

Oracle® Solaris 11.1 ソフトウェア
パッケージの追加および更新

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

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

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

U.S. GOVERNMENT END USERS:

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

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

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

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

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

目次

はじめに	7
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	13
発行元、リポジトリ、およびパッケージアーカイブ	15
リポジトリの起点とミラー	15
イメージとブート環境	16
パッケージのファセットとバリエーション	16
インストール権限	17
2 IPS のグラフィカルユーザーインターフェース	19
パッケージマネージャーの使用	19
パッケージマネージャーのコマンド行オプション	20
Web Install の使用	20
Update Manager の使用	22
Update Manager のコマンド行オプション	24
3 ソフトウェアパッケージに関する情報の取得	25
パッケージのインストール状態情報の表示	25
パッケージの説明またはライセンスの表示	28
パッケージマニフェストからの情報の表示	29
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	30
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	31
ライセンス要件の表示	32

パッケージの検索	32
特定のファイルを提供するパッケージの識別	33
どのパッケージがどのSMFサービスを提供するかを表示する	34
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
4 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
ブート環境オプション	39
新しいパッケージのインストール	40
新しいブート環境へのパッケージのインストール	43
パッケージの拒否	45
パッケージの更新	45
パッケージの問題の修正	46
パッケージのインストールの検証	47
検証エラーの修正	47
ファイルの復元	48
パッケージのアンインストール	49
非大域ゾーンの操作	49
非大域ゾーンへのパッケージのインストール	49
複数の非大域ゾーンの同時更新	52
5 インストールされるイメージの構成	55
発行元の構成	55
発行元情報の表示	55
パッケージ発行元の追加、変更、削除	57
オプションのコンポーネントのインストールの制御	60
バリエーション値の表示と変更	62
ファセット値の表示と変更	63
指定したバージョンへのパッケージのロック	64
incorporation によって指定されたバージョン制約の緩和	66
デフォルトのアプリケーション実装の指定	68
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	69

イメージの更新	72
イメージ更新のベストプラクティス	72
インストールするバージョンの指定	74
更新前のバージョン制約の指定	74
使用可能なパッケージの制約	75
イメージのダウングレード	75
その他の更新コマンドオプション	75
イメージと発行元のプロパティの構成	76
ブート環境ポリシーイメージのプロパティ	76
パッケージの署名のプロパティ	78
追加のイメージのプロパティ	80
イメージのプロパティの設定	82
イメージの作成	83
操作履歴の表示	84

はじめに

『Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新』では、Oracle Solaris Image Packaging System (IPS) 機能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 コマンドを使用すると、Oracle Solaris 11 オペレーティングシステムのソフトウェアパッケージを一覧表示、検索、インストール、更新、および削除できます。単一の IPS コマンドで、イメージを新しいオペレーティングシステムリリースに更新できます。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。

さらに、IPS コマンドを使用して、IPS パッケージリポジトリをコピーおよび作成し、IPS パッケージを作成することもできます。それらのツールについては、[8 ページの「関連ドキュメント」](#)を参照してください。

IPS を使用するには、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 11.1 システムのインストール](#)』を参照してください。

対象読者

このドキュメントは、ソフトウェアのインストールと管理、システムイメージの管理などを行うシステム管理者のために作成されています。

内容の紹介

- [第1章「Image Packaging System の概要」](#)では、Image Packaging System と、パッケージ、発行元、リポジトリなどのコンポーネントについて説明します。
- [第2章「IPS のグラフィカルユーザーインターフェース」](#)では、パッケージマネージャーと Update Manager の使用方法 (Web Install の使用方法を含む) について説明します。
- [第3章「ソフトウェアパッケージに関する情報の取得」](#)では、パッケージの検索方法と、パッケージに関する情報の表示方法について説明します。
- [第4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、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方法について説明します。

- 第5章「インストールされるイメージの構成」では、パッケージの発行元の構成やインストール可能なパッケージの制限などの、イメージ全体に適用される特性の構成方法について説明します。

関連ドキュメント

次のマニュアルに加えて、パッケージマネージャーのオンラインヘルプや、pkg(1M)とbeadm(1M)のマニュアルページを参照してください。

- 『Oracle Solaris 11.1 パッケージリポジトリのコピーおよび作成』
- 『Oracle Solaris 11.1 ブート環境の作成と管理』
- 『Oracle Solaris 11.1 システムのインストール』
- 『Oracle Solaris 11.1 での Image Packaging System を使用したソフトウェアのパッケージ化および配布』

Oracle サポートへのアクセス

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

表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表 P-1 表記上の規則

字体	説明	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% su Password:
<i>aabbcc123</i>	プレースホルダ: 実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm filename と入力します。

表 P-1 表記上の規則 (続き)

字体	説明	例
AaBbCc123	書名、新しい単語、および強調する単語を示します。	『ユーザーズガイド』の第6章を参照してください。 キャッシュは、ローカルに格納されるコピーです。 ファイルを保存しないでください。 注: いくつかの強調された項目は、オンラインでは太字で表示されます。

コマンド例のシェルプロンプト

次の表に、Oracle Solaris OS に含まれるシェルの UNIX システムプロンプトおよびスーパーユーザーのプロンプトを示します。コマンド例のシェルプロンプトは、そのコマンドを標準ユーザーで実行すべきか特権ユーザーで実行すべきかを示します。

表 P-2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	#
C シェル	machine_name%
C シェルのスーパーユーザー	machine_name#

Image Packaging System の概要

Oracle Solaris Image Packaging System (IPS) は、Oracle Solaris 11 オペレーティングシステムのソフトウェアパッケージを一覧表示、検索、インストール、更新、および削除するためのフレームワークです。単一の IPS コマンドで、イメージを新しいオペレーティングシステムリリースに更新できます。

Image Packaging System

Oracle Solaris 11 ソフトウェアは IPS パッケージの形態で配布されます。IPS パッケージは、IPS 発行元が提供する 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 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。発行元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。
- パッケージ発行元を一覧表示、追加、および削除します。検索の優先順やステッキネスなどの、発行元の属性を変更します。署名ポリシーなどの発行元のプロパティを設定します。
- イメージを新しいオペレーティングシステムリリースに更新します。
- 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。
- パッケージを作成および公開します。
- ブート環境を作成します。

IPS を使用するには、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 11.1 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、このガイドでこれから使用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、キー/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、[pkg\(5\)](#) を参照してください。

グループパッケージと *incorporation* パッケージは、ファイルなどの内容は提供しませんが、関連パッケージのセットをインストールするのに役立ちます。

incorporation は、指定された一連のパッケージのバージョンを制限するパッケージです。たとえば、インストールされた *incorporation* のパッケージがバージョン 1.4.3 である場合、1.4.3 未満または 1.4.4 以上のバージョンはインストールできません。ただし、1.4.3.7 など、単にドット形式の数値を拡張したバージョンはインストールできます。*incorporation* は、*incorporation* 対象のパッケージを強制的に同期的にアップグレードします。*incorporation* 対象のパッケージは削除できますが、パッケージをインストールまたは更新する場合、バージョンが制限されます。関連情報については、[66 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。

entire という名前のパッケージは、ほかの *incorporation* パッケージ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に指定したパッケージは、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のマニフェストでは group 依存関係が指定されます。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(それらのパッケージが回避リストに含まれている場合は除く)。回避リストについては、69 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。

group/feature/amp パッケージは、たとえば、Apache Web サーバー、MySQL データベース、および PHP を提供します。group/system/solaris-desktop パッケージは、デスクトップシステムに適した一連のパッケージを提供します。group/system/solaris-large-server パッケージは、メディアツールやウィンドウ表示テーマなどのデスクトップパッケージは提供しません。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、31 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。

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、発行元、パッケージ名、およびバージョン文字列で構成されます。スキーム、発行元、およびバージョン文字列はオプションです。IPS コマンドを使用するときは、パッケージ名のうち、パッケージを一意に識別する最小部分を使用できます。

FMRI の形式:

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

FMRI の例:

```
pkg://solaris/driver/network/ethernet/bge@0.5.11,5.11-0.175.1.0.0.21.0:20120723T161616Z
```

スキーム pkg

発行元 solaris

発行元を指定する場合、発行元名の前に `pkg://` または `//` を付ける必要があります。

パッケージ名 `driver/network/ethernet/bge`

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発行元は省略する場合、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: `0.175.1.0.0.21.0`

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、`0.175` は Oracle Solaris 11 を示します。

更新リリース番号: `1`

この Oracle Solaris リリースの更新リリース番号。更新の値は、Oracle Solaris リリースの最初の顧客出荷で `0`、そのリリースの最初の更新で `1`、そのリリースの `2` 番目の更新で `2`、のようになります。この例で、`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更新 (SRU) 番号。SRU にはバグの修正だけが含まれ、新機能は含まれません。Oracle サポートリポジトリは、サポート契約を結んでいるシステムでのみ利用可能です。

予約済み: 0

このフィールドは現在 Oracle Solaris パッケージには使用されていません。

SRU ビルド番号: 21

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: 20110921T002716Z

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、このパッケージが発行された時間を ISO-8601 基本形式 YYYYMMDDTHHMMSSZ で表しています。

発行元、リポジトリ、およびパッケージアーカイブ

発行元は、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。発行元は、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。発行元は、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、パッケージ仕様に発行元の名前が含まれない場合、そのパッケージに対して、検索順序の先頭の発行元が検索されます。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 番目の発行元が検索されるというように、パッケージが見つかるか、すべての発行元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、発行元の情報と、その発行元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

イメージ内の特定の発行元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。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、クライアントは発行元のカタログを取得するために起点にアクセスします。発行元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発行元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。

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。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、[76 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。

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、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なプロパティであり、パッケージ内のアクションに設定されるタグです。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。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。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。

