

## Ajout et mise à jour de logiciels dans Oracle® Solaris 11.2

ORACLE®

Référence: E53744  
Juillet 2014

Copyright © 2007, 2014, Oracle et/ou ses affiliés. Tous droits réservés.

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 expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, 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 livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

#### U.S. GOVERNMENT END USERS:

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

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 un risque de 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 des applications dangereuses.

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

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

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

# Table des matières

---

<b>Utilisation de la présente documentation .....</b>	<b>9</b>
<b>1 Présentation d'Image Packaging System .....</b>	<b>11</b>
Image Packaging System .....	11
Concepts d'IPS .....	12
Packages IPS .....	12
Identificateurs de ressource de gestion des pannes .....	14
Editeurs, référentiels et archives de packages .....	16
Sources et miroirs du référentiel .....	16
Images et environnements d'initialisation .....	17
Facettes et variantes de package .....	18
Privilèges d'installation .....	18
<b>2 Obtention d'informations sur les packages logiciels .....</b>	<b>21</b>
Affichage des informations d'état d'installation du package .....	21
Packages installés .....	22
Packages pour installation .....	22
Packages récents .....	22
Mise à jour des packages pour lesquels des mises à jour sont disponibles .....	23
Tous les Packages disponibles .....	23
Packages renommés et obsolètes .....	24
Packages figés sur une version spécifique .....	25
Affichage des descriptions ou des licences de package .....	25
Affichage de la description du package, Taille, FMRI complet .....	26
Affichage de licences de package .....	26
Affichage des informations à partir du fichier manifeste du package .....	27
Etablissement de la liste des fichiers installés par un package .....	27
Attributs de l'affichage des fichiers installés par un package .....	28
Affichage des autres attributs et objets du système de fichiers .....	28

Etablissement de la liste de tous les packages installables dans un package de groupe .....	29
Affichage des conditions d'octroi de licence .....	30
Recherche de packages .....	30
La comparaison des commandes <code>pkg search</code> et <code>pkg contents</code> .....	30
Spécification de la requête de recherche .....	31
Identification du package fournissant un fichier spécifique .....	33
Identification du package fournissant un service SMF spécifié .....	33
Etablissement de la liste des packages par classification ou catégorie .....	34
Affichage des packages dépendants .....	34
Etablissement de la liste des packages compris dans un package de groupe .....	35
<b>3 Installation et mise à jour des packages logiciels .....</b>	<b>37</b>
Aperçu d'une opération .....	37
Installation et mise à jour de packages .....	39
Options d'installation communes .....	40
Installation d'un nouveau package .....	44
Installation d'un package dans un nouvel environnement d'initialisation .....	46
Rejet d'un package .....	48
Mise à jour d'un package .....	48
Mise à niveau vers une version antérieure d'un package .....	49
Dépannage des packages installés .....	50
Comparaison des commandes <code>pkg fix</code> et <code>pkg revert</code> .....	50
Vérification des packages et résolution d'erreurs de vérification .....	50
Restauration d'un fichier .....	53
Désinstallation de packages .....	55
Réinstallation d'une image .....	55
Utilisation de zones non globales .....	56
Relation entre la planification des zones globales et non globales .....	57
Le référentiel système et de services proxy .....	58
Mise à jour simultanée de plusieurs zones non globales .....	60
<b>4 Mise à jour ou mise à niveau d'une image Oracle Solaris .....</b>	<b>63</b>
Généralités sur l'image de mise à jour .....	63
Meilleures pratiques de mise à jour d'images .....	64
Vérifiez les versions disponibles .....	64
Afficher un aperçu de l'opération de mise à jour .....	65
Indiquez un nouvel environnement d'initialisation .....	66
Spécification de la version à installer .....	66

