

Adding and Updating Oracle® Solaris 11 Software Packages

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Contents

Preface	7
1 Introduction to the Image Packaging System	11
Image Packaging System	11
Installation Privileges	12
IPS Concepts	12
IPS Packages	12
Fault Management Resource Identifiers	13
Publishers, Repositories, and Package Archives	14
Repository Origins and Mirrors	14
Images and Boot Environments	15
Package Facets and Variants	15
2 IPS Graphical User Interfaces	17
Using Package Manager	17
Package Manager Command Line Options	17
Using Web Install	18
Using Update Manager	20
Update Manager Command Line Options	21
3 Getting Information About Software Packages	23
Showing Package Install State Information	23
Displaying Package Descriptions or Licenses	25
Showing Information From the Package Manifest	26
Listing Files Installed By a Package	27
Listing All Installable Packages In a Group Package	28
Displaying License Requirements	28

Searching for Packages	28
Identifying Which Package Delivers a Specific File	29
Listing Packages By Category	29
Showing Dependent Packages	30
Listing All Packages In a Group Package	30
4 Installing and Updating Software Packages	33
Previewing an Operation	34
Installing and Updating Packages	35
Boot Environment Options	35
Installing a New Package	36
Installing a Package into a New Boot Environment	37
Rejecting a Package	39
Updating a Package	39
Fixing Package Problems	40
Verifying Package Installation	41
Fixing Verification Errors	41
Restoring a File	42
Uninstalling Packages	42
5 Configuring Installed Images	43
Configuring Publishers	43
Displaying Publisher Information	43
Adding, Modifying, or Removing Package Publishers	44
Locking Packages to a Specified Version	46
Marking Packages To Be Avoided	47
Controlling Installation of Optional Components	47
Displaying a Variant	49
Changing a Variant	49
Displaying a Facet	50
Changing a Facet	50
Updating an Image	51
Configuring Image and Publisher Properties	52
Boot Environment Policy Image Properties	52
Properties For Signing Packages	54

Additional Image Properties	56
Setting Image Properties	57
Creating an Image	58
Viewing Operation History	60

Preface

Adding and Updating Oracle Solaris 11 Software Packages describes the software installation functions of the Oracle Solaris Image Packaging System (IPS) feature. IPS commands enable you to list, search, install, update, and remove software packages for the Oracle Solaris 11 operating system. A single IPS command can update your image to a new operating system release. IPS commands enable you to restrict which packages can be installed or which versions of packages can be installed.

IPS commands also enable you to copy and create IPS package repositories, and create IPS packages. See [“Related Documentation” on page 8](#) for information about those tools.

To use IPS, you must be running the Oracle Solaris 11 OS. To install the Oracle Solaris 11 OS, see [Installing Oracle Solaris 11 Systems](#).

Who Should Use This Book

This book is for system administrators who install and manage software and manage system images.

How This Book Is Organized

- [Chapter 1, “Introduction to the Image Packaging System,”](#) describes the Image Packaging System and components such as packages, publishers, and repositories.
- [Chapter 2, “IPS Graphical User Interfaces,”](#) explains how to use Package Manager and Update Manager, including how to use Web Install.
- [Chapter 3, “Getting Information About Software Packages,”](#) shows how to search for packages and display information about packages.
- [Chapter 4, “Installing and Updating Software Packages,”](#) shows how to install, update, and uninstall packages.
- [Chapter 5, “Configuring Installed Images,”](#) shows how to configure characteristics that apply to an entire image, such as configuring package publishers or restricting which packages can be installed.

Related Documentation

In addition to these books, see the Package Manager online help and the pkg(1M) and beadm(1M) man pages.

- *Image Packaging System Man Pages*
- *Copying and Creating Oracle Solaris 11 Package Repositories*
- *Creating and Administering Oracle Solaris 11 Boot Environments*
- *Installing Oracle Solaris 11 Systems*

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Description	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your .login file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name%</code> su Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <i>rm filename</i> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P-2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	\$
Bash shell, Korn shell, and Bourne shell for superuser	#
C shell	machine_name%
C shell for superuser	machine_name#

Introduction to the Image Packaging System

The Oracle Solaris Image Packaging System (IPS) is a framework that enables you to list, search, install, update, and remove software packages for the Oracle Solaris 11 operating system. A single IPS command can update your image to a new operating system release.

Image Packaging System

Oracle Solaris 11 software is distributed in IPS packages. IPS packages are stored in IPS package repositories, which are populated by IPS publishers. IPS packages are installed into Oracle Solaris 11 images. A subset of the capabilities that are available through the IPS command-line interface is available through the Package Manager graphical user interface.

IPS tools provide the following capabilities. See [“IPS Concepts” on page 12](#) for definitions of terms such as publisher and repository.

- List, search, install, restrict installation, update, and remove software packages.
- List, add, and remove package publishers. Change publisher attributes such as search priority and stickiness. Set publisher properties such as signature policy.
- Update an image to a new operating system release.
- Create copies of existing IPS package repositories. Create new package repositories.
- Create and publish packages.
- Create boot environments.

To use IPS, you must be running the Oracle Solaris 11 OS. To install the Oracle Solaris 11 OS, see [Installing Oracle Solaris 11 Systems](#).

Installation Privileges

The commands discussed in [Chapter 3, “Getting Information About Software Packages,”](#) do not require any special privilege to use. Tasks such as installing and updating IPS packages, setting publishers, and modifying images require more privilege.

Use one of the following methods to gain more privilege:

- Use the `profiles` command to list the rights profiles that are assigned to you. If you have the Software Installation rights profile, you can use the `pfexec` command to install and update packages.

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

Other rights profiles also provide installation privilege, such as the System Administrator rights profile.

- Depending on the security policy at your site, you might be able to use the `sudo` command with your user password to execute a privileged command.

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

- Use the `roles` command to list the roles that are assigned to you. If you have the root role, you can use the `su` command with the root password to assume the root role.

IPS Concepts

This section defines terms and concepts that are used in the remainder of this guide.

IPS Packages

An IPS *package* is defined by a text file called a *manifest*. A package manifest describes package *actions* in a defined format of key/value pairs and possibly a data payload. Package actions include files, directories, links, drivers, dependencies, groups, users, and license information. Package actions represent the installable objects of a package. Actions called “set” actions define package metadata such as classification, summary, and description.

You can search for packages by specifying package actions and action keys. See [pkg\(5\)](#) for descriptions of package actions.

An *incorporation* is a package that constrains the versions of a specified set of packages. For example, if a package in an installed incorporation is version 1.4.3, then no version less than 1.4.3 or greater than or equal to 1.4.4 can be installed. However, versions that merely extend the dotted sequence, such as 1.4.3.7, could be installed. Incorporations force the incorporated packages to upgrade synchronously. An incorporated package could be removed, but if the package is installed or updated, the version is constrained.

A *group* package specifies the set of packages that constitute a feature or tool. Packages specified in a group package do not specify the package version. The group package is a content management tool, not a version management tool.

Fault Management Resource Identifiers

Each package is represented by a Fault Management Resource Identifier (FMRI). The full FMRI for a package consists of the scheme, a publisher, the package name, and a version string in the following format. The scheme, publisher, and version string are optional. When using IPS commands, you can use the smallest portion of the package name that uniquely identifies the package.

Format:

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

Example:

```
pkg://solaris/editor/vim@7.3.254,5.11-0.174.0.0.0.504:20110921T002716Z
```

Scheme pkg

Publisher solaris

If the publisher is specified, then the publisher name must be preceded by `pkg://` or `//`.

Package name editor/vim

The package namespace is hierarchical and arbitrarily deep. In IPS commands, you can specify the smallest portion of the package name that uniquely identifies the package. If you specify the full package name but omit the publisher, the full package name can be preceded by `pkg:/` or `/` but not by `pkg://` or `//`. If you specify an abbreviated package name, do not use any other characters to the left of the package name.

Version The package version has four parts:

Component version 7.3.254

For components tightly bound to the operating system, this is usually the value of `uname -r` for that version of the operating system.

