Oracle® Solaris 11 ソフトウェアパッケージの追加および更新

Part No: E25785
2012 年 3 月、E25785-03

120411@25097

目次

はじめに

『Oracle Solaris 11 ソフトウェアパッケージの追加および更新』では、Oracle Solaris Image Packaging System (IPS) 機能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 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。

- 『Image Packaging System のマニュアルページ』
- 『Oracle Solaris 11 パッケージリポジトリのコピーおよび作成』
- 『Oracle Solaris 11 ブート環境の作成と管理』
- 『Oracle Solaris 11 システムのインストール』

Oracle サポートへのアクセス

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

表記上の規則

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

表 P–1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% **su** password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm *filename* と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。

表 P–1 表記上の規則 *(続き)*

字体または記号	意味	例
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	`sun% grep '^#define \` `XV_VERSION_STRING'`

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

- C シェル

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

- C シェルのスーパーユーザー

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

- Bash シェル、Korn シェル、および Bourne シェル

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

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

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

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

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

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

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

◆ ◆ ◆ 第 1 章

1 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ィッキネスなどの、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ーを設定します。
- 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 システムのインストール』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。

- `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。Software Installation 権利プロファイルがある場合、`pfexec` コマンドを使用して、パッケージをインストールおよび更新できます。

  ```
  $ pfexec pkg install editor/gnu-emacs
  ```

  System Administrator 権利プロファイルなど、その他の権利プロファイルもインストール権限を提供します。
- 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。

  ```
  $ sudo pkg install editor/gnu-emacs
  ```
- `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 役割を持つ場合、root パスワードで `su` コマンドを使用して、root 役割になることができます。

IPS の概念

ここでは、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。

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列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。

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。

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省略可能です。 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。

書式:

scheme://*publisher*/*package_name*@*version*:*date*T*time*Z

例:

```
pkg://solaris/editor/vim@7.3.254,5.11-0.174.0.0.0.0.504:20110921T002716Z
```

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　`editor/vim`

パッケージ名前空間は階層的で、任意の深さです。IPS コマンドでは、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指定できます。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	`7.3.254` 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、これは通常、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	`0.174.0.0.0.0.504` 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。
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、ISO-8601 基本形式 (*YYYYMMDD*T *HHMMSS*Z) のパッケージが発行された日時です。

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。パッケージ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、56 ページの「ブート環境ポリシーイメージのプロパティー」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アント

ソフトウェアには、省略可能なコンポーネントや、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。省略可能な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、省略可能な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アントと呼びます。

ファセットとバリアントはイメージの特殊なプロパティーであり、個々のパッケージには設定できません。

パッケージマニフェスト内の各アクションには、ファセットおよびバリアントタグを指定することができます。1 つのアクションに複数のファセットタグおよびバリアントタグを付けることができます。

アクションのファセットおよびバリアント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アントの値の比較により、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。

- ファセットまたはバリアントタグのないアクションは常にインストールされます。
- ファセットタグのあるアクションは、イメージ上のタグに一致するすべてのファセットまたはファセットパターンが `false` に設定されていないかぎり、インストールされます。いずれかのファセットが `true` に設定されているか、明示的に設定されていない (`true` はデフォルト) 場合、アクションがインストールされます。
- バリアントタグのあるアクションは、すべてのバリア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アントタグの両方があるアクションは、ファセットとバリア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アントの値を表示または変更するには、50 ページの「省略可能な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。

◆ ◆ ◆ 第 2 章

2 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	**# packagemanager**

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
`--image-dir` または `-R` *dir*	*dir* 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、`/aux0/example_root` に格納されたイメージを操作します。 `# packagemanager -R /aux0/example_root`
`--update-all` または `-U`	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、20 ページの「Update Manager の使用」を参照してください。
`--info-install` または `-i` *file*`.p5i`	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 `.p5i` ファイルを指定します。指定するファイルの拡張子が `.p5i` である必要があります。詳細は、18 ページの「Web Install の使用」を参照してください。
`--help` または `-h`	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