Spécification d'une contrainte de version avant la mise à jour .....	67
Installation d'une incorporation personnalisée .....	67
Créer un package d'incorporation personnalisé .....	68
Installer le package de contrôle de mise à niveau .....	71
Mettre à jour le package de contrôle de mise à niveau .....	72
Mettre l'image à niveau .....	73
Retour à la version antérieure d'une image .....	74
<b>5 Configuration des images installées .....</b>	<b>75</b>
Configuration des éditeurs .....	75
Affichage des informations sur les éditeurs .....	76
Ajout, modification ou suppression des éditeurs de packages .....	77
Spécification d'un proxy .....	81
Contrôle de l'installation des composants optionnels .....	83
Influence des valeurs de facettes et des variantes sur l'installation d'un package .....	84
Valeurs de facettes variante et exemple .....	84
Affichage et modification des valeurs de variables .....	85
Affichage et modification des valeurs de facettes .....	86
Verrouillage de packages en une version spécifiée .....	87
Assouplissement des contraintes de version spécifiées par les incorporations .....	89
Spécification d'une implémentation d'application par défaut .....	91
Identification des participants à une médiation .....	91
Modification de l'application préférée .....	92
Annulation de l'installation de certains packages compris dans un package de groupe .....	94
Configuration des propriétés d'image et d'éditeur .....	96
Propriétés d'image de la stratégie d'environnement d'initialisation .....	96
Propriétés de signature de packages .....	98
Propriétés de l'image supplémentaires .....	101
Définitions des propriétés d'image .....	102
Création d'une image .....	103
Viewing Operation History .....	105
<b>A Installation d'un package et mettre à jour Dépannage .....</b>	<b>107</b>
Initiale Troubleshooting Etapes .....	107
Vérifiez la version installée de pkg:/entire .....	108
Vérifiez les origines d'un éditeur de votre serveur de contenu configuré .....	108
Réessayez l'installation .....	111

Impossible d'accéder à l'éditeur ou au référentiel .....	112
Impossible d'accéder au référentiel de packages .....	112
Problème de certificat SSL .....	113
Emplacement introuvable .....	114
Service indisponible. ....	115
Aucune mise à jour disponible .....	116
Les packages ne peuvent pas être installés .....	117
Impossible de répondre aux contraintes .....	117
Mise à jour d'un package contraint par une incorporation .....	118
Mise à jour d'une incorporation quand une dépendance appropriée ne peut pas être trouvée .....	122
Mise à jour d'une incorporation lorsqu'une dépendance installée n'est pas autorisée .....	124
Package requis introuvable .....	125
Package requis rejeté .....	125
Packages non mis à jour comme prévu .....	127
Impossible d'installer le package lié de synchronisation .....	128
Impossible d'installer des zones non globales .....	129
Impossible de modifier l'image .....	130
Les fichiers ont été récupérés .....	130
Minimiser les métadonnées d'images stockées .....	131
Augmenter les performances d'installation d'un package .....	131
<b>B Interfaces graphiques utilisateur IPS .....</b>	<b>133</b>
Utilisation du Gestionnaire de packages .....	133
Options de ligne de commande Gestionnaire de packages .....	134
Utilisation de l'installation Web .....	134
Utilisation du Gestionnaire de mise à jour .....	136
Options le Gestionnaire de mises à jour de ligne de commande .....	138
<b>Index .....</b>	<b>141</b>

## Liste des exemples

---

<b>EXEMPLE 5-1</b>	Indiquez un nouvel éditeur .....	78
<b>EXEMPLE 5-2</b>	Configuration d'importation Publisher .....	78
<b>EXEMPLE 5-3</b>	Spécifiez un certificat et une clé d'éditeur .....	80
<b>EXEMPLE 5-4</b>	Une clé et un certificat révoquer d'un éditeur .....	80
<b>EXEMPLE 5-5</b>	Ajout et suppression de packages dans la liste Avoid .....	94
<b>EXEMPLE A-1</b>	Déverrouillage et mise à jour de l'environnement Java Runtime .....	118
<b>EXEMPLE A-2</b>	Mise à jour de pkg:/entire : quand une dépendance est déverrouillée et mise à jour séparément. ....	122



## Utilisation de la présente documentation

---

- **Présentation** : décrit les fonctionnalités d'installation logicielle de la Oracle Solaris IPS (Image Packaging System) d '. Les commandes IPS et de recherche vous permettent de lister et packages de logiciels, de supprimer un logiciel, installation et effectuez une mise à niveau vers une nouvelle version système d'exploitation Oracle Solaris.
- **Public visé** Ce manuel s'adresse aux administrateurs système qui installent et gèrent des logiciels et gèrent des images système.
- **Connaissances requises** : expérience dans l'administration des systèmes Oracle Solaris.

## Bibliothèque de la documentation des produits

Les informations de dernière minute et les problèmes connus pour ce produit sont inclus dans la bibliothèque de documentation accessible à l'adresse : <http://www.oracle.com/pls/topic/lookup?ctx=E56338>.

## Accès aux services de support Oracle

Les clients Oracle ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

## Commentaires

Faites part de vos commentaires sur cette documentation à l'adresse : <http://www.oracle.com/goto/docfeedback>.



