

Guide d'Oracle Solaris 11 Express Image Packaging System

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.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. 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. UNIX est une marque déposée concédée sous licence par X/Open Company, Ltd.

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.

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.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. 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. UNIX est une marque déposée concédée sous licence par X/Open Company, Ltd.

Table des matières

Préface	5
1 Introduction à Image Packaging System	9
Image Packaging System	9
IPS - Concepts	10
Packages IPS	10
Identificateurs de ressource de gestion des pannes	10
Éditeurs et référentiels	11
Sources et miroirs du référentiel	11
Images et environnements d'initialisation	12
Variantes et facettes de package	13
2 Interfaces graphiques IPS	15
Utilisation du Gestionnaire de packages	15
Options de ligne de commande du Gestionnaire de packages	16
Utilisation de l'Installation Web	16
Utilisation du Gestionnaire de mises à jour	18
Options de ligne de commande du Gestionnaire de mises à jour	19
3 Utilisation de packages	21
Tâches de gestion des packages	21
Installation et gestion des packages	23
▼ Installation ou mise à jour d'un package	23
▼ Mise à jour d'un package	24
▼ Visualisation d'une action d'installation sans installer	24
▼ Vérification de l'installation d'un package	25
▼ Correction des erreurs d'installation de package	25

▼ Désinstallation des packages	26
▼ Désinstallation d'un package à partir d'un environnement d'initialisation inactif	26
▼ Recherche de packages	26
▼ Affichage du contenu d'un package	28
▼ Affichage des informations sur les packages	28
▼ Présentation sous forme de liste des informations d'état du package	29
▼ Affichage d'une variante	31
▼ Modification d'une variante	31
▼ Affichage d'une facette	32
▼ Modification d'une facette	32
▼ Affichage et suppression de l'historique des opérations	33
Gestion des éditeurs de package	33
▼ Affichage des informations concernant l'éditeur	33
▼ Ajout, modification ou suppression d'un éditeur de package	34
Installation de packages signés	37
Propriétés de l'image pour les packages signés	37
Propriétés de l'éditeur pour les packages signés	38
Configuration des propriétés de signature de package	38
4 Création et gestion des images	41
Pourquoi créer une image ?	41
Meilleures pratiques pour la création d'images	42
Créer une image	43
Afficher, définir et supprimer des propriétés de l'image	43
Mise à jour d'une image	44
A Aide-mémoire de la commande	47
Commandes IPS	47
Page de manuel pkg(1)	48
Glossaire	69

Préface

Le *Guide d'Oracle Solaris 11 Express Image Packaging System* décrit Oracle Solaris Image Packaging System (IPS). L'outil IPS vous permet d'installer, mettre à jour et gérer des packages de logiciels pour le système d'exploitation Oracle Solaris. IPS vous permet également de créer vos propres packages de logiciels, de créer et de gérer des référentiels de packages et de mettre en miroir des référentiels de packages existants.

Utilisateurs de ce manuel

Ce manuel s'adresse aux administrateurs système qui installent et gèrent des logiciels et gèrent des images système.

Organisation de ce document

- [Chapitre 1, “Introduction à Image Packaging System”](#) décrit Image Packaging System et des composants tels que des packages, des éditeurs et des référentiels.
- [Chapitre 2, “Interfaces graphiques IPS”](#) explique comment utiliser le Gestionnaire de packages et le Gestionnaire de mises à jour, y compris l'Installation Web .
- [Chapitre 3, “Utilisation de packages”](#) fournit des exemples montrant comment gérer des packages de logiciels et des éditeurs de package.
- [Chapitre 4, “Création et gestion des images”](#) décrit comment créer et mettre à jour des images et comment définir des propriétés d'image.
- [Annexe A, “Aide-mémoire de la commande”](#) décrit brièvement les commandes IPS et montre la page de manuel pkg(1).
- [Glossaire](#) définit les termes IPS.

Documentation connexe

En plus de ces manuels, reportez-vous à l'aide en ligne du Gestionnaire de packages et aux pages de manuel `pkg(1M)` et `beadm(1M)`.

- *Démarrage d'Oracle Solaris 11 Express* explique comment installer le SE Oracle Solaris et comprend suffisamment d'informations sur IPS pour vous permettre d'installer des logiciels supplémentaires après l'installation initiale et maintenir à jour votre logiciel.
- *Gestion des environnements d'initialisation avec Oracle Solaris 11 Express* fournit des informations complètes sur les environnements d'initialisation ainsi que sur leur utilisation et leur gestion.
- *Guide du programme d'installation automatisée Oracle Solaris 11 Express* explique comment configurer des installations automatiques de clients sur un réseau, y compris comment spécifier des packages à installer et les éditeurs à utiliser.

Documentation, support et formation

Pour obtenir des ressources supplémentaires, reportez-vous aux sites Web suivants :

- [Documentation \(http://docs.sun.com\)](http://docs.sun.com)
- [Assistance \(http://www.oracle.com/us/support/systems/index.html\)](http://www.oracle.com/us/support/systems/index.html)
- [Formation \(http://education.oracle.com\)](http://education.oracle.com) - Cliquez sur le lien Sun dans la barre de navigation à gauche.

Ressources logicielles Oracle

Le site [Oracle Technology Network \(http://www.oracle.com/technetwork/index.html\)](http://www.oracle.com/technetwork/index.html) propose une large gamme de ressources pour les logiciels Oracle :

- Discutez de problèmes techniques et trouvez des solutions sur les [forums de discussion \(http://forums.oracle.com\)](http://forums.oracle.com).
- Passez à la pratique grâce aux didacticiels étape par étape avec [Oracle By Example \(http://www.oracle.com/technetwork/tutorials/index.html\)](http://www.oracle.com/technetwork/tutorials/index.html).
- Téléchargez un [échantillon de code \(http://www.oracle.com/technology/sample_code/index.html\)](http://www.oracle.com/technology/sample_code/index.html).

Conventions typographiques

Le tableau ci-dessous décrit les conventions typographiques utilisées dans ce manuel.

TABLEAU P-1 Conventions typographiques

Type de caractères	Signification	Exemple
AaBbCc123	Noms des commandes, fichiers et répertoires, ainsi que messages système.	Modifiez votre fichier <code>.login</code> . Utilisez <code>ls -a</code> pour afficher la liste de tous les fichiers. <code>nom_machine%</code> Vous avez reçu du courrier.
AaBbCc123	Ce que vous entrez, par opposition à ce qui s'affiche à l'écran.	<code>nom_machine%</code> su Mot de passe :
<i>aaBBCc123</i>	Paramètre fictif : à remplacer par un nom ou une valeur réel(le).	La commande permettant de supprimer un fichier est <code>rm nom_fichier</code> .
<i>AaBbCc123</i>	Titres de manuel, nouveaux termes et termes importants.	Reportez-vous au chapitre 6 du <i>Guide de l'utilisateur</i> . Un <i>cache</i> est une copie des éléments stockés localement. <i>N'enregistrez pas</i> le fichier. Remarque : en ligne, certains éléments mis en valeur s'affichent en gras.

Invites de shell dans les exemples de commandes

Le tableau suivant présente l'invite système UNIX par défaut et l'invite superutilisateur pour les shells faisant partie du SE Oracle Solaris. L'invite système par défaut qui s'affiche dans les exemples de commandes dépend de la version Oracle Solaris.

TABLEAU P-2 Invites de shell

Shell	Invite
Bash shell, korn shell et bourne shell	\$
Bash shell, korn shell et bourne shell pour superutilisateur	#
C shell	<code>nom_machine%</code>

TABLEAU P-2 Invites de shell (Suite)

Shell	Invite
C shell pour superutilisateur	nom_machine#

Introduction à Image Packaging System

Oracle Solaris Image Packaging System (IPS) est une structure permettant d'installer, de mettre à jour et de supprimer des packages de logiciels pour le système d'exploitation Oracle Solaris. IPS vous permet également de créer vos propres packages de logiciels, de créer et de gérer des référentiels de packages et de mettre en miroir des référentiels de packages existants.

Image Packaging System

Le logiciel Oracle Solaris est distribué en packages IPS. Les packages IPS sont stockés dans des référentiels de packages IPS qui sont alimentés par des éditeurs IPS. Un sous-ensemble de capacités accessibles via l'interface de ligne de commande IPS est disponible par le biais des interfaces graphiques IPS du Gestionnaire de packages et du Gestionnaire de mises à jour.

Les outils IPS offrent les capacités suivantes :

- lister, rechercher, installer, mettre à jour et supprimer des packages de logiciels ;
- lister, ajouter et supprimer des éditeurs de package ; modifier des attributs d'éditeur comme la priorité de recherche et la persistance ;
- mettre à jour votre système vers une nouvelle version logicielle ;
- créer des miroirs de référentiels de packages existants ;
- créer des référentiels de packages ;
- créer et éditer des packages ;
- lister, créer, renommer, activer et supprimer des environnements d'initialisation.

Pour utiliser IPS, le système d'exploitation Oracle Solaris 11 Express doit être en cours d'exécution. Pour installer la version Oracle Solaris 11 Express, reportez-vous au guide *Démarrage d'Oracle Solaris 11 Express*.

IPS - Concepts

Cette section définit les termes et concepts utilisés dans le reste du guide.

Packages IPS

IPS gère les logiciels en unités de packages. Un *package* IPS est un ensemble de répertoires, de fichiers, de liens, de pilotes, de dépendances, de groupes, d'utilisateurs et d'informations de licence dans un format défini. Cet ensemble représente les objets d'un package pouvant être installés. Les packages ont des attributs tels qu'un nom de package et une description.

Vous pouvez utiliser des commandes pour voir des informations sur le package ou afficher le fichier manifest d'un package.

IPS prend en charge les packages IPS et les packages SVR4.

Il n'existe pas de format sur disque standard pour un package IPS. IPS n'a pas d'équivalent pour les fichiers `.rpm`, les packages SVR4 ou les fichiers `.nbn`.

Identificateurs de ressource de gestion des pannes

Chaque package IPS est représenté par un FMRI (Fault Management Resource Identifiers, identificateurs de ressources de gestion des pannes).

Le FMRI inclut des informations descriptives sur le package telles que son nom, les informations de version et sa date, comme indiqué dans l'exemple ci-dessous :

```
pkg://solaris/library/libc@5.11,5.11-0.75:20071001T163427Z
```

- Méthode : pkg
- Éditeur : solaris
- Catégorie : library
- Nom du package : libc
- Chaîne de version
 - Version du composant : 5.11
 - Version de compilation : 5.11
 - Version de branche : 0.75
 - Moment auquel le package a été publié, au format de base ISO-8601 : 20071001T163427Z

Éditeurs et référentiels

- Un *éditeur* est un nom de domaine direct qui identifie une personne, un groupe de personnes ou une organisation qui publie un ou plusieurs packages.
- Un *référentiel* est un emplacement dans lequel les clients publient et récupèrent des packages. L'emplacement est indiqué par un URI. Un référentiel peut également être qualifié de serveur de dépôt.
- Un référentiel contient des packages d'un seul éditeur.
- Un éditeur peut publier des packages sur plusieurs référentiels.

Plusieurs référentiels peuvent contenir des packages portant le même nom. Les éditeurs peuvent être configurés selon un ordre de recherche préféré dans chaque image. Lorsqu'un package est sélectionné pour être installé, le système recherche en premier l'éditeur préféré pour ce package, puis le second éditeur préféré jusqu'à ce qu'il trouve le package ou que tous les éditeurs aient été recherchés.

Un URI de référentiel ne doit pas nécessairement contenir le nom de l'éditeur. Par exemple, l'éditeur `solaris` publie vers des référentiels hébergés au niveau de `oracle.com`.

Sources et miroirs du référentiel

Un référentiel a une source et zéro miroir ou plus.

- Une *source* de référentiel est l'emplacement d'un référentiel de packages qui contient à la fois des *métadonnées* du package (fichiers manifest et catalogues du package) et le *contenu* du package (fichiers du package).
- Un *miroir* est l'emplacement d'un référentiel de packages qui contient uniquement le contenu du package.

Les miroirs fournissent un sous-ensemble de données fournies par les sources. Les miroirs peuvent être utilisés pour télécharger des fichiers de package. Des métadonnées de package sont téléchargées à partir de la source. Les clients IPS accèdent au référentiel source pour obtenir un catalogue d'éditeur, même lorsque les clients téléchargent le contenu de package à partir d'un miroir.

Images et environnements d'initialisation

- Une *image* est un emplacement dans lequel des packages peuvent être installés.

Une image peut être de trois types différents :

- Les images complètes peuvent fournir un système complet.
 - Les images partielles sont liées à une image complète (image parent) mais ne suffisent pas pour fournir un système complet.
 - Les images d'utilisateur contiennent uniquement des packages réadressables.
- Un *environnement d'initialisation* (EI) est une instance amorçable d'une image. Vous pouvez disposer de plusieurs EI sur votre système, chacun pouvant avoir différentes versions de logiciels installées. Lorsque vous initialisez votre système, vous avez la possibilité de démarrer dans n'importe quel EI sur votre système.

Un nouvel EI est automatiquement créé lorsque vous exécutez l'une des opérations suivantes :

- Installation du &OSOS.
- Mise à jour de packages de système clé particuliers, tels que certains pilotes et autres composants de noyau. Ceci peut se produire lorsque vous installez, désinstallez, mettez à jour, ou modifiez une variante ou une facette.
- Utilisation de la commande `beadm create`.

Il arrive souvent qu'un EI soit créé lorsque vous exécutez la commande `pkg update` pour mettre à jour tous les packages pour lesquels des mises à jour sont disponibles.

Si un EI est créé, le système effectue les opérations suivantes :

1. Il crée un clone de l'EI actuel qui est une image amorçable.

Un clone de l'EI contient l'intégralité du contenu, ordonné dans une arborescence sous le jeu de données root principal de l'EI d'origine. Les systèmes de fichiers partagés ne se trouvent pas sous le jeu de données root et ne sont pas clonés. Au lieu de cela, l'EI accède aux systèmes de fichiers partagés d'origine.