ト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リティー保護された発行元である場合、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ーの「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ー 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。

  ```
  # packagemanager --update-all
  ```
- `pkg` コマンドを使用します。

  ```
  # pkg update
  ```

次のいずれかの方法で 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**

「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更新します。
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/packagemanger
   package/pkg/updatemanager
   ```

   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
`--image-dir` または `-R` *dir*	*dir* 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、`/aux0/example_root` にあるイメージを更新します。 **`# pm-updatemanager -R /aux0/example_root`**
`--help` または `-h`	コマンドの使用方法に関する情報を表示します。

◆ ◆ ◆ 第 3 章

ソフトウェアパッケージに関する情報の取得

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。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アントは表示されません。

```
/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 (isv.com)              1.0                      i--
system/dtrace/dtrace-toolkit      0.99-0.174.0.0.0.0.17765 i--
```

丸括弧で囲まれた発行元名は、isv.com 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す。

I 列内の「i」は、これらのパッケージがこのイメージにインストールされていることを示します。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           3.0.15-0.174.0.0.0.0.0     ---
isvtoolkit (isv.com)              1.0                        i--
system/dtrace/dtrace-toolkit      0.99-0.174.0.0.0.0.17765 i--
```

この出力は image/nvidia/cg-toolkit 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           3.0.15-0.174.0.0.0.0.0     ---
isvtoolkit (isv.com)              1.0                        i--
system/dtrace/dtrace-toolkit      0.99-0.174.0.0.0.0.17765 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変更されています。web/amp パッケージの名前が group/feature/amp に変更されており、group/feature/amp がインストールされていないため、パッケージは -f オプションを使用しないで表示されます。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、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                 0.5.11-0.174.0.0.0.0.2559 ---
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。その情報を取得するには、26 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 コマンドを使用します。

-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                 5.1.37-0.174.0.0.0.0.504 ---
```

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、48 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    1.8.4-0.174.0.0.0.0.504  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@5.1.37,5.11-0.174.0.0.0.0.504:20110920T230125Z ---
```

-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。

-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。

```
/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: 0.174.0.0.0.0.2559
Packaging Date: Wed Sep 21 19:12:55 2011
          Size: 5.45 kB
          FMRI: pkg://solaris/group/feature/amp@0.5.11,5.11-0.174.0.0.0.0.2559:20110921T191255Z
```

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: Fri Aug 26 14:55:51 2011
         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 -r --license x11/server/xorg
Copyright (c) 2011, Oracle and/or its affiliates. All rights reserved.
The following software...
```

`-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。

```
/usr/bin/pkg contents [-Hmr] [-a attribute=pattern ...] [-g path_or_uri ...]
    [-o attribute ...] [-s sort_key] [-t action_type ...] [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 e1000g
PATH
kernel
kernel/drv
kernel/drv/amd64
kernel/drv/amd64/e1000g
kernel/drv/e1000g.conf
usr/share/man/man7d
usr/share/man/man7d/e1000g.7d
```

`-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。

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。

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

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

```
# pkg contents -t file -o owner,group,mode,pkg.size,path e1000g
OWNER GROUP MODE PKG.SIZE PATH
root  sys   0755   420912 kernel/drv/amd64/e1000g
root  sys   0644     4238 kernel/drv/e1000g.conf
root  bin   0444       20 usr/share/man/man7d/e1000.7d
root  bin   0444    12813 usr/share/man/man7d/e1000g.7d
```

パッケージマニフェストを表示した場合、e1000g パッケージに 7 つ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 3 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あり、デバッグファイルを含みません。x86 アーキテクチャーのデバッグファイルは上に表示されず、SPARC アーキテクチャーに対しては、デバッグファイルもデバッグファイル以外のファイルも表示されません。イメージにデバッグファイルを含めるかどうかを変更するには、デバッグイメージバリアントを変更します。50 ページの「省略可能な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。 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 -o fmri -H -rt depend -a type=group solaris-desktop
archiver/gnu-tar
audio/audio-utilities
...
```