## Présentation d'Image Packaging System

---

Le Oracle Solaris IPS (Image Packaging System) est un cadre qui vous permet d'afficher la liste des packages logiciels, installation et de recherche et supprimer le logiciel, et effectuez une mise à niveau vers une nouvelle version système d'exploitation Oracle Solaris. Les commandes IPS vous permettent de limiter les packages qui peuvent être installés ou les différentes versions de packages qui peuvent être installées.

### Image Packaging System

Les logiciels d'Oracle Solaris 11 sont répartis dans des packages IPS. Les packages IPS sont stockés dans des référentiels de packages IPS qui sont alimentés par des éditeurs IPS. Les packages IPS sont installés dans des images Oracle Solaris 11. Un sous-ensemble de capacités accessibles via l'interface de ligne de commande IPS est disponible par le biais de l'interface utilisateur graphique du Gestionnaire de packages.

Les outils IPS offrent les capacités suivantes. Reportez-vous à la section [“Concepts d'IPS” à la page 12](#) pour obtenir les définitions des termes tels que référentiel et éditeur.

- Lister, rechercher, installer, restreindre l'installation mettre à jour et supprimer des packages de logiciels.
- Lister, ajouter et supprimer des éditeurs de packages. Modifier des attributs d'éditeur comme la priorité de recherche et le caractère résident/non permanent. Définir les propriétés de l'éditeur comme la stratégie de signature.
- Mettre à niveau un système d'exploitation image vers une nouvelle version.
- Crée des copies de référentiels de packages IPS existants. Créer des référentiels de packages. Reportez-vous à la section [“Copie et création de référentiels de packages dans Oracle Solaris 11.2”](#).
- Créer et publier des packages. Voir [“Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2”](#).
- Créer des environnements d'initialisation et d'autres types d'image.

Pour utiliser IPS, vous devez exécutez les OS Oracle Solaris 11. Pour installer le système d'exploitation Oracle Solaris 11, reportez-vous à la section [“Installation des systèmes Oracle Solaris 11.2”](#).

## Concepts d'IPS

Cette section définit les termes et les concepts liés à IPS.

### Packages IPS

Un *package* IPS est défini par un fichier texte appelé *fichier manifeste*. Un fichier manifeste décrit les *actions* de package dans un format défini de paires clé/valeur et éventuellement une charge utile des données. Les actions de package sont les fichiers, répertoires, liens, pilotes, dépendances, groupes, utilisateurs et informations de licence. Les actions de package représentent les objets installables d'un package. Des actions appelées actions set définissent des métadonnées des packages telles que leur classification, leur récapitulatif et leur description.

Vous pouvez rechercher des packages en spécifiant des actions de package et des clés d'action. Reportez-vous à “ [Package Content: Actions](#) ” du manuel “ [Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2](#) ” ou la page de manuel [pkg\(5\)](#) à propos des descriptions des actions sur les packages.

Les packages d'incorporation et les packages de groupe ne distribuent pas de contenu tel que des fichiers. Spécifier des dépendances et packages de groupes d'incorporation qui vous aident à installer des ensembles de packages apparentés.

### Les packages d'incorporation

Un package *d'incorporation* indique les versions d'autres packages pouvant être installées. L'installation d'un package d'incorporation ne permet pas d'installer tous les autres packages. Un package d'incorporation garantit que si vous installez un package de dépendance incorporée du package d'incorporation, seule la version du package dépendant déterminé peut être installé. Par exemple, si un package spécifiée comme une dépendance incorporée dans une incorporation installée de package a la valeur de la version 1.4.3, aucune version du package ne peut être installé disposant d'une version valeur inférieure à 1.4.3 ou supérieure ou égale à 1.4.4. Une version du package portant un numéro de version, 1.4.3.7 par exemple, peut être installée.

Un package d'incorporation spécifie souvent plusieurs dépendances incorporées pour définir une *surface* compatible dans la version du package. Les packages contenant de tels ensembles de dépendances incorporées qui sont souvent appelés *incorporations*. Les incorporations sont généralement utilisées pour définir des ensembles de packages de logiciels qui ne sont pas créés séparément en même temps que l'autre et d'attributs de gestion des versions. La dépendance incorporate d'Oracle Solaris est fortement sollicitée pour vous assurer que des versions compatibles de logiciels sont installées en même temps.