2. Il met à jour les packages dans l'EI clone sans mettre à jour les packages dans l'EI actuel.

Si des zones sont configurées dans l'EI actuel, ces zones existantes sont configurées dans le nouvel EI. Cependant, les packages ne sont pas installés ou mis à jour dans ces zones. Vous devez mettre à jour manuellement chaque zone dans le nouvel EI.

3. Définit le nouvel environnement d'initialisation qui sera utilisé comme environnement par défaut lors de la prochaine initialisation du système. L'environnement d'initialisation actuel figure toujours parmi les choix d'initialisation possibles.

Plusieurs commandes IPS offrent les options suivantes pour prendre en charge la création d'EI. Reportez-vous à la page de manuel correspondante.

- be-name *BName* Assigne *BName* au dernier EI créé. Cette option est ignorée si aucun nouvel EI n'est créé.
- deny-new-be Ne crée pas d'EI, même si l'installation d'un nouvel EI est requise par les composants. Si cette option est spécifiée et qu'un nouvel EI est requis, l'opération n'est pas effectuée.
- require-new-be Crée un EI, même s'il n'est pas requis.

Si un nouvel EI est requis mais que l'espace est insuffisant pour le créer, vous pouvez peut-être supprimer des EI existants inutiles. Pour plus d'informations sur les EI, reportez-vous au guide [Gestion des environnements d'initialisation avec Oracle Solaris 11 Express](#).

Variantes et facettes de package

Les logiciels peuvent avoir des composants optionnels et des composants incompatibles. Les environnements linguistiques et la documentation sont des exemples de composants optionnels. Les binaires SPARC ou x86 et les binaires de débogage et de non-débogage sont des exemples de composants incompatibles.

Dans IPS, les composants optionnels sont appelés des *facettes* et les composants incompatibles sont appelés des *variantes*. Les variantes et les facettes apparaissent comme des repères sur des *actions* IPS.

Les actions représentent les objets d'un système pouvant être installés. Les actions sont décrites dans le fichier manifest d'un package. Chaque action est essentiellement constituée de son nom et d'un attribut clé. Ensemble, elles se rapportent à un objet unique suivant un historique des versions.

Les variantes et les facettes déterminent si une action particulière est sélectionnée ou désélectionnée pour l'installation.

La liste suivante fournit des exemples de repères de facettes et de variantes et leurs valeurs possibles :

Nom	Valeurs
facet.local.*	true, false
facet.doc.man	true, false
facet.doc	true, false
facet.devel.*	true, false
variant.arch	sparc, i386, zos

Nom	Valeurs
<code>variant.debug.*</code>	<code>true, false</code>

Les actions qui ne sont pas marquées avec des facettes ou des variantes sont incluses.

Une action marquée avec une variante non sélectionnée est exclue. Une action qui est marquée avec une ou plusieurs facettes est exclue si aucune des facettes n'est sélectionnée.

- Les repères de variante sont évalués avec l'opérante AND. Si l'un des repères de variante ne correspond pas, l'action n'est pas installée.
- Les repères de facette sont évalués avec OU. Si l'un des repères de facette ne correspond pas, l'action n'est pas exclue.

Une seule action peut avoir plusieurs repères de facette et de variante. Un fichier d'en-tête spécifique à une architecture, utilisé par des développeurs, est un exemple de composant à repères multiples de facette et de variante.

Les variantes et les facettes sont définies au niveau de l'image. Une image avec une variante définie sur une propriété particulière peut posséder uniquement des actions qui correspondent à la variante installée. Par exemple, vous ne pouvez pas installer un package x86 sur une image SPARC.

Les administrateurs système peuvent effectuer les opérations suivantes sur les facettes et les variantes :

- Affichage de la valeur de toutes les variantes ou facettes définies sur l'image actuelle.
- Modification de variantes et facettes sur l'image actuelle. Cette opération entraînera probablement la mise à jour de packages et peut nécessiter un nouvel EI.

Interfaces graphiques IPS

IPS comprend deux outils qui fournissent des interfaces graphiques.

- Le Gestionnaire de packages assure la plupart des opérations de package et d'éditeur et certaines opérations liées à l'environnement d'initialisation (EI). Si vous êtes un nouvel utilisateur du SE Oracle Solaris et des technologies IPS, utilisez le Gestionnaire de packages pour télécharger et installer rapidement des packages.
- Le Gestionnaire de mises à jour met à jour tous les packages pour lesquels une mise à jour est disponible.

Utilisation du Gestionnaire de packages

Le Gestionnaire de packages fournit un sous-ensemble des tâches qui peuvent être effectuées à partir de la ligne de commande :

- lister, rechercher, installer, mettre à jour et supprimer des packages ;
- ajouter et configurer des sources de package ;
- activer, renommer et supprimer des EI.

Démarrez le Gestionnaire de packages de l'une des manières suivantes :

Barre d'outils	Cliquez sur l'icône du Gestionnaire de packages dans la barre d'outils. L'icône du Gestionnaire de packages représente une boîte entourée d'une flèche circulaire.
Icône du bureau	Cliquez deux fois sur l'icône du Gestionnaire de packages sur le bureau.
Barre de menu	Choisissez Système > Administration > Gestionnaire de packages.
Ligne de commande	<code>\$ /usr/lib/pm-launch packagemanager</code>

Pour consulter la documentation complète du Gestionnaire de packages, choisissez Aide>Contenu dans la barre de menu du Gestionnaire de packages.

Options de ligne de commande du Gestionnaire de packages

Les options suivantes sont prises en charge pour la commande `packagemanager(1)`.

TABLEAU 2-1 Options de commande du Gestionnaire de packages

Option	Description
<code>--image-dir</code> ou <code>-R dir</code>	S'exécute sur l'image résidant au niveau de <i>dir</i> . Par défaut, la commande s'exécute sur l'image actuelle. La commande suivante s'exécute sur l'image stockée au niveau de <code>/aux0/example_root</code> : <code>\$ /usr/lib/pm-launch packagemanager -R /aux0/example_root</code>
<code>--update-all</code> ou <code>-U</code>	Met à jour tous les packages installés pour lesquels une mise à jour est disponible. Spécifier cette option équivaut à choisir l'option Mises à jour dans l'interface graphique du Gestionnaire de packages. Pour plus d'informations sur la mise à jour de tous les packages, reportez-vous à la section "Utilisation du Gestionnaire de mises à jour" à la page 18.
<code>--info-install</code> ou <code>-i file.p5i</code>	Spécifie un fichier <code>.p5i</code> pour exécuter le Gestionnaire de packages en mode Installation Web. Le fichier spécifié doit avoir l'extension <code>.p5i</code> . Pour de plus amples informations, reportez-vous à la section "Utilisation de l'Installation Web" à la page 16.
<code>--help</code> ou <code>-h</code>	Affiche les informations d'utilisation de la commande.

Utilisation de l'Installation Web

Reportez-vous à l'aide du Gestionnaire de packages pour obtenir des informations détaillées sur le processus d'Installation Web.

Le Gestionnaire de packages prend en charge l'installation de packages à l'aide d'un processus d'Installation Web en un seul clic. Le processus d'Installation Web utilise un fichier `.p5i`. Un fichier `.p5i` contient des informations pour ajouter des éditeurs et des packages pouvant être installés à partir de ces éditeurs. Les informations contenues dans le fichier `.p5i` sont lues et utilisées par le processus d'Installation Web.

Exportation de fichiers à l'aide de l'Installation Web

Si vous souhaitez que d'autres utilisateurs puissent installer des packages que vous avez installés sur votre système, vous pouvez exporter leurs instructions d'installation à l'aide du processus d'Installation Web. Le processus d'Installation Web crée un fichier `.p5i` qui se compose des instructions d'installation des packages et éditeurs à installer.

Pour exporter vers un fichier `.p5i` les instructions d'installation correspondant aux packages et éditeurs que vous avez sélectionnés, procédez comme suit :

1. À partir du menu déroulant Éditeur du Gestionnaire de packages, sélectionnez l'éditeur dont vous souhaitez inclure les packages dans le fichier `.p5i`.
2. Dans le volet Liste des packages du Gestionnaire de packages sélectionnez le package dont vous souhaitez distribuer les instructions d'installation.
3. Choisissez Fichier>Sélections d'exportation pour afficher la fenêtre Confirmation des sélections d'exportation.
4. Cliquez sur le bouton OK pour confirmer les sélections. La fenêtre Sélections d'exportation s'affiche.
5. Un nom par défaut est attribué au fichier `.p5i`. Vous pouvez modifier ce nom de fichier, mais ne modifiez pas l'extension `.p5i`.
6. Un emplacement par défaut est attribué au fichier `.p5i`. Vous pouvez modifier l'emplacement.
7. Cliquez sur le bouton Enregistrer pour enregistrer le nom de fichier et l'emplacement.

Utilisation de l'Installation Web pour ajouter des éditeurs et installer des packages

Le processus d'Installation Web vous permet d'installer des packages à partir d'un fichier `.p5i`. Ce fichier peut se trouver sur votre bureau ou sur un site Web.

1. Utilisez l'une des méthodes suivantes pour démarrer le Gestionnaire de packages en mode Installation Web :
 - Cliquez sur un fichier `.p5i` sur votre bureau.
 - Démarrez le Gestionnaire de packages à partir de la ligne de commande et spécifiez un fichier `.p5i` :


```
$ /usr/lib/pm-launch packagemanager ./wifile.p5i
```
 - Allez à un emplacement d'URL contenant un lien vers un fichier `.p5i`.
Si le fichier `.p5i` réside sur un serveur Web où le type MIME est enregistré, il suffit de cliquer sur le lien vers le fichier `.p5i`.
Si le fichier `.p5i` réside sur un serveur Web où ce type MIME n'est pas enregistré, enregistrez le fichier `.p5i` sur votre bureau, puis cliquez sur ce dernier.
2. La fenêtre Installer/Mettre à jour s'affiche. L'étiquette en haut de la fenêtre est : « Package Manager Web Installer/The following will be added to your system. » (Programme d'installation Web du gestionnaire de packages/Les éléments suivants seront ajoutés à votre système.) Les éditeurs et les packages à installer apparaissent dans une liste. Cliquez sur le bouton Continuer pour poursuivre l'installation.
3. Si l'éditeur de package spécifié n'est pas encore configuré sur votre système, la fenêtre Ajouter un éditeur s'affiche. Le nom et l'URI de l'éditeur sont déjà spécifiés.

Lorsque les éditeurs à ajouter sont sécurisés, une clé et un certificat SSL sont nécessaires. Spécifiez leur emplacement sur le système.

La boîte de dialogue Ajout de l'éditeur terminé s'affiche si l'opération a réussi. Cliquez sur le bouton OK pour poursuivre l'installation.

4. Si un fichier .p5i contient des packages provenant d'un éditeur désactivé, Installation Web ouvre une boîte de dialogue Activer un éditeur. Utilisez cette boîte de dialogue pour activer l'éditeur de sorte à pouvoir installer les packages.

La fenêtre Installer/Mettre à jour est maintenant la même que lorsque vous sélectionnez l'option Installer/Mettre à jour dans le Gestionnaire de packages.

L'application se ferme lorsque tous les packages sont installés.

Utilisation du Gestionnaire de mises à jour

Le Gestionnaire de mises à jour met à jour tous les packages installés vers la version la plus récente autorisée par les contraintes imposées sur le système par les packages installés et la configuration d'éditeur. Cette fonction est la même que les fonctions suivantes :

- Dans l'interface graphique du Gestionnaire de packages, choisissez le bouton Mises à jour ou l'option de menu Package>Mises à jour.
- Utilisez la CLI du Gestionnaire de packages.

```
$ /usr/lib/pm-launch packagemanager --update-all
```

- Utilisez la commande pkg.

```
# pkg update
```

Démarrez le Gestionnaire de mises à jour de l'une des manières suivantes :

Barre d'état Lorsque des mises à jour sont disponibles, une notification doit apparaître dans la barre d'état. Cliquez sur l'endroit indiqué dans la notification. L'icône du Gestionnaire de mises à jour représente une pile de trois boîtes.

Barre de menu Choisissez Système>Administration>Gestionnaire de mises à jour.

Ligne de commande \$ /usr/lib/pm-launch pm-updatemanager

La fenêtre Mises à jour s'affiche et le processus de mise à jour démarre :

1. Le système actualise tous les catalogues.
2. Le système évalue tous les packages installés afin de déterminer ceux pour lesquels des mises à jour disponibles.
 - Si aucun package n'a de mise à jour disponible, le message « Aucune mise à jour disponible » s'affiche et le processus s'arrête.

- Si des mises à jour sont disponibles, les packages à mettre à jour sont répertoriés pour être vérifiés. Il s'agit de votre dernière possibilité de cliquer sur le bouton Annuler pour annuler la mise à jour.
3. Cliquez sur le bouton Continuer pour poursuivre la mise à jour. Le système télécharge et installe toutes les mises à jour de package.

Les packages suivants sont mis à jour en premier si des mises à jour correspondantes sont disponibles. Les autres packages sont ensuite mis à jour.

```
package/pkg
package/pkg/package manager
package/pkg/updates manager
```

Par défaut, chaque package est mis à jour depuis l'éditeur à partir duquel il a été initialement installé. Si l'éditeur initial est non persistant, une version plus récente du package compatible avec cette image peut être installée à partir d'un autre éditeur. Utilisez la fenêtre Gérer les éditeurs du Gestionnaire de packages ou la commande `pkg set-publisher` pour définir un éditeur comme persistant ou non persistant.

Un nouvel environnement d'initialisation peut être créé si certains packages sont mis à jour. Voir [“Images et environnements d'initialisation”](#) à la page 12.

Si une erreur se produit au cours du processus de mise à jour, le panneau Détails contenant les informations sur l'erreur se déploie. Un indicateur du statut d'erreur s'affiche en regard de l'étape ayant échoué.