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せん。
- 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せん。
- 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、[60 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3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と更新、発行元の設定、イメージの変更などのタスクには、強力な権限が必要です。権限を強めるには、次のいずれかの方法を使用します。

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

```
$ pfexec pkg install editor/gnu-emacs
$ pfexec beadm activate solaris11_1-2
```

役割 `roles` コマンドを使用して、自分に割り当てられている役割を一覧表示します。root 役割を持っている場合は、`su` コマンドと root パスワードを使用して、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ーポリシーに応じて、自分のユーザーパスワードで `sudo` コマンドを使用し、特権コマンドを実行できる場合があります。

IPS のグラフィカルユーザーインタフェース

IPS には、2つのグラフィカルユーザーインタフェース (GUI) ツールが含まれます。

- パッケージマネージャーでは、パッケージおよび発行元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。
- Update Manager 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を使用して、コマンド行から実行できるタスクの一部を実行できます。

- 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起動します。

ツールバー	ツールバーにあるパッケージマネージャーのアイコンをクリックします。パッケージマネージャーのアイコンは、円を描く矢印が箱に描かれた形をしています。
デスクトップアイコン	デスクトップにあるパッケージマネージャーの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」>「管理」>「パッケージマネージャー」の順に選択します。
コマンド行	<code>\$ pfexec packagemanager &</code>

パッケージマネージャーのドキュメント全体を参照するには、パッケージマネージャーのメニューバーから「ヘルプ」>「使い方」を選択します。

パッケージマネージャーのコマンド行オプション

`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 2-1 パッケージマネージャーのコマンドオプション

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、 22 ページの「Update Manager の使用」 を参照してください。
<code>--info-install</code> または <code>-i file.p5i</code>	Web Install モードでパッケージマネージャーを実行するための <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、 20 ページの「Web Install 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 Install の使用

Web Install 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 Install 処理を使用してパッケージをインストールできます。Web Install 処理では `.p5i` ファイルを使用します。`.p5i` ファイルには、発行元と、それらの発行元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 Install 処理によって読み取られ、使用されます。

Web Install を使用したファイルのエクスポート

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 Install 処理を使用して、それらのパッケージファイルのインス

ツール命令をエクスポートすることができます。Web Install 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発行元のインストール命令で構成されます。

選択したパッケージとその発行元のインストール命令を .p5i ファイルにエクスポートするには、次の手順を実行します。

1. パッケージマネージャーの「発行元」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発行元を選択します。
2. 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. 「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. 「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. .p5i ファイルのデフォルトの名前が表示されます。この名前は変更できますが、拡張子 .p5i は変更しないでください。
6. .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. 「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 Install を使用した発行元の追加とパッケージのインストール

Web Install 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することができます。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. 次のいずれかの方法を使用して、Web Install モードでパッケージマネージャーを起動します。
 - デスクトップ上の .p5i ファイルを選択します。
 - コマンド行からパッケージマネージャーを起動し、.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。
2. 「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発行元とパッケージの一覧が表示されます。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。

3. 指定されたパッケージ発行元がシステムでまだ構成されていない場合、「発行元を追加」ウィンドウが表示されます。発行元の名前と URI はすでに入力されています。

追加する発行元がセキュアな発行元である場合、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。

発行元が正常に追加されると、「発行元を追加しました」ダイアログが表示されます。インストールを続けるには「OK」ボタンをクリックします。

4. 無効な発行元からのパッケージが .p5i ファイルに含まれている場合、「発行元の有効化」ダイアログが開きます。このダイアログを使用して発行元を有効にし、パッケージをインストールできるようにします。

ここで「インストール/更新」ウィンドウに表示される内容は、パッケージマネージャの「インストール/更新」オプションを選択したときと同じです。

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。

Update Manager の使用

Update Manager 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発行元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の GUI で、「更新」ボタンまたは「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。

```
$ pfexec packagemanager --update-all
```

- `pkg` コマンドを使用します。

```
$ pfexec pkg update
```

次のいずれかの方法で Update Manager を起動します。

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。Update Manager アイコンは3つの箱を積み重ねた形をしています。

メニューバー 「システム」>「管理」>「Update Manager」の順に選択します。

コマンド行 `# pm-updatemanager`

自動 Update Manager 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。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、構成された発行元から利用可能な更

新パッケージを定期的を確認します (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。Update Manager の GUI を開くには、通知アイコンを選択します。

「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。
3. 更新を続けるには「続行」ボタンをクリックします。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発行元から更新されます。元の発行元が非固定の場合、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、別の発行元からのパッケージをインストールすることができます。パッケージ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「発行元の管理」ウィンドウまたは `pkg set-publisher` コマンドを使用します。

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。BE の名前を編集したら、「今すぐ再起動」ボタンをクリックしてシステムをすぐに再起動します。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。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。

Update Manager のコマンド行オプション

`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 2-2 Update Manager のコマンドオプション

オプション	説明
<code>--image-dir</code> または <code>-R dir</code>	<p><i>dir</i> 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。</p> <p>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。</p> <pre># pm-updatemanager -R /aux0/example_root</pre>
<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。

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、または更新できるかどうか
- 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
- 特定のカテゴリ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

以下のコマンドを実行するために特別な権限は必要ありません。

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。このイメージに一致しないアーキテクチャーやゾーンタイプのパッケージバリエーションは表示されません。

```
/usr/bin/pkg list [-Hafnsuv] [-g path_or_uri ...] [--no-refresh]
                 [pkg_fmri_pattern ...]
```

`pkg list` コマンドは、パッケージごとに1行の情報を表示します。

```
$ pkg list '*toolkit'
NAME (PUBLISHER)          VERSION          IFO
isvtoolkit (isvpub)      1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.1.0.0.21.0 i--
```

丸括弧で囲まれた発行元名は、`isvpub` の発行元が、このイメージの発行元の検索順序で先頭の発行元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、検索順序の先頭の発行元によって発行されていません。

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。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  3.0.15-0.175.1.0.0.14.0 ---
isvtoolkit (isvpub)      1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.1.0.0.21.0 i--
```

この出力は `image/nvidia/cg-toolkit` パッケージを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。

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を一覧表示するには、`-af` オプションを使用します。これらのパッケージのうち最新のバージョンのみを一覧表示するには、`@latest` を指定します。

```
$ pkg list -af '*toolkit@latest'
NAME (PUBLISHER)          VERSION          IFO
developer/dtrace/toolkit 0.99-0.173.0.0.0.1.0 --r
image/nvidia/cg-toolkit  3.0.15-0.175.1.0.0.14.0 ---
isvtoolkit (isvpub)      1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.1.0.0.21.0 i--
```

この出力は `developer/dtrace/toolkit` パッケージをこのイメージにインストールできないことを示しています。O 列内の「r」は、このパッケージの名前が変更されていることを示します。`developer/dtrace/toolkit` パッケージの名前が `system/dtrace/dtrace-toolkit` に変更され、`system/dtrace/dtrace-toolkit` がすでにインストールされています。

次の例では、`web/amp` パッケージの名前が `group/feature/amp` に変更されています。`web/amp` パッケージをインストールするコマンドを指定した場合は、`group/feature/amp` パッケージが自動的にインストールされます。

```
$ pkg list -a amp
NAME (PUBLISHER)          VERSION          IFO
group/feature/amp        0.5.11-0.175.0.0.0.21.0 ---
web/amp                  0.5.11-0.174.0.0.0.0.0 --r
```

`pkg list` コマンドは、名前が変更されたパッケージの新しい名前を報告しません。前の例では、`pkg list` コマンドへの入力として指定されたパターンが新旧両方の名前にたまたま一致したため、推論を導き出すことができました。一般に、名前が変更されたパッケージの新しい名前を表示するには、[28 ページの「パッケージの説明またはライセンスの表示」](#) に示すように `pkg info` を使用します。

-n オプションは、既知の各パッケージの最新バージョンを一覧表示します。O 列の「o」は、パッケージが廃止されていることを示します。廃止されているパッケージはインストールできません。

```
$ pkg list -n '*mysql-5?'
NAME (PUBLISHER)          VERSION          IFO
database/mysql-50        5.0.91-0.171    --o
database/mysql-51        5.1.37-0.175.1.0.0.21.0  ---
```

この出力は database/mysql-50 パッケージをこのイメージにインストールできないことを示しています。このパッケージの名前は変更されていません。mysql-50 パッケージをインストールするコマンドを指定した場合、mysql-51 パッケージはインストールされません。この場合、どのパッケージもインストールされません。

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、[64 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 mercurial
NAME (PUBLISHER)          VERSION          IFO
developer/versioning/mercurial  2.2.1-0.175.1.0.0.21.0  if-
```

-s オプションは、パッケージ名とサマリーのみを一覧表示します。

```
$ pkg list -ns mysql-51 feature/amp
NAME (PUBLISHER)  SUMMARY
database/mysql-51  MySQL 5.1 Database Management System
group/feature/amp  AMP (Apache, MySQL, PHP) Deployment Kit for Oracle Solaris
```

-v オプションは完全なパッケージ FMRI を一覧表示します。

```
$ pkg list -nv mysql-51
FMRI          IFO
pkg://solaris/database/mysql-51@5.1.37,5.11-0.175.1.0.0.21.0:20120723T165236Z  ---
```

-u オプションは、新しいバージョンが使用可能な、インストールされているすべてのパッケージを一覧表示します。

```
$ pkg list -u 'compress/*'
NAME (PUBLISHER)          VERSION          IFO
compress/bzip2            1.0.6-0.175.1.0.0.19.0  i--
compress/gzip             1.4-0.175.1.0.0.19.0    i--
compress/p7zip            9.20.1-0.175.1.0.0.19.0  i--
compress/unzip            6.0-0.175.1.0.0.19.0    i--
compress/zip              3.0-0.175.1.0.0.19.0    i--
```