Build version 5.11

	The build version must follow a comma (.). The build version specifies the version of the operating system on which the contents of the package were built.
Branch version	0.174.0.0.0.0.504
	The branch version must follow a hyphen (-). The branch version provides vendor-specific information.
Time stamp	20110921T002716Z
	The time stamp must follow a colon (:). The time stamp is the time the package was published in ISO-8601 basic format: <i>YYYYMMDDTHHMMSSZ</i> .

Publishers, Repositories, and Package Archives

A *publisher* identifies a person or organization that provides one or more packages. Publishers can distribute their packages using package repositories or package archives. Publishers can be configured into a preferred search order. When a package installation command is given and the package specification does not include the publisher name, the first publisher in the search order is searched for that package. If the package is not found, the second publisher in the search order is searched, and so forth until the package is found or all publishers have been searched.

A *repository* is a location where packages are published and from where packages are retrieved. The location is specified by a Universal Resource Identifier (URI). A *catalog* is the list of all the packages in a repository.

A *package archive* is a file that contains publisher information and one or more packages provided by that publisher.

Repository Origins and Mirrors

An *origin* is a package repository that contains both package *metadata* (such as catalogs, manifests, and search indexes) and package *content* (files). If multiple origins are configured for a given publisher in an image, the IPS client attempts to choose the best origin from which to retrieve package data.

A *mirror* is a package repository that contains only package content. IPS clients access the origin to obtain a publisher's catalog, even when the clients download package content from a mirror. If a mirror is configured for a publisher, the IPS client prefers the mirror for package content

retrieval. If multiple mirrors are configured for a given publisher in an image, the IPS client attempts to choose the best mirror from which to retrieve package content. If all mirrors are unreachable, do not have the required content, or are slower, the IPS client retrieves the content from an origin.

Images and Boot Environments

An *image* is a location where IPS packages can be installed and where other IPS operations can be performed.

A *boot environment* (BE) is bootable instance of an image. You can maintain multiple BEs on your system, and each BE can have different software versions installed. When you boot your system, you have the option to boot into any of the BEs on the system. A new BE can be created automatically as a result of package operations. You can also explicitly create a new BE. Whether a new BE is created depends on image policy as described in “[Boot Environment Policy Image Properties](#)” on page 52.

Package Facets and Variants

Software can have components that are optional and components that are mutually exclusive. Examples of optional components include locales and documentation. Examples of mutually exclusive components include SPARC or x86 and debug or non-debug binaries. In IPS, optional components are called *facets* and mutually exclusive components are called *variants*.

Facets and variants are special properties of the image and cannot be set on individual packages.

Individual actions in package manifests can have facet and variant tags. A single action can have multiple facet and variant tags.

The values of facet and variant tags on an action compared with the values of facets and variants set in the image determine whether that package action can be installed.

- Actions with no facet or variant tags are always installed.
- Actions with facet tags are installed unless all of the facets or facet patterns matching the tags are set to `false` on the image. If any facet is set to `true` or is not explicitly set (`true` is the default), then the action is installed.
- Actions with variant tags are installed only if the values of all the variant tags are the same as the values set in the image.
- Actions with both facet and variant tags are installed if both the facets and the variants allow the action to be installed.

To view or modify the values of the facets and variants set on the image, see [“Controlling Installation of Optional Components”](#) on page 47.

IPS Graphical User Interfaces

IPS includes two Graphical User Interface (GUI) tools.

- Package Manager provides most package and publisher operations and some boot environment (BE) operations. If you are new to the Oracle Solaris OS and IPS technologies, you can use Package Manager to quickly identify and install packages.
- Update Manager updates all packages in the image that have updates available.

Using Package Manager

Package Manager provides a subset of the tasks that can be performed from the command line:

- List, search, install, update, and remove packages
- Add and configure package sources
- Activate, rename, and remove BEs

Start Package Manager in one of the following ways:

Tool bar Click the Package Manager icon in the tool bar. The Package Manager icon is a box with a circling arrow.

Desktop icon Double-click the Package Manager icon on the desktop.

Menu bar Select System>Administration>Package Manager.

Command line # `packagemanager`

For complete Package Manager documentation, select Help>Contents from the Package Manager menu bar.

Package Manager Command Line Options

The following options are supported for the `packagemanager(1)` command.

TABLE 2-1 Package Manager Command Options

Option	Description
<code>--image-dir</code> or <code>-R dir</code>	Operate on the image rooted at <i>dir</i> . The default behavior is to operate on the current image. The following command operates on the image stored at <code>/aux0/example_root</code> : # <code>packagemanager -R /aux0/example_root</code>
<code>--update-all</code> or <code>-U</code>	Update all installed packages that have updates available. Specifying this option is the same as selecting the Updates option in the Package Manager GUI. See “Using Update Manager” on page 20 for more information about updating all packages.
<code>--info-install</code> or <code>-i file.p5i</code>	Specify a <code>.p5i</code> file to run Package Manager in Web Install mode. The specified file must have the extension <code>.p5i</code> . See the “Using Web Install” on page 18 for more information.
<code>--help</code> or <code>-h</code>	Display command usage information.

Using Web Install

See the Package Manager Help for detailed information about the Web Install process.

Package Manager supports installing packages using a simple one-click Web Install process. The Web Install process uses a `.p5i` file. A `.p5i` file contains information to add publishers and add packages that can be installed from these publishers. The information in the `.p5i` file is read and used by the Web Install process.

Exporting Files Using Web Install

If you want other users to be able to install packages that you have installed on your system, you can export the installation instructions for those package files using the Web Install process. The Web Install process creates a `.p5i` file that consists of installation instructions for those packages and publishers to be installed.

To export the installation instructions for your selected packages and their publishers to a `.p5i` file, perform the following steps:

1. From the Package Manager Publisher drop-down menu, select the publisher from which you want to include the packages in the `.p5i` file.
2. In the Package Manager package list pane, select the package whose installation instructions you want to distribute.
3. Select File>Export Selections to display the Export Selections Confirmation window.
4. Click the OK button to confirm the selections. The Export Selections window is displayed.

5. A default name for the .p5i file is provided. You can change this file name, but do not change the .p5i extension.
6. A default location for the .p5i file is provided. You can change the location.
7. Click the Save button to save the file name and location.

Using Web Install to Add Publishers and Install Packages

The Web Install process enables you to install packages through a .p5i file. This file might be on your desktop or on a web site.

1. Use one of the following methods to start Package Manager in Web Install mode:
 - Select a .p5i file on your desktop.
 - Start Package Manager from the command line and specify a .p5i file:


```
# packagemanager ./wifile.p5i
```
 - Go to a URL location that contains a link to a .p5i file.

If the .p5i file is located on a web server that has registered this MIME type, just click the link to the .p5i file.

If the .p5i file is located on a web server that has not registered this MIME type, save the .p5i file to your desktop and then select it.
2. The Install/Update window is displayed. The label at the top of the window is: “Package Manager Web Installer/The following will be added to your system.” The publishers and packages to be installed are listed. Click the Proceed button to continue with the installation.
3. If the specified package publisher is not already configured on your system, the Add Publisher window is displayed. The name and URI of the publisher are already entered.

If the publishers to be added are secure publishers, an SSL key and certificate are required. Browse to locate the SSL Key and SSL Certificate on your system.

If the publisher is added successfully, the Adding Publisher Complete dialog displays. Click the OK button to continue with the installation.
4. If a .p5i file contains packages from a disabled publisher, Web Install opens an Enable Publisher dialog. Use this dialog to enable the publisher so that you can install the packages.

The Install/Update window now looks the same as when you select the Package Manager Install/Update option.

The application closes when all packages are installed.

Using Update Manager

Update Manager updates all installed packages to the newest version allowed by the constraints imposed on the system by installed packages and publisher configuration. This function is the same as the following functions:

- In the Package Manager GUI, select the Updates button or the Package>Updates menu option.

- Use the `packagemanager` command.

```
# packagemanager --update-all
```

- Use the `pkg` command.

```
# pkg update
```

Start Update Manager in one of the following ways:

Status bar When updates are available, you should see a notification in the status bar. Click where indicated in the notification. The Update Manager icon is a stack of three boxes.

Menu bar Select System>Administration>Update Manager.