-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。

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-accept,must-display,license,pkg.name *incorporation
MUST-ACCEPT MUST-DISPLAY LICENSE                          PKG.NAME
true        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 [-HIaflpr] [-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、pkg search クエリー用語は大文字と小文字を除いて正確に一致します。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。

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_type*: *key*:*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* フィールドでは明示的ワイルドカードがサポートされます。*action_type* および

key は正確に一致する必要があります。パッケージのアクションとキー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。

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。

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 -l -H -o pkg.name /lib/libpower.so.1
system/kernel/power
```

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 ':set:info.classification:Source Code Management'
INDEX               ACTION VALUE                                PACKAGE
info.classification set    Development/Source Code Management pkg:/developer/versioning/subversion@1.6.16
info.classification set    Development/Source Code Management pkg:/developer/versioning/git@1.7.3.2-0.174
info.classification set    Development/Source Code Management pkg:/developer/versioning/sccs@0.5.11-0.174
info.classification set    Development/Source Code Management pkg:/library/perl-5/subversion@1.6.16-0.174
info.classification set    Development/Source Code Management pkg:/library/java/subversion@1.6.16-0.174.0
info.classification set    Development/Source Code Management pkg:/library/python-2/subversion@1.6.16-0.1
info.classification set    Development/Source Code Management pkg:/developer/xopen/xcu4@0.5.11-0.174.0.0.
info.classification set    Development/Source Code Management pkg:/developer/quilt@0.47-0.174.0.0.0.0.504
info.classification set    Development/Source Code Management pkg:/developer/versioning/cvs@1.12.13-0.174
info.classification set    Development/Source Code Management pkg:/developer/versioning/mercurial@1.8.4-0
```

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 -o pkg.name -H ':set:info.classification:Source Code Management'
developer/versioning/subversion
developer/versioning/git
```

```
developer/versioning/sccs
library/perl-5/subversion
library/java/subversion
library/python-2/subversion
developer/xopen/xcu4
developer/quilt
developer/versioning/cvs
developer/versioning/mercurial
```

依存パッケージの表示

次の例は、指定したパッケージに依存す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 -l -H -o pkg.name 'depend: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@1.10.99 への exclude 依存関係があることを示しています。

```
$ pkg search -l -o pkg.name,fmri 'depend:exclude:'
PKG.NAME                                FMRI
x11/server/xorg/driver/xorg-video-ati   pkg:/x11/server/xorg@1.10.99
x11/server/xorg/driver/xorg-video-intel pkg:/x11/server/xorg@1.10.99
x11/server/xvnc                         pkg:/x11/server/xorg@1.10.99
desktop/remote-desktop/tigervnc         pkg:/x11/server/xorg@1.10.99
x11/server/xserver-common               pkg:/x11/server/xorg@1.10.99
...
```

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。 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 -o fmri -H '*/solaris-desktop:depend:group:'
archiver/gnu-tar
audio/audio-utilities
...
```

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-desktop パッケージに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グループ依存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アントを含めることができます。この検索からの出力を、28 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。

◆ ◆ ◆ 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

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設定する、パッケージの署名プロパティーを設定するなどのイメージの設定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。この章で示す手順と結果では、デフォルトのイメージ構成を前提としています。