注-より新しいバージョンがパッケージリポジトリにあるパッケージの数は、このイメージ内で更新可能なパッケージの数より大きい場合があります。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発行元の構成によってイメージに課せられるものです。このイメージで更新可能なパッケージを調べるには、`pkg update -nv`を使用します。

`-g` オプションは、操作のパッケージデータのソースとして使用するリポジトリまたはパッケージアーカイブを指定する場合に使用します。

`--no-refresh` オプションを使用した場合、`pkg` は、イメージの発行元のリポジトリに接続を試みて、使用可能なパッケージの最新の一覧を取得しません。

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、パッケージに関する情報を表示します。これには、名前、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、および完全な FMRI が含まれます。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せん。

```
/usr/bin/pkg info [-lr] [-g path_or_uri ...] [--license] [pkg_fmri_pattern ...]
```

`info` サブコマンドと `list` サブコマンドはどちらも、パッケージ名、発行元、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していないかどうかを表示します。`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。

`-r` オプションは、使用可能な最新のバージョンを表示し、構成された発行元のリポジトリから、現在インストールされていないパッケージについての情報を取得します。

```
$ pkg info -r group/feature/amp
Name: group/feature/amp
Summary: AMP (Apache, MySQL, PHP) Deployment Kit for Oracle Solaris
Description: Provides a set of components for deployment of an AMP (Apache,
MySQL, PHP) stack on Oracle Solaris
Category: Meta Packages/Group Packages (org.opensolaris.category.2008)
Web Services/Application and Web Servers (org.opensolaris.category.2008)
```

```

State: Not installed
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.175.1.0.0.21.0
Packaging Date: July 23, 2012 06:20:57 PM
Size: 5.46 kB
FMRI: pkg://solaris/group/feature/amp@0.5.11,5.11-0.175.1.0.0.21.020120723T182057Z

```

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ます。次の例は、developer/dtrace/toolkit パッケージの新しい名前が system/dtrace/dtrace-toolkit であることを示しています。

```

$ pkg info -r developer/dtrace/toolkit
Name: developer/dtrace/toolkit
Summary:
State: Not installed (Renamed)
Renamed to: pkg://system/dtrace/dtrace-toolkit@0.99,5.11-0.173.0.0.0.0.0
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

```

--license オプションはパッケージのライセンステキストを表示します。この情報はきわめて長くなることがあります。上に示す情報(--license オプションを使用しない)は表示されません。

```

$ pkg info --license x11/server/xorg
Copyright (c) 2012, Oracle and/or its affiliates. All rights reserved.
...

```

-g オプションは、操作のパッケージデータのソースとして使用するリポジトリまたはパッケージアーカイブを指定する場合に使用します。

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。

```

/usr/bin/pkg contents [-Hmr] [-a attribute=pattern ...] [-g path_or_uri ...]
[-o attribute ...] [-s sort_key] [-t action_name ...] [pkg_fmri_pattern ...]

```

`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。

次の例は、`pkg contents` のデフォルトの動作を示しています。表示するアクションと属性を指定するには、オプションを使用します。

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

`-m` オプションは、パッケージマニフェスト全体を表示します。

`-r` オプションは、使用可能な最新のバージョンを表示し、構成された発行元のリポジトリから、現在インストールされていないパッケージについての情報を取得します。

`-g` オプションは、操作のパッケージデータのソースとして使用するリポジトリまたはパッケージアーカイブを指定する場合に使用します。

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

表示するアクションのタイプを指定する場合は、`-t` オプションを使用します。コ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。

出力に表示する属性を指定する場合は、`-o` オプションを使用します。コ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。この例では、`pkg.size` 疑似属性は、ファイルのサイズを示しています。`file` アクションには `size` 属性がありません。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。

指定したアクション属性によってアクションをソートする場合は、`-s` オプションを使用します。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` オプションは複数回指定できます。

```
$ pkg contents -t file -o owner,group,mode,pkg.size,path -s path zip
OWNER GROUP MODE PKG.SIZE PATH
root bin 0555 228600 usr/bin/zip
root bin 0555 107944 usr/bin/zipcloak
root bin 0555 101856 usr/bin/zipnote
root bin 0555 106252 usr/bin/zipsplit
root bin 0444 86036 usr/share/man/man1/zip.1
root bin 0444 2548 usr/share/man/man1/zipcloak.1
root bin 0444 2239 usr/share/man/man1/zipnote.1
root bin 0444 1680 usr/share/man/man1/zipsplit.1
```

パッケージマニフェストを表示した場合、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。上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ません。このイメージは x86 アーキテクチャーです。SPARC アーキテクチャー用のファイルは表示されません。バリエーションとファセットについては、[60 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。

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 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-small-server グループパッケージは、サーバーに小さいパッケージのセットをインストールするために使用できる代替パッケージでもあります。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-large-server
archiver/gnu-tar
compress/bzip2
...
text/texinfo
web/wget
```

-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、[12 ページの「IPS 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-large-server'
archiver/gnu-tar GNU version of the tar archiving utility
compress/bzip2 high-quality block-sorting file compressor - utilities
```

```
compress/gzip      GNU Zip (gzip)
...
text/texinfo       Documentation system for on-line information and printed output
web/wget           wget - GTNU wget
```

ライセンス要件の表示

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

```
$ pkg contents -rt license -a must-accept=true \
-o must-display,license,pkg.name '*incorporation'
MUST-DISPLAY LICENSE          PKG.NAME
true          usr/src/pkg.license_files/lic_OTN consolidation/osnet/osnet-incorporation
```

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。

```
/usr/bin/pkg search [-HIafldr] [-o attribute ...] [-s repo_uri] query
```

`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` コマンドはクエリーに一致するパッケージの名前を返します。

デフォルトでは、*query* は、大文字と小文字を除いて完全一致する一連の用語として解釈されます。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。

クエリーは次の構造化された形式で表現できます。

```
pkg_name:action_name:index:token
```

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。*pkg_name* および *token* フィールドには、明示的なワイルドカードを含めることができます。*action_name* および *index* の値は、正確に一致する必要があります。*action_name* の値は、アクションの名前です。*index* の値は、アクションの属性の名前です。パッケージのアクションと属性の一覧については、[pkg\(5\)](#) のマニュアルページの「アクション」を参照してください。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* の役立つ値の例には、 *file* および *dir* アクションの *basename* や *path*、 *depend* アクションの依存関係タイプ (*require* や *group* など)、ドライバアクションの *driver_name* や *alias* などがあります。

一般に、 *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、次の部分的な *driver* アクションでは、 *alias* は *index* に指定できる属性名であり、 *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。たとえば、次に示す部分的な *set* アクションの場合、 *index* には *pkg.summary* を指定し、 *token* には *Broadcom* を指定できます。

```
set name=pkg.summary value="Broadcom 57xx 1GbE NIC Driver"
```

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*、 および *pkg.summary* を含むものがあります。 [pkg\(5\)](#) のマニュアルページの「設定アクション」を参照してください。

デフォルトでは、このイメージに対して構成されているすべての発行元に関連付けられたリポジトリが検索されます。このイメージにインストールされているパッケージのみを検索するには、 *-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。

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。

特定の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
```

どのパッケージがどのSMFサービスを提供するかを表示する

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 value, pkg.name 'org.opensolaris.smf.fmri:*network/http*'
VALUE                                PKG.NAME
['svc:/network/http', 'svc:/network/http:apache22'] web/server/apache-22
['svc:/network/http', 'svc:/network/http:tomcat6'] web/java-servlet/tomcat
['svc:/network/http', 'svc:/network/http:squid'] web/proxy/squid
['svc:/network/http', 'svc:/network/http:lighttpd14'] web/server/lighttpd-14
```

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
```

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「ソースコード管理」があるすべてのパッケージを識別しています。

```
$ pkg search 'info.classification:source code management'
INDEX      ACTION VALUE                                PACKAGE
info.classification set    Development/Source Code Management pkg:/developer/versioning/sccs@0.5.11-0.175
info.classification set    Development/Source Code Management pkg:/developer/xopen/xcu4@0.5.11-0.175.1.0.
info.classification set    Development/Source Code Management pkg:/developer/versioning/git@1.7.9.2-0.175
info.classification set    Development/Source Code Management pkg:/developer/versioning/mercurial-27@2.2.
info.classification set    Development/Source Code Management pkg:/library/python-2/subversion@1.7.5-0.17
info.classification set    Development/Source Code Management pkg:/developer/versioning/mercurial-26@2.2.
info.classification set    Development/Source Code Management pkg:/library/java/subversion@1.7.5-0.175.1.
info.classification set    Development/Source Code Management pkg:/developer/quilt@0.60-0.175.1.0.0.21.0
info.classification set    Development/Source Code Management pkg:/developer/versioning/cvs@1.12.13-0.175
info.classification set    Development/Source Code Management pkg:/developer/versioning/subversion@1.7.5-
info.classification set    Development/Source Code Management pkg:/developer/versioning/mercurial@2.2.1-0
info.classification set    Development/Source Code Management pkg:/library/perl-5/subversion@1.7.5-0.175.
```

この例は、大量の繰り返された情報が表示されているため、実際に必要とする情報がわかりにくくなっています。

次の例は、-o オプションを使用して、パッケージの名前のみを表示し、-H オプションを使用して、列見出しを省略しています。