Command line # `pm-updatemanager`

The Updates window displays, and the update process starts:

1. The system refreshes all catalogs.
2. The system evaluates all installed packages to determine which packages have updates available.
 - If no packages have updates available, the message “No Updates Available” is displayed and processing stops.
 - If package updates are available, the packages to be updated are listed for your review. This is your last chance to click the Cancel button to abort the update.
3. Click the Proceed button to continue with the update. The system downloads and installs all package updates.

The following packages are updated first if they have updates available. Then any other packages are updated.

```
package/pkg  
package/pkg/packagemanager  
package/pkg/updatemanager
```

By default, each package is updated from the publisher from which it was originally installed. If the original publisher is non-sticky, then a newer version of the package that is

compatible with this image could be installed from another publisher. Use the Package Manager Manage Publishers window or the `pkg set -publisher` command to set a publisher as sticky or non-sticky.

A new BE might be created, depending on which packages are updated and depending on your image policy.

If an error occurs at any time during the update process, the Details panel expands and the details of the error are displayed. An error status indicator is shown next to the failed stage.

4. If the system created a new BE for the update, you can edit the default BE name. When you are satisfied with the BE name, click the Restart Now button to restart your system immediately. Click the Restart Later button to restart your system at a later time. You must restart to boot into the new BE. The new BE will be your default boot choice. Your current BE will be available as an alternate boot choice.

Update Manager Command Line Options

The following options are supported for the `pm-updatemanager(1)` command.

TABLE 2-2 Update Manager Command Options

Option	Description
<code>--image-dir</code> or <code>-R dir</code>	Operate on the image rooted at <i>dir</i> . The default behavior is to operate on the current image. The following command updates the image at <code>/aux0/example_root</code> : # pm-updatemanager -R /aux0/example_root
<code>--help</code> or <code>-h</code>	Display command usage information.

Getting Information About Software Packages

This chapter describes commands that give you the following kinds of information about packages:

- Whether the package is installed or can be updated
- The description, size, and version of the package
- Which packages are part of a group package
- Which packages are in a particular category
- Which package delivers a specified file

No special privileges are needed to run any of these commands.

Showing Package Install State Information

The `pkg list` command tells you whether a package is installed in the current image and whether an update is available. With no options or operands, this command lists all packages that are installed in the current image. To narrow your results, provide one or more package names. You can use wildcards in the package names. Package variants for an architecture or zone type that does not match this image are not listed.

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

The `pkg list` command displays one line of information for each package.

```
$ 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--
```

The publisher name in parentheses indicates that the `isv.com` publisher is not the first publisher in the publisher search order in this image. The `dtrace-toolkit` package that is installed in this image is published by the publisher that is the first publisher in the search order.

The “i” in the I column indicates that these packages are installed in this image. To list packages that are installed and the newest versions of packages that are not installed but could be installed in this image, use the `-a` option.

```
$ 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--
```

This output indicates that the `image/nvidia/cg-toolkit` can be installed in this image.

To list all matching packages, including packages that cannot be installed in this image, use the `-af` option. To list only the newest versions of these packages, specify `@latest`.

```
$ 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--
```

This output indicates that the `developer/dtrace/toolkit` package cannot be installed in this image. The “r” in the O column indicates that this package has been renamed. The `developer/dtrace/toolkit` package has been renamed to `system/dtrace/dtrace-toolkit`, and `system/dtrace/dtrace-toolkit` is already installed.

In the following example, the `web/amp` package has been renamed. The package displays without using the `-f` option because the `web/amp` package has been renamed to `group/feature/amp`, and `group/feature/amp` is not installed. If you specify the command to install the `web/amp` package, the `group/feature/amp` package is installed automatically.

```
$ 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
```

The `pkg list` command does not tell you the new name of a renamed package. To get that information, use the `pkg info` command as shown in [“Displaying Package Descriptions or Licenses” on page 25](#).

The `-n` option lists the newest version of each known package. An “o” in the O column indicates that the package is obsolete. You cannot install a package that is obsolete.

```
$ 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 ---
```

This output indicates that the `database/mysql-50` package cannot be installed in this image. This package has not been renamed. If you specify the command to install the `mysql-50` package, the `mysql-51` package is not installed. No packages are installed in this case.

An “f” in the F column indicates the package is frozen. If a package is frozen, you can only install or update to packages that match the frozen version. See [“Locking Packages to a Specified Version” on page 46](#) for information about freezing packages.

```
$ pkg list mercurial
NAME (PUBLISHER)          VERSION          IFO
developer/versioning/mercurial  1.8.4-0.174.0.0.0.504  if-
```

The `-s` option lists only the package name and summary.

```
$ 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
```

The `-v` option lists the full package FMRI.

```
$ pkg list -nv mysql-51
FMRI
pkg://solaris/database/mysql-51@5.1.37,5.11-0.174.0.0.0.504:20110920T230125Z --- IFO
```

The `-u` option lists all installed packages that have newer versions available.

Use the `-g` option to specify the repository or package archive to use as the source of package data for the operation.

When you use the `--no-refresh` option, `pkg` does not attempt to contact the repositories for the image's publishers to retrieve the newest list of available packages.

Displaying Package Descriptions or Licenses

The `pkg info` command displays information about a package, including the name, installed state, version, packaging date, package size, and the full FMRI. With no options or operands, this command displays information about all packages that are installed in the current image. To narrow your results, provide one or more package names. You can use wildcards in the package names.

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

Both the `info` and `list` subcommands display the package name, publisher, and version information. The `pkg list` command shows whether an update exists for the package, whether an update can be installed in this image, and whether a package is obsolete or renamed. The `pkg info` command displays the package summary, description, category, and size, and can separately display the license information.

The `-r` option displays the newest available versions, retrieving information for any packages not currently installed from the repositories of the configured publishers.

```
$ 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
```

Use the `pkg info` command to find the new name of a renamed package. The following example shows that the new name of the `developer/dtrace/toolkit` package is `system/dtrace/dtrace-toolkit`.

```
$ 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
```

The `--license` option displays the license texts for the packages. This information can be quite lengthy. The information shown above (without the `--license` option) is not displayed.

```
$ pkg info -r --license x11/server/xorg
Copyright (c) 2011, Oracle and/or its affiliates. All rights reserved.
The following software...
```

Use the `-g` option to specify the repository or package archive to use as the source of package data for the operation.

Showing Information From the Package Manifest

The `pkg contents` command displays the file system content of packages. With no options or operands, this command displays path information for all packages that are installed in the current image. Use command options to specify particular package content to display. To narrow your results, provide one or more package names. You can use wildcards in the package names.

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

Both the contents and search subcommands query the contents of packages. The `pkg contents` command displays actions and attributes of packages. The `pkg search` command lists the packages that match the query.

The following example shows the `pkg contents` default behavior. Use options to specify which actions and attributes to display.

```
$ 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
```

The `-m` option displays the entire package manifest.

The `-r` option displays the newest available versions, retrieving information for any packages not currently installed from the repositories of the configured publishers.

Use the `-g` option to specify the repository or package archive to use as the source of package data for the operation.

Use the `-s` option to sort actions by the specified action attribute. By default, output is sorted by path or by the first attribute specified by the `-o` option. The `-s` option can be specified multiple times.

Listing Files Installed By a Package

Use the `-t` option to specify the type of actions to display. You can specify multiple types in a comma-separated list, or you can specify the `-t` option multiple times.

Use the `-o` option to specify the attributes to display in the output. You can specify multiple attributes in a comma-separated list, or you can specify the `-o` option multiple times. See the [pkg\(5\)](#) man page for a list of package actions and attributes. In this example, the `pkg.size` pseudo attribute shows the size of the file; the `file` action does not have a size attribute.

```
# 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
```

If you view the package manifest, you see that the `e1000g` package has seven file actions. The three that are not shown in the above output are files that cannot be installed in this image. This image is an x86 architecture, and does not include debug files. The debug file for the x86 architecture is not shown above, and neither the debug nor the non-debug file is shown for the SPARC architecture. You can change whether an image includes debug files by changing the debug image variant. See [“Controlling Installation of Optional Components” on page 47](#).