Les packages nommés dépendances incorporé dans les packages d'incorporation peuvent eux-mêmes être le package d'incorporation. De cette manière, de nombreux packages peuvent être altérés par un package d'incorporation même si ces paramètres ne sont pas nommés dans le manifeste du package d'incorporation. Les packages dont l'installation est affectée par un package d'incorporation sont *contraints* par cette incorporation. La mise à jour d'un package d'incorporation B-incorporation incorporée par A-incorporation entraîne la mise à jour d'A-incorporation et de tous les autres packages contraints par A-incorporation.

Les incorporations forcent les packages contraints à être mis à jour de manière synchrone pour conserver une image fonctionnelle. En général, il est conseillé de ne pas installer ou mettre à jour un package contraint par un package d'incorporation. Au lieu de cela, vous devez mettre à jour le package d'incorporation. Un package incorporé pourrait être supprimé, mais si le package est installé ou mis à jour, la version est contrainte. Reportez-vous à la section [“Assouplissement des contraintes de version spécifiées par les incorporations”](#) à la page 89 pour plus d'informations.

Le package `pkg://solaris/entire` est une incorporation spéciale qui spécifie les dépendances incorporé sur de nombreux autres packages d'incorporation pour contraindre les versions de la plupart des logiciels système installés dans l'image.



---

**Attention** - Ne supprimez pas le package nommé `entire`. Le package `entire` restreint les versions des packages du système de manière à ce que le jeu de packages résultant constitue une image prise en charge. La mise à jour correcte du système et la sélection du package approprié dépendent de cette incorporation. Supprimer le package `entire` aurait pour effet de générer un système non pris en charge.

---

## Packages de groupe

Un package de *groupe* spécifie l'ensemble de packages qui constituent une fonction ou un outil. L'installation d'un package de groupe installe tous les packages de dépendance du groupe dans ce package de groupe. Les packages spécifiés comme dépendances de groupe dans un package de groupe ne spécifient pas la version du package. Le profil de groupe est un outil de gestion de contenu, pas un outil de gestion de versions.

Les packages de groupe fournissent les packages cités dans ces dépendances groupe, à moins que les packages concernés ne se trouvent sur la liste "avoid". Reportez-vous à la section [“Annulation de l'installation de certains packages compris dans un package de groupe”](#) à la page 94 pour plus d'informations sur la liste "avoid" d'une image.

Le package `group / feature / storage-server`, par exemple, fournit des pilotes, des services, systèmes de fichiers, composants E / S, bibliothèques et utilitaires liés au stockage, si cela n'est pas déjà fait. Le package `group/system/solaris-minimal-server` fournit l'ensemble de serveur minimum requis pour l'environnement Oracle Solaris pris en charge. Reportez-vous

à la section [“Etablissement de la liste de tous les packages installables dans un package de groupe” à la page 29](#) pour un exemple illustrant la procédure d'établissement de la liste des packages fournis par un package de groupe.

La désinstallation d'un package de groupe n'occasionne pas automatiquement la désinstallation de tous les packages cités dans les dépendances de group. Les packages requis par les autres logiciels toujours installés ne seront pas désinstallés lorsque vous désinstallez le package de groupe.

## Identificateurs de ressource de gestion des pannes

Chaque package est représenté par un FMRI (Fault Management Resource Identifiers, identificateurs de ressources de gestion des pannes). Le FMRI complet pour un package se compose d'un schéma, d'un éditeur, du nom du package, et d'une chaîne de version au format suivant :

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

Le schéma, l'éditeur et la chaîne de version sont facultatifs. Dans les opérandes de commande dans IPS vous pouvez utiliser la plus petite partie du nom du package qui identifie de façon unique le package, et vous pouvez utiliser les caractères de style (3C) ? et \* comme caractères génériques glob pour remplacer un ou plusieurs packages.

L'exemple suivant d'un package FMRI concerne la bibliothèque suri :

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

Scheme

```
pkg
```

Editeur

```
solaris
```

Si l'éditeur est spécifié, le nom de l'éditeur doit être précédé de pkg:// ou //.

Nom de package

```
system/library/storage/suri
```

Les noms des packages ont une structure hiérarchique et sont formés d'un nombre quelconque de composants séparés par des barres obliques (/). Dans les commandes IPS, les composants initiaux d'un nom de package peuvent être omis si le nom du package utilisé dans la commande identifie le package de façon unique. Si vous spécifiez le nom complet du package sans l'éditeur, le nom complet du package peut être précédé de pkg:/

ou /, mais pas de pkg:// ou //. Si vous spécifiez un nom de package abrégé, n'utilisez aucun autre caractère sur la gauche du nom du package.

version

La version du package dispose de quatre parties :

Version du composant : 0.5.11

Pour les composants étroitement liés au système d'exploitation, la version du composant inclut généralement la valeur uname -r de la version concernée du système d'exploitation. Pour un composant possédant son propre cycle de développement, la version du composant correspond à un numéro de version contenant des points, tel que 2.4.10.