```
$ pkg search -Ho pkg.name 'info.classification:source code management'
developer/versioning/sccs
developer/xopen/xcu4
developer/versioning/git
developer/versioning/mercurial-27
library/python-2/subversion
developer/versioning/mercurial-26
library/java/subversion
developer/quilt
developer/versioning/cvs
developer/versioning/subversion
developer/versioning/mercurial
library/perl-5/subversion
```

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 contents コマンドを使用して、i86pc パッケージと system/hal パッケージのタイプ require の depend アクションを表示した場合、両方のパッケージに system/kernel/power が表示されます。

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

次の例は、多くのパッケージで pkg:/x11/server/xorg@1.12.99 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/xvnc                          pkg:/x11/server/xorg@1.12.99
x11/server/xorg                          pkg:/x11/server/xorg@1.12.99
x11/server/xorg/driver/xorg-video-mga    pkg:/x11/server/xorg@1.12.99
x11/server/xorg/driver/xorg-video-vesa   pkg:/x11/server/xorg@1.12.99
x11/server/xorg/driver/xorg-input-vmouse pkg:/x11/server/xorg@1.12.99
...
```

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-small-server グループパッケージは、サーバーに小さいパッケージの

セットをインストールするために使用できる代替パッケージでもあります。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 -Hfo fmri '*/solaris-large-server:depend:group:*'
archiver/gnu-tar
compress/bzip2
...
text/texinfo
web/wget
```

この例で、`-o pkg.name` は、クエリーの `pkg_name` 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-desktop
```

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

デフォルトで、検索では、このイメージにインストール可能なパッケージのみが返されます。この例では、検索は一致するパッケージを返さず、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、類似の `pkg contents` コマンドの結果の数より多くなります。これらの検索結果には、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 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を含めることができます。この検索からの出力を、[31 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」](#) に示す `pkg contents` コマンドからの出力と比較します。

ヒント - 一般に、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。

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、発行元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3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

[49 ページの「非大域ゾーンの操作」](#)では、非大域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、[17 ページの「インストール権限」](#)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。

```
$ pfexec pkg install -nv group/feature/amp
  Packages to install: 5
  Estimated space available: 112.19 GB
  Estimated space to be consumed: 374.19 MB
  Create boot environment: No
  Create backup boot environment: No
  Services to change: 2
  Rebuild boot archive: No
Changed packages:
solaris
  database/mysql-51
  None -> 5.1.37,5.11-0.175.1.0.0.21.0:20120723T165236Z
  database/mysql-common
  None -> 0.5.11,5.11-0.175.1.0.0.21.0:20120723T165447Z
  group/feature/amp
  None -> 0.5.11,5.11-0.175.1.0.0.21.0:20120723T182057Z
  web/server/apache-22/module/apache-dtrace
  None -> 0.3.1,5.11-0.175.1.0.0.21.0:20120723T174611Z
  web/server/apache-22/module/apache-fcgid
  None -> 2.3.6,5.11-0.175.1.0.0.21.0:20120723T174613Z
Services:
  restart_fmri:
  svc:/system/manifest-import:default
  svc:/system/rbac:default
```

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。使用される追加の領域の量はMバイトではなく、Gバイト単位です。この操作には、長い時間を必要とする可能性があり、このイメージとパッケージリポジトリ間で大量のネットワークトラフィックが発生する可能性があります。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、[76 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。

```
$ pfexec pkg change-facet -nv 'facet.locale.*=true'
  Packages to update:      831
  Variants/Facets to change: 1
  Estimated space available: 112.19 GB
```

```

Estimated space to be consumed: 2.96 GB
  Create boot environment:      No
Create backup boot environment: Yes
  Rebuild boot archive:        No
Changed variants/facets:
  facet facet.locale.*: True
Changed packages:
solaris
...

```

パッケージのインストールおよび更新

`pkg install` コマンドは、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、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。`pkg install` コマンドには1つ以上のパッケージ名を指定する必要があります。

`pkg update` コマンドは、インストール済みのパッケージを更新します。`pkg update` コマンドに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。`pkg update` コマンドには、すでにインストールされているパッケージの名前を0個以上指定します。パッケージ名を指定しないと、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\(1\)](#) のマニュアルページの `file` アクションのそれぞれの属性を参照してください。

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 が作成されることがあります。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、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。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、[76 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 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。

```

--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。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、76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、76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の使用は、暗黙的に--require-new-beを示します。

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発行元の検索順序で先頭の発行元から、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発行元から、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。

イメージで複数の発行元が有効になっている場合は、発行元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で発行元を指定したりすることによって、パッケージを提供する発行元を制御できます。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、13ページの「障害管理リソース識別子」を参照してください。発行元のスティッキネスと検索順序の設定については、55ページの「発行元の構成」を参照してください。

```
/usr/bin/pkg install [-nvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-index] [--no-refresh] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  [--reject pkg_fmri_pattern ...] pkg_fmri_pattern ...
```

`pkg_fmri_pattern` で発行元を指定しない場合は、一致するパッケージを提供する最初の発行元がインストール元として使用されます。該当する発行元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発行元を確認するには、`pkg list -a` コマンドを使用します。

次のコマンドは、パッケージ `atool` のインストール可能なバージョンが構成済み発行元から入手可能であるが、検索順序の最初にある発行元がこのイメージでインストールできないバージョンを持っていることを示しています。

```
$ 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発行元から「`atool`」という `pkg_fmri_pattern` 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。

```
$ pfexec 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 `pkg_fmri_pattern` を指定します。

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

実際のインストールを実行する前に、何がインストールされるかを確認するには、`-nv` オプションを使用します。

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発行元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発行元が優先されます。

- 指定された `pkg_fmri_pattern` に一致するパッケージが、イメージで有効になっている発行元から利用可能な場合に、その同じ発行元が -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、イメージで有効になっている発行元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発行元によって提供されるパッケージがある場合は、起点なしでイメージ構成に追加されます。
- 指定された `pkg_fmri_pattern` に一致するパッケージが、イメージで有効になっている発行元から利用可能な場合に、その同じ発行元が -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 発行元から btool 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 発行元からも入手可能ですが、devtool 発行元はこのイメージで構成されていません。イメージ内に構成されている発行元からパッケージが利用可能な場合は、構成済みの発行元が -g の発行元より優先されるため、次のコマンドは solaris 発行元からパッケージをインストールしようとします。

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

devtool 発行元からパッケージをインストールするには、`pkg_fmri_pattern` に発行元の名前を指定します。

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

次の例で、isvpub は起点 `/export/isvrepo` でイメージ内に構成されている発行元です。isvpub 発行元は `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発行元にその起点は指定されていません。同じ発行元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。

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

57 ページの「[パッケージ発行元の追加、変更、削除](#)」にある発行元のスティッキネスの説明も参照してください。

大域ゾーンと同時に n 個の非大域ゾーンにパッケージをインストールするには、-c オプションを使用します。-c オプションの使用例については、52 ページの「[複数の非大域ゾーンの同時更新](#)」を参照してください。

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しない

と、パッケージのライセンスに同意が必要になった場合、インストール操作は失敗します。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。

`--no-index` オプションを指定すると、操作の正常終了後に、検索インデックスが更新されません。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 `install`、`update`、および `uninstall` 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、指定された各発行元の発行元メタデータを更新できます。発行元を指定しない場合、すべての発行元を対象にリフレッシュが実行されます。

`-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、イメージの発行元のリポジトリに接続しません。

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、それが次回、`--no-be-activate` オプションを指定しない場合のリポート時のデフォルトブート環境になります。

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、[76 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、次回のリポート時に、新しいBEがデフォルトのブート選択肢になりません。

新しいBEを強制的に作成するか、新しいBEにわかりやすい名前を付ける場合は、`--be-name` オプションを使用します。

```
$ pfexec pkg install --be-name sllamp group/feature/amp
  Packages to install: 5
  Create boot environment: Yes
  Create backup boot environment: No
```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
----------	------	-------	-----------	-------

```
Completed                               5/5      271/271    52.3/52.3    0B/s

PHASE                                ITEMS
Installing new actions                 410/410
Updating package state database        Done
Updating image state                   Done
Creating fast lookup database          Done
Reading search index                  Done
Updating search index                  5/5
```

A clone of s11_1 exists and has been updated and activated.
 On the next boot the Boot Environment s11amp will be
 mounted on '/'. Reboot when ready to switch to this updated BE.

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

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 s11amp BE にインストールされます。

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
--           -
s11amp      R           -      20.75G  static 2012-08-06 15:36
solaris     -           -      44.81M  static 2010-11-07 17:45
solaris11_1 N           /      30.04M  static 2012-07-25 17:10
```

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。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。

```
$ pfexec beadm mount s11amp /mnt
$ pkg -R /mnt list group/feature/amp
NAME (PUBLISHER)      VERSION                IFO
group/feature/amp    0.5.11-0.175.1.0.0.21.0  i--
```

s11amp BE を必ずアンマウントしてください。

```
$ beadm list
BE           Active Mountpoint Space  Policy Created
--           -
s11amp      R           /mnt    20.75G  static 2012-08-06 15:36
solaris     -           -      44.81M  static 2010-11-07 17:45
solaris11_1 N           /      30.05M  static 2012-07-25 17:10
$ pfexec beadm unmount s11amp
```

パッケージの拒否

指定した `pkg_fmri_pattern` に一致する名前を持つパッケージをインストールしないようにする場合は、`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。グループ依存関係のターゲットである拒否対象パッケージは回避リストに登録されます。回避リストについては、[69 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。

```
$ pfexec pkg install -nv --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発行元から、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。

イメージで複数の発行元が有効になっている場合は、発行元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 で発行元を指定したりすることによって、パッケージを提供する発行元を制御できます。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、[13 ページの「障害管理リソース識別子」](#)を参照してください。発行元のスティッキネスと検索順序の設定については、[55 ページの「発行元の構成」](#)を参照してください。

```
/usr/bin/pkg update [-fnvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-index] [--no-refresh] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  [--reject pkg_fmri_pattern ...] [pkg_fmri_pattern ...]
```

パッケージの最新バージョンを明示的に要求するには、`pkg_fmri_pattern` のバージョン部分に `latest` を使用します。