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

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、12 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。

```
# pkg install -nv group/feature/amp
           Packages to install: 8
     Estimated space available: 112.19 GB
Estimated space to be consumed: 452.42 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 -> 5.1.37,5.11-0.174.0.0.0.0.504:20110920T230125Z
  group/feature/amp
    None -> 0.5.11,5.11-0.174.0.0.0.0.2559:20110921T191255Z
  web/php-52
    None -> 5.2.17,5.11-0.174.0.0.0.0.504:20110921T041858Z
  web/php-52/extension/php-apc
    None -> 3.0.19,5.11-0.174.0.0.0.0.504:20110921T041245Z
  web/php-52/extension/php-mysql
    None -> 5.2.17,5.11-0.174.0.0.0.0.504:20110921T041411Z
  web/server/apache-22/module/apache-dtrace
    None -> 0.3.1,5.11-0.174.0.0.0.0.504:20110921T042357Z
  web/server/apache-22/module/apache-fcgid
    None -> 2.3.6,5.11-0.174.0.0.0.0.504:20110921T042430Z
  web/server/apache-22/module/apache-php5
    None -> 5.2.17,5.11-0.174.0.0.0.0.504:20110921T042738Z
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、56 ページの「ブート環境ポリシーイメージのプロパティー」を参照してください。

```
# 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(5) 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するか、ファイルを元に戻した場合に、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。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、56 ページの「ブート環境ポリシーイメージのプロパティー」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、56 ページの「ブート環境ポリシーイメージのプロパティー」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、56 ページの「ブート環境ポリシーイメージのプロパティー」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。

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、45 ページの「発行元の構成」を参照してください。

```
/usr/bin/pkg install [-nvq] [-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 ...
```

特定の発行元からのパッケージをインストールするには、*pkg_fmri_pattern* に発行元名を指定します。次の例では、`isv.com` が発行元の名前です。

```
# pkg install pkg://isv.com/developer/isvtool
```

パッケージの特定のバージョンをインストールするには、*pkg_fmri_pattern* にバージョン情報を指定します。

```
# pkg list -avH vim
pkg://solaris/editor/vim@7.3.254,5.11-0.174.0.0.0.0.504:20110921T002716Z    ---
# pkg install vim@7.3.254,5.11-0.174
```

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。

```
# pkg install vim@latest
```

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。`install` または `update` の実行後に、発行元によって提供され、イメージ内に見つからない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。

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更新が実行されます。

`-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。

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、56 ページの「ブート環境ポリシーイメージのプロパティー」を参照してください。

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

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

新しい 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。

```
# pkg install --be-name s11amp group/feature/amp
           Packages to install:   8
       Create boot environment: Yes
Create backup boot environment:  No

DOWNLOAD                                  PKGS       FILES    XFER (MB)
Completed                                  8/8     640/640    70.9/70.9

PHASE                                       ACTIONS
Install Phase                               942/942

PHASE                                         ITEMS
Package State Update Phase                    8/8
Image State Update Phase                      2/2

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

A clone of solaris-174 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」BE は現在ブートしており、「R」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 2011-09-23 13:58
solaris      -      -          44.81M  static 2010-11-07 17:45
solaris-151a -      -          158.12M static 2010-11-12 14:37
solaris-174  N      /          30.04M  static 2011-09-02 12:38
```

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。

```
# beadm mount s11amp /mnt
# pkg -R /mnt list group/feature/amp
NAME (PUBLISHER)       VERSION                        IFO
group/feature/amp      0.5.11-0.174.0.0.0.0.2559    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 2011-09-23 13:58
solaris      -      -          44.81M  static 2010-11-07 17:45
solaris-151a -      -          158.12M static 2010-11-12 14:37
solaris-174  N      /          30.05M  static 2011-09-02 12:38
# beadm unmount s11amp
# beadm list
BE           Active Mountpoint Space   Policy Created
--           ------ ---------- -----   ------ -------
s11amp       R      -          20.75G  static 2011-09-23 13:58
solaris      -      -          44.81M  static 2010-11-07 17:45
solaris-151a -      -          158.12M static 2010-11-12 14:37
solaris-174  N      /          30.06M  static 2011-09-02 12:38
```