Numéro de version : 5.11

Le numéro de version doit suivre une virgule (.). Le numéro de version indique la version du système d'exploitation sur lequel le contenu de ce package a été construit.

Version de branche : 0.175.2.0.0.34.0

La version de branche doit suivre un trait d'union (-). La version de branche fournit des informations spécifiques au fournisseur.

Les packages affichent les informations suivantes dans la partie relative à la version de branche de la chaîne de version d'un package FMRI :

Numéro de version principale : 0.175

Le numéro de version de développement marketing ou principal. Dans cet exemple, 0.175 représente Oracle Solaris 11.

Numéro de version de mise à jour : 2

Le numéro de version de mise à jour pour cette version Oracle Solaris. La valeur de mise à jour est 0 pour le premier exemplaire client d'une version Oracle Solaris 11, pour la première mise à jour de cette version, 2 pour la deuxième mise à jour de cette version et ainsi de suite. Dans cet exemple, 1 représente Oracle Solaris 11.1.

Numéro SRU : 0

Le numéro de mise à jour du référentiel support (SRU) pour cette version de mise à jour. Les SRU incluent uniquement les corrections de bogues ; ils n'incluent pas de nouvelles fonctions. Le référentiel support d'Oracle n'est disponible que pour les systèmes pour lesquels un contrat de support a été souscrit.

Réservé : 0

Ce champ n'est pas actuellement utilisé pour les packages Oracle Solaris.

Numéro de version SRU : 34

Le numéro de version du SRU ou le numéro de remaniement de la version principale.

Numéro de version de développement : 0

Le numéro de version des différentes versions de développement.

Horodatage : 20140303T145535Z

L'horodatage doit suivre un signe deux-points (:). L'horodatage est l'heure à laquelle le package a été publié au format de base ISO-8601 : AAAAMMJJTHHMMSSZ.

## Editeurs, référentiels et archives de packages

Un *éditeur* identifie une personne ou une organisation qui fournit un ou plusieurs packages. Les éditeurs peuvent distribuer leurs packages par le biais de référentiels de packages ou d'archives de packages. Les éditeurs peuvent être configurés selon un ordre de recherche préféré. Lorsqu'une commande d'installation de package est donnée et que la spécification du package n'inclut pas le nom de l'éditeur, le premier éditeur dans l'ordre de recherche est recherché pour ce package. Si aucune correspondance n'est trouvée pour le modèle de FMRI de package spécifié, le deuxième éditeur dans l'ordre de recherche est parcouru, et ainsi de suite jusqu'à ce que le package soit trouvé ou que tous les éditeurs aient été parcourus.

Un *référentiel* est un emplacement dans lequel les packages sont publiés et à partir duquel les packages sont récupérés. L'emplacement est désigné à l'aide d'un URI. Un *catalogue* est la liste de tous les packages dans un référentiel.

Une *archive de package* est un fichier qui contient les informations de l'éditeur et un ou plusieurs packages fournis par cet éditeur.

## Sources et miroirs du référentiel

Une *source* est un référentiel de package qui contient les *métadonnées* (notamment les catalogues, fichiers manifestes et index de recherche) et le *contenu* de package (fichiers). Si plusieurs origines sont configurées pour un éditeur donné dans une image, le client IPS tente de choisir la meilleure origine à partir de laquelle récupérer les données des packages.

Un *miroir* est un référentiel de packages qui contient uniquement le contenu du package. Les clients qui installent et mettent à jour des packages à partir d'un référentiel miroir doivent télécharger les métadonnées à partir d'un référentiel d'origine. Les clients IPS accèdent à l'origine pour obtenir le catalogue d'un éditeur, même lorsque les clients téléchargent le contenu des packages à partir d'un miroir. Si un miroir est configuré pour un éditeur, le client IPS préfère le miroir pour la récupération du contenu du package. Si plusieurs miroirs sont configurés pour un éditeur donné dans une image, le client IPS tente de choisir le meilleur miroir à partir duquel récupérer le contenu du package. Si tous les miroirs sont inaccessibles, ne disposent pas du

contenu requis ou sont plus lents, le client IPS récupère le contenu à partir d'une origine. Pour les éditeurs et des référentiels, reportez-vous à la page de manuel [pkg\(5\)](#).