```
$ pfexec pkg update vim@latest
```

現在インストールされているバージョンより古いパッケージバージョンを指定して、インプレースダウングレードを実行できます。保持される構成ファイルのうち、ダウングレードされるパッケージの一部であり、元のバージョンがインストールされたあとに変更されたファイルは、拡張子 `.update` を使用して名前が変更されます。どのファイルを保持するかをパッケージシステムが決定する方法

と、パッケージのアップグレード中にこれらの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項目を参照してください。

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。`-g` オプションの効果の詳細と例については、[40 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。

大域ゾーンと同時に n 個の非大域ゾーンのパッケージを更新するには、`-c` オプションを使用します。例については、[52 ページ](#)の「[複数の非大域ゾーンの同時更新](#)」を参照してください。

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、それが次回、`--no-be-activate` オプションを指定しない場合のリポート時のデフォルトブート環境になります。

更新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しないと、パッケージのライセンスに同意が必要になった場合、更新操作は失敗します。この操作の一環として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。

`--no-index` オプションを指定すると、操作の正常終了後に、検索インデックスが更新されません。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 `install`、`update`、および `uninstall` 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、指定された各発行元の発行元メタデータを更新できます。発行元を指定しない場合、すべての発行元を対象にリフレッシュが実行されます。

`pkg-fmri` が指定されていないとき、または指定された `pkg-fmri` 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、[72 ページ](#)の「[イメージの更新](#)」を参照してください。

パッケージの問題の修正

パッケージのインストール後に発生する可能性のある問題の例として、パッケージによって提供されるファイルが破損することがあります。このセクションで示す例では、`/usr/share/auto_install/manifest/default.xml` ファイルが削除されました。

失われているファイルを提供したパッケージを特定するには、`pkg search` コマンドを使用します。

```
$ pkg search -Hlo pkg.name /usr/share/auto_install/manifest/default.xml
system/install/auto-install/auto-install-common
```

パッケージのインストールの検証

現在の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。

```
/usr/bin/pkg verify [-Hqv] [pkg_fmri_pattern ...]
```

関連する発行元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、[78 ページの「パッケージの署名のプロパティ」](#) の `signature-policy` を参照してください。

検証の出力からヘッダーを省略する場合は、`-H` オプションを使用します。致命的なエラーが検出された場合に、エラーを返す以外に何も出力しない場合は、`-q` オプションを使用します。パッケージに関する情報メッセージを含める場合は、`-v` オプションを使用します。

```
$ pfexec pkg verify -v system/install/auto-install/auto-install-common
PACKAGE                               STATUS
pkg://solaris/system/install/auto-install/auto-install-common  ERROR
    file: usr/share/auto_install/manifest/default.xml
    Missing: regular file does not exist
```

検証エラーの修正

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

```
/usr/bin/pkg fix [--accept] [--licenses] [pkg_fmri_pattern ...]
```

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。

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しないと、パッケージのライセンスに同意が必要になった場合、修正操作は失敗します。この操作の一環として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。

```
$ pfexec pkg fix --accept system/install/auto-install/auto-install-common
Verifying: pkg://solaris/system/install/auto-install/auto-install-common  ERROR
    file: usr/share/auto_install/manifest/default.xml
```

```

Missing: regular file does not exist
Created ZFS snapshot: 2012-08-06-23:32:03
Repairing: pkg://solaris/system/install/auto-install/auto-install-common
Creating Plan (Evaluating mediators):

```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
Completed	1/1	1/1	0.0/0.0	0B/s

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

ファイルの復元

ファイルをそれらの配布時の状況に復元する場合は、`pkg revert` コマンドを使用します。

```

/usr/bin/pkg revert [-nv] [--no-be-activate] [--no-backup-be | --require-backup-be] [--backup-be-name name]
[--deny-new-be | --require-new-be] [--be-name name] (--tagged tag-name ... | path-to-file ...)

```

特定の *tag-name* でタグ付けされたすべてのファイル、または個別のファイルを元に戻すことができます。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デフォルト値に戻すと、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。

次の例では、変更されたシステム構成ファイルのサンプルが、その元の内容に復元されます。

```

$ pfexec pkg revert /usr/share/auto_install/sc_profiles/static_network.xml
   Packages to update: 1
   Create boot environment: No
   Create backup boot environment: No

```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
Completed	1/1	1/1	0.0/0.0	0B/s

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

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。

```
/usr/bin/pkg uninstall [-nvq] [-C n] [--no-index] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name] pkg_fmri_pattern ...
```

パッケージがグループ依存関係の対象である場合、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。回避リストについては、[69 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。

`--no-index` オプションを指定すると、操作の正常終了後に、検索インデックスが更新されません。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 `install`、`update`、および `uninstall` 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、指定された各発行元の発行元メタデータを更新できます。発行元を指定しない場合、すべての発行元を対象にリフレッシュが実行されます。

大域ゾーンと同時に n 個の非大域ゾーンのパッケージをアンインストールするには、`-c` オプションを使用します。`-c` オプションの使用例については、[52 ページの「複数の非大域ゾーンの同時更新」](#)を参照してください。

非大域ゾーンの操作

ほとんどの IPS コマンドは、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。Oracle Solaris 10 ブランドゾーンは Oracle Solaris 11 非大域ゾーンとは異なります。Oracle Solaris 10 ブランドゾーンは IPS コマンドで無視されます。このドキュメントでは、「非大域ゾーン」は Oracle Solaris 11 非大域ゾーンを意味します。

大域ゾーンと非大域ゾーンの重要な違いは、パッケージ発行元の使用方法です。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発行元構成の変更は、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。

非大域ゾーンへのパッケージのインストール

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。大域ゾーンで `pkg update` コマンドを引数なしで実行すると、大域ゾーンと各非大域ゾーンが更新されます。例について

は、52 ページの「複数の非大域ゾーンの同時更新」を参照してください。大域ゾーンでインストール、更新、またはアンインストールコマンドにパッケージ名を指定すると、IPS は各非大域ゾーンを確認し、大域ゾーンとの互換性を保つために必要な場合のみ変更を加えます。大域ゾーンでファセットとバリエーションを変更すると、非大域ゾーンにも影響することがあります。

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、`-n` 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。結果として大域ゾーンとの互換性が保たれる場合は、異なるパッケージをインストールすることも、同じパッケージの異なるバージョンをインストールすることもできます。非大域ゾーンのイメージでは、異なるパッケージを回避すること、異なるバージョンでパッケージを凍結すること、異なるデフォルト実装を選択するようにメタデータを設定すること、および異なるファセットを設定することができます。

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でなければなりません。`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。

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発行元構成の変更は、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。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します。

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-proxyd: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、大域ゾーンの発行元を示しています。

```
global:~$ pkg publisher
PUBLISHER      TYPE      STATUS P LOCATION
solaris         origin    online F http://pkg.oracle.com/solaris/release/
solaris         origin    online F file:///export/repoSolaris11/
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、これらの同じ発行元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>
solaris (syspub) origin    online F <system-repository>
isvpub (syspub) origin    online F <system-repository>
```

P列のTは、起点にプロキシが存在することを意味します。詳細情報を取得するには、次のいずれかのコマンドを使用します。

```
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
solaris true true true origin online http://localhost:1008/solaris/omitted/ -
z1:~$ pkg publisher solaris
Publisher: solaris
Alias:
Origin URI: http://localhost:1008/solaris/91b04f12f39930ae8e27f5636b7a342e8f460133/
SSL Key: None
SSL Cert: None
Origin URI: http://pkg.oracle.com/solaris/release/
Proxy: http://localhost:1008
SSL Key: None
SSL Cert: None
Client UUID: c92e7a92-dce5-11e1-b7e5-8800209e4377
Catalog Updated: August 2, 2012 05:10:48 PM
Enabled: Yes
```

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、発行元の場所が <system-repository> の場合、発行元の起点やプロパティ、および発行元の検索順序を変更することはできません。

発行元に到達できない場合は、`http_proxy` 環境変数を設定するか、`pkg set-publisher` コマンドに `--proxy` オプションを指定することで、大域ゾーンにプロキシを設定できます。`--proxy` オプションについては、[pkg\(1\)](#) のマニュアルページおよび [57 ページの「パッケージ発行元の追加、変更、削除」](#) を参照してください。`http_proxy` および `https_proxy` 環境変数の設定手順については、『[Oracle Solaris 11.1 の管理: Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管](#)』

理』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。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。

大域ゾーンですでに構成されている特定の発行元のパッケージを一覧表示する場合、次の形式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 '//isvpub/*'
NAME (PUBLISHER)   VERSION   IFO
isvtool (isvpub)   1.0-0     ---
```

大域ゾーンで構成されていないが、ネットワークまたはファイルシステムによって非大域ゾーンからアクセスできるリポジトリの場合、file:///export/myrepoが非大域ゾーンからアクセスできるリポジトリの場所であれば、次のコマンドはどちらも同じパッケージを一覧表示します。

```
z1:~$ pkg list -af -g file:///export/myrepo
z1:~$ pkgrepo list -s file:///export/myrepo
```

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。-c n オプションと PKG_CONCURRENCY=n 環境変数は、n が 1 以上の場合に、最大 n 個のイメージを並列で更新するように指定します。n のデフォルト値は 1 です。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ともに並列で更新されます。

-c オプションが指定された場合、PKG_CONCURRENCY 環境変数は無視されます。-c オプションと PKG_CONCURRENCY 環境変数は、pkg install、pkg uninstall、pkg change-variant、pkg change-facet、および pkg update で使用できます。

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。必要なのは非大域ゾーンをマウントすることだけです。

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。

```
global:~$ pfexec pkg update -C 0 --be-name s1lu1
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.
```