Listing All Installable Packages In a Group Package

The Oracle Solaris 11 GUI installer installs the `solaris-desktop` group package. The text installer and the default AI manifest in an Automated Installer installation install the `solaris-large-server` group package. The `solaris-small-server` group package is an alternative you can use to install a smaller set of packages on a server. You can use the following command to display the set of packages that is included in each group.

```
$ pkg contents -o fmri -H -rt depend -a type=group solaris-desktop
archiver/gnu-tar
audio/audio-utilities
...
```

The `-t` option matches depend actions in the package. The `-a` option matches the depend actions that are type `group`. The `-o` option displays only the `fmri` attribute of the group depend action.

Displaying License Requirements

This example displays all the incorporation packages that require you to accept the package license.

```
$ 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
```

Searching for Packages

Use the `pkg search` command to search for packages whose data matches the specified pattern.

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

Like the `pkg contents` command, the `pkg search` command examines the contents of packages. While the `pkg contents` command returns the contents, the `pkg search` command returns the names of packages that match the query.

By default, `pkg search` query terms are matched exactly except for case. Use the `-I` option to specify a case-sensitive search. You can use `?` and `*` wildcards in query terms.

You can specify more than one query term. By default, multiple terms are joined with AND. You can explicitly join two terms with OR.

Queries can be expressed in the following structured form:

```
pkg_name:action_type:key:token
```

Missing fields are implicitly wildcarded. Explicit wildcards are supported in the *pkg_name* and *token* fields. The *action_type* and *key* must match exactly. See the [pkg\(5\)](#) man page for a list of package actions and keys.

By default, repositories associated with all publishers configured for this image are searched. Use the `-l` option to search only packages that are installed in this image. Use the `-s` option to specify the URI of the repository to search.

By default, matches are displayed only for currently installed or newer package versions. Use the `-f` option to display all matched versions.

Identifying Which Package Delivers a Specific File

The following example shows that the `libpower` library came from the `system/kernel/power` package.

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

Listing Packages By Category

The following example identifies all packages that have “Source Code Management” in the value of their `info.classification` attribute.

```
# 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.
Development/Source Code Management pkg:/developer/quilt@0.47-0.174.0.0.0.504
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
```

This example shows a large amount of repeated information that obscures the information that was really wanted.

The following example uses the `-o` option to show only the names of the packages and uses the `-H` option to omit the column heading.

```
# 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
```

Showing Dependent Packages

These examples show the packages that depend on the specified package.

The following example shows packages that have a require dependency on the system/kernel/power package. If you used the pkg contents command to display depend actions of type require for the i86pc and system/hal packages, you would see that system/kernel/power is listed for both packages.

```
$ pkg search -l -H -o pkg.name 'depend:require:system/kernel/power'  
system/kernel/dynamic-reconfiguration/i86pc  
system/hal
```

The following example shows that many packages have an exclude dependency on pkg:/x11/server/xorg@1.10.99.

```
$ 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  
...
```

Listing All Packages In a Group Package

The Oracle Solaris 11 GUI installer installs the solaris-desktop group package. The text installer and the default AI manifest in an Automated Installer installation install the solaris-large-server group package. The solaris-small-server group package is an alternative you can use to install a smaller set of packages on a server. You can use the following search form to display the set of packages that is included in each group.

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

In this example, `-o pkg.name` would return only the name of the package specified in the `pkg_name` field of the query:

```
group/system/solaris-desktop
```

The `-o fmri` option returns the FMRI of the packages that are specified in the `solaris-desktop` package as group type dependencies.

By default, `search` returns only packages that are installable in this image. In this example, `search` is not returning matching packages but rather is returning the value of an attribute of an action in a specified package. That attribute value happens to be a package name in this example. The number of results from this command is larger than the number of the results from the similar `pkg contents` command because these search results include the names of all packages that are named in group depend actions in the specified package, not just installable packages. For example, package variants might be included that are not installable in this image. Compare the output from this search to the output from the `pkg contents` command shown in [“Listing All Installable Packages In a Group Package” on page 28](#).

Tip – In general, use the `pkg contents` command to show the contents of a specified package, and use the `pkg search` command to show packages that match a query. If you know which package delivers the content that you are interested in, use the `pkg contents` command.

Installing and Updating Software Packages

Package installation and update are affected by image configuration such as constraining some packages to a particular version, configuring publisher search order, and setting package signing properties. Image configuration is discussed in [Chapter 5, “Configuring Installed Images.”](#) The instructions and results shown in this chapter assume default image configuration.

How to determine which packages are already installed, which packages are available to install, and which packages have updates available is covered in [Chapter 3, “Getting Information About Software Packages.”](#)

This chapter shows how to perform the following tasks:

- Run a trial installation to see whether the installation would succeed and what would be installed
- Install, update, and uninstall packages
- Validate packages
- Fix problems with installed packages
- Restore an installed file to its original content
- Uninstall packages

Installing, updating, and uninstalling packages require increased privileges. See [“Installation Privileges” on page 12](#) for more information.

Previewing an Operation

Many of the commands shown in this chapter and in [Chapter 5, “Configuring Installed Images,”](#) have an `-n` option that enables you to see what the command will do without making any changes.

Tip – Best practice is to use the `-n` option whenever it is available. Use the `-n` option with one or more verbose options (`-nv`, `-nvv`) and review the effects of the command before you execute the command without the `-n` option.

The following example shows information about a package installation that is not actually performed:

```
# 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 -> 5.1.37,5.11-0.174.0.0.0.504:20110920T230125Z
  group/feature/amp
  None -> 0.5.11,5.11-0.174.0.0.0.2559:20110921T191255Z
  web/php-52
  None -> 5.2.17,5.11-0.174.0.0.0.504:20110921T041858Z
  web/php-52/extension/php-apc
  None -> 3.0.19,5.11-0.174.0.0.0.504:20110921T041245Z
  web/php-52/extension/php-mysql
  None -> 5.2.17,5.11-0.174.0.0.0.504:20110921T041411Z
  web/server/apache-22/module/apache-dtrace
  None -> 0.3.1,5.11-0.174.0.0.0.504:20110921T042357Z
  web/server/apache-22/module/apache-fcgid
  None -> 2.3.6,5.11-0.174.0.0.0.504:20110921T042430Z
  web/server/apache-22/module/apache-php5
  None -> 5.2.17,5.11-0.174.0.0.0.504:20110921T042738Z
Services:
  restart_fmri:
    svc:/system/manifest-import:default
    svc:/system/rbac:default
```

The following command produces a large amount of output since so many packages would be affected. Notice the amount of additional space that would be used is in gigabytes, not megabytes. This operation might require a large amount of time and cause a large amount of network traffic between this image and the package repository. Notice that a new BE would not be created by default, but a backup BE would be created. See [“Boot Environment Policy Image Properties” on page 52](#) for information about when BEs are created.

```
# 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
  ...
```

Installing and Updating Packages

The `pkg install` command installs packages that are not currently installed and updates packages that are already installed. The `pkg install` command requires one or more package names.

The `pkg update` command updates installed packages. If you specify a package that is not already installed to the `pkg update` command, the system does not install that package. The `pkg update` command takes zero or more names of packages that are already installed. Specifying no package names updates all packages that are installed in the image.

See the `preserve` and `overlay` attributes of the `file` action in the [pkg\(5\)](#) man page to understand how files with these attributes will be handled during installation and update.

Boot Environment Options

A new BE or a backup BE might be created when you install, update, or uninstall a package or revert a file. Within the constraints of the image policy regarding BEs, you can control the creation of new and backup BEs using the options described below. See [“Boot Environment Policy Image Properties” on page 52](#) for information about new BEs and backup BEs and how to set image policy regarding BEs.

Use the BE options to force a new BE or backup BE to be created or not created, to give the BE a custom name, and to specify that the new BE should not be activated.

<code>--no-be-activate</code>	If a BE is created, do not set it as the active BE on the next boot. Use the beadm(1M) command to show and change the active BE.
<code>--no-backup-be</code>	Do not create a backup BE.
<code>--require-backup-be</code>	Create a backup BE if a new BE will not be created. Without this option, a backup BE is created based on image policy. See “Boot Environment Policy Image Properties” on page 52 for an explanation of when backup BEs are created automatically.