4. Si le système a créé un EI, vous pouvez modifier le nom de l'EI par défaut. Lorsque vous êtes satisfait du nom de l'EI, cliquez sur le bouton Redémarrer maintenant pour redémarrer votre système immédiatement. Cliquez sur le bouton Redémarrer plus tard pour redémarrer votre système à une date ultérieure. Vous devez redémarrer pour initialiser dans le nouvel EI. Le nouvel EI sera votre choix d'initialisation par défaut. Votre EI actuel sera disponible en tant que choix d'initialisation alternatif.

Options de ligne de commande du Gestionnaire de mises à jour

Les options suivantes sont prises en charge pour la commande `pm-updates manager(1)`.

TABLEAU 2-2 Options de commande du Gestionnaire de mises à jour

Option	Description
<code>--image-dir</code> ou <code>-R dir</code>	<p>S'exécute sur l'image résidant au niveau de <i>dir</i>. Par défaut, la commande s'exécute sur l'image actuelle.</p> <p>La commande suivante met à jour l'image au niveau de <code>/aux0/example_root</code> :</p> <pre>\$ /usr/lib/pm-launch pm-updatemanager -R /aux0/example_root</pre>
<code>--help</code> ou <code>-h</code>	Affiche les informations d'utilisation de la commande.

Utilisation de packages

Ce chapitre vous explique comment installer et gérer des packages sur votre système et comment gérer les éditeurs de package.

Voir aussi [Annexe A](#), “Aide-mémoire de la commande”.

Tâches de gestion des packages

Le tableau suivant présente une liste des tâches de gestion des packages et de références aux instructions de réalisation de ces tâches. Utilisez la commande `pkg help` pour afficher toutes les sous-commandes et options de la commande `pkg`.

TABLEAU 3-1 Liste des tâches de gestion des packages

Tâche de gestion des packages	Instructions
Ajouter ou mettre à jour le logiciel (packages).	“Installation ou mise à jour d'un package” à la page 23 et “Mise à jour d'un package” à la page 24
Exécuter une simulation d'installation des packages pour afficher les changements système consécutifs à cette installation, sans y procéder réellement.	“Visualisation d'une action d'installation sans installer” à la page 24
Vérifier si le package a été correctement installé et corriger les erreurs d'installation signalées.	“Vérification de l'installation d'un package” à la page 25 et “Correction des erreurs d'installation de package” à la page 25
Supprimer des packages.	“Désinstallation des packages” à la page 26 et “Désinstallation d'un package à partir d'un environnement d'initialisation inactif” à la page 26
Rechercher un package installé sur le système ou effectuer une recherche dans un référentiel distant.	“Recherche de packages” à la page 26

TABLEAU 3-1 Liste des tâches de gestion des packages (Suite)

Tâche de gestion des packages	Instructions
Afficher des informations sur des packages.	“Affichage du contenu d'un package” à la page 28, “Affichage des informations sur les packages” à la page 28 et “Présentation sous forme de liste des informations d'état du package” à la page 29
Afficher et modifier une variante ou une facette.	“Affichage d'une variante” à la page 31, “Modification d'une variante” à la page 31, “Affichage d'une facette” à la page 32 et “Modification d'une facette” à la page 32
Afficher l'historique des opérations.	“Affichage et suppression de l'historique des opérations” à la page 33
Afficher des informations sur les éditeurs.	“Affichage des informations concernant l'éditeur” à la page 33
Gérer les éditeurs et les référentiels.	“Ajout, modification ou suppression d'un éditeur de package” à la page 34
Configurer une image pour installer des packages signés.	“Installation de packages signés” à la page 37

Si vous recevez des messages d'erreur concernant des difficultés pour contacter un référentiel ou un éditeur, vérifiez les éléments suivants.

- Assurez-vous d'être connecté à Internet.
- Vérifiez que l'URI d'origine de l'éditeur est correct. Vérifiez que vous pouvez voir les packages dans le référentiel lorsque vous utilisez l'URI à l'aide d'un navigateur Web.
- Si votre système est connecté à un serveur proxy, assurez-vous que ce dernier fonctionne correctement.
- Assurez-vous que la résolution de noms est correctement configurée.
- Avez-vous besoin d'un certificat et d'une clé SSL pour accéder au référentiel ?

Si vos opérations d'empaquetage sont lentes, votre disque est peut-être plein. Il peut s'avérer nécessaire de supprimer des environnements d'initialisation inutilisés pour récupérer de l'espace. Utilisez la commande `beadm list` pour répertorier les environnements d'initialisation existants. Reportez-vous à la page de manuel `beadm(1M)`, au guide *Gestion des environnements d'initialisation avec Oracle Solaris 11 Express* ou à l'aide en ligne du Gestionnaire de packages pour obtenir des instructions sur la suppression d'environnements d'initialisation.

Installation et gestion des packages

IPS vous permet de télécharger et d'installer des packages de logiciels à partir d'un référentiel de packages IPS.

▼ Installation ou mise à jour d'un package

Par défaut, la version la plus récente d'un package compatible avec le reste de l'image est installée à partir de l'éditeur préféré. Si le package est déjà installé, il est mis à jour en installant la version la plus récente du package compatible avec le reste de l'image à partir de l'éditeur qui a fourni la version actuellement installée. Si vous avez plusieurs éditeurs configurés, vous pouvez contrôler quel éditeur fournit un package en définissant la persistance et l'ordre de recherche des éditeurs ou en spécifiant l'éditeur dans l'identificateur de ressource de gestion des pannes de package. Vous pouvez également spécifier la version que vous souhaitez installer dans l'identificateur de ressource de gestion des pannes de package. Reportez-vous à [“Identificateurs de ressource de gestion des pannes”](#) à la page 10 pour obtenir la description d'un identificateur de ressource de gestion des pannes de package. Un environnement d'initialisation peut être créé lorsque vous installez, mettez à jour ou désinstallez un package. Reportez-vous à [“Images et environnements d'initialisation”](#) à la page 12 pour plus d'informations sur la création d'environnement d'initialisation et une description des options `--be-name`, `--require-new-be` et `--deny-new-be`.

- Utilisez la commande `pkg install` pour installer ou mettre à jour un package.

Vous pouvez spécifier plusieurs modèles `pkg-fmri`.

```
# pkg install pkg-fmri
```

Exemple 3-1 Installation d'un package

Dans cet exemple, le package `installadm` est installé. La sortie affiche l'état du téléchargement, le nombre de packages installés, le nombre de fichiers installés, ainsi que la taille du téléchargement en mégaoctets.

```
# pkg install install/installadm
DOWNLOAD   PKGS      FILES    XFER (MB)
Completed   9/9      1067/1067  6.1/6.1

PHASE                ACTIONS
Install Phase        1458/1458

PHASE                ITEMS
Package State Update Phase  9/9
Image State Update Phase    2/2
```

Exemple 3-2 Installation d'un package à partir d'un éditeur spécifique

Pour installer un package d'un éditeur spécifique, indiquez le nom de l'éditeur dans la commande *pkg-fmri*.

```
# pkg install pkg://example.com/developer/sunstudio12u1
```

Exemple 3-3 Installation d'une version spécifique d'un package

Pour installer une version spécifique d'un package, spécifiez les informations de version dans la commande *pkg-fmri*.

```
# pkg install pkg:/developer/sunstudio12u1@12.1.1,5.11-0.111:20100306T002245Z
```

▼ Mise à jour d'un package

- Vous pouvez utiliser les sous-commandes `install` ou `update` pour mettre à jour un package.

La sous-commande `install` installe le package s'il n'est pas déjà installé sur l'image. Si vous voulez être sûr d'actualiser uniquement les packages déjà installés, et ne pas installer de nouveaux packages, utilisez la sous-commande `update`.

Vous pouvez spécifier plusieurs modèles *pkg-fmri*.

```
# pkg update pkg-fmri
```



Attention – Si vous utilisez la commande `pkg update` sans *pkg-fmri* spécifié, tous les packages installés qui ont des mises à jour disponibles sont mis à jour. Voir [“Mise à jour d'une image” à la page 44](#).

Si une version plus récente d'un package installé est disponible et est compatible avec le reste de l'image, le package est mis à jour vers cette version. Reportez-vous à [“Installation ou mise à jour d'un package” à la page 23](#) pour une description plus complète des packages et des éditeurs les installant.

▼ Visualisation d'une action d'installation sans installer

Vous pouvez vérifier exactement ce qui sera installé avant de procéder à toute installation. Grâce à l'option `-n`, vous pouvez exécuter la commande `install` sans apporter de modification persistante. L'option `-n` affiche les modifications qui seraient apportées si vous exécutiez la commande sans l'option `-n`. Vous pouvez également utiliser l'option `-n` avec les sous-commandes `update` et `uninstall`.

- Utilisez l'option `-n` avec la commande `pkg install` pour voir ce qui serait installé sans procéder à une réelle installation.

L'option `-v` donne des informations détaillées.

```
# pkg install -nv pkg-fmri
```

Exemple 3-4 Visualisation d'une action d'installation sans installer

La commande suivante donne des informations sur ce qui serait installé, mais sans procéder à une réelle installation.

```
# pkg install -n pkg:/developer/sunstudio12u1
```

▼ Vérification de l'installation d'un package

- Utilisez la commande `pkg verify` pour valider l'installation d'un package.

```
$ pkg verify pkg-fmri
```

Vous pouvez spécifier plusieurs modèles `pkg-fmri`. Utilisez l'option `-v` pour afficher des messages d'information. Utilisez l'option `-q` pour afficher uniquement des messages d'erreur.

Exemple 3-5 Vérification de l'installation d'un package

```
$ pkg verify -v e1000g
Verifying: PACKAGE          STATUS
pkg://solaris/driver/network/e1000g  OK
```

▼ Correction des erreurs d'installation de package

- Utilisez la commande `pkg fix` pour corriger les erreurs d'installation de package signalées par la commande `pkg verify`.

```
$ pkg fix --accept pkg-fmri
```

Si vous ne spécifiez pas l'option `--accept` et qu'un ou plusieurs packages requièrent l'acceptation d'une licence, l'opération de correction échoue. Utilisez l'option `--licenses` pour afficher toutes les licences pour les packages installés ou mis à jour par cette opération de correction.

▼ Désinstallation des packages

- Utilisez la commande `pkg uninstal` pour désinstaller les packages existants.

Vous pouvez spécifier plusieurs modèles *pkg-fMRI*. Vous devez spécifier au moins un modèle *pkg-fMRI*.

```
# pkg uninstall pkg-fMRI
```

Utilisez l'option `-r` pour chercher de façon récursive les packages qui contiennent des dépendances requises sur le package *pkg-fMRI*. Reportez-vous à “[Images et environnements d'initialisation](#)” à la page 12 pour plus d'informations sur la création d'environnement d'initialisation et une description des options `--be-name`, `--require-new-be` et `--deny-new-be`.

▼ Désinstallation d'un package à partir d'un environnement d'initialisation inactif

- 1 Montez l'environnement d'initialisation inactif à partir duquel le package doit être désinstallé.

```
# beadm mount inactive-be mntpt
```

- 2 Désinstallez le package.

```
# pkg -R mntpt uninstall pkg-fMRI
```

- 3 Démontez l'environnement d'initialisation précédemment monté.

```
# beadm unmount inactive-be
```

▼ Recherche de packages

- Utilisez la commande `pkg search` pour rechercher des packages dont les données correspondent au modèle spécifié.

Vous pouvez spécifier plus d'un *modèle*. Plusieurs modèles requièrent l'opérateur AND.

```
$ pkg search pattern
```

Par défaut, la recherche est effectuée sur les référentiels associés à tous les éditeurs configurés pour cette image. Utilisez l'option `-l` pour rechercher uniquement les packages installés dans cette image. Par défaut, les correspondances s'affichent uniquement pour les versions installées ou les plus récentes des packages. Utilisez l'option `-f` pour afficher toutes les versions correspondantes.

Exemple 3-6 Recherche d'un package dans l'image installée

L'exemple suivant illustre la recherche du package `bash` dans l'image installée. La colonne INDEX indique où la correspondance a été trouvée dans les données.

```
$ pkg search -l bash
INDEX      ACTION VALUE                                PACKAGE
pkg.fmri   set      solaris/shell/bash                        pkg:/shell/bash@4.0.28-0.149
basename   file     usr/bin/bash                              pkg:/shell/bash@4.0.28-0.149
basename   dir      etc/bash                                  pkg:/shell/bash@4.0.28-0.149
basename   dir      usr/share/bash                            pkg:/shell/bash@4.0.28-0.149
```

Exemple 3-7 Recherche d'un package dans un référentiel spécifié

Dans cet exemple, deux référentiels qui ne sont pas configurés pour cette image sont explorés.

```
$ pkg search -s http://pkg.example1.com/release \
-s http://pkg.example2.com/release ksh
```

Exemple 3-8 Recherche d'un package offrant un fichier spécifique

Cet exemple montre que la bibliothèque `libdhcpagent` provient du package `system/library`.

```
$ pkg search -l /lib/libdhcpagent.so.1
INDEX      ACTION VALUE                                PACKAGE
path       file     lib/libdhcpagent.so.1                    pkg:/system/library@0.5.11-0.149
```

Exemple 3-9 Recherche à l'aide de caractères génériques et de directives booléennes

Plusieurs modèles requièrent l'opérateur AND par défaut. Vous pouvez également spécifier OR, * et ?.

```
$ pkg search netbeans AND plug*in OR ide
```

Exemple 3-10 Recherche avec champs

La chaîne de recherche peut être spécifiée comme l'ensemble de champs suivant :

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

Les champs manquants sont implicitement considérés comme des caractères génériques. Des caractères génériques explicites peuvent être utilisés dans les champs `pkg_name` et `token`. Les champs `action_type` et `key` doivent correspondre exactement. Reportez-vous à la section Actions de la page de manuel `pkg(5)` pour voir une liste de tous les types d'action. Parmi les exemples de clés se trouvent `basename`, `description` et `driver_name`.

```
$ pkg search -l 'depend::package/pkg'
INDEX      ACTION VALUE                                PACKAGE
incorporate depend package/pkg@0.5.11-0.150                pkg:/consolidation/ips/ips-incorporation@0.5.11-0.150
```

```
require depend package/pkg pkg:/package/pkg/package-manager@0.5.11-0.150
require depend package/pkg pkg:/system/zones/brand/ipkg@0.5.11-0.150
```

▼ Affichage du contenu d'un package

- Utilisez la commande `pkg contents` pour afficher les attributs d'action d'un package.