インストールされるイメージの構成

この章では、パッケージ発行元の構成、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を示します。

- 55 ページの「発行元の構成」
- 60 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」
- 64 ページの「指定したバージョンへのパッケージのロック」
- 66 ページの「incorporation によって指定されたバージョン制約の緩和」
- 68 ページの「デフォルトのアプリケーション実装の指定」
- 69 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」
- 72 ページの「イメージの更新」
- 76 ページの「イメージと発行元のプロパティの構成」
- 83 ページの「イメージの作成」
- 84 ページの「操作履歴の表示」

発行元の構成

ソフトウェアをインストールし、更新するには、パッケージリポジトリに接続する必要があります。

発行元情報の表示

このイメージに対して構成されているパッケージ発行元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。発行元は、パッケージ FMRI に発行元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。

```
/usr/bin/pkg publisher [-HPn] [-F format] [publisher ...]
```

デフォルトで、solaris 発行元は、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。発行元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/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、[15 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がプロキシ設定されることはありません。値が F になっている HTTP リポジトリは、`$http_proxy` 環境変数が現在設定されている場合 (この場合も `pkg publisher` の出力には F と表示される) を除き、プロキシ設定されていません。値が T になっている HTTP リポジトリは、`pkg set-publisher` 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、そのミラーまたは起点に設定されているプロキシが P 列に表示されます。例については、[49 ページの「非大域ゾーンへのパッケージのインストール」](#)を参照してください。

それらの発行元の詳細構成を表示するには、名前発行元を指定します。

```
$ 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: July 25, 2012 11:40:03 PM
Enabled: Yes
Properties:
    proxied-urls = []
```

発行元の検索順序の先頭の発行元のみを表示する場合は、`-P` オプションを使用します。有効になっている発行元のみを表示する場合は、`-n` オプションを使用します。`-H` オプションは、出力のヘッダーを省略します。

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

パッケージ発行元の追加、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。

- 新しい発行元を構成する。
- 発行元の起点とミラーを設定する。
- 発行元を有効または無効にする。新しく追加された発行元は、デフォルトで有効になります。無効な発行元は、パッケージリストの作成時やインストール、アンインストール、または更新パッケージ操作で使用されません。無効な発行元のプロパティを設定または表示することはできます。1つだけの発行元が有効な場合、その発行元は無効にできません。
- 発行元のスティッキネスを設定する。新しく追加された発行元は、デフォルトで固定です。発行元が非固定の場合は、この発行元からインストールされたパッケージを別の発行元から更新できます。
- 発行元の検索順序を設定する。新しく追加された発行元は、デフォルトで検索順序の最後になります。発行元の検索順序は、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発行元が非固定の場合は、更新するパッケージを検索するために発行元の検索順序が使用されます。

一致するパッケージを提供する最初の発行元がインストール元として使用されません。該当する発行元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発行元からインストールするには、発行元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。

- 発行元の SSL キーおよび証明書を指定する。
- 発行元プロパティを設定または設定解除したり、発行元プロパティ値を追加または削除する。79 ページの「[パッケージの署名プロパティの構成](#)」を参照してください。

`pkg set-publisher` コマンドには 2 つの形式があります。次の形式では、発行元の名前が必要です。

```
/usr/bin/pkg set-publisher [-Ped] [-k ssl_key] [-c ssl_cert]
  [-g origin_to_add | --add-origin origin_to_add ...]
  [-G origin_to_remove | --remove-origin origin_to_remove ...]
  [-m mirror_to_add | --add-mirror mirror_to_add ...]
  [-M mirror_to_remove | --remove-mirror mirror_to_remove ...]
  [--enable] [--disable] [--no-refresh] [--reset-uuid]
  [--non-sticky] [--sticky] [--search-after publisher]
  [--search-before publisher] [--search-first]
  [--approve-ca-cert path_to_CA]
  [--revoke-ca-cert hash_of_CA_to_remove]
  [--unset-ca-cert hash_of_CA_to_remove]
  [--set-property name_of_property=value]
  [--add-property-value name_of_property=value_to_add]
  [--remove-property-value name_of_property=value_to_remove]
```

```
[--unset-property name_of_property_to_delete]
[--proxy proxy_to_use] publisher
```

次の形式では、リポジトリ URI を指定してあるため、発行元の名前はオプションです。

```
/usr/bin/pkg set-publisher -p repo_uri [-Ped]
[-k ssl_key] [-c ssl_cert] [--non-sticky] [--sticky]
[--search-after publisher] [--search-before publisher]
[--search-first] [--approve-ca-cert path_to_CA]
[--revoke-ca-cert hash_of_CA_to_remove]
[--unset-ca-cert hash_of_CA_to_remove]
[--set-property name_of_property=value]
[--add-property-value name_of_property=value_to_add]
[--remove-property-value name_of_property=value_to_remove]
[--unset-property name_of_property_to_delete]
[--proxy proxy_to_use] [publisher]
```

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` という新しい発行元を追加し、この発行元が検索順序の先頭になるように設定します。指定した発行元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。

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

次のコマンドは、`isvpub` 発行元を有効にし、それを検索順序で `devtool` 発行元の前に設定します。

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

指定したリポジトリ URI から発行元の構成情報を取得するには、`-p` オプションを使用します。発行元を指定した場合、一致する発行元のみが追加または更新されます。発行元を指定しない場合、すべての発行元が必要に応じて追加または更新されます。`-p` オプションは、`-g`、`--add-origin`、`-G`、`--remove-origin`、`-m`、`--add-mirror`、`-M`、`--remove-mirror`、`--disable`、`--enable`、`--no-refresh`、または `--reset-uuid` の各オプションとは組み合わせることができません。

発行元の起点 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。新しい起点 URI を追加するには、`-g` オプションを使用します。古い起点 URI を削除するには、`-G` オプションを使用します。

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

次のコマンドは、`solaris` 発行元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発行元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 file:///export/repoSolaris11/
```

```
$ pfexec pkg set-publisher -g http://pkg.oracle.com/solaris/release/ solaris
$ pkg publisher
PUBLISHER          TYPE      STATUS P LOCATION
solaris            origin   online F file:///export/repoSolaris11/
solaris            origin   online F http://pkg.oracle.com/solaris/release/
```

指定した発行元のミラーとして 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、[15 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。指定した発行元のミラーとして URI を削除するには、`-M` オプションを使用します。

```
$ pfexec 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/
```

クライアントの SSL キーを指定するには、`-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、この発行元に対する `pkg publisher` コマンドの出力に一覧表示されます。[55 ページの「発行元情報の表示」](#)を参照してください。

```
$ pfexec pkg set-publisher -k /root/creds/example.key -c /root/creds/example.cert \
--approve-ca-cert /tmp/example_file.pem isvpub
```

指定した証明書を取り消された証明書として扱うには、`--revoked-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、この発行元に対する `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。

`-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、イメージの発行元のリポジトリに接続しません。

このイメージをその発行元に対して識別する新しい一意識別子を選択する場合は、`--reset-uuid` オプションを使用します。

指定された起点 (`-g`) またはミラー (`-m`) の内容の取得元の永続的な Web プロキシ URI を指定するには、`--proxy` オプションを使用します。プロキシ値は、発行元の構成の一部として格納されます。実行時に、`$http_proxy` または関連する環境変数が、このプロキシ設定をオーバーライドします。プロキシ環境変数の詳細は、`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。

発行元を削除するには、`pkg unset-publisher` コマンドを使用します。

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

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。

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、パッケージ内のアクションに設定されるタグです。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。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。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、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やほか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はアン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。

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。

バリエーションタグの例は `variant.arch=sparc` です。ファセットタグの例は `facet.devel=true` です。ファセットとバリエーションは、`facet.` や `variant.` を先頭に付けずに参照されることがよくあります。

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。

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

ほとんどのバリエーションは任意の数の値を設定できます。たとえば、archバリエーションには、i386、sparc、ppc、arm、またはディストリビューションがサポートしているようなアーキテクチャーでも設定できます。(Oracle Solaris では i386 と sparc のみを使用されます。)例外は debug バリエーションです。debug バリエーションは、true または false のみ設定でき、ほかの値では動作が不定になります。ファイル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 バリエーション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
```

バリエーションを使用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 バリエーションは、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.* バリエーションはデフォルトで false です。

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ーションが、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ーションタグのないアクションは常にインストールされません。
- 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せん。
- バリエーションタグのあるアクションは、すべてのバリエーション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ーションタグの両方があるアクションは、ファセットとバリエーション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。

独自のファセットおよびバリエーションタグを作成できます。Oracle Solaris では、次のタグが一般に使用されます。

バリエーション名	取り得る値
variant.arch	sparc、i386
variant.opensolaris.zone	global、nonglobal
variant.debug.*	true、false

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
```

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。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。

```
/usr/bin/pkg variant [-H] [variant_nsmr ...]
```

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

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。

```
/usr/bin/pkg change-variant [-nvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  variant_name=value ...
```

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、[76 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。

大域ゾーンと同時に n 個の非大域ゾーンのバリエーションを変更するには、`-c` オプションを使用します。`-c` オプションの使用例については、52 ページの「複数の非大域ゾーンの同時更新」を参照してください。

`-n` オプションを使用すると、実際の変更を行わずに、この操作を `-n` なしで実行した場合に何が変更されるかを確認できます。

```
$ pfxec 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

設定されているファセットの値を表示するには、`pkg facet` コマンドを使用します。

```
/usr/bin/pkg facet [-H] [facet_name ...]
```

```
$ pkg facet
FACETS          VALUE
facet.locale.en_US True
facet.locale.en True
facet.locale.*  False
$ pkg facet -H 'facet.locale.*'
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。