<code>--backup-be-name <i>name</i></code>	If a backup BE is created, name it <i>name</i> instead of a default name. Use of <code>--backup-be-name</code> implies <code>--require-backup-be</code> .
<code>--deny-new-be</code>	Do not create a new BE. The install, update, uninstall, or revert operation is not performed if a new BE is required.
<code>--require-new-be</code>	Create a new BE. Without this option, a BE is created based on image policy. See “Boot Environment Policy Image Properties” on page 52 for an explanation of when BEs are created automatically. This option cannot be combined with <code>--require-backup-be</code> .
<code>--be-name <i>name</i></code>	If a BE is created, name it <i>name</i> instead of a default name. Use of <code>--be-name</code> implies <code>--require-new-be</code> .

Installing a New Package

By default, the newest version of a package that is compatible with the rest of the image is installed from the first publisher in the publisher search order that offers the package.

If the package is already installed, the package is updated by installing the newest version of the package that is compatible with the rest of the image from the publisher that provided the currently installed version.

If the image has more than one publisher enabled, you can control which publisher provides a package by setting publisher stickiness and search order or by specifying the publisher in the package FMRI. You can also specify the version you want to install in the package FMRI. See [“Fault Management Resource Identifiers” on page 13](#) for a description of a package FMRI. See [“Configuring Publishers” on page 43](#) for information about setting publisher stickiness and search order.

```
/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 ...
```

To install a package from a specific publisher, specify the publisher name in the *pkg_fmri_pattern*. In the following example, `isv.com` is the name of the publisher.

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

To install a specific version of a package, specify the version information in the *pkg_fmri_pattern*.

```
# pkg list -avH vim
pkg://solaris/editor/vim@7.3.254,5.11-0.174.0.0.0.504:20110921T002716Z    ---
# pkg install vim@7.3.254,5.11-0.174
```

To explicitly request the latest version of a package, use `latest` for the version portion of `pkg_fmri_pattern`.

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

Use the `-g` option to temporarily add the specified package repository or package archive to the list of sources in the image from which to retrieve package data. After `install` or `update`, any packages provided by publishers not found in the image are added to the image configuration without an origin.

Use the `--accept` option to indicate that you agree to and accept the terms of the licenses of the packages that are updated or installed. If you do not provide this option, and any package licenses require acceptance, the installation operation fails. Use the `--licenses` option to display all of the licenses for the packages that are installed or updated as part of this operation.

When you specify the `--no-index` option, the search indices are not updated after the operation has completed successfully. Specifying this option might save some time if you are installing a large number of packages. When you are finished with all `install`, `update`, and `uninstall` operations, you can use `pkg refresh` to update the list of available packages and publisher metadata for each publisher specified. If no publishers are specified, the refresh is performed for all publishers.

When you specify the `--no-refresh` option, the repositories for the image's publishers are not contacted to retrieve the newest list of available packages and other metadata.

Installing a Package into a New Boot Environment

Tip – Explicitly specifying a new BE is the safest way to install or update. See “[Boot Environment Policy Image Properties](#)” on page 52 for information about when BEs are created.

The new BE is a clone of the current BE with the specified install, uninstall, or update changes applied. The current BE is not modified. The system is not automatically restarted. The new BE is the default boot selection the next time you restart the system. The current BE is still available to be booted.

If you specify the `--no-be-activate` option, the new BE is not the default boot selection the next time you reboot.

Use the `--be-name` option to force a new BE to be created or to give the new BE a meaningful name.

```
# 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
```

The `pkg list` command reports that the `group/feature/amp` package is not installed because the `group/feature/amp` package is not installed in the current BE. The `group/feature/amp` package is installed in the new `s11amp` BE.

Use the `beadm list` command to check that the system has a new active BE named `s11amp`. The “N” BE is currently booted; the “R” BE is the default on reboot. Use the `beadm activate` command to change which BE is the default on reboot.

```
# 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
```

Check that the `group/feature/amp` package is installed in the new BE. The “i” in the I column indicates that the `group/feature/amp` package is installed.

```
# 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--
```

Remember to unmount the s11amp BE.

```
# 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
```

Rejecting a Package

Use the `--reject` option of the `pkg install` command to prevent packages with names that match the specified `pkg_fmri_pattern` from being installed. If matching packages are already installed, they are removed as part of this operation. Rejected packages that are the target of group dependencies are placed on the avoid list. See [“Marking Packages To Be Avoided” on page 47](#) for information about the avoid list.

```
# pkg install -nv --reject cvs developer-gnu
```

Updating a Package

You can use either the `install` or `update` subcommand to update an installed package to the newest version of the package that is compatible with the rest of the image from the publisher that provided the currently installed version. To avoid unintentionally installing a package that was not already installed, use the `pkg update` command to update packages.

If the image has more than one publisher enabled, you can control which publisher provides a package by setting publisher stickiness and search order or by specifying the publisher in the package FMRI. You can also specify the version you want to install in the package FMRI. See [“Fault Management Resource Identifiers” on page 13](#) for a description of a package FMRI. See [“Configuring Publishers” on page 43](#) for information about setting publisher stickiness and search order.

```
/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 ...]
```

To explicitly request the latest version of a package, use `latest` for the version portion of *pkg_fmri_pattern*.

```
# pkg update vim@latest
```

You can specify a package version older than the version that is currently installed to perform an in-place downgrade. Any preserved configuration files that are part of packages to be downgraded and that have been changed since the original version was installed are renamed with the extension `.update`. For more information about how the package system determines which files to preserve, and how these files are preserved during package upgrades, see “File Actions” in the `pkg(5)` man page.

Use the `-g` option to temporarily add the specified package repository or package archive to the list of sources in the image from which to retrieve package data. After `install` or `update`, any packages provided by publishers not found in the image are added to the image configuration without an origin.

Use the `--accept` option to indicate that you agree to and accept the terms of the licenses of the packages that are updated. If you do not provide this option, and any package licenses require acceptance, the update operation fails. Use the `--licenses` option to display all of the licenses for the packages that are updated as part of this operation.

When you specify the `--no-index` option, the search indices are not updated after the operation has completed successfully. Specifying this option might save some time if you are installing a large number of packages. When you are finished with all `install`, `update`, and `uninstall` operations, you can use `pkg refresh` to update the list of available packages and publisher metadata for each publisher specified. If no publishers are specified, the refresh is performed for all publishers.

If you use the `pkg update` command with no *pkg_fmri* specified, or if the *pkg_fmri* specified is an asterisk character (`*`), all installed packages that have updates available are updated. Use the `pkg list -u` command to display the list of packages that have updates available. When you specify the `-f` option when updating all installed packages, the client up-to-date check is not executed.

Fixing Package Problems

An example of a problem that could occur after a package is installed is that a file delivered by the package becomes corrupted. In the example shown in this section, the `/usr/share/auto_install/manifest/default.xml` file has been deleted.

Use the `pkg search` command to determine which package delivered the missing file:

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


Verifying Package Installation

Use the `pkg verify` command to validate the installation of packages in the current image.

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

If current signature policy for related publishers is not ignore, the signatures of each package are validated based on policy. See `signature-policy` in “[Properties For Signing Packages](#)” on [page 54](#) for an explanation of how signature policies are applied.

Use the `-H` option to omit the headers from the verification output. Use the `-q` option to print nothing but return failure if any fatal errors are found. Use the `-v` option to include informational messages regarding packages.

```
# 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
```

Fixing Verification Errors

Use the `pkg fix` command to fix package installation errors reported by the `pkg verify` command.

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

Verification of installed package content is based on a custom content analysis that might return different results than those of other programs.

Use the `--accept` option to indicate that you agree to and accept the terms of the licenses of the packages that are updated or installed. If you do not provide this option, and any package licenses require acceptance, the `fix` operation fails. Use the `--licenses` option to display all of the licenses for the packages that are updated as part of this operation.

```
# 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
Image State Update Phase

ITEMS
2/2

Restoring a File

Use the `pkg revert` command to restore files to their as-delivered condition.