---

**Remarque** - Même si un référentiel qui est spécifié sous la forme d'un référentiel miroir est entier, avec à la fois le contenu et des métadonnées, les utilisateurs n'ont pas accès au référentiel de contenu dans le miroir, sauf si la même version du même package existe également dans un référentiel d'origine pour ce même éditeur.

---

## Images et environnements d'initialisation

Une *image* est un emplacement dans lequel des packages IPS peuvent être installés et où d'autres opérations IPS peuvent être effectuées.

Un *environnement d'initialisation* est une instance amorçable d'une image. Vous pouvez gérer plusieurs environnements d'initialisation physiques ou virtuels sur un système, chacun pouvant avoir différentes versions de logiciels BE installés, dans les versions de système d'exploitation différent. Lorsque vous initialisez votre système, vous avez la possibilité de démarrer dans n'importe quel environnement d'initialisation sur votre système. Un nouvel environnement d'initialisation peut être créé automatiquement à la suite des opérations de packages. La création d'un nouvel environnement d'initialisation dépend de la stratégie d'image, telle que décrite dans la section "[Propriétés d'image de la stratégie d'environnement d'initialisation](#)" à la page 96. Vous pouvez également créer explicitement un nouveau BE en spécifiant les options décrite dans la section "[Options d'environnement d'initialisation](#)" à la page 40. Reportez-vous à la section "[Création et administration d'environnements d'initialisation Oracle Solaris 11.2](#)" pour plus d'informations sur l'utilisation de la commande `beadm` de création d'un BE.

Les packages ne peuvent être installés que dans les systèmes de fichiers qui font partie d'un BE. Par exemple, sur une installation par défaut d'Oracle Solaris 11, seuls les jeux de données sous `rpool/ROOT/BENAME/` sont prises en charge pour les opérations sur les packages.

Une zone Oracle Solaris est un autre exemple d'une image. Une *zone* non globale est un environnement de système d'exploitation virtualisé, créé au sein d'une instance unique du système d'exploitation Oracle Solaris appelée la zone globale. La zone globale est l'image parent, et les zones non globales à l'intérieur de cette zone globale sont des images enfant de cette zone globale. Dans la sortie de commande IPS, les zones non globales sont parfois appelées des images liées car elles sont liées à leur image de zone globale parente.

Les commandes IPS exécutées dans une zone globale peuvent avoir une incidence sur les zones non globales, comme le décrit la section "[Utilisation de zones non globales](#)" à la page 56. Les commandes IPS exécutées dans une zone globale n'ont aucune incidence sur les zones kernel (zones marquées `solaris-kz`) ou les zones Oracle Solaris 10 (zones marquées `solaris10`). Dans ce guide, "zone non globale" signifie une zone Oracle Solaris 11 marquée `solaris non globale`. Reportez-vous à "[Présentation d'Oracle Solaris Zones](#)" pour plus d'informations sur les zones.

## Facettes et variantes 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. Un composant facultatif dans IPS est appelé *facette* et un composant mutuellement exclusif est appelé une *variante*.

Les facettes et les variantes sont des propriétés spéciales définies sur l'image. Les balises de facettes et variantes définies sur des actions sont également dans un fichier manifeste. L'interaction des valeurs des balises de facettes et de variantes définies sur une action et des valeurs des facettes et des variantes définies sur l'image déterminent si l'action de package concernée peut être installée. Par exemple, si vous définissez une facette d'environnement linguistique donnée sur `false` dans l'image, toutes les actions de fichiers qui spécifient cette facette ne sont pas installées et les actions de fichiers actuellement installées qui spécifient cette facette sont désinstallées.

La plupart des variantes peuvent avoir des valeurs aléatoires. Les balises de facettes définies sur une action peuvent uniquement avoir la valeur `true`. Les propriétés de facette définies dans l'image peuvent uniquement avoir la valeur `true` ou `false`.

L'algorithme suivant décrit la manière dont les facettes et variantes définies sur une image déterminent si une action donnée est installée.

- Les actions sans balise de facette ou de variante sont toujours installées.
- Les actions avec balises de facettes sont installées, sauf si toutes les facettes ou tous les modèles de facettes correspondant aux balises sont définis sur `false` dans l'image. Si une seule facette est définie sur `true` ou n'est pas explicitement définie (`true` est la valeur par défaut), l'action est installée.
- Les actions avec balises de variantes sont uniquement installées si les valeurs des balises de variantes sont les mêmes que celles définies dans l'image.
- Les actions comportant à la fois des balises de facettes et des balises de variantes sont installées si les facettes et les variantes autorisent l'installation des actions concernées.