```
/usr/bin/pkg change-facet [-nvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  facet_name=[True|False|None] ...
```

大域ゾーンと同時に n 個の非大域ゾーンのファセットを変更するには、`-c` オプションを使用します。`-c` オプションの使用例については、52 ページの「複数の非大域ゾーンの同時更新」を参照してください。

`-n` オプションを使用すると、実際の変更を行わずに、この操作を `-n` なしで実行した場合に何が変更されるかを確認できます。

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せん。この操作には、長い時間を必要とする可能性があります。このイメージとパッケージリポジトリ間で大量のネットワークトラフィックが発生する可能性があります。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、[76 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。

```
$ pfexec 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。パッケージを凍結するときの 1 つの例は、大域ゾーンの更新時に、非大域ゾーン内のパッケージを更新しない場合です。

```
/usr/bin/pkg freeze [-n] [-c reason] [pkg_fmri_pattern] ...
```

`pkg_fmri_pattern` で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。`pkg_fmri_pattern` 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。

`pkg_fmri_pattern` に指定された発行元を使用して、一致するパッケージが検索されません。ただし、発行元の情報は凍結の一環として記録されません。パッケージは発行元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、バージョン、パッケージがいつ凍結されたか、関連付けられた理由があればその理由)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。

パッケージが凍結されている理由を記録するには、`-c` オプションを使用します。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。

操作を試しに実行し、凍結されるパッケージの一覧を表示しますが、実際にはどのパッケージも凍結しない場合は、`-n` オプションを使用します。

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。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。

```
$ pfexec 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、凍結に関するメッセージが表示されます。

```
$ pfexec 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。制約を緩和するには、`pkg unfreeze` コマンドを使用します。

```
/usr/bin/pkg unfreeze [-n] [pkg_name_pattern] ...
```

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します。バージョンを提供しても無視されます。

凍結解除を試しに実行し、凍結解除されるパッケージの一覧を表示しますが、実際にはどのパッケージも凍結解除しない場合は、`-n` オプションを使用します。

incorporation によって指定されたバージョン制約の緩和

Oracle Solaris 11 OS に含まれるどのパッケージにも、incorporation パッケージへの依存関係があります。incorporation パッケージは、incorporation 対象のパッケージのバージョンを制約して、更新にわたり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 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。そのような incorporation 対象のパッケージには、incorporation パッケージで `version-lock.pkg_name` ファセット属性が指定されません。`version-lock.pkg_name` 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.pkg_name` 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 1.0.0.10-0.175.1.0.0.19.0  i--
library/security/openssl 1.0.0.10-0.175.1.0.0.18.0  ---
$ pfexec pkg update library/security/openssl@1.0.0.10-0.175.1.0.0.18.0
Creating Plan (Solver setup): |
pkg update: No matching version of library/security/openssl can be installed:
Reject: pkg:///solaris/library/security/openssl@1.0.0.10,5.11-0.175.1.0.0.18.0:20120611T201116Z
Reason: This version is excluded by installed incorporation
pkg:///solaris/consolidation/userland/userland-incorporation@0.5.11,5.11-0.175.1.0.0.19.0:20120625T163952Z
```

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、76 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。

```
$ pfexec 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
$ pfexec pkg update library/security/openssl@1.0.0.10-0.175.1.0.0.18.0
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  1.0.0.10-0.175.1.0.0.18.0  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。

```

$ pfmexec 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 list library/security/openssl
NAME (PUBLISHER)      VERSION                                IFO
library/security/openssl  1.0.0.10-0.175.1.0.0.18.0  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` ファセットを `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。

```

$ pfmexec pkg install editor/hexedit@1.2.12-0.175.1.0.0.21.0
Creating Plan (Solver setup): -
pkg install: No matching version of editor/hexedit can be installed:
  Reject: pkg://solaris/editor/hexedit@1.2.12,5.11-0.175.1.0.0.21.0:20120723T170720Z
  Reason: All versions matching 'require' dependency
  pkg://system/library@0.5.11,5.11-0.175.1.0.0.20.0 are rejected
  Reject: pkg://solaris/system/library@0.5.11,5.11-0.175.1.0.0.20.0:20120709T163421Z

pkg://solaris/system/library@0.5.11,5.11-0.175.1.0.0.21.0:20120723T163000Z
Reason: This version is excluded by installed incorporation
pkg://solaris/consolidation/osnet/osnet-incorporation@0.5.11,5.11-0.175.1.0.0.19.0:20120625T152525Z

```

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` ファセットを `false` に設定すると、`incorporation` に

組み込まれているパッケージの同期は保ちながら、incorporation をシステムの残りの部分からロック解除することができます。

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を同じイメージで提供場合があります。アプリケーションの複数のバージョンが同じメディエーションに含まれて提供されていれば、デフォルトまたは優先のバージョンを簡単に設定し直すことができます。「メディエーション」とは、あるアプリケーションの異なる実装へのリンクのセットで、各リンクは同じメディエータ名と同じリンクパスを持ちますが、ターゲットリンクパスが異なります。

イメージ内のすべてのメディエータを表示するには、あるいは、指定したメディエータの現在選択されているバージョンを表示するには、`pkg mediator` コマンドを使用します。

```
/usr/bin/pkg mediator [-aH] [-F format] [mediator ...]
```

指定したメディエータ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。

```
usr/bin/pkg set-mediator [-nv] [-I implementation]
  [-V version] [--no-be-activate]
  [--no-backup-be | --require-backup-be]
  [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  mediator ...
```

次の例では、2つの異なるバージョンの Java Runtime Environment がインストールされています。

```
$ pkg list 'runtime/java*'
NAME (PUBLISHER)      VERSION                               IFO
runtime/java         1.6.0.33-0.175.1.0.0.18.1          i--
runtime/java/jre-6   1.6.0.33-0.175.1.0.0.18.1          i--
runtime/java/jre-7   1.7.0.5-0.175.1.0.0.18.0           i--
```

次のコマンドでは、java メディエーションが定義されていること、およびバージョン 1.7 がデフォルトのバージョンであることが示されています。

```
$ pkg mediator
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java     system  1.7   system
php      system  5.2   system
python   vendor  2.6   vendor
```

次のコマンドは、設定できるすべての java メディエーションを表示します。jre-6 と jre-7 の両方に、`/usr/bin/java` からの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、`/usr/bin/java` リンクのターゲットは `jdk1.6` です。jre-7

パッケージでは、`/usr/bin/java` リンクのターゲットは `jdk1.7` です。前のコマンドは、現在 `/usr/bin/java` リンクのターゲットはバージョン 1.7であることを示しています。このメディエーションでは優先実装は指定されておらず、パッケージシステムはより高いバージョンの実装を優先実装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
```

次のコマンドは、バージョン 1.6 を優先実装に設定する方法を示しています。これは、`/usr/bin/java` を呼び出すと JRE version 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。

```
$ pkg mediator java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java     system  1.7     system
$ pfexec 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
```

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。

```
/usr/bin/pkg avoid [pkg_fmri_pattern ...]
```

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。

`pkg_fmri_pattern` 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。

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ます。その依存関係が削除された場合、パッケージはアンインストールされます。

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

```
/usr/bin/pkg unavoid [pkg_fmri_pattern ...]
```

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、この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。

次のコマンドの出力は、`group/feature/amp` グループパッケージがインストールされておらず、そのグループパッケージに含まれているパッケージはどれもインストールされていないことを示しています。これらの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ている可能性があります。

```
$ pkg list -a group/feature/amp
NAME (PUBLISHER)      VERSION                IFO
group/feature/amp    0.5.11-0.175.1.0.0.21.0  ---
$ pkg list -a 'pkg contents -o fmri -H -rt depend -a type=group group/feature/amp'
NAME (PUBLISHER)      VERSION                IFO
database/mysql-51     5.1.37-0.175.1.0.0.21.0  ---
web/php-52            5.2.17-0.175.1.0.0.21.0  ---
web/php-52/extension/php-apc  3.0.19-0.175.1.0.0.21.0  ---
web/php-52/extension/php-mysql  5.2.17-0.175.1.0.0.21.0  ---
web/server/apache-22  2.2.22-0.175.1.0.0.21.0  ---
web/server/apache-22/module/apache-dtrace  0.3.1-0.175.1.0.0.21.0  ---
web/server/apache-22/module/apache-fcgid  2.3.6-0.175.1.0.0.21.0  ---
web/server/apache-22/module/apache-php5    5.2.17-0.175.1.0.0.18   --r
```

次のコマンドは、このグループパッケージに属するパッケージの1つを回避リストに登録します。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。

```
$ pfexec pkg avoid web/server/apache-22/module/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。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。

```
$ pfexec pkg install group/feature/amp
...
$ pkg list -a 'pkg contents -o fmri -H -rt depend -a type=group group/feature/amp'
NAME (PUBLISHER)                                VERSION                                IFO
database/mysql-51                               5.1.37-0.175.1.0.0.21.0             i--
web/php-52                                       5.2.17-0.175.1.0.0.21.0             i--
web/php-52/extension/php-apc                    3.0.19-0.175.1.0.0.21.0             i--
web/php-52/extension/php-mysql                  5.2.17-0.175.1.0.0.21.0             i--
web/server/apache-22                             2.2.22-0.175.1.0.0.21.0             i--
web/server/apache-22/module/apache-dtrace       0.3.1-0.175.1.0.0.21.0             i--
web/server/apache-22/module/apache-fcgid        2.3.6-0.175.1.0.0.21.0             ---
web/server/apache-22/module/apache-php5        5.2.17-0.175.1.0.0.18              i-r
$ 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。

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

```
$ pfexec pkg avoid web/server/apache-22/module/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
$ pfexec pkg uninstall web/server/apache-22/module/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。