```
/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 ...)
```

Either all files tagged with a particular *tag-name*, or individual files can be reverted. File ownership and protections are also restored.



Caution – Reverting some editable files to their default values can make the system unbootable, or cause other malfunctions.

Uninstalling Packages

Use the `pkg uninstall` command to remove installed packages.

```
/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 ...
```

If a package is the subject of a group dependency, uninstalling the package places it on the avoid list. See [“Marking Packages To Be Avoided” on page 47](#) for information about the avoid list.

When you specify the `--no-index` option, the search indices are not updated after the operation has completed successfully. Specifying this option might save some time if you are installing a large number of packages. When you are finished with all `install`, `update`, and `uninstall` operations, you can use `pkg refresh` to update the list of available packages and publisher metadata for each publisher specified. If no publishers are specified, the refresh is performed for all publishers.

Configuring Installed Images

This chapter shows how to configure characteristics that apply to an entire image, such as configuring package publishers, restricting which packages can be installed, setting package signing policy, and configuring BE policy.

Configuring Publishers

To install and update software, you need to be able to contact a package repository.

Displaying Publisher Information

Use the `pkg publisher` command to display information about package publishers configured for this image. The publishers are listed in the order in which they are searched to find packages when the publisher is not specified in the package FMRI.

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

By default, the `solaris` publisher is configured on a newly installed Oracle Solaris 11 system. Use the `pkg publisher` command to check the origin of your publisher.

```
$ 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/
```

Specify publishers by name to display detailed configuration for those publishers.

```
$ 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
```

Use the `-P` option to display only the first publisher in the publisher search order. Use the `-n` option to display only enabled publishers. The `-H` option omits headers in the output.

Adding, Modifying, or Removing Package Publishers

Use the `pkg set-publisher` command to perform the following operations:

- Configure a new publisher.
- Set publisher origins and mirrors.
- Enable or disable a publisher. A newly-added publisher is enabled by default. A disabled publisher is not used when populating the package list or in install, uninstall, or update package operations. The properties for a disabled publisher can still be set and viewed. If only one publisher is enabled, that publisher cannot be disabled.
- Set publisher stickiness. A newly-added publisher is sticky by default. If a publisher is non-sticky, then a package that was installed from this publisher could be updated from another publisher.
- Set publisher search order. A newly-added publisher is last in the search order by default. The publisher search order is used to find packages to install. The publisher search order is used to find packages to update if the publisher that the package was originally installed from is non-sticky.
- Specify SSL keys and certificates for a publisher.
- Set and unset a publisher property, and add and remove a publisher property value. See [“Configuring Package Signature Properties” on page 55](#).

```
/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]
```

The following command adds a new publisher named `data.com` with an origin URI specified with the `-g` option and sets this publisher to be first in the search order. Use the `-P` option or the `--search-first` option to set the specified publisher first in the search order.

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

The following command enables the `example.com` publisher and sets it ahead of the `isv.com` publisher in the search order.

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

Use the `-p` option to retrieve publisher configuration information from the specified repository URI. If a publisher is specified, then only the matching publisher is added or updated. If no publisher is specified, all publishers are added or updated as appropriate. The `-p` option cannot be combined with the `-g`, `--add-origin`, `-G`, `--remove-origin`, `-m`, `--add-mirror`, `-M`, `--remove-mirror`, `--disable`, `--enable`, `--no-refresh`, or `--reset-uuid` options.

To change the origin URI for a publisher, add the new URI and remove the old URI. Use the `-g` option to add a new origin URI. Use the `-G` option to remove the old origin URI.

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

Use the `-m` option to add a URI as a mirror for the specified publisher. See [“Repository Origins and Mirrors” on page 14](#) for an explanation of the difference between an origin and a mirror. Use the `-M` option to remove a URI as a mirror for the specified publisher.

```
# 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/
```

Use the `-k` option to specify the client SSL key. Use the `-c` option to specify the client SSL certificate. Use the `--approve-ca-cert` option to add the specified certificate as a CA certificate that is trusted. The hashes of the user approved CA certificates are listed in the output of the `pkg publisher` command for this publisher. See [“Displaying Publisher Information” on page 43](#).

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

Use the `--revoked-ca-cert` option to treat the specified certificate as revoked. The hashes of the user revoked CA certificates are listed in the output of the `pkg publisher` command for this publisher.

Use the `--unset-ca-cert` option to remove the specified certificate from the list of approved and the list of revoked certificates.

When you specify the `--no-refresh` option, the repositories for the image's publishers are not contacted to retrieve the newest list of available packages and other metadata.

Use the `--reset-uuid` option to choose a new unique identifier that identifies this image to its publisher.

Use the `pkg unset-publisher` command to remove a publisher.

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

Locking Packages to a Specified Version

Use the `pkg freeze` command to constrain a package version. One example of a time to freeze a package is when you do not want the package in a non-global zone to be updated when the global zone is updated.

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

If no version is provided in *pkg_fmri_pattern*, the named package must be installed and is constrained to the version installed on the system. If a version is provided in *pkg_fmri_pattern*, then this constraint, or `freeze`, acts as if an `incorporate` dependency were installed where the `fmri` attribute had the value of the provided package version.

When a package that is frozen is installed or updated, it must end up at a version that matches the version at which it was frozen. For example, if a package was frozen at 1.2, then it could be updated to 1.2.1, 1.2.9, 1.2.0.0.1, and so on. That package could not end up at 1.3, or 1.1.

A publisher specified in the *pkg_fmri_pattern* is used to find matching packages. However, publisher information is not recorded as part of the freeze. A package is frozen with respect to its version only, not its publisher.

Freezing a package that is already frozen replaces the frozen version with the newly specified version.

If no packages are specified, information about currently frozen packages is displayed: package names, versions, when the package was frozen, and any associated reasons.

Freezing a package does not prevent removal of the package. No warning is displayed if the package is removed.

Use the `-c` option to record the reason the package is being frozen. The reason is shown if a freeze prevents an installation or update from succeeding.

Use the `-n` option to perform a trial run of the operation, displaying the list of packages that would be frozen without freezing any packages.

A freeze is never lifted automatically by the packaging system. To relax a constraint, use the `pkg unfreeze` command.

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

Remove the constraints that freezing imposes from the specified packages. Any versions provided are ignored.

Use the `-n` option to perform a trial run of the unfreeze, displaying the list of packages that would be unfrozen without unfreezing any packages.

Marking Packages To Be Avoided

Use the `pkg avoid` command to avoid the specified packages if they are the target of a group dependency.

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

With no arguments, the `pkg avoid` command displays each avoided package along with any packages that have a group dependency on that package.

With `pkg_fmri_pattern` specified, the `pkg avoid` command places the package names that currently match the specified patterns on the avoid list. Only packages that are not currently installed can be avoided. If a package is currently the target of a group dependency, uninstalling the package places it on the avoid list.

If a package is on the avoid list, installing it removes it from that list. Packages that are on the avoid list are installed if needed to satisfy a required dependency. If that dependency is removed, the package is uninstalled.

Use the `pkg unavoid` command to remove the specified packages from the avoid list.

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

Packages on the avoid list that match an installed package's group dependency cannot be removed from the avoid list using this subcommand. To remove a package from the avoid list that matches a group dependency, install the package.

Controlling Installation of Optional Components

Software can have components that are optional and components that are mutually exclusive. Examples of optional components include locales and documentation. Examples of mutually exclusive components include SPARC or x86 and debug or non-debug binaries. In IPS, optional components are called *facets* and mutually exclusive components are called *variants*.

Facets and variants are special properties of the image and cannot be set on individual packages. To view the current values of the facets and variants set on the image, use the `pkg facet` and `pkg variant` commands. To modify the values of the facets and variants set on the image, use the `pkg change-facet` and `pkg change-variant` commands. See the [pkg\(1\)](#) man page and the examples below.

Facets and variants are specified as tags on package actions. Each facet and variant tag has a name and a value. A single action can have multiple facet and variant tags. An example of a component with multiple facet and variant tags is an architecture-specific header file that is used by developers, or a component that is only for a SPARC global zone.

An example of a variant tag is `variant.arch=sparc`. An example of a facet tag is `facet.devel=true`. Facets and variants are often referred to without the leading `facet.` and `variant.`.

Facets are boolean: They can be set only to `true` (enabled) or `false` (disabled). By default, all facets are considered to be set to `true` in the image. A facet tag on an action should only have the value `true`; other values have undefined behavior. A facet set on the image can be a full facet such as `doc.man` or a pattern such as `locale.*`. This is useful when you want to disable a portion of the facet namespace, and only enable individual facets within it. For example, you could disable all locales and then only enable one or two specific locales, as shown in the following example:

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

Most variants can have any number of values. For example, the `arch` variant can be set to `i386`, `sparc`, `ppc`, `arm`, or whatever architectures the distribution supports. (Only `i386` and `sparc` are used in Oracle Solaris.) The exception are the debug variants. The debug variants can only be set to `true` or `false`; other values have undefined behavior. If a file action has both non-debug and debug versions, both versions must have the applicable debug variant explicitly set, as shown in the following example:

```
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
```

The variant value must be set on the image in order for a package using the variant to be installed. The `arch` and `zone` variants are set by the program that creates the image and installs its initial contents. The `debug.*` variants are `false` in the image by default.

The facets and variants set on the image affect whether a particular action is installed.

- Actions with no facet or variant tags are always installed.
- Actions with facet tags are installed unless all of the facets or facet patterns matching the tags are set to `false` on the image. If any facet is set to `true` or is not explicitly set (`true` is the default), then the action is installed.
- Actions with variant tags are installed only if the values of all the variant tags are the same as the values set in the image.

- Actions with both facet and variant tags are installed if both the facets and the variants allow the action to be installed.

You can create your own facet and variant tags. The following tags are in common use in Oracle Solaris.

Variant Name	Possible Values
variant.arch	sparc, i386
variant.opensolaris.zone	global, nonglobal
variant.debug.*	true, false

The following list shows a small sample of the facet tags that are used in Oracle Solaris:

```
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
```

You can display the values of variants and facets that are set on the current image, and you can change variants and facets in the current image. Changing variants and facets might update a large number of packages and might require a new BE. Use `-nv` to review what changes will be made before you make any changes.

Displaying a Variant

Use the `pkg variant` command to display the values of variants that are set.

```
/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
```

Changing a Variant

Use the `pkg change-variant` command to change the value of a variant.

```
/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 ...
```

The following command produces a large amount of output since so many packages would be affected. Notice that a new BE would not be created by default, but a backup BE would be created. See “[Boot Environment Policy Image Properties](#)” on page 52 for information about when BEs are created.

Use the `-n` option to see what would change if you performed the operation without `-n`, but make no actual changes.

```
# 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
...
```

Displaying a Facet

Use the `pkg facet` command to display the values of facets that are set.

```
/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
```

Changing a Facet

Use the `pkg change -facet` command to change the value of a facet.

```
/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] ...
```

Use the `-n` option to see what would change if you performed the operation without `-n`, but make no actual changes.

If the facet value is set to `None`, the facet specification is removed from the current image.

The following command produces a large amount of output since so many packages would be affected. Notice the amount of additional space that would be used is in gigabytes, not megabytes. This operation might require a large amount of time and cause a large amount of network traffic between this image and the package repository. Notice that a new BE would not be created by default, but a backup BE would be created. See [“Boot Environment Policy Image Properties” on page 52](#) for information about when BEs are created.

```
# 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
...
```

Updating an Image

Use the `pkg update` command with no `pkg-fmri` specified, or with an asterisk character (*) as the `pkg-fmri`, to update all installed packages that have updates available to the newest version allowed by the constraints imposed on the system by installed packages and publisher configuration. If non-global zones are configured in the current image, these zones are also updated. See [“About Packages and Zones” in Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management](#).

```
/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 ...]
```

Use the `pkg list -u` command to display the list of packages that have updates available. The `pkg update` operation often results in a new BE being automatically created. Use `pkg update -nv` to review what changes would be made to the current image or in a new BE. Use the `--be-name` option to force a new BE to be created or to give the new BE a meaningful name.

Tip – Explicitly specifying a new BE is the safest way to install or update. See “[Boot Environment Policy Image Properties](#)” on page 52 for information about when BEs are created.

By default, each package is updated from the publisher that provided the currently installed version. You can control the publisher that provides packages by specifying publisher stickiness and search order. See “[Adding, Modifying, or Removing Package Publishers](#)” on page 44.

When you specify the `-f` option when updating all installed packages, the client up-to-date check is not executed.

Use the `-g` option to temporarily add the specified package repository or package archive to the list of sources in the image from which to retrieve package data. After update, any packages provided by publishers not found in the image are added to the image configuration without an origin.

Use the `--accept` option to indicate that you agree to and accept the terms of the licenses of the packages that are updated. If you do not provide this option, and any package licenses require acceptance, the update operation fails. Use the `--licenses` option to display all of the licenses for the packages that are updated as part of this operation.

When you specify the `-no-refresh` option, the repositories for the image's publishers are not contacted to retrieve the newest list of available packages and other metadata.

When you specify the `--no-index` option, the search indices are not updated after the operation has completed successfully. Specifying this option might save some time if you are installing a large number of packages. When the update operation finishes, you can use `pkg refresh` to update the list of available packages and publisher metadata for each publisher specified. If no publishers are specified, the refresh is performed for all publishers. If a new BE was created, do this publisher refresh in the new BE.

Configuring Image and Publisher Properties

To implement image policies, set image properties. This section describes image and publisher properties and how to set these properties. See also “Image Properties” in the `pkg(1)` man page for descriptions of image properties.

Boot Environment Policy Image Properties

An image is a location where IPS packages can be installed and where other IPS operations can be performed.

A boot environment (BE) is bootable instance of an image. You can maintain multiple BEs on your system, and each BE can have different software versions installed. When you boot your system, you have the option to boot into any of the BEs on the system. A new BE can be created automatically as a result of package operations. You can also explicitly create a new BE. Whether a new BE is created depends on image policy, as discussed in this section

By default, a new BE is automatically created when you perform one of the following operations:

- Update particular key system packages such as some drivers and other kernel components. This can happen when you install, uninstall, update, change variant, or change facet. Often a new BE is created when you execute the `pkg update` command to update all packages that have updates available.
- Specify any of the following options: `--be-name`, `--require-new-be`, `--backup-be-name`, `--require-backup-be`.
- Set the `be-policy` image policy to `always-new`. Under this policy, all package operations are performed in a new BE set as active on the next boot.

When a new BE is created, the system performs the following steps:

1. Creates a clone of the current BE.

The clone BE includes everything hierarchically under the main root dataset of the original BE. Shared file systems are not under the root dataset and are not cloned. Instead, the new BE accesses the original shared file systems.
2. Updates the packages in the clone BE, but does not update any packages in the current BE.

If non-global zones are configured in the current BE, these existing zones are configured in the new BE.
3. Sets the new BE as the default boot choice the next time the system is booted, unless `--no-be-activate` is specified. The current BE remains as an alternate boot choice.

If a new BE is required but not enough space is available to create a new BE, you might be able to delete existing unneeded BEs. For more information about BEs, see [Creating and Administering Oracle Solaris 11 Boot Environments](#).

See “Setting Image Properties” on page 57 for instructions to set the image properties described below.

`be-policy`

Specifies when a boot environment is created during packaging operations. The following values are allowed:

- | | |
|-------------------------|--|
| <code>default</code> | Apply the default BE creation policy: <code>create-backup</code> . |
| <code>always-new</code> | Require a reboot for all package operations by performing them in a new BE set as active on the next boot. A backup BE is not created unless explicitly requested. |

This policy is the safest, but is more strict than most sites need since no packages can be added without a reboot.

`create-backup`

For package operations that require a reboot, this policy creates a new BE set as active on the next boot. If packages are modified or content that could affect the kernel is installed and the operation affects the live BE, a backup BE is created but not set as active. A backup BE can also be explicitly requested.

This policy is potentially risky only if newly installed software causes system instability, which is possible, but relatively rare.

`when-required`

For package operations that require a reboot, this policy creates a new BE set as active on the next boot. A backup BE is not created unless explicitly requested.

This policy carries the greatest risk since if a packaging change to the live BE makes further changes impossible, a recent fallback BE might not exist.

Properties For Signing Packages

If you are installing signed packages, set the image properties and publisher properties described in this section to verify package signatures.

Image Properties for Signed Packages

Configure the following image properties to use signed packages.

`signature-policy`

The value of this property determines what checks will be performed on manifests when installing, updating, modifying, or verifying packages in the image. The final policy applied to a package depends on the combination of image policy and publisher policy. The combination will be at least as strict as the stricter of the two policies taken individually. By default, the package client does not check whether certificates have been revoked. To enable those checks, which might require the client to contact external web sites, set the `check-certificate-revocation` image property to `true`. The following values are allowed:

<code>ignore</code>	Ignore signatures for all manifests.
<code>verify</code>	Verify that all manifests with signatures are validly signed, but do not require all installed packages to be signed. This is the default value.
<code>require-signatures</code>	Require that all newly installed packages have at least one valid signature. The <code>pkg fix</code> and <code>pkg verify</code> commands also warn if an installed package does not have a valid signature.

`require-names` Follow the same requirements as `require-signatures` but also require that the strings listed in the `signature-required-names` image property appear as a common name of the certificates used to verify the chains of trust of the signatures.

`signature-required-names`

The value of this property is a list of names that must be seen as common names of certificates while validating the signatures of a package.

`trust-anchor-directory`

The value of this property is the path name of the directory that contains the trust anchors for the image. This path is relative to the image.

The default value is `ignore`.

Publisher Properties for Signed Packages

Configure the following publisher properties to use signed packages from a particular publisher.

`signature-policy`

The function of this property is identical to the function of the `signature-policy` image property except that this property only applies to packages from the specified publisher.

`signature-required-names`

The function of this property is identical to the function of the `signature-required-names` image property except that this property only applies to packages from the specified publisher.

Configuring Package Signature Properties

Use the `set-property`, `add-property-value`, `remove-property-value`, and `unset-property` subcommands to configure package signature properties for this image.

Use the `--set-property`, `--add-property-value`, `--remove-property-value`, and `--unset-property` options of the `set-publisher` subcommand to specify signature policy and required names for a particular publisher.

The following example configures this image to require all packages to be signed. This example also requires the string “oracle.com” to be seen as a common name for one of the certificates in the chain of trust.

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

The following example configures this image to require all signed packages to be verified.

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

The following example configures this image to require that all packages installed from the publisher `example.com` must be signed.

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

The following example adds a required signature name. This example adds the string `trustedname` to the image's list of common names that must be seen in a signature's chain of trust to be considered valid.