Par défaut, seul l'attribut de chemin est indiqué. Utilisez l'option `-o` pour spécifier d'autres valeurs d'attributs à afficher. Vous pouvez spécifier plusieurs modèles *pkg-fmri*. Si vous ne spécifiez aucun modèle *pkg-fmri*, les données sont affichées pour tous les packages installés. Si le package n'est pas installé, vous devez utiliser l'option `-r` pour récupérer des données de tous les éditeurs configurés pour cette image. Lorsque vous utilisez l'option `-r`, vous devez spécifier un ou plusieurs modèles *pkg-fmri*.

```
$ pkg contents pkg-fmri
```

Exemple 3-11 Présentation sous forme de liste du contenu d'un package

Dans cet exemple, la taille et le chemin de chaque fichier du package `e1000g` sont répertoriés :

```
$ pkg contents -t file -o pkg.size,path network/e1000g
PKG.SIZE PATH
471656 kernel/drv/amd64/e1000g
323612 kernel/drv/e1000g
4238 kernel/drv/e1000g.conf
```

▼ Affichage des informations sur les packages

- Utilisez la commande `pkg info` pour afficher des informations sur un package.

Vous pouvez spécifier plusieurs modèles *pkg-fmri*. Si vous ne spécifiez aucun modèle *pkg-fmri*, les informations relatives à tous les packages installés s'affichent. Si le package n'est pas installé, vous devez utiliser l'option `-r` pour récupérer des données de tous les éditeurs configurés pour cette image. Lorsque vous utilisez l'option `-r`, vous devez spécifier un ou plusieurs modèles *pkg-fmri*.

```
$ pkg info pkg-fmri
```

Exemple 3-12 Affichage des informations sur un package spécifique

Cet exemple affiche des informations sur le package `openoffice`. Dans cet exemple, l'option `-r` est utilisée pour afficher les informations, même si le package n'est pas installé.

```
$ pkg info -r openoffice
Name: openoffice
Summary: OpenOffice.org 3.1.0
Category: Applications/Office
```

```

State: Not installed
Publisher: solaris
Version: 3.1.0
Build Release: 5.11
Branch: 0.111
Packaging Date: May 18, 2009 06:27:12 AM
Size: 430.65 MB
FMRI: pkg://solaris/openoffice@3.1.0,5.11-0.111:20090518T062712Z

```

Exemple 3-13 Affichage des informations de copyright et de licence relatives à un package

Cet exemple affiche les informations de copyright et de licence relatives au package `firefox`.

```

$ pkg info --license firefox
Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.

Copyright (c) 1998-2010 by Contributors. All rights reserved.
Firefox and the Firefox logos are trademarks of the Mozilla Foundation. All rights reserved.

```

▼ Présentation sous forme de liste des informations d'état du package

Les deux sous-commandes `info` et `list` affichent le nom et l'éditeur du package et certaines informations de version. En plus du nom et de l'éditeur du package, la commande `pkg info` affiche le résumé, la catégorie et la taille du package. La commande `pkg list` indique si une mise à jour existe pour le package, si une mise à jour peut être installée dans cette image et si le package est obsolète ou renommé. La commande `pkg info` affiche plusieurs lignes d'informations sur chaque package. La commande `pkg list` affiche une ligne d'informations sur chaque package.

- Utilisez la commande `pkg list` pour afficher les informations d'état sur un package, par exemple si une mise à jour est disponible.

```
$ pkg list pkg-fmri
```

Vous pouvez spécifier plusieurs modèles `pkg-fmri`. Si vous ne spécifiez aucun modèle `pkg-fmri`, tous les packages installés sont répertoriés. La commande `pkg list -u` permet d'afficher la liste de tous les packages installés qui ont de nouvelles versions disponibles.

La commande `pkg list` affiche les informations suivantes :

NAME (PUBLISHER)	Nom du package. S'il n'est pas le préféré, l'éditeur figure entre parenthèses dans la liste après le nom du package.
VERSION	Version et versions de branche du package. Si vous spécifiez l'option <code>-v</code> , la colonne VERSION n'est pas affichée. Au lieu de cela, l'identificateur de ressource de gestion des pannes du package est indiqué dans la colonne NAME.
STATE	Etat du package. L'état est <code>installed</code> (installé) ou <code>known</code> (connu).

UFOXI	Indicateurs qui donnent des informations sur la manière dont le package se rapporte à d'autres packages dans l'image.
U	Un u dans la colonne U indique qu'une version plus récente de ce package est disponible. La nouvelle version sera peut-être impossible à installer à cause des dépendances du package ou d'autres contraintes. Reportez-vous à l'option -a dans l' Exemple 3-16 .
O	Un o dans la colonne O indique que ce package est obsolète. Un r dans la colonne O indique que ce package a été renommé.

Exemple 3-14 Répertorier un package installé

Dans cet exemple, le package est installé à partir de l'éditeur préféré et dispose d'une mise à jour disponible.

```
$ pkg list firefox
NAME (PUBLISHER)    VERSION    STATE    UFOXI
web/browser/firefox 0.5.11-0.150 installed u----
```

Exemple 3-15 Liste des versions les plus récentes d'un package

Utilisez l'option -n pour répertorier les versions les plus récentes d'un package depuis l'ensemble des éditeurs configurés.

```
$ pkg list -n firefox
NAME (PUBLISHER)    VERSION    STATE    UFOXI
web/browser/firefox (example.com) 0.5.11-0.151 known    -----
web/browser/firefox 0.5.11-0.150 installed u-----
```

Exemple 3-16 Répertorier un package non installé

Utilisez l'option -a pour répertorier les packages installés et la version la plus récente disponible pour installation. Les packages sont disponibles pour installation s'ils sont autorisés par les incorporations installées et par les variantes de l'image. Dans cet exemple, le package n'est pas encore installé mais il est disponible pour installation à partir de l'éditeur préféré. Le package sunstudio12u1 a été renommé developer/sunstudio12u1.

```
$ pkg list -a sunstudio12u1
NAME (PUBLISHER)    VERSION    STATE    UFOXI
developer/sunstudio12u1 12.1.1-0.111 known    -----
sunstudio12u1      12.1.1-1   known    --r--
```

Exemple 3–17 Liste des noms et résumés de packages

Utilisez l'option `-s` pour afficher uniquement le nom et le résumé du package.

```
$ pkg list -s developer/sunstudio12u1
NAME (PUBLISHER)          SUMMARY
developer/sunstudio12u1  Sun Studio - C, C++, & Fortran compilers and Tools
```

▼ Affichage d'une variante

Une variante est un composant incompatible d'un package tel qu'une architecture. Les variantes apparaissent comme des repères sur des actions IPS et déterminent si cette action est installable. Si une action comporte des repères de variante, tous les repères de variante doivent correspondre aux critères de sélection pour installer l'action.

- Utilisez la commande `pkg variant` pour afficher les valeurs des variantes d'une image.

```
$ pkg variant variant_spec
```

Vous pouvez spécifier plusieurs *variant_spec*.

Exemple 3–18 Affichage des valeurs de toutes les variantes

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

▼ Modification d'une variante

- Utilisez la commande `pkg change-variant` pour modifier la valeur d'une variante.

```
# pkg change-variant -n --accept variant_spec=instance
```

Utilisez l'option `-n` pour voir ce qui serait modifié si vous aviez effectué l'opération sans `-n`, mais sans effectuer de modification réelle. Un environnement d'initialisation peut être créé. Voir [“Images et environnements d'initialisation”](#) à la page 12.

Exemple 3–19 Modification d'une variante

```
# pkg change-variant -n --accept variant.debug=false
```

▼ Affichage d'une facette

Une facette constitue un composant optionnel d'un package tel qu'un environnement linguistique. Les facettes apparaissent comme des repères sur des actions IPS et déterminent si cette action est installable. Si une action comporte des repères de facette, au moins un repère de facette doit correspondre aux critères de sélection pour installer l'action.

- Utilisez la commande `pkg facet` pour afficher les valeurs actuelles de toutes les facettes définies dans l'image active.

```
$ pkg facet facet_spec
```

Vous pouvez spécifier plusieurs `facet_spec`.

Exemple 3–20 Affichage de toutes les facettes de l'image active

```
$ pkg facet
FACETS      VALUE
facet.devel FALSE
```

▼ Modification d'une facette

- Utilisez la commande `pkg change-facet` pour modifier les valeurs actuelles d'une facette.

```
# pkg change-facet -n --accept facet_spec=True|False|None
```

Utilisez l'option `-n` pour voir ce qui serait modifié si vous aviez effectué l'opération sans `-n`, mais sans effectuer de modification réelle. Un environnement d'initialisation peut être créé. Voir [“Images et environnements d'initialisation” à la page 12](#).

Si la valeur de la facette est `None`, sa spécification est supprimée de l'image active.

Exemple 3–21 Modification d'une facette dans l'image actuelle

```
# pkg change-facet facet.devel=True
$ pkg facet facet.devel
FACETS      VALUE
facet.devel TRUE
```


▼ Affichage et suppression de l'historique des opérations

- 1 Utilisez la commande `pkg history` pour afficher l'historique des commandes dans l'image active.

Utilisez l'option `-l` pour afficher plus d'informations, notamment le résultat de la commande, le temps d'exécution de la commande, la version et le nom du client utilisé, le nom de l'utilisateur qui a effectué l'opération et les erreurs survenues lors de l'exécution de la commande. Utilisez l'option `-n` pour afficher uniquement le nombre spécifié d'opérations les plus récentes.

```
$ pkg history
TIME                OPERATION                CLIENT  OUTCOME
2010-10-16T16:32:46 update-publisher        pkg     Succeeded
2010-10-16T16:34:42 refresh-publishers      pkg     Succeeded
2010-10-16T16:36:04 rebuild-image-catalogs  pkg     Succeeded
2010-10-16T16:38:17 install                    pkg     Succeeded
```

- 2 Utilisez la commande `pkg purge-history` pour supprimer toutes les informations d'historique des commandes.

```
# pkg purge-history
```

Gestion des éditeurs de package

IPS permet de configurer les éditeurs et les référentiels de packages.

▼ Affichage des informations concernant l'éditeur

- Utilisez la commande `pkg publisher` pour afficher des informations sur les éditeurs de package configurés pour cette image.

Exemple 3–22 Affichage de tous les éditeurs

Si aucune option ou aucun argument n'est spécifié, tous les éditeurs configurés pour cette image sont affichés.

```
$ pkg publisher
PUBLISHER                TYPE  STATUS  URI
solaris                   (preferred)  origin  online  http://pkg.oracle.com/solaris/release
example.com                (non-sticky)  origin  online  http://pkg.example.com/release
```

Exemple 3–23 Affichage de l'éditeur préféré uniquement

```
$ pkg publisher -P
PUBLISHER          TYPE    STATUS  URI
solaris            (preferred)  origin  online  http://pkg.oracle.com/solaris/release
```

Exemple 3–24 Affichage des informations relatives à un éditeur spécifique

Dans cet exemple, les informations sur l'éditeur solaris sont affichées.

```
$ pkg publisher solaris
Publisher: solaris
Alias:
Origin URI: http://pkg.oracle.com/solaris/release
SSL Key: None
SSL Cert: None
Client UUID: 99e58a54-1119-11df-98e6-00262236a2ae
Catalog Updated: October 17, 2010 08:11:06 PM
Enabled: Yes
```

▼ Ajout, modification ou suppression d'un éditeur de package

- Utilisez la commande `pkg set-publisher` pour effectuer les opérations suivantes :
 - Ajout d'un éditeur. Pour supprimer un éditeur, utilisez la commande `pkg unset-publisher`. L'éditeur préféré ne peut pas être supprimé.
 - Spécification de l'éditeur préféré.
 - Activation ou désactivation d'un éditeur. L'éditeur préféré ne peut pas être désactivé. Un éditeur récemment ajouté est activé par défaut.
 - Définition de la persistance d'un éditeur. Un éditeur récemment ajouté est persistant par défaut. Si un éditeur n'est pas persistant, un package installé à partir de cet éditeur peut être mis à jour à partir d'un autre éditeur.
 - Définition de l'ordre de recherche des éditeurs. L'éditeur préféré est affiché en premier dans l'ordre de recherche. Un éditeur récemment ajouté est affiché par défaut en dernier dans l'ordre de recherche.
 - Ajout ou suppression d'un miroir d'éditeur.
 - Spécification des certificats et clés SSL pour un éditeur.
 - Modification de l'URI d'un éditeur.
 - Définition et rétablissement de la propriété d'un éditeur, et ajout et suppression de la valeur de propriété d'un éditeur. Voir [“Installation de packages signés”](#) à la page 37.

Exemple 3–25 Ajout d'un éditeur

Utilisez l'option `-g` pour spécifier l'URI d'origine de l'éditeur.

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

Exemple 3–26 Spécification de l'éditeur préféré

Utilisez l'option `-P` pour spécifier un éditeur comme préféré. L'éditeur spécifié est placé au début de l'ordre de recherche. Vous pouvez spécifier l'option `-P` lorsque vous ajoutez un éditeur ou vous pouvez modifier un éditeur existant.

```
# pkg set-publisher -P example.com
```

Exemple 3–27 Activation ou désactivation d'un éditeur

Utilisez l'option `-d` pour désactiver un éditeur. L'éditeur préféré ne peut pas être désactivé. Un éditeur désactivé n'est pas utilisé dans les opérations de package telles que `list` ou `install`. Vous pouvez modifier les propriétés des éditeurs désactivés.

