- ゾーンプロキシサービス
 - zones-proxy-client, 59
 - zones-proxyd, 59
- ソフトウェアリポジトリ, 11

た

- 大域ゾーン, 17, 41, 57

は

- バージョン制約, 12
 - 緩和, 91
- バージョンの制約, 122
- バージョン文字列コンポーネント, 14
- パッケージ, 12
 - FMRI, 14, 44
 - incorporation, 12, 122
 - pkg:/entire, 112, 122
 - SMF サービスアクチュエータ, 42
 - version-lock. ファセット, 91
 - アンインストール, 55
 - 依存関係, 35
 - 一覧表示, 21
 - イメージのクリーンアップ, 56
 - インストール, 44
 - インストール可能, 44
 - インストール可能な, 22
 - インストールされたファイルの修正, 53
 - インストールされていないパッケージの無視, 49, 55

- インストール済みの, 22
 - インストール済みパッケージの検証, 51
 - インストール済みパッケージの修正, 50
 - インストールソース, 45
 - インストール中の拒否, 48, 129
 - オプションのコンポーネント, 18, 86
 - 親の依存関係, 57
 - 回避リスト, 97
 - 削除, 44, 56
 - 追加, 48, 55
 - カスタム incorporation パッケージ, 68
 - 完全な FMRI, 26
 - グループ, 14, 29, 35, 97
 - 検索, 30
 - 公開, 11
 - 更新が利用可能な, 23
 - 更新またはアップグレード, 49
 - 最新の, 22
 - 削除, 55
 - 作成, 11
 - 識別子, 14
 - 署名プロパティ, 102
 - 使用可能なすべての, 23
 - すべて更新, 63
 - 制約された, 12, 122
 - 説明, 25
 - 相互に排他的なコンポーネント, 18, 86
 - ダウングレード, 50
 - 提供されたサービス, 33
 - 提供されたファイル, 33
 - 凍結, 25, 56, 90, 133
 - 名前, 14
 - 名前の変更, 24
 - 廃止, 24
 - パブリッシャー, 16, 44
 - バリエーション, 18, 86
 - バージョンが制約された, 12, 90, 122, 133
 - ファイルシステムの内容, 27
 - ファセット, 18, 86
 - ライセンス, 27, 30, 43
 - リポジトリ, 16
 - パッケージアーカイブ, 16, 45
 - パッケージに関する情報の更新, 21, 43
 - パッケージパブリッシャー 参照 パブリッシャー
 - パッケージマネージャー, 139
 - パッケージリポジトリ 参照 リポジトリ
 - パブリッシャー, 16, 77
 - 起点, 78
 - 検索順序, 22
 - 構成, 79
 - プロキシ, 84
 - プロパティ, 100
 - 署名付きパッケージの使用, 103
 - 追加と削除, 103
 - バリエーション, 18, 86
 - 非大域ゾーン, 17, 41, 57
 - 更新, 61
 - パッケージのインストール, 57
 - パッケージパブリッシャー, 59, 117
 - ファイル
 - 編集可能な preserve または overlay, 40
 - ファセット, 18, 86, 91
 - ブート環境 (BE), 17
 - プロキシ
 - SMF サービスプロパティ, 85
 - 環境変数, 85
 - パッケージパブリッシャー, 84
 - プロキシサービス, 59
 - プロパティ, 100
 - BE 作成ポリシー, 100
 - 署名付きパッケージの使用, 102
 - 設定, 106
 - 追加と削除, 103
 - 表示, 106
- ま**
- ミラーリポジトリ, 17
 - メディアエーション, 94
 - メディアエーター, 94
- ら**
- リポジトリ, 11, 16, 45
 - 起点, 17
 - 内容の更新, 21
 - ミラー, 17
 - リンクされたイメージ, 17, 41, 61
 - pkg uninstall, 55
 - pkg update, 49

B

BE, 17
pkg コマンドオプション, 40
 アクティブ化, 40
 イメージポリシープロパティ設定, 100
 名前の指定, 40
 パッケージのインストール中の作成, 47
 必要, 40
beadm コマンド, 48

E

entire incorporation パッケージ, 13

F

FMRI, 14

I

incorporation, 12, 91, 122
incorporation パッケージ, 12, 91, 112, 122
 カスタム, 68

O

Oracle Solaris 10 ゾーン, 17

P

p5i ファイル, 140
pkg.sysrepo コマンド, 59
pkg avoid コマンド, 97
pkg change-facet コマンド, 89, 91
pkg change-variant コマンド, 88
pkg contents コマンド, 27
 pkg search との比較, 31
pkg exact-install コマンド, 56
pkg facet コマンド, 89
pkg fix コマンド, 51
 pkg revert との比較, 50
pkg freeze コマンド, 90, 133

pkg history コマンド, 108
pkg image-create コマンド, 107
pkg info コマンド, 25
 pkg list との比較, 26
pkg install コマンド, 44
 --be-name オプション, 47
 --reject オプション, 48, 129
 pkg update との比較, 39
pkg list コマンド, 21, 44
 pkg info との比較, 26
pkg mediator コマンド, 94
pkg publisher コマンド, 78
pkg purge-history コマンド, 108
pkg refresh コマンド, 21, 43
pkg revert コマンド, 53
 pkg fix との比較, 50
pkg search コマンド, 30
 pkg contents との比較, 31
pkg set-mediator コマンド, 96
pkg set-publisher コマンド, 79, 84
 プロパティの追加と削除, 103
pkg unavoid コマンド, 97
pkg unfreeze コマンド, 90, 133
pkg uninstall コマンド, 55
pkg unset-mediator コマンド, 96
pkg update コマンド
 pkg install との比較, 39
 イメージのアップグレード, 63
 パッケージの更新またはアップグレード, 49
 パッケージのダウングレード, 50
pkg variant コマンド, 88
pkg verify コマンド, 50, 51
pkg:/entire パッケージ, 112, 122
pkg コマンド
 プレビュー, 38
 履歴の表示, 108
pkg コマンドのレビュー, 38
pkgrepo コマンド, 70

S

SMF サービス, 42
SMF プロキシサービス, 59
solaris-kz ブランドゾーン, 17

solaris10 ブランドゾーン, 17
solaris ブランドゾーン, 17

U

URI (Universal Resource Identifier), 16

W

Web インストール, 140