```
# pkg add-property-value signature-require-names trustedname
```

The following example removes a required signature name. This example removes the string `trustedname` from the image's list of common names that must be seen in a signature's chain of trust to be considered valid.

```
# pkg remove-property-value signature-require-names trustedname
```

The following example adds a required signature name for a specified publisher. This example adds the string `trustedname` to the `example.com` publisher's list of common names that must be seen in a signature's chain of trust to be considered valid.

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

Additional Image Properties

`ca-path`

Specifies a path name that points to a directory where CA certificates are kept for SSL operations. The format of this directory is specific to the underlying SSL implementation. To use an alternate location for trusted CA certificates, change this value to point to a different directory. See the `CApath` portions of `SSL_CTX_load_verify_locations(3openssl)` for requirements for the CA directory.

The default value is `/etc/openssl/certs`.

`check-certificate-revocation`

If this is set to `True`, the package client attempts to contact any CRL distribution points in the certificates used for signature verification to determine whether the certificate has been revoked since being issued.

The default value is `False`.

`flush-content-cache-on-success`

If this is set to `True`, the package client removes the files in its content-cache when install or update operations complete. For update operations, the content is removed only from the source BE. When a packaging operation next occurs in the destination BE, the package client flushes its content cache if this option has not been changed.

This property can be used to keep the content-cache small on systems with limited disk space. This property can cause operations to take longer to complete.

The default value is `True`.

`mirror-discovery`

This property tells the client to discover link-local content mirrors using mDNS and DNS-SD. If this property is set to `True`, the client attempts to download package content from mirrors it dynamically discovers. To run a mirror that advertises its content via mDNS, see [pkg.depotd\(1m\)](#).

The default value is `False`.

`send-uuid`

Send the image's Universally Unique Identifier (UUID) when performing network operations. Although users can disable this option, some network repositories might refuse to talk to clients that do not supply a UUID.

The default value is `True`.

`use-system-repo`

This property indicates whether the image should use the system repository as a source for image and publisher configuration and as a proxy for communicating with the publishers provided. See [pkg.sysrepo\(1m\)](#) for information about system repositories.

The default value is `ignore`.

Setting Image Properties

Use the `set-property`, `add-property-value`, `remove-property-value`, and `unset-property` subcommands to configure properties for this image.

```
/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 ...
```

Displaying the Values of Image Properties

Use the `pkg property` command to view the properties of an image.

```
$ 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

The preferred-authority and publisher-search-order properties can be set using pkg set-publisher command options. See [“Adding, Modifying, or Removing Package Publishers” on page 44](#).

Setting the Value of an Image Property

Use the pkg set-property command to set the value of an image property or add and set a property.

The following example sets the value of the mirror-discovery property.

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

Resetting the Value of an Image Property

Use the pkg unset-property command to reset the values of the specified properties to their default values.

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

Creating an Image

An image is a location where IPS packages and their associated files, directories, links, and dependencies can be installed, and where other IPS operations can be performed.

An image can be one of three types:

- Full images are capable of providing a complete system. In a full image, all dependencies are resolved within the image itself, and IPS maintains the dependencies in a consistent manner. After you have completed an installation of the Oracle Solaris OS, the root file system and its contents are contained in a full image.
- Partial images are linked to a full image (the parent image), but do not provide a complete system on their own. A non-global zone is a partial image. Use the `-z` or `--zone` option to set an appropriate variant. In a zone image, IPS maintains the non-global zone consistent with its global zone as defined by dependencies in the packages. See [Part II, “Oracle Solaris Zones,”](#) in *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management* to learn about non-global zones.
- User images contain only relocatable packages.

```
/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
```

At the location given by *dir*, create an image suitable for package operations. The default image type is user (`-U` or `--user`). The image type can be set to a full image (`-F` or `--full`) or to a partial image (`-P` or `--partial`) linked to the full image enclosing the given *dir* path.

To run the new image in a non-global zone context, use the `-z` or `--zone` option to set an appropriate variant.

A package repository URI must be provided using the `-p` or `--publisher` option. If a publisher name is also provided, then only that publisher is added when the image is created. If a publisher name is not provided, then all publishers known by the specified repository are added to the image. An attempt to retrieve the catalog associated with this publisher is made following the initial creation operations.

Use the `-g` option to specify additional origins. Use the `-m` option to specify mirrors.

For publishers using client SSL authentication, use the `-c` or `-k` options to register a client key and client certificate. This key and certificate are used for all publishers added during image creation.

Use the `-f` option to force the creation of an image over an existing image. Use this option with caution.

When you specify the `--no-refresh` option, the repositories for the image's publishers are not contacted to retrieve the newest list of available packages and other metadata.

Use the `--variant` option to set the specified variant to the indicated value. Use the `--facet` option to set the specified facet to the indicated value.

Viewing Operation History

Use the `pkg history` command to view the command history in the current image.

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

Use the `-l` option to display more information, including the outcome of the command, the time the command completed, the version and name of the client used, the name of the user who performed the operation, and any errors encountered while executing the command.

Use the `-n` option to display only the specified number of most recent operations.

```
$ 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
```

Use the `-o` option to display output using the specified comma-separated list of column names. See the list of column names in [pkg\(1\)](#).

```
# 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
```

Use the `-t` option to log records for a comma-separated list of timestamps, formatted with `%Y-%m-%dT%H:%M:%S` (see `strftime(3C)`). To specify a range of times, use a hyphen (`-`) between a start and finish timestamp. The keyword `now` can be used as an alias for the current time. If the timestamps specified contain duplicate timestamps or overlapping date ranges, only a single instance of each duplicate history event is printed.

Use the `pkg purge-history` command to delete all command history information.

```
# pkg purge-history
```