```
$ pfexec 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。

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

イメージの更新

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を、インストール済みパッケージの依存関係および発行元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するには、*pkg-fmri* を指定せずに、または *pkg-fmri* としてアスタリスク文字 (*) を付けて、`pkg update` コマンドを使用します。現在のイメージに非大域ゾーンがマウントされている場合は、これらのゾーンも更新されます。52 ページの「複数の非大域ゾーンの同時更新」を参照してください。

```
/usr/bin/pkg update [-fnvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-index] [--no-refresh] [--no-be-activate]
  [--no-backup-be | --require-backup-be] [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  [--reject pkg_fmri_pattern ...] [pkg_fmri_pattern ...]
```

イメージ更新のベストプラクティス

`pkg update` コマンドを使用する前に、構成済み発行元起点から入手可能なバージョンを確認し、実際に更新を実行せずに `-nv` オプションを使用して更新されるパッケージの一覧を表示します。

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、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 発行元から 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、13 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 発行元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発行元から更新されます。発行元のスティッ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発行元を制御することができます。[57 ページ](#)の「[パッケージ発行元の追加、変更、削除](#)」を参照してください。

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、実際には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。

```
$ pfexec 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 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 に属し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 に属し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発行元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たり、新しいパッケージがインストールされたりする可能性があります。

前の例は、-n オプションなしでこのコマンドを再実行した場合、この更新で新しい 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、[76 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。この動作を希望しない場合は、pkg update コマンドで --no-be-activate オプションを使用します。そのあと、新しい環境を使用する準備ができれば、beadm activate コマンドを使用して新しい BE をアクティブにします。

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。

```
$ pfexec 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が entire incorporation に属していない場合があります。これらのパッケージは個別にインストールされた可能性があるため、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。

```
$ pfexec 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、64 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。

使用可能なパッケージの制約

インストールまたは更新できるバージョンを制御するための別の方法として、独自のローカル IPS パッケージリポジトリを指定し、そのリポジトリの内容を制御します。たとえば、独自のリポジトリには Oracle Solaris 11.1 パッケージを除く、Oracle Solaris 11 11/11 のすべてのサポート更新を含めることができます。

独自のリポジトリを作成する場合は、必ず solaris 発行元起点を更新してください。

詳細は、『Oracle Solaris 11.1 パッケージリポジトリのコピーおよび作成』を参照してください。

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。

その他の更新コマンドオプション

インストールされているすべてのパッケージを更新する場合に、-f オプションを指定すると、クライアントの最新状態チェックが実行されません。

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。-g オプションの効果の詳細と例については、40 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。

大域ゾーンと同時に n 個の非大域ゾーンを更新するには、-c オプションを使用します。例については、52 ページの「複数の非大域ゾーンの同時更新」を参照してください。

更新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しないと、パッケージのライセンスに同意が必要になった場合、更新操作は失敗します。この操作の一環として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。

`-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、イメージの発行元のリポジトリに接続しません。

`--no-index` オプションを指定すると、操作の正常終了後に、検索インデックスが更新されません。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。更新操作が終了したら、`pkg refresh` を使用して、使用可能なパッケージと、指定した各発行元の発行元メタデータのリストを更新できます。発行元を指定しない場合、すべての発行元を対象にリフレッシュが実行されます。新しい BE が作成された場合、新しい BE でこの発行元のリフレッシュを実行します。

イメージと発行元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、イメージと発行元のプロパティと、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「Image Properties」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更新する。これは、インストール、アンインストール、更新、パリアントの変更、またはファセットの変更を行うときに発生する可能性があります。

`pkg update` コマンドを実行して、利用可能な更新があるすべてのパッケージを更新すると、多くの場合、新しい BE が作成されます。

- `--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 内のパッケージを更新しますが、現在の BE 内のパッケージは更新しません。

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。

新しい BE が必要な一方で、新しい BE 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 11.1 ブート環境の作成と管理](#)』を参照してください。

下に説明するイメージプロパティの設定の手順については、[82 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、リブ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発行元プロパティを、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発行元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。

<code>ignore</code>	すべてのマニフェストの署名を無視します。
<code>verify</code>	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。
<code>require-signatures</code>	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、 <code>pkg fix</code> および <code>pkg verify</code> コマンドでも警告が表示されます。
<code>require-names</code>	<code>require-signatures</code> と同じ要件に従いますが、 <code>signature-required-names</code> 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発行元プロパティ

次の発行元プロパティを、特定の発行元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発行元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発行元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

このイメージの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。

特定の発行元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。

```
$ pfexec 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。

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

次の例は、このイメージで、発行元 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。

```
$ pfexec 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。

```
$ pfexec pkg add-property-value signature-require-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。

```
$ pfexec pkg remove-property-value signature-require-names trustedname
```

次の例は、指定した発行元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発行元 example.com の共通名の一覧に文字列 trustedname を追加します。

```
$ pfexec pkg set-publisher --add-property-value \  
signature-require-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。

flush-content-cache-on-success

True に設定すると、パッケージクライアントは、インストールまたは更新操作の完了時にその内容キャッシュ内のファイルを削除します。更新操作の場合、内容はソース BE からのみ削除されます。出力先 BE でパッケージ操作が次に発生したとき、このオプションが変更されていなければ、パッケージクライアントはその内容キャッシュをフラッシュします。

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、イメージおよび発行元の構成のソースとして、および提供された発行元と通信するためのプロキシとしてイメージで使用す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

このイメージの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。

```
/usr/bin/pkg property [-H] [propname ...] /usr/bin/pkg set-property propname propvalue /usr/bin/pkg
add-property-value propname propvalue /usr/bin/pkg remove-property-value propname propvalue /usr/bin/
pkg unset-property propname ...
```

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
```

`preferred-authority` および `publisher-search-order` プロパティは、`pkg set-publisher` コマンドオプションを使用して設定できます。[57 ページ](#)の「[パッケージ発行元の追加、変更、削除](#)」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。

```
$ pfexec 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。

```
$ pfexec 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。

イメージには次の 3 つの種類があります。

- フルイメージは完全なシステムを提供することができます。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。
- 部分イメージはフルイメージ(親イメージ)にリンクされますが、それ自体では完全なシステムを提供しません。非大域ゾーンは部分イメージです。-z または --zone オプションを使用して、適切なバリエーションを設定します。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。非大域ゾーンについては、『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』のパート II 「Oracle Solaris ゾーン」を参照してください。
- 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。

```
/usr/bin/pkg image-create [-FPUfz] [--force]
  [--full | --partial | --user] [--zone]
  [-k ssl_key] [-c ssl_cert] [--no-refresh]
  [--variant variant_name=value ...]
  [-g path_or_uri | --origin path_or_uri ...]
  [-m uri | --mirror uri ...]
  [--set-property name_of_property=value]
  [--facet facet_name=(True|False) ...]
  [(-p | --publisher) [name=]repo_uri] dir
```

dir によって指定された場所に、パッケージ操作に適したイメージを作成します。デフォルトのイメージタイプはユーザー (-U または -user) です。イメージタイプはフルイメージ (-F または --full)、または指定された *dir* パスを包含するフルイメージにリンクされた部分イメージ (-P または *partial*) に設定できます。

非大域ゾーンコンテキストで新しいイメージを実行するには、-z または --zone オプションを使用して、適切なバリエーションを設定します。

パッケージリポジトリの URI は、-p または --publisher オプションを使用して提供する必要があります。発行元の名前も提供した場合、イメージの作成時にその発行元のみが追加されます。発行元の名前を提供しない場合、指定されたリポジトリによって認識されているすべての発行元がイメージに追加されます。この発行元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。

追加の起点を指定する場合は、`-g` オプションを使用します。ミラーを指定する場合は、`-m` オプションを使用します。

クライアント SSL 認証を使用する発行元の場合、`-c` オプションまたは `-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発行元のために使用されます。

既存のイメージを上書きするイメージを強制的に作成する場合は、`-f` オプションを使用します。このオプションは慎重に使用してください。

`-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、イメージの発行元のリポジトリに接続しません。

指定したバリエーションを指示した値に設定する場合は、`--variant` オプションを使用します。指定したファセットを指示した値に設定する場合は、`--facet` オプションを使用します。

操作履歴の表示

現在のイメージ内のコマンド履歴を表示するには、`pkg history` コマンドを使用します。

```
/usr/bin/pkg history [-HNl] [-t [time | time-time],...]
                    [-o column,...] [-n number]
```

コマンドの結果、コマンドが完了した時間、使用されたクライアントのバージョンと名前、操作を実行したユーザーの名前、コマンドの実行中に発生したエラーを含む詳細情報を表示するには、`-l` オプションを使用します。

指定した数の最新の操作のみを表示するには、`-n` オプションを使用します。

```
$ pkg history -n4
START                OPERATION                CLIENT                OUTCOME
2012-08-06T16:32:03  fix                    pkg                  Succeeded
2012-08-06T16:41:47  revert                 pkg                  Succeeded
2012-08-06T17:56:22  set-property          pkg                  Succeeded
2012-08-06T17:56:53  unset-property        pkg                  Succeeded
```

列名の指定したコンマ区切りリストを使用して出力を表示する場合は、`-o` オプションを使用します。[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
2012-08-06T16:32:03  0:00:27            fix                    Succeeded
2012-08-06T16:41:47  0:00:43            revert                 Succeeded
2012-08-06T17:56:22  0:00:00            set-property          Succeeded
2012-08-06T17:56:53  0:00:00            unset-property        Succeeded
```

%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でレコードを記録する場合は、`-t` オプションを使用します (`strftime(3C)` 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、現在の日時の別名として使用できます。指定されたタイムスタンプに、重複したタイムスタンプまたは重複する日付範囲が含まれる場合、重複した各履歴イベントの1つのインスタンスのみが出力されます。

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。

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。

```
$ pfexec pkg purge-history
```