Pour afficher ou modifier les valeurs des facettes et des variantes définies sur l'image, reportez-vous à la section [“Contrôle de l'installation des composants optionnels”](#) à la page 83.

## Privilèges d'installation

L'utilisation des commandes abordées au [Chapitre 2, Obtention d'informations sur les packages logiciels](#) ne nécessite aucun privilège spécial. Les packages de tâches telles que l'installation, la mise à niveau des IPS, la configuration de l'éditeur système d'exploitation et propriétés de l'image et nécessitent davantage de privilèges. Utilisez l'une des méthodes suivantes pour

obtenir le privilège dont vous avez besoin pour créer et configurer des référentiels de packages. Reportez-vous à “ [Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2](#) ” pour plus d'informations sur les profils et des rôles, y compris les procédures permettant de déterminer les profils ou rôles qui vous intéressent.

**Profils de droits**

Utilisez la commande `profiles` pour répertorier les profils de droits d'accès qui vous sont affectés. Si vous disposez du profil de droits Software Installation (Installation de logiciels), vous pouvez utiliser les commandes `pkg` et `beadm` pour installer et mettre à jour les packages et gérer les environnements d'initialisation.

**Rôles**

Utilisez la commande `roles` pour répertorier les rôles qui vous sont affectés. Si vous disposez du rôle `root`, vous pouvez utiliser la commande `su` avec le mot de passe `root` pour prendre le rôle `root`.

**Commande sudo**

En fonction de la stratégie de sécurité de votre site, vous pouvez être en mesure d'utiliser la commande `sudo` avec votre mot de passe utilisateur pour exécuter une commande privilégiée.



## Obtention d'informations sur les packages logiciels

---

Ce chapitre décrit les commandes qui vous donnent les types suivants d'informations sur les packages :

- Si le package est installé ou s'il peut être mis à jour.
- La description, la taille et la version du package.
- Quels packages font partie d'un package de groupe.
- Quel package fournit un fichier spécifié.
- Quel package fournit un service SMF spécifié.
- Quels sont les packages dépendants du package spécifié.
- Quels packages font partie d'une catégorie donnée.

Si le contenu des référentiels des éditeurs configurés ont éventuellement changé, mettez à jour votre liste des packages disponibles au début de votre session, afin de vous assurer que vous recevez les informations les plus récentes. Pour mettre à jour votre liste des packages, exécutez la commande `pkg refresh`. Par exemple, si un référentiel a été mis à jour avec de nouveaux packages, ces nouveaux packages risquent de ne pas s'afficher tant que vous n'avez pas exécuté `pkg refresh`.

Pour obtenir la liste complète de toutes les options pour les commandes abordées dans ce chapitre, reportez-vous à la page de manuel [pkg\(1\)](#).

### Affichage des informations d'état d'installation du package

La commande `pkg list` vous indique si un package est installé dans l'image en cours, et si une mise à jour est disponible. Sans options ou opérandes, cette commande `pkg list` répertorie tous les packages installés sur l'image actuelle. Pour affiner les résultats de la recherche, vous devez fournir un ou plusieurs noms de package. Vous pouvez utiliser des caractères génériques dans les noms de package. Placez les caractères génériques entre apostrophes afin que l'argument soit directement transmis à la commande `pkg` et que le shell ne le développe pas.

## Packages installés

La commande `pkg list` affiche une ligne d'informations sur chaque package correspondant, comme indiqué dans l'exemple suivant. Le "i" dans la colonne I indique que ces packages sont installés dans cette image.

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

Le nom de l'éditeur entre parenthèses indique que l'éditeur `isvpub` n'est pas le premier dans l'ordre de recherche des éditeurs dans cette image. Le package `dtrace-toolkit` installé dans cette image est publié par l'éditeur qui est le premier éditeur dans l'ordre de recherche.

## Packages pour installation

Pour afficher la liste des packages installés et les versions les plus récentes des packages qui ne sont pas installés mais pourraient l'être dans cette image, utilisez l'option `-a`.

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

Cette sortie indique que le package `image/nvidia/cg-toolkit` peut être installé dans cette image.