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、49 ページの「回避するパッケージのマーキング」を参照してください。

```
# pkg install -nv --reject cvs 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、45 ページの「発行元の構成」を参照してください。

```
/usr/bin/pkg update [-fnvq] [-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。

```
# 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。install または update の実行後に、発行元によって提供され、イメージ内に見つからない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。

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更新が実行されます。

pkg update コマンドを *pkg-fmri* を指定しないで使用するか、指定した *pkg-fmri* がアスタリスク文字 (*) である場合、使用可能な更新があるインストールされたすべてのパッケージが更新されます。使用可能な更新があるパッケージのリストを表示するには、pkg list -u コマンドを使用します。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。

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 -l -Ho 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 ...]
```

関連するパブリッシャーの現在の署名ポリシーが ignore でない場合、各パッケージの署名がポリシーに基づいて検証されます。署名ポリシーが適用されるしくみについては、57 ページの「パッケージの署名のプロパティー」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。

```
# 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。

```
# 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: 2011-09-28-05:34:02
Repairing: pkg://solaris/system/install/auto-install/auto-install-common

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

PHASE                                        ACTIONS
Update Phase                                     1/1

PHASE                                          ITEMS
Image State Update Phase                         2/2
```

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。

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] [--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、49 ページの「回避するパッケージのマーキング」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更新が実行されます。

◆ ◆ ◆ 第 5 章

5 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設定、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、BE ポリシーの設定などのイメージ全体に適用する特性を設定する方法を示します。

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できる必要があります。

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] [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  URI
solaris                      origin  online  http://pkg.oracle.com/solaris11/release/
isv.com        (non-sticky) origin  online  file:/export/isv-repo/
example.com    (disabled)   origin  online  http://pkg.example.com/
```

それらの発行元の詳細構成を表示するには、名前で発行元を指定します。

```
$ pkg publisher solaris
       Publisher: solaris
           Alias:
      Origin URI: http://pkg.oracle.com/solaris11/release/
         SSL Key: None
        SSL Cert: None
     Client UUID: 00000000-3db4-fcc2-0111-000000000000
 Catalog Updated: Thu Sep 22 21:06:03 2011
         Enabled: Yes
Signature Policy: verify
```

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。

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ー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。
- 発行元の SSL キーおよび証明書を指定する。
- 発行元プロパティーを設定または設定解除したり、発行元プロパティー値を追加または削除する。59 ページの「パッケージの署名プロパティーの構成」を参照してください。

```
/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 ...]
    [-p repo_uri] [--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]
[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 data.com 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。

```
# pkg set-publisher -P -g http://pkg.data.com/release/ data.com
```

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

```
# pkg set-publisher --enable --search-before isv.com example.com
```

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。パブリッシャーを指定した場合、一致するパブリッシャーのみが追加または更新されます。パブリッシャーを指定しない場合、すべてのパブリッシャーが必要に応じて追加または更新されます。-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。

```
# pkg set-publisher -G '*' -g http://pkg.example.com/support/ example.com
```

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。

```
# pkg set-publisher -m http://pkg.data.com/release2/ data.com
# pkg publisher
PUBLISHER                   TYPE    STATUS   URI
data.com                    origin  online   http://pkg.data.com/release/
data.com                    mirror  online   http://pkg.data.com/release2/
```

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。45 ページの「発行元情報の表示」を参照してください。

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

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。

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

```
# pkg unset-publisher isv.com
```

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す。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す。ただし、パブリッシャー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。

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。

回避するパッケージのマーキング

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、それらを回避するには、`pkg avoid` コマンドを使用します。

```
/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、要求された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。

省略可能な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、省略可能なコンポーネントや、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。省略可能な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、省略可能な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アントと呼びます。

ファセットとバリアントはイメージの特殊なプロパティーであり、個々のパッケージには設定できません。イメージに設定されているファセットとバリアント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アントの現在の値を変更するには、pkg change-facet コマンドと pkg change-variant コマンドを使用します。pkg(1) のマニュアルページと下の例を参照してください。