Utilisez l'option `-e` pour activer un éditeur.

```
# pkg set-publisher -d example2.com
```

Exemple 3–28 Spécification de la persistance d'un éditeur

Utilisez l'option `--non-sticky` pour rendre un éditeur non persistant. Un éditeur récemment ajouté est persistant par défaut. Si un éditeur n'est pas persistant, un package installé à partir de cet éditeur peut être mis à jour à partir d'un autre éditeur.

Utilisez l'option `--sticky` pour rendre un éditeur persistant.

```
# pkg set-publisher --non-sticky example.com
```

Exemple 3–29 Modification de l'ordre de recherche des éditeurs

L'éditeur préféré est affiché en premier dans l'ordre de recherche. Un éditeur récemment ajouté est affiché par défaut en dernier dans l'ordre de recherche. L'ordre de recherche des éditeurs sert à trouver des packages à installer. L'ordre de recherche des éditeurs est utilisé pour trouver des packages à mettre à jour si l'éditeur à partir duquel le package a été installé initialement est non persistant.

Dans l'exemple suivant, l'éditeur `example2.com` est défini pour être exploré avant l'éditeur `example1.com`. Si `example1.com` est l'éditeur préféré actuel, `example2.com` devient l'éditeur préféré.

```
# pkg set-publisher --search-before example1.com example2.com
```

Dans l'exemple suivant, l'éditeur `example2.com` est défini pour être exploré après l'éditeur `example1.com`. Si `example2.com` est l'éditeur préféré actuel, `example1.com` devient l'éditeur préféré.

```
# pkg set-publisher --search-after example1.com example2.com
```

Exemple 3–30 Ajout et suppression d'un miroir d'éditeur

Utilisez l'option `-m` pour ajouter un URI comme miroir pour l'éditeur spécifié. Reportez-vous à [“Sources et miroirs du référentiel” à la page 11](#) pour obtenir une explication sur la différence entre une source et un miroir. Utilisez l'option `-M` pour supprimer un URI servant de miroir à l'éditeur spécifié.

```
# pkg set-publisher -m http://pkg.example.org/release example.com
$ pkg publisher
PUBLISHER                TYPE    STATUS  URI
example.com              (preferred) origin  online  http://pkg.example.com/release
example.com              (non-sticky) mirror  online  http://pkg.example.org/release
```

Exemple 3–31 Spécification des clés et certificats SSL

Utilisez l'option `-k` pour spécifier la clé SSL client. Utilisez l'option `-c` pour spécifier le certificat SSL client.

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

Utilisez l'option `--approve-ca-cert` pour ajouter le certificat spécifié comme certificat d'AC fiable. Les hachages des certificats d'AC approuvés par l'utilisateur sont répertoriés dans la sortie de la sous-commande `publisher` pour cet éditeur. Voir [Exemple 3–24](#).

```
# pkg set-publisher --approve-ca-cert /tmp/example_file.pem example.com
```

Utilisez l'option `--revoked-ca-cert` pour traiter le certificat spécifié comme révoqué. Les hachages des certificats d'AC révoqués par l'utilisateur sont répertoriés dans la sortie de la sous-commande `publisher` pour cet éditeur.

```
# pkg set-publisher --revoked-ca-cert a12345 example.com
```

Utilisez l'option `--unset-ca-cert` pour supprimer le certificat spécifié de la liste des certificats approuvés et de la liste des certificats révoqués.

```
# pkg set-publisher --unset-ca-cert a12345 example.com
```

Exemple 3-32 Modification de l'URI d'origine d'un éditeur

Pour modifier l'URI d'origine d'un éditeur, ajoutez le nouvel URI et supprimez l'ancien. Utilisez l'option `-g` pour ajouter un nouvel URI d'origine. Utilisez l'option `-G` pour supprimer l'ancien URI d'origine.

```
# pkg set-publisher -g http://pkg.example.com/support \
-G http://pkg.example.com/release example.com
```

Installation de packages signés

Si vous installez des packages signés, définissez les propriétés de l'image et de l'éditeur décrites dans cette section pour vérifier les signatures des packages.

Propriétés de l'image pour les packages signés

Configurez les propriétés de l'image suivantes pour utiliser des packages signés.

signature-policy

La valeur de cette propriété détermine quelles vérifications sont effectuées sur des fichiers manifest lors de l'installation d'un package dans cette image. La stratégie finale appliquée à un package dépend de la combinaison des stratégies de l'image et de l'éditeur. La combinaison sera au moins aussi stricte que la plus stricte des deux stratégies prises individuellement. Les valeurs suivantes sont valides pour cette propriété.

ignore	Ignore les signatures pour tous les fichiers manifest.
verify	Vérifie que tous les fichiers manifest avec signatures sont valablement signés, mais ne nécessite pas que tous les packages installés soient signés.
require-signatures	Demande à ce que tous les nouveaux packages installés disposent au moins d'une signature valide. Les commandes <code>pkg fix</code> et <code>pkg verify</code> avertissent également lorsqu'un package installé ne possède pas de signature valide.
require-names	Suit les mêmes exigences que <code>require-signatures</code> mais nécessite aussi que les chaînes répertoriées dans la propriété d'image <code>signature-required-names</code> s'affichent en tant que nom commun des certificats utilisés pour vérifier les chaînes de confiance des signatures.

signature-required-names

La valeur de cette propriété est une liste de noms qui doivent être considérés comme des noms communs de certificats lors de la validation des signatures d'un package.

`trust-anchor-directory`

La valeur de cette propriété est le nom du chemin d'accès au répertoire contenant les ancres de confiance pour l'image. Ce chemin est relatif à l'image.

Propriétés de l'éditeur pour les packages signés

Configurez les propriétés de l'éditeur suivantes pour utiliser des packages signés d'un éditeur particulier.

`signature-policy`

La fonction de cette propriété est identique à la fonction de la propriété d'image `signature-policy`, si ce n'est que cette propriété s'applique uniquement aux packages de l'éditeur spécifié.

`signature-required-names`

La fonction de cette propriété est identique à la fonction de la propriété d'image `signature-required-names`, si ce n'est que cette propriété s'applique uniquement aux packages de l'éditeur spécifié.

Configuration des propriétés de signature de package

Utilisez les sous-commandes `set-property`, `add-property-value`, `remove-property-value` et `unset-property` pour configurer les propriétés de signature de package pour cette image.

Utilisez les options `--set-property`, `--add-property-value`, `--remove-property-value` et `--unset-property` de la sous-commande `set-publisher` pour spécifier la stratégie de signature et les noms requis pour un éditeur particulier.

EXEMPLE 3-33 Demande de vérification de tous les packages signés

Configurez cette image pour vérifier tous les packages signés.

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

EXEMPLE 3-34 Demande de signature de tous les packages

Configurez cette image pour exiger que tous les packages soient signés. Nécessite également que la chaîne `oracle.com` soit considérée comme un nom commun pour l'un des certificats dans la chaîne de confiance.

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

EXEMPLE 3-35 Demande de signature de tous packages d'un éditeur spécifié

Configurez cette image pour exiger que tous les packages installés à partir de l'éditeur `example.com` soient signés.

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

EXEMPLE 3-36 Ajout d'un nom de signature requis

Cet exemple ajoute la chaîne `trustedname` à la liste de noms communs de l'image qui doivent être vus dans la chaîne de confiance d'une signature pour qu'elle soit considérée valide.

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

EXEMPLE 3-37 Suppression d'un nom de signature requis

Cet exemple supprime la chaîne `trustedname` de la liste de noms communs de l'image qui doivent être vus dans la chaîne de confiance d'une signature pour qu'elle soit considérée valide.

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

EXEMPLE 3-38 Ajout d'un nom de signature requis pour un éditeur spécifié

Cet exemple ajoute la chaîne `trustedname` à la liste de noms communs de l'éditeur `example.com` qui doivent être vus dans la chaîne de confiance d'une signature pour qu'elle soit considérée valide.

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


Création et gestion des images

IPS vous permet de créer et de gérer des images sur votre système Oracle Solaris.

Pourquoi créer une image ?

Une image est un emplacement sur le système dans lequel les packages et les fichiers, répertoires, liens et dépendances qui leur sont associés peuvent être installés.

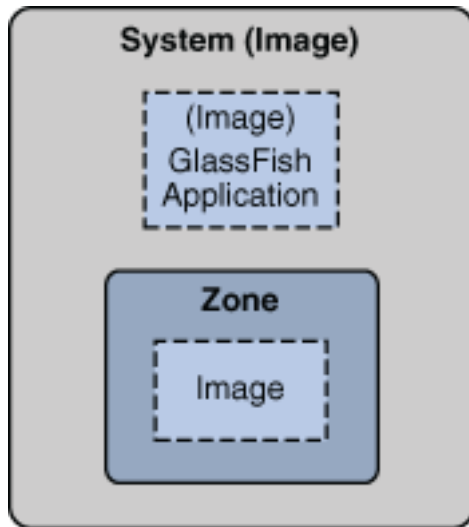
Une image peut être de trois types différents :

- Les images complètes peuvent fournir un système complet. Un système Oracle Solaris installé constitue une image complète.
- Les images partielles sont liées à une image complète (image parent) mais ne suffisent pas pour fournir un système complet.
- Les images d'utilisateur contiennent uniquement des packages réadressables.

Après l'installation du SE Oracle Solaris, une image complète contient le système de fichiers root et son contenu. Vous souhaitez peut-être créer des images pour fournir une séparation logique entre différentes applications logicielles.

La figure suivante illustre des images partielles dans une image système complète.

FIGURE 4-1 Images partielles dans une image système



Reportez-vous à la section Zones du document *System Administration Guide: Oracle Solaris Zones, Oracle Solaris 10 Containers, and Resource Management* pour en savoir plus sur les zones.

Meilleures pratiques pour la création d'images

- **Où créer une image.** Ne créez pas plusieurs images dans un même répertoire. Ne créez pas d'image sous la racine (/).
- **Quels packages utiliser.** Les informations concernant l'éditeur et le référentiel de packages doivent être définies lors de la création d'une image. Fournissez ces informations à l'aide de l'option -p comme cela est illustré dans [“Créer une image” à la page 43](#).
- **Quelle image modifier.** Si votre système dispose de plusieurs images, spécifiez celle que vous souhaitez modifier à l'aide de la variable d'environnement `PKG_IMAGE` ou de l'option -R.
- **Versions multiples d'un package.** IPS part du principe qu'une seule version d'un package est installée sur une image spécifique. Pour installer plusieurs versions d'un package, le nom du package doit inclure le numéro de version et les dépendances doivent refléter le numéro de version et être liées à la version de package appropriée.

Créer une image

La commande `pkg image -create` permet de créer une zone de gestion des packages et de spécifier l'éditeur de l'image. Une image utilisateur est créée par défaut.

- Assurez-vous de disposer des droits d'écriture dans le répertoire où vous tentez de créer une image.
- Utilisez l'option `-p` ou l'option `--preferred` pour spécifier un URI d'éditeur préféré. Une fois l'image créée, le catalogue de l'éditeur préféré est récupéré.

EXEMPLE 4-1 Création d'une image complète

Cet exemple spécifie l'option `-F` pour créer une image complète. L'option `-p` définit `example.com` comme éditeur préféré, avec le référentiel situé à l'adresse `http://pkg.example.com:10000`. L'image est créée dans le répertoire `/aux0/example_root`.

```
# pkg image-create -F -p example.com=http://pkg.example.com:10000 \
/aux0/example_root
```

Afficher, définir et supprimer des propriétés de l'image

Pour consulter la description des propriétés de l'image, reportez-vous à la section « PROPRIÉTÉS DE L'IMAGE » à la «[Page de manuel pkg\(1\)](#)» à la [page 48](#).

EXEMPLE 4-2 Affichage des valeurs des propriétés de l'image

Utilisez la commande `pkg property` pour afficher les propriétés d'une image :

```
$ pkg property
PROPERTY                                VALUE
ca-path                                  /etc/openssl/certs
flush-content-cache-on-success          False
mirror-discovery                         False
preferred-publisher                      solaris
publisher-search-order                   ['solaris']
send-uuid                                 True
signature-policy                         ignore
signature-required-names                 []
trust-anchor-directory                   etc/certs/CA
```

La plupart de ces propriétés sont gérées à l'aide des sous-commandes `set -property`, `unset -property`, `add -property -value` et `remove -property -value` et de la commande `pkg`. Les propriétés `preferred-publisher` et `publisher-search-order` sont définies à l'aide de la commande `pkg set -publisher`. Voir [Exemple 3-26](#) et [Exemple 3-29](#).

EXEMPLE 4-3 Définition de la valeur d'une propriété d'image

Utilisez la commande `pkg set -property` pour définir la valeur d'une propriété d'image ou ajouter et définir une propriété.

Dans cet exemple, la valeur de la propriété `mirror-discovery` est définie.

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

EXEMPLE 4-4 Réinitialisation de la valeur d'une propriété d'image

Utilisez la commande `pkg unset -property` pour rétablir les valeurs par défaut des propriétés spécifiées.

Dans cet exemple, le paramétrage de la propriété `mirror-discovery` est annulé.

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

Mise à jour d'une image

Utilisez la commande `pkg update` pour mettre à jour tous les packages installés vers leur dernière version. Le comportement de la commande `pkg update` est similaire à celui du Gestionnaire de mises à jour. Voir [“Utilisation du Gestionnaire de mises à jour”](#) à la page 18.

EXEMPLE 4-5 Mise à jour de l'environnement d'initialisation actif

Cette commande remplace tous les packages installés sur l'image actuelle par leur toute dernière version autorisée par les contraintes imposées par les packages installés et la configuration d'éditeur.

```
# pkg update --accept
```

Par défaut, chaque package est mis à jour depuis l'éditeur qui a fourni la version actuellement installée. Vous pouvez contrôler l'éditeur qui fournit les packages en spécifiant la persistance de l'éditeur et l'ordre de recherche. Voir [Exemple 3-28](#) et [Exemple 3-29](#).

Si vous spécifiez l'option `--accept`, vous indiquez que vous acceptez les conditions de licence des packages qui sont mis à jour. Si vous ne spécifiez pas l'option `--accept` alors que les licences de package exigent l'acceptation, aucun package n'est mis à jour.