## Packages récents

Pour obtenir la liste de tous les packages correspondants, y compris ceux qui ne peuvent pas être installés dans cette image, utilisez l'option `-n`.

```
$ pkg list -n '*toolkit'  
NAME (PUBLISHER)          VERSION          IFO  
developer/dtrace/toolkit  0.99-0.173.0.0.0.1.0  --r  
image/nvidia/cg-toolkit   3.0.15-0.175.2.0.0.17.0 ---  
isvtoolkit (isvpub)       1.0             i--  
system/dtrace/dtrace-toolkit 0.99-0.175.2.0.0.35.0 ---
```

Les packages `developer /dtrace / toolkit` et `system / dtrace / dtrace-toolkit` ne peuvent pas être installés dans cette image. Vous le savez parce que ces packages n'ont pas été

affichés avec l'option `-a`, Le package `developer / dtrace / toolkit` a été renommé. Reportez-vous à [“Packages renommés et obsolètes” à la page 24](#) pour plus d'informations.

Une version plus récente du package `dtrace-toolkit` que celle qui est actuellement installée dans cette image existe à partir des éditeurs configurés. La version plus récente peut être installée si vous mettez également à jour les autres packages dont le package `dtrace-toolkit` est une dépendance, comme indiqué dans la ligne "Raison" dans l'exemple suivant. Les horodatages ont été masqués à partir de la sortie de la commande suivante dans un souci de concision. Les commandes `pkg update` et `pkg install` sont abordées dans [Chapitre 3, Installation et mise à jour des packages logiciels](#).

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

## Mise à jour des packages pour lesquels des mises à jour sont disponibles

L'option `-u` permet d'afficher la liste de tous les packages installés qui ont de nouvelles versions disponibles. Le nombre des packages pour lesquels il existe des versions plus récentes peut être plus grand que le nombre de packages pouvant être mis à jour dans cette image, comme indiqué dans [“Packages récents” à la page 22](#). Les packages peuvent uniquement être mis à jour avec les versions autorisées en raison des contraintes imposées par les dépendances de package installées et par la configuration de l'éditeur.

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

## Tous les Packages disponibles

Pour obtenir la liste de tous les packages correspondants, y compris ceux qui ne peuvent pas être installés dans cette image, utilisez l'option `-af`. L'option `-f` ne peut pas être utilisée sans l'option `-a`. Il peut être utile d'indiquer une partie de la chaîne de version, ce qui permet de limiter les résultats. La saisie de la chaîne de version spéciale `@latest` présente le même résultat que l'option `-n`.

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

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

## Packages renommés et obsolètes

Dans l'exemple suivant, la lettre "r" dans la colonne O indique que le package a été renommé, et la lettre "o" dans la colonne O indique que ce package est obsolète.

```
$ pkg list -n developer/dtrace/toolkit database/mysql-50 web/amp
NAME (PUBLISHER)          VERSION          IFO
database/mysql-50        5.0.91-0.171    --o
developer/dtrace/toolkit 0.99-0.173.0.0.1.0 --r
web/amp                   0.5.11-0.174.0.0.0.0 --r
```

Aucun de ces packages ne peut être installé. Si vous tentez d'installer un package obsolète, l'installation échoue et vous avertis qu'aucune mise à jour n'est nécessaire pour cette image. Si vous tentez d'installer un package renommé, le système tente d'installer le package portant le nom utilisé pour renommer le package.

Utilisez la commande `pkg info` pour trouver le nouveau nom d'un package renommé. Utilisez l'option `-r` pour interroger les référentiels de packages configurés car le package n'est pas installé. Reportez-vous à la ligne "Renommé", comme indiqué dans l'exemple suivant :

```
$ pkg info -r web/amp
Name: web/amp
Summary:
State: Not installed (Renamed)
Renamed to: group/feature/amp@0.5.11-0.174.0.0.0.0
consolidation/ips/ips-incorporation
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.174.0.0.0.0
Packaging Date: September 21, 2011 07:15:02 PM
Size: 5.45 kB
FMRI: pkg://solaris/web/amp@0.5.11,5.11-0.174.0.0.0.0:20110921T191502Z
```

Si vous tentez d'installer le package `web/amp`, le package `group/feature/amp` est installé à sa place, s'il n'est pas déjà installé et s'il peut être installé dans cette image.

Dans l'exemple suivant, le package "Renommé vers" est déjà installé. par conséquent, le système de création de package signale qu'aucune mise à jour n'est nécessaire.

```
$ pkg info -r developer/dtrace/toolkit
Name: developer/dtrace/toolkit
Summary:
State: Not installed (Renamed)
Renamed to: pkg:/system/dtrace/dtrace-toolkit@