ファセットとバリアントはパッケージアクションのタグとして指定します。各ファセットタグおよびバリアント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アント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アント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。

バリアントタグの例は variant.arch=sparc です。ファセットタグの例は facet.devel=true です。ファセットとバリアントは、facet. と variant. を先頭に付けずに参照されることがあります。

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。これは、ファセット名前空間の一部を無効にし、その中の個々のファセットのみを有効にする場合に役立ちます。たとえば、次の例に示すように、すべてのロケールを無効にしてから、1 つか 2 つの特定のロケールのみを有効にすることができます。

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

ほとんどのバリアントは任意の数の値を設定できます。たとえば、arch バリアントには、i386、sparc、ppc、arm、またはディストリビューションでサポートされているどんなアーキテクチャーでも設定できます。(Oracle Solaris では i386 および sparc のみ使用されます。)例外は debug バリアントです。debug バリアントには、true または false のみを設定でき、ほかの値では動作が不確定になります。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 バリアントが明示的に設定されている必要があります。

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

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

バリアントを使用するパッケージをインストールするために、バリアント値をイメージに設定する必要があります。arch および zone バリアントは、イメージを作成し、その初期コンテンツをインストールするプログラムによって設定されます。イメージ内の debug.* バリアントは、デフォルトで false です。

イメージに設定されたファセットとバリアントは、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します。

- ファセットまたはバリアントタグのないアクションは常にインストールされます。
- ファセットタグのあるアクションは、イメージ上のタグに一致するすべてのファセットまたはファセットパターンが false に設定されていないかぎり、インストールされます。いずれかのファセットが true に設定されているか、明示的に設定されていない (true はデフォルト) 場合、アクションがインストールされます。
- バリアントタグのあるアクションは、すべてのバリア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アントタグの両方があるアクションは、ファセットとバリア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。

独自のファセットおよびバリアント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。

バリアント名	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アントとファセットの値を表示し、現在のイメージのバリアントとファセットを変更できます。バリアント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。

バリアントの表示

設定されているバリアント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_spec ...]
```

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

バリアントの変更

バリアント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] [-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_spec=instance ...
```

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、56 ページの「ブート環境ポリシーイメージのプロパティー」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。

```
# pkg change-variant -nv --accept variant.debug.*=true
            Packages to update:       831
     Variants/Facets to change:         4
     Estimated space available: 112.19 GB
Estimated space to be consumed: 220.76 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.en: None
    facet facet.locale.*: None
Changed packages:
solaris
  ...
```

ファセットの表示

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_spec ...]
```

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

ファセットの変更

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] [-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_spec=[True|False|None] ...
```

-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、56ページの「ブート環境ポリシーイメージのプロパティー」を参照してください。

```
# 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
  ...
```

イメージの更新

インストールされているパッケージと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、使用可能な更新があるすべてのインストールされているパッケージを更新するには、*pkg-fmri*を指定しないか、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。非大域ゾーンが現在のイメージに構成されている場合、これらのゾーンも更新されます。『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「パッケージとゾーンについて」を参照してください。

```
/usr/bin/pkg update [-fnvq] [-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 ...]
```

使用可能な更新があるパッケージのリストを表示するには、`pkg list -u`コマンドを使用します。`pkg update`操作により、多くの場合、新しいBEが自動的に作成されます。現在のイメージまたは新しいBEに行われる変更を確認するには、`pkg update -nv`を使用します。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。

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、56 ページの「ブート環境ポリシーイメージのプロパティー」を参照してください。

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。46 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。`update` の実行後、発行元によって提供され、イメージ内に見つからない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。

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更新が実行されます。新しい BE が作成された場合、新しい BE でこの発行元の更新を実行します。

イメージと発行元のプロパティーの構成

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ーを設定します。このセクションでは、イメージと発行元のプロパティーと、これらのプロパティーの設定方法について説明します。イメージプロパティーの説明については、`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ー