Si des packages particuliers sont mis à jour, comme des composants d'un certain noyau, un autre EI est créé. Voir [“Images et environnements d'initialisation”](#) à la page 12.

EXEMPLE 4-6 Mise à jour d'une image dans un environnement d'initialisation inactif

Dans cet exemple, une image est mise à jour dans un EI monté au niveau de `/mnt`.

EXEMPLE 4-6 Mise à jour d'une image dans un environnement d'initialisation inactif (Suite)

```
# beadm mount OracleSolaris-1 /mnt  
# pkg -R /mnt update --accept --be-name OracleSolaris-3
```


Aide-mémoire de la commande

Cet aide-mémoire répertorie et décrit brièvement les commandes dans Image Packaging System. Cette annexe présente également la page de manuel complète pour la commande `pkg(1)`.

Commandes IPS

La page de manuel `pkg(5)` décrit le système Image Packaging System. La page de manuel `pkg(5)` définit un package IPS et traite des noms, des versions et des FMRI des packages IPS. La page de manuel `pkg(5)` définit une action IPS et décrit différents types d'actions. La page de manuel `pkg(5)` décrit les mécanismes de positionnement, les propriétés et les stratégies d'image.

Le logiciel Image Packaging System offre les commandes suivantes :

- | | |
|--------------------------------|---|
| <code>packagemanager(1)</code> | Ouvre l'outil Gestionnaire de packages qui est l'interface graphique d'IPS. Reportez-vous à la section “Utilisation du Gestionnaire de packages” à la page 15 pour plus d'informations. |
| <code>pkg(1)</code> | Liste, recherche, installe, met à jour et désinstalle des packages. Gère des éditeurs de package.

Pour de plus amples informations, reportez-vous aux sections suivantes : <ul style="list-style-type: none">▪ Chapitre 3, “Utilisation de packages”▪ Chapitre 4, “Création et gestion des images”▪ “Page de manuel pkg(1)” à la page 48 |
| <code>pkg.depotd(1M)</code> | Crée des référentiels de packages et gère des référentiels de packages existants ou en fournit l'accès réseau. |
| <code>pkgdepend(1)</code> | Analyse les dépendances de package. Utilisé par les éditeurs de package pour générer et résoudre les dépendances des packages. |

pkgdiff(1)	Compare deux fichiers manifest de package et signale les différences.
pkgfmt(1)	Formate un fichier manifest de package, en triant les actions par type et en plaçant les attributs clé en premier.
pkglint(1)	Effectue des opérations de vérification sur des fichiers manifest de packages IPS, y compris des tests pour des actions dupliquées, des attributs manquants et des droits d'accès à un fichier inhabituels.
pkgmogrify(1)	Lors de l'édition de fichiers manifest de package, simplifie les transformations normales nécessaires à l'automatisation de la génération de logiciel et de la republication de package.
pkgrecv(1)	Récupère le contenu d'un package à partir d'un référentiel de packages. Le package récupéré peut alors être modifié et republié à l'aide de la commande pkgsend.
pkgrepo(1)	Crée et gère des référentiels de packages.
pkgsend(1)	Publie des packages dans un référentiel de packages.
pkgsign(1)	Ajoute une action de signature, comprenant une clé et des certificats spécifiés, à un package d'un référentiel sans changer l'horodatage du package.
pm-updatemanager(1)	Ouvre l'outil Gestionnaire de packages qui est l'interface graphique d'IPS. Reportez-vous à la section “Utilisation du Gestionnaire de packages” à la page 15 pour plus d'informations.

Page de manuel pkg(1)

NAME

pkg - image packaging retrieval client

SYNOPSIS

```
/usr/bin/pkg [options] command [cmd_options] [operands]

/usr/bin/pkg install [-nvq] [--accept] [--licenses] [--no-index]
  [--no-refresh] [--deny-new-be | --require-new-be] [--be-name name]
  pkg_fmri_pattern ...
/usr/bin/pkg uninstall [-nrq] [--no-index]
  [--deny-new-be | --require-new-be] [--be-name name]
  pkg_fmri_pattern ...

/usr/bin/pkg update [-fnvq] [--accept] [--be-name name]
  [--deny-new-be | --require-new-be] [--licenses] [--no-index]
  [--no-refresh]

/usr/bin/pkg refresh [--full] [publisher ...]

/usr/bin/pkg contents [-Hmr] [-a attribute=pattern ...]
```



```

    [-o attribute ...] [-s sort_key] [-t action_type ...]
    [pkg_fmri_pattern ...]
/usr/bin/pkg info [-lr] [--license] [pkg_fmri_pattern ...]
/usr/bin/pkg list [-Hafnsuv] [--no-refresh] [pkg_fmri_pattern ...]
/usr/bin/pkg search [-HIaflpr] [-o attribute ...] [-s repo_uri]
    query

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

/usr/bin/pkg image-create [-FPUfz] [--force]
    [--full|--partial|--user] [--zone] [-k ssl_key] [-c ssl_cert]
    [--no-refresh] [--variant <variant_spec>=<instance> ...]
    [-g uri|--origin=uri ...] [-m uri|--mirror=uri ...]
    [--facet <facet_spec>=[True|False] ...]
    (-p|--publisher) [<name>=<repo_uri> dir

/usr/bin/pkg variant [-H] [<variant_spec>]
/usr/bin/pkg change-variant [-nvq] [--accept]
    [--deny-new-be | --require-new-be] [--be-name name]
    [--licenses] <variant_spec>=<instance> ...

/usr/bin/pkg facet [-H] [<facet_spec>]
/usr/bin/pkg change-facet [-nvq] [--accept] [--be-name name]
    [--deny-new-be | --require-new-be]
    [--licenses] <facet_spec>=[True|False|None] ...

/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 ...
/usr/bin/pkg property [-H] [propname ...]

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

/usr/bin/pkg history [-Hl] [-n number]
/usr/bin/pkg purge-history

/usr/bin/pkg rebuild-index

/usr/bin/pkg version

```

/usr/bin/pkg help

DESCRIPTION

pkg is the retrieval client for the image packaging system. With a valid configuration, pkg can be invoked to create locations for packages to be installed, called 'images', and install packages into those images. Packages are published by publishers, who may make their packages available at one or more repositories. pkg, then, retrieves packages from a publisher's repository and installs them into an image.

A publisher is a forward domain name that can be used to identify a person, group of persons, or an organization as the source of one or more packages. The name of a publisher does not have to be contained within the URIs that identify the locations of publisher repositories. For example, the name of a publisher might be "example.com", but its repositories may be hosted at "example.org" or "example.net".

A repository is a location where clients can publish and retrieve package content (files contained within the package such as programs, documents, etc.) and metadata (information about the package such as its name, description, etc.). As an example, a publisher named "example.org" may have their repository located at the URI "http://example.org/repository".

pkg can also uninstall packages, refresh publisher metadata (such as catalogs), validate package installation in an image, and query the image for various tokens. These queries can also be made of pkg(5) repositories.

Images can be of three types: full images, capable of providing a complete system; partial images, which are linked to a full image (parent image), but do not provide a complete system on their own; and user images, which contain only relocatable packages. (See NOTES on user images.)

pkg(1) attempts to determine, based on its working directory, in what image it has been invoked. If no image metadata can be found in the parent directories, the invocation will fail.

OPTIONS

The following options are supported:

- R dir
Operate on the image rooted at dir, rather than the one discovered automatically.
- help or -?
Displays a usage message.

SUBCOMMANDS

The following subcommands are supported:

```
image-create [-FPUfz] [--force] [--full|--partial|--user] [--zone]
[-k ssl_key] [-c ssl_cert] [--no-refresh]
[--variant <variant_spec>=<instance> ...]
[-g uri|--origin=uri ...] [-m uri|--mirror=uri ...]
[--facet <facet_spec>=[True|False] ...]
```

(-p|--publisher) [<name>=<repo_uri> dir

Create, at location given by dir, an image suitable for package operations. The default image type is user, as given by the -U (--user) option. The image type may 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. Additional origins can be specified using -g or --origin, while additional mirrors can be specified using -m or --mirror.

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

For publishers using client SSL authentication, a client key and client certificate may be registered via the -c and -k options, and will be used for all publishers added during image creation.

If the image is to be run within nonglobal zone context, then the -z (--zone) option can be used to set an appropriate filter.

With -f (--force), force the creation of an image over an existing image. This option should be used with care.

With --no-refresh, do not attempt to contact the repositories for the image's publishers to retrieve publisher metadata (e.g. catalogs).

With --variant, set the specified variant to the indicated value.

With --facet, set the specified facet to the indicated value.

refresh [--full] [publisher ...]

Retrieve updates to the metadata (e.g. catalogs) for each publisher specified. When given no arguments, retrieves updates for each publisher registered within the image.

With --full, retrieve all publisher metadata instead of attempting an incremental update.

install [-nvq] [--accept] [--licenses] [--no-index] [--no-refresh] [--deny-new-be | --require-new-be] [--be-name] pkg_fmri_pattern ...

Installs and updates packages to the newest version that match pkg_fmri_pattern allowed by the packages installed in the image.

Some configuration files may be renamed or replaced during the install process. For more information on how the package system determines which files to preserve, and how they are preserved during package operations, see "File Actions" in pkg(5).

With the -n option, execute the requested operation but make no persistent changes to the image.

With the `-v` option, issue verbose progress messages during the requested operation. With the `-q` option, issue no progress messages during the requested operation.

With `--accept`, you 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 operation will fail.

With `--licenses`, display all of the licenses for the packages that will be installed or updated as part of this operation.

With `--no-index`, do not update the search indices after the operation has completed successfully.

With `--no-refresh`, do not attempt to contact the repositories for the image's publishers to retrieve publisher metadata (e.g. catalogs).

With `--be-name`, rename the newly created boot environment to be the argument given. This option is only valid if a new boot environment is created during the operation. See also `beadm(1m)`.

With `--require-new-be`, always create a new boot environment. Without this option, a boot environment is created automatically if needed.

With `--deny-new-be`, disallow creation of a new boot environment; the operation will not be performed if a new boot environment is required.

```
uninstall [-nrpq] [--no-index] [--deny-new-be | --require-new-be]
  [--be-name name] pkg_fmri_pattern ...
```

Removes installed packages that match `pkg_fmri_pattern`.

With `-r`, recursively uninstall any packages that contain 'require' dependencies on the initial package. (Packages containing 'optional' or 'incorporate' dependencies are not included in the removal.)

For all other options, refer to the `install` command above for usage and their effects.

```
update [-fnvq] [--accept] [--be-name name] [--licenses]
  [--no-index] [--no-refresh] [--deny-new-be | --require-new-be]
  [pkg_fmri_pattern ...]
```

With no arguments, or if `'*'` is one of the patterns provided, update all installed packages in the current image to the newest version allowed by the constraints imposed on the system by installed packages and publisher configuration.

If `pkg_fmri_pattern` is provided, update will replace packages that are installed, and that match `pkg_fmri_pattern`, with the

newest version allowed by the pattern(s) and the constraints imposed on the system by installed packages and publisher configuration. Versions older or newer than what is already installed may be specified to perform in place downgrades or upgrades of specific packages. Please note that updating specific packages across package rename or obsolete boundaries is not supported.

Any preserved configuration files that are part of packages to be downgraded by update and that have been changed since the original version was installed will be renamed using the extension '.update'. For more information on how the package system determines which files to preserve, and how these files are preserved during package upgrades, see "File Actions" in pkg(5).

With the -f option, don't execute the client up to date check when updating all installed packages.

For all other options, refer to the install command above for usage and their effects.

```
info [-lr] [--license] [pkg_fmri_pattern ...]
Display information about packages in a human-readable form.
Multiple FMRI patterns may be specified; with no patterns,
display information on all installed packages in the image.

With -l, use the data available from locally installed packages.
This is the default.

With -r, retrieve the data from the repositories of the image's
configured publishers. Note that you must specify one or more
package patterns in this case.

With --license, print out the license text(s) for the package.
This may be combined with -l or -r.
```

```
contents [-Hmr] [-a attribute=pattern ...] [-o attribute ...]
[-s sort_key] [-t action_type ...] [pkg_fmri_pattern ...]
Display the contents (action attributes) of packages in the
current image. By default, only the path attribute is displayed,
but the attribute set may be determined with the -o option. The
-o option may be specified multiple times, or multiple attributes
may be specified as the argument to one -o option by separating
the attribute names with commas. Only actions which have the
requested attributes will be displayed. The -m option may
also be used, as a shorthand for '-Ho action.raw'.
```

The -a option allows you to limit the output to those actions which have an attribute named in the option argument the value of which matches the (glob) pattern in the option argument (following the attribute name with an equals sign). If multiple -a options are given, then actions matching any of them will be displayed.

The -s option specifies the attribute by which the listing should be sorted.

The `-t` option limits the action types which will be displayed.

The `-H` option causes the headers to be omitted.

The `-r` option retrieves the requested data from the repositories of the image's configured publishers. This option is intended to be used when the named packages are not already installed.

With no arguments, the output includes all installed packages. Alternatively, multiple FMRI patterns may be specified, which restricts the display to the contents of the matching packages. When using `-r`, one or more `pkg_fmri_patterns` must be specified.

Several special "pseudo" attribute names are available for convenience:

<code>action.hash</code>	Corresponds to the value of the action's hash, if the action carries a payload.
<code>action.key</code>	Corresponds to the value of the action's key attribute. For example, for a file action, this is the path to the file.
<code>action.name</code>	Corresponds to the name of the action. For example, for a file action, this is "file"
<code>action.raw</code>	Corresponds to the complete contents of the action as represented in the package manifest. This corresponds to the lines of output of 'pkg contents -m'
<code>pkg.fmri</code>	Corresponds to the full form FMRI of the package containing the action, such as <code>pkg://extra/virtualbox@3.0.0,5.11-0.101:20090702T175410Z</code>
<code>pkg.name</code>	Corresponds to the name of the package containing the action, such as "SUNWcs"
<code>pkg.publisher</code>	Corresponds to the publisher of the package containing the action, such as "opensolaris.org"
<code>pkg.shortfmri</code>	Corresponds to the short form FMRI of the package containing the action, such as <code>pkg://opensolaris.org/SUNWzone@0.5.11-0.79</code>

The contents and search subcommands are related: both are used to query the system for the contents of packages. The contents subcommand displays actions in one or more packages, filtering the output based on the options chosen by the user. The search subcommand approaches the query from the other direction, looking for packages which contain a user-supplied token.

Each subcommand is capable of formulating some queries of which the other is capable. Care should be taken in choosing the subcommand, as a given query may be more naturally formulated in one than in the other.

`search [-HIaflpr] [-o attribute ...] [-s repo_uri] query`
 Search for matches to the query, and display the results. Which tokens are indexed are action-dependent, but may include content hashes and pathnames. By default, queries are interpreted as a series of terms to be matched exactly. The '?' and '*' characters can be used as glob(3C)-style wildcards, allowing more flexible query matches.

With `-H`, omit the headers.

With `-I`, use a case-sensitive search.

By default, and with `-a`, perform the search and display information about the matching actions.

By default, search prunes results from packages older than the currently installed version and from package versions excluded by current incorporations. Use `-f` to show all results, regardless of package version.

With `-l`, search the image's installed packages.

With `-o`, the columns of the results may be controlled. The `-o` option may be specified multiple times, or multiple attributes may be specified as the argument to one `-o` option by separating the attribute names with commas. In addition to the "pseudo" attributes outlined above, more are defined for search results:

`search.match` Corresponds to the string which matched the search query.

`search.match_type` Corresponds to the attribute which contained the string that matched the search query.

With `-p`, display packages which have some actions that match each query term. Using this option is equivalent to putting '<>' around each term in the query. (For a description of the '<>' operator, please see below.)

By default, and with `-r`, search the repositories corresponding to the image's publishers.

With `-s`, search the pkg(5) repository located at the given URI. This may be specified multiple times.

Both `-l` and `-r` (or `-s`) may be specified together, in which case both local and remote searches will be performed.

In addition to simple token matching and wildcard search, a more complicated query language is supported. Phrases may be searched for by using ' or ". Note: Please make sure to take your shell into account so that pkg actually sees the ' or ".

Boolean search using AND and OR is supported. Field, or structured, queries are supported. The syntax for these is `pkg name:action_type:key:token`. Missing fields are implicitly wildcarded. A search for `:basename:pkg` would match all actions

types in all packages with a key of basename and which matched the token 'pkg'. Explicit wildcards are supported in the pkg_name and token fields, action_type and key must match exactly.

To convert actions to the packages which contain those actions, use '<>'. With the -a option, Searching for 'token' results in information about the actions matching token, while searching for '<token>' results in a list of packages containing actions which matched token.

list [-Hafnsuv] [--no-refresh] [pkg_fmri_pattern ...]

Display a list of packages in the current image, including state and other information. By default, package variants for a different architecture or zone type are excluded.

The usual output is in four columns:

NAME (PUBLISHER)	VERSION	STATE	UFOXI
SUNWcs	0.5.11-0.126	installed	----
web/firefox/plugin/flash (extra)	10.0.32.18-0.111	installed	----

The first column contains the name of the package. If the publisher from which it is installed (or available, if not installed) is not the preferred publisher, then the publisher name is listed in parentheses after the package name. The second column contains the release and branch versions of the package (see pkg(5)). The third column contains the state of the package as it exists on the system. Possible values are "installed" and "known". The last column contains a set of flags that show how the package relates to other packages:

- a "u" in the "U" column shows that a newer version is available, although it may not be possible to install this newer version due to package dependencies or constraints;
- an "f" in the "F" column shows that this version has been frozen (not implemented);
- an "o" in the "O" column shows that it is obsolete, while an "r" shows that it has been renamed (a form of obsolescence);
- an "x" in the "X" column shows that it is prevented from being installed because some other package has excluded it (not implemented); and
- an "i" in the "I" column shows that it has been constrained by an incorporation (not implemented).

With -H, omit the headers from the listing.

With -a, list installed packages and the newest version of packages that are available for installation. Packages are considered to be available for installation if they are allowed by the installed incorporations and by the image's variants. If one or more patterns are specified, then the newest version matching the specified pattern and is also allowed by any installed incorporations and the image's

variants will be listed. Without `-a`, list only installed packages.

With `-f` and `-a`, list all versions of all packages for all variants regardless of incorporation constraints or installed state.

With `-n`, display the newest versions of all known packages, regardless of installed state.

With `-s`, display a one-line short-form giving the package name and description. This option may be used with `-a`, `-n`, `-u` or `-v`.

With `-u`, list only packages with newer versions available.

With `-v`, show full package FMRIs, including publisher and complete version, all in the first column (the `VERSION` column disappears). This option may be used with `-a`, `-n`, or `-u`.

With `--no-refresh`, do not attempt to contact the repositories for the image's publishers to retrieve publisher metadata (e.g. catalogs).

`verify [-Hqv] [pkg_fmri_pattern ...]`

Validate the installation of packages in the current image. Please note that verification of installed package content is based on a custom content analysis that may return different results than those of other programs.

With `-H`, omit the headers from the verification output.

With `-q`, print nothing, but return failure if there are any fatal errors.

With `-v`, include informational messages regarding packages.

`variant [-H] [<variant_spec> ...]`

Display the current values of all variants, or with arguments, only the variants specified

With `-H`, omit the headers from the listing.

`change-variant [-nvq] [--accept] [--be-name name] [--licenses] <variant_spec>=<instance> ...`

Change the specified variants in the current image.

With the `-n` option, plan the requested operation but make no actual changes.

With the `-v` option, issue verbose progress messages during the requested operation. With the `-q` option, be completely silent.

With `--accept`, you 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 operation will fail.

With `--licenses`, display all of the licenses for the packages that will be installed or updated as part of this operation.

With `--be-name`, rename the newly created boot environment to be the argument given. This option is only valid if a new boot environment is created during image update. See also `beadm(1m)`.

With `--require-new-be`, always create a new boot environment. Without this option, a new boot environment is only created if needed.

With `--deny-new-be`, disallow creation of a new boot environment; the operation will not be performed if a new boot environment is required.

`facet [-H] [<facet_spec> ...]`

Without arguments, displays the current values of all facets. With argument(s), evaluate if each facet would be true or false and print the result.

With `-H`, omit the headers from the listing.

`change-facet [-nvq] [--accept] [--be-name name] [--licenses] <facet_spec>=[True|False|None] ...`

Change the specified facets in the current image.

With the `-n` option, plan the requested operation but make no actual changes.

With the `-v` option, issue verbose progress messages during the requested operation. With the `-q` option, be completely silent.

With `--accept`, you 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 operation will fail.

With `--licenses`, display all of the licenses for the packages that will be installed or updated as part of this operation.

With `--be-name`, rename the newly created boot environment to be the argument given. This option is only valid if a new boot environment is created during the operation. See also `beadm(1m)`.

With `--require-new-be`, always create a new boot environment. Without this option, a new boot environment is only created if needed.

With `--deny-new-be`, disallow creation of a new boot environment; the operation will not be performed if a new boot environment is required.

Facets may be set to True or False. Setting one to None removes that facet specification from the current image.

`fix [--accept] [--licenses] [pkg_fmri_pattern ...]`

Fix any errors reported by `pkg verify`. Please note that verification of installed package content is based on a custom content analysis that may return different results

than those of other programs.

With `--accept`, you 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 operation will fail.

With `--licenses`, display all of the licenses for the packages that will be installed or updated as part of this operation.

`set-property` *propname* *propvalue*

Update an existing image property or add a new image property; except for `preferred-publisher`, which can only be changed using `set-publisher`.

`add-property-value` *propname* *propvalue*

Add a value to an existing image property or add a new image property; except for `preferred-publisher`, which can only be changed using `set-publisher`.

`remove-property-value` *propname* *propvalue*

Remove a value from an existing image property; except for `preferred-publisher`, which can only be changed using `set-publisher`.

`unset-property` *propname* ...

Remove an existing image property or properties; except for `preferred-publisher`, which can only be changed using `set-publisher`.

`property` [-H] [*propname* ...]

Display image property information. With no argument, display the names and values for all image properties. If a specific list of property names is requested, display the names and values for those properties.

With `-H`, omit the headers from the listing.

```
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]
[--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]
```

Update an existing publisher or add an additional package publisher. If no options affecting search order are specified, new publishers are appended to the search order and are thus searched last.

With `-P`, set the specified publisher as the preferred publisher, i.e. first in the search order. When installing new packages, this publisher will be searched first. Updates to already installed packages will come from the same publisher that originally provided the package so long as that publisher remains sticky.

With `--non-sticky`, specify that higher ranked publishers than this one may provide updates to packages originally installed from this publisher.

With `--sticky`, return to the default behavior of always sourcing updates from the same publisher that provided the package originally.

With `--search-before`, alter the publisher search order so that the publisher being modified is now searched before the specified publisher.

With `--search-after`, alter the publisher search order so that the publisher being modified is now searched after the specified publisher.

With `--approve-ca-cert`, add the given certificate as a CA certificate that is trusted. The hashes of the user approved CA certificates are listed in the output of the detailed pkg publisher view for a publisher.

With `--revoked-ca-cert`, treat the certificate with the given hash as revoked. The hashes of the user revoked CA certificates are listed in the output of the detailed pkg publisher view for a publisher.

With `--unset-ca-cert`, remove the certificate with the given hash from the list of approved and the list of revoked certificates.

With `--set-property`, update an existing publisher property or add a new publisher property.

With `--add-property-value`, add a value to an existing publisher property or add a new publisher property.

With `--remove-property-value`, remove a value from an existing publisher property.

With `--unset-property`, remove an existing publisher property.

With `-c` and `-k`, specify client SSL certificate and key respectively.

With `-g` (`--add-origin`), add the URI as an origin for the given publisher. This should be the location of a package repository.

With `-G` (`--remove-origin`), remove the URI from the list of origins for the given publisher.

With `--no-refresh`, do not attempt to contact the publisher specified to retrieve its metadata (e.g. catalog).

With `--reset-uuid`, choose a new unique identifier that identifies this image to its publisher.

With `-m` (`--add-mirror`), add the URI as a mirror for the given publisher.

With `-M` (`--remove-mirror`), remove the URI from the list of mirrors for the given publisher.

With `-p`, retrieve publisher configuration information from the specified repository URI. If a publisher is specified, then only the matching one will be added or updated. If no publisher is specified, all will be added or updated as appropriate. This option may not be combined with the `-g`, `--add-origin`, `-G`, `--remove-origin`, `-m`, `--add-mirror`, `-M`, `--remove--mirror`, `--disable`, `--enable`, `--no-refresh`, or `--reset-uuid` options.

With `-e` (`--enable`), enable the publisher; with `-d` (`--disable`), disable the publisher. A disabled publisher is not used when populating the package list or in certain package operations (install, uninstall, and update). However, the properties for a disabled publisher can still be set and viewed. If only one publisher exists, it cannot be disabled.

`unset-publisher publisher ...`

Remove the configuration associated with the given publisher or publishers.

`publisher [-HPn] [publisher ...]`

Display publisher information. With no arguments, display the list of all publishers, their origin URIs, and mirrors in order of search preference. If specific publishers are requested, display the configuration values, including mirrors, associated with those publishers.

With `-H`, omit the headers from the listing.

With `-P`, display only the preferred publisher.

With `-n`, display only enabled publishers.

`history [-HL] [-n number]`

Display the command history of the applicable image.

With `-H`, omit the headers from the listing.

With `-l`, display log records in long format, which, in addition to the standard format, includes 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 that were encountered while executing the command.

With `-n`, display only the specified number of most recent entries.

`purge-history`

Deletes all existing history information.

`rebuild-index`

Rebuilds the index used by 'pkg search'. This is a recovery operation not intended for general use.

version

Display a unique string identifying the version of pkg(1). This string is not guaranteed to be comparable in any fashion between versions.

IMAGE PROPERTIES

The following properties are part of the image and may be set using the set-property subcommand. The values of these properties are viewable with the property subcommand.

ca-path

(string) A pathname that points to a directory where CA certs are kept for SSL operations. The format of this directory is specific to the underlying SSL implementation. If the administrator would like to use an alternate location for trusted CA certificates, this value should be changed to point to a different directory. Please see the 'CApath' portions of SSL_CTX_load_verify_locations(3openssl) for requirements about the CA directory.

Default value: /etc/openssl/certs

flush-content-cache-on-success

(boolean) If this is set to True, the package client will remove 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, it will flush its content cache, provided this option has not been changed.

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

Default value: False

mirror-discovery

(boolean) Mirror-discovery tells the client to discover link-local content mirrors using mDNS and DNS-SD. If this is set to True, the client will attempt to download package content from mirrors it dynamically discovers. To run a mirror that advertises its content via mDNS, see pkg.depotd(1m).

Default value: False

send-uuid

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

Default value: True

signature-policy

(string) Determine what checks will be performed on manifests

when installing a package into this 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. The following are the valid values for this property.

ignore

Ignore signatures for all manifests.

verify

Verify that all manifests with signatures are validly signed, but do not require all installed packages to be signed.

require-signatures

Require that all newly installed packages have at least one valid signature. 'pkg fix' and 'pkg verify' will 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' property appear as a common name of the certificates used to verify the chains of trust of the signatures.

signature-required-names

(list of strings) A list of names which must be seen as common names of certificates while validating the signatures of a package.

trust-anchor-directory

(string) The pathname of the directory that contains the trust anchors for the image. This path is relative to the image.

PUBLISHER PROPERTIES

The following properties are part of the image and may be set using the set-property option of the set-publisher subcommand.

signature-policy

(string) This property functions identically to the image property of the same name except it only applies to packages from the particular publisher.

signature-required-names

(list of strings) This property functions identically to the image property of the same name except it only applies to packages from the particular publisher.