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、バリアントの変更、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 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ーの設定の手順については、61 ページの「イメージのプロパティーの設定」を参照してください。

`be-policy`
パッケージ操作中にいつブート環境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ー

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ーと発行元プロパティーを、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ー

次のイメージプロパティーを、署名付きパッケージを使用するように構成します。

`signature-policy`
: このプロパティーの値により、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ーを `true` に設定します。次の値が許可されます。

  `ignore`
  : すべてのマニフェストの署名を無視します。

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

    これがデフォルト値です。

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

  `require-names`
  : `require-signatures` と同じ要件に従いますが、`signature-required-names` イメージプロパティー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`
: このプロパティーの値は、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。

`trust-anchor-directory`
: このプロパティーの値は、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。

署名付きパッケージの発行元プロパティー

次の発行元プロパティーを、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
　このプロパティーの機能は、このプロパティー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ーの機能と同じです。

signature-required-names
　このプロパティーの機能は、このプロパティー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ーの機能と同じです。

パッケージの署名プロパティーの構成

このイメージのパッケージの署名プロパティーを設定するには、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。

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

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

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

次の例は、このイメージで、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。

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

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

```
# pkg add-property-value signature-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。

```
# 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。

```
# 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ー

ca-path
: SSL 操作用の 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 CApath 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。

  このプロパティーを使用して、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。このプロパティーを使用すると、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
: このプロパティーは、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。このプロパティーを 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の実行方法については、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。

use-system-repo
このプロパティー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るべきかどうかを指定します。システムリポジトリについては、pkg.sysrepo(1m) を参照してください。

デフォルト値は ignore です。

イメージのプロパティーの設定

このイメージのプロパティーを設定するには、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ーの値の表示

イメージのプロパティーを表示するには、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
display-copyrights              True
flush-content-cache-on-success  False
mirror-discovery                False
preferred-authority             solaris
publisher-search-order          ['solaris', 'opensolaris.org', 'extra']
pursue-latest                   True
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 プロパティーは、pkg set-publisher コマンドオプションを使用して設定できます。46 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ーの値の設定

イメージプロパティーの値を設定したり、プロパティー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 プロパティーの値を設定します。

```
# pkg set-property mirror-discovery True
# pkg property -H mirror-discovery
mirror-discovery True
```

イメージプロパティーの値のリセット

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

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

イメージの作成

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

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、適切なバリアントを設定します。ゾーンイメージでは、IPS は、パッケージ内の依存関係によって定義されているとおりに、非大域ゾーンと大域ゾーンの一貫性を維持します。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_spec=instance ...]
    [-g path_or_uri | --origin path_or_uri ...]
    [-m uri | --mirror uri ...]
    [--facet facet_spec=(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、適切なバリアントを設定します。

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。パブリッシャーの名前も提供した場合、イメージの作成時にそのパブリッシャーのみが追加されます。パブリッシャーの名前を提供しない場合、指定されたリポジトリによって認識されているすべてのパブリッシャーがイメージに追加されます。このパブリッシャーに関連付けられたカタログは、初期作成操作に続いて取得が試みられます。

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パブリッシャー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。

指定したバリアント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 [-Hl] [-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刻、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
2011-09-07T12:15:52   update                  pkg               Succeeded
2011-09-26T18:53:12   refresh-publishers      pkg               Succeeded
2011-09-26T18:53:50   rebuild-image-catalogs  pkg               Succeeded
2011-09-27T09:05:34   update                  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
2011-09-07T12:15:52   0:13:56   update                  Succeeded
2011-09-26T18:53:12   0:01:22   refresh-publishers      Succeeded
2011-09-26T18:53:50   0:00:44   rebuild-image-catalogs  Succeeded
2011-09-27T09:05:34   0:20:08   update                  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。

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

```
# pkg purge-history
```