EXAMPLES

Example 1: Create a new, full image, with publisher example.com, stored at /aux0/example_root.

```
$ pkg image-create -F -p example.com=http://pkg.example.com:10000 \
  /aux0/example_root
```

Example 2: Create a new, full image, with publisher example.com, that also has an additional mirror, two additional origins and is stored at /aux0/example_root.

```
$ pkg image-create -F -p example.com=http://pkg.example.com:10000 \
-g http://alternate1.example.com:10000/ \
-g http://alternate2.example.com:10000/ \
-m http://mirror.example.com:10000/ \
/aux0/example_root
```

Example 3: Install the latest version of the widget package in the current image.

```
$ pkg install application/widget
```

Example 4: List the contents of the SUNWzfs package. Display the action name, the mode of the file (if defined), the size (if defined), the path, and the target (if a link). Limit the action to types dir, file, link, and hardlink, since specifying the action.name attribute, which is available for all actions, will display a line for all actions, which is not desired here.

```
$ pkg contents -t dir,file,link,hardlink \
-o action.name,mode,pkg.size,path,target SUNWzfs
```

NAME	MODE	SIZE	PATH	TARGET
dir	0755		etc	
dir	0755		etc/fs	
dir	0755		etc/fs/zfs	
link			etc/fs/zfs/mount	../../../../sbin/zfs
link			etc/fs/zfs/umount	../../../../sbin/zfs
dir	0755		etc/zfs	
dir	0755		lib	
dir	0755		lib/amd64	
link			lib/amd64/libzfs.so	libzfs.so.1
file	0755	469616	lib/amd64/libzfs.so.1	
file	0644	62057	lib/amd64/llib-lzfs.ln	
link			lib/libzfs.so	libzfs.so.1
....				

Example 5: List the contents of SUNWfirefox and SUNWthunderbird, limiting the display to just the package name and path attributes of actions whose "path" attribute ends in ".desktop" or ".png".

```
$ pkg contents contents -o pkg.name,path -a path=\*.desktop \
-a path=\*.png SUNWfirefox SUNWthunderbird
```

PKG.NAME	PATH
SUNWfirefox	usr/lib/firefox/chrome/icons/default/default16.png
SUNWfirefox	usr/lib/firefox/chrome/icons/default/default32.png
SUNWfirefox	usr/lib/firefox/chrome/icons/default/default48.png
SUNWfirefox	usr/lib/firefox/icons/document.png
SUNWfirefox	usr/lib/firefox/icons/mozicon128.png
SUNWfirefox	usr/lib/firefox/res/html/folder.png
SUNWfirefox	usr/share/applications/firefox.desktop
SUNWthunderbird	usr/share/applications/thunderbird.desktop
SUNWfirefox	usr/share/pixmaps/firefox-icon.png
SUNWthunderbird	usr/share/pixmaps/thunderbird-icon.png

Example 6: Search the package database for the token "bge".

```
$ pkg search bge
```

INDEX	ACTION	VALUE	PACKAGE
basename	file	kernel/drv/bge	pkg:/SUNWbge@0.5.11-0.79


```
driver_name driver bge                                pkg:/SUNWbge@0.5.11-0.79
```

The token shows up in the package SUNWbge both as the basename for the file action representing /kernel/drv/bge and as a driver name.

Example 7: Search for installed packages which depend on SUNWipkg.

```
$ pkg search -l 'depend::SUNWipkg'
INDEX      ACTION VALUE                                PACKAGE
incorporate depend SUNWipkg@0.5.11-0.111    pkg:/entire@0.5.11-0.111
require    depend SUNWipkg@0.5.11-0.111    pkg:/slim_install@0.1-0.111
require    depend SUNWipkg@0.5.11-0.111    pkg:/SUNWipkg-brand@0.5.11-0.111
```

Example 8: Search for all incorporate dependencies in installed packages.

```
$ pkg search -l 'depend:incorporate:'
INDEX      ACTION VALUE                                PACKAGE
incorporate depend BRMCmbnx@0.5.11-0.111    pkg:/entire@0.5.11-0.111
incorporate depend BRMCmbnx@0.5.11-0.111    pkg:/entire@0.5.11-0.111
....
```

Example 9: Add new publisher example.org, with a repository located at <http://www.example.org/repo>:

```
$ pkg set-publisher -g http://www.example.org/repo example.org
```

Example 10: Add new publisher example.com, with a secure repository located at <https://secure.example.com/repo>, and a key and cert stored in the directory /root/creds:

```
$ pkg set-publisher -k /root/creds/example.key \
  -c /root/creds/example.cert -g https://secure.example.com/repo \
  example.com
```

Example 11: Add new publisher with a repository located at /export/repo using automatic configuration:

```
$ pkg set-publisher -p file:/export/repo
```

Example 12: Add new publisher example.org with a repository located at /export/repo/example.com using manual configuration:

```
$ pkg set-publisher -g file:/export/repo example.com
```

Example 13: Configure an image to verify all signed packages.

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

Example 14: Configure an image to require all packages to be signed and the string "opensolaris.org" has to be seen as a common name for one of the certificates in the chain of trust.

```
$ pkg set-property signature-policy require-names opensolaris.org
```

Example 15: Configure an image so that all packages installed from publisher foo must be signed.

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

Example 16: Add the string "foo" 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 foo
```

Example 17: Remove the string "foo" from publisher test's list of common names that must be seen to validate a signature.

```
$ pkg set-publisher --remove-property-value signature-require-names=foo \  
test
```

Example 18: Add the certificate stored in /tmp/example_file.pem as a trusted CA certificate for the publisher test.

```
$ pkg set-publisher --approve-ca-cert /tmp/example_file.pem
```

Example 19: Revoke the certificate with the hash a12345 for publisher test, preventing it from validating any signatures for packages from test.

```
$ pkg set-publisher --revoke-ca-cert a12345
```

Example 20: Make pkg forget that the certificate a12345 was ever added or revoked by the user.

```
$ pkg set-publisher --unset-ca-cert a12345
```

Example 21: Downgrade the installed package foo@1.1 to an older version:

```
$ pkg update foo@1.0
```

EXIT STATUS

The following exit values are returned:

- 0 Command succeeded.
- 1 An error occurred.
- 2 Invalid command line options were specified.
- 3 Multiple operations were requested, but only some of them succeeded.
- 4 No changes were made - nothing to do.
- 5 The requested operation cannot be performed on a live image.
- 6 The requested operation cannot be completed as the licenses for the packages being installed or updated have not been accepted.
- 7 The image is currently in use by another process and cannot be modified.

FILES

A pkg(5) image can be located arbitrarily within a larger file

system. In the following, the token `$IMAGE_ROOT` is used to distinguish relative paths. For a typical system installation, `$IMAGE_ROOT` is equivalent to `"/"`.

`$IMAGE_ROOT/var/pkg` Metadata directory for a full or partial image.

`$IMAGE_ROOT/.org.opensolaris/pkg` Metadata directory for a user image.

Within a particular image's metadata, certain files and directories can contain information useful during repair and recovery. We use the token `$IMAGE_META` to refer to the top-level directory containing the metadata. `$IMAGE_META` is typically one of the two paths given above.

`$IMAGE_META/lost+found` Location of conflicting directories and files moved during a package operation.

`$IMAGE_META/publisher` Contains a directory for each publisher. Each directory stores publisher-specific metadata.

Other paths within the `$IMAGE_META` directory hierarchy are `Private`, and are subject to change.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWipkg pkg:/package/pkg
Interface Stability	None / Under Development

SEE ALSO

`pkgsend(1)`, `pkg.depotd(1M)`, `glob(3C)`, `attributes(5)`, `pkg(5)`

NOTES

The image packaging system is an under-development feature. Command names, invocation, formats, and operations are all subject to change. Development is hosted in the OpenSolaris community at:

<http://hub.opensolaris.org/bin/view/Project+pkg/>

At present, user images are not restricted to relocatable packages--but they will be.

The `pkg(1)` command recognizes use of the `http_proxy` and `https_proxy` environment variables to select a suitable HTTP or HTTPS proxy server. At present, particular care is needed when using local repository URIs--such as `http://localhost:10000/--with the http_proxy environment variable; this behavior may change in a future version of image packaging.`

At present, pkg(1), on directory removal, will move unpackaged contents of that directory to \$IMAGE_META/lost+found.

Glossaire

action	Objet installable tel qu'un fichier, un répertoire, un lien ou une dépendance. Les actions sont décrites dans le fichier manifest d'un package. Chaque action est essentiellement constituée de son nom et d'un attribut clé. Ensemble, elles se rapportent à un objet unique suivant un historique des versions.
attribut	Paramètre d'un package ou d'une action.
balise	Représentation des paramètres d'un fichier, dans IPS.
catalogue	Ensemble des packages compris dans un référentiel de packages IPS. Les packages dans un catalogue sont associés à un éditeur spécifique.
clone	Copie exacte. Pour l'installation, un clone peut être une copie exacte d'un système d'exploitation, d'un système de fichiers ou d'un volume. Cette copie est 100 % compatible avec l'original.
contenu	Fichiers de package.
éditeur	Nom de domaine direct identifiant une personne, un groupe de personnes ou une organisation qui publie un ou plusieurs packages.
éditeur préféré	Éditeur de package recherché en premier lorsque plusieurs éditeurs non persistants sont configurés. Si plusieurs éditeurs sont configurés, vous pouvez utiliser la commande <code>pkg</code> pour définir l'éditeur préféré. Vous ne pouvez pas désactiver ou supprimer l'éditeur préféré.
environnement d'initialisation	Instance amorçable d'un environnement Oracle Solaris composée d'un ensemble de points de montage, de systèmes de fichiers, de jeux de données ZFS et éventuellement de zones non globales. Un EI est un ensemble de systèmes de fichiers obligatoires, indispensables au fonctionnement du SE Oracle Solaris. L'EI actif est celui qui est en cours d'initialisation. Un EI inactif est disponible pour l'initialisation lors du prochain redémarrage.
facette	Composant optionnel d'un package tel qu'un environnement linguistique. Les facettes apparaissent comme des repères sur des actions IPS et déterminent si cette action est installable. Si une action comporte des repères de facette, au moins un repère de facette doit correspondre aux critères de sélection pour installer l'action.
fichier manifest de package	Fichier d'informations définissant un package.
identificateur de ressource de gestion des pannes (FMRI)	Identificateur pour chaque package. Le FMRI inclut l'éditeur du package, le nom et la version. La commande <code>pkg</code> utilise des FMRI, ou des parties de FMRI, pour agir sur les packages.

image	Emplacement dans lequel des packages peuvent être installés.
image ISO	Ensemble de logiciels qui contient un système d'exploitation complet dans un seul fichier. Une image ISO peut être utilisée pour créer un CD ou DVD d'initialisation à utiliser pour installer le système.
image USB	Ensemble de logiciels qui contient un système d'exploitation complet dans un seul fichier. Une image USB peut être utilisée pour créer un périphérique USB amovible utilisé pour installer le système.
instantané	Image en lecture seule d'un système de fichiers ou d'un EI à un instant t. Un instantané n'est pas amorçable.
jeu de données	Nom générique pour les entités ZFS suivantes : clones, systèmes de fichiers, instantanés ou volumes. Chaque jeu de données est identifié par un nom unique dans l'espace de noms ZFS.
métadonnées	Fichiers manifest et catalogues du package.
miroir	<ol style="list-style-type: none">1. Dans IPS, emplacement d'un référentiel de packages contenant uniquement le contenu du package. Un miroir de référentiel n'inclut pas de métadonnées de package.2. Périphérique virtuel stockant des copies identiques de données sur un ou plusieurs disques, dans ZFS. En cas d'échec d'un disque dans un miroir, tout autre disque du miroir peut fournir les mêmes données.
mise à niveau	<p>Installation qui fusionne des fichiers mis à jour avec des fichiers existants et conserve les modifications dans la mesure du possible.</p> <p>Une mise à niveau du SE Oracle Solaris fusionne la nouvelle version du SE Oracle Solaris avec les fichiers existant sur le ou les disques du système. En règle générale, les modifications que vous avez apportées à la version précédente du système d'exploitation sont conservées.</p>
origin	Emplacement du référentiel incluant à la fois les métadonnées et le contenu des packages. Les clients IPS accèdent au référentiel source pour obtenir un catalogue d'éditeur, même lorsque les clients téléchargent le contenu de package à partir d'un miroir.
package	Ensemble de répertoires, de fichiers, de liens, de pilotes, de dépendances, de groupes, d'utilisateurs et d'informations de licence dans un format défini.
package réadressable	Package qui permet à un utilisateur de spécifier le chemin d'accès à l'installation du package.
pool	Groupe logique de périphériques décrivant la disposition et les caractéristiques physiques du stockage disponible. L'espace pour les jeux de données est alloué à partir d'un pool.
référentiel	Emplacement dans lequel les clients publient et récupèrent des packages. L'emplacement est indiqué par un URI. Un référentiel peut également être qualifié de serveur de dépôt.
restauration	Retour à l'EI exécuté avant une transaction spécifique. Utilisez la fonction de restauration lorsque vous activez un environnement et que l'environnement d'initialisation désigné échoue ou se comporte de manière inattendue. La restauration est appelée traitement de secours (fallback) dans Solaris Live Upgrade.
système de fichiers ZFS	Jeu de données ZFS de type <code>filesystem</code> monté dans l'espace de noms système standard et se comportant comme les autres systèmes de fichiers.

URI (Universal Resource Identifier)	Emplacement d'une machine ou d'une ressource sur Internet.
utilitaire beadm	Interface utilisateur pour la gestion des environnements d'initialisation dans le SE Oracle Solaris. La commande beadm remplace les commandes de Solaris Live Upgrade.
variante	Composant incompatible d'un package tel qu'une architecture. Les variantes apparaissent comme des repères sur des actions IPS et déterminent si cette action est installable. Si une action comporte des repères de variante, tous les repères de variante doivent correspondre aux critères de sélection pour installer l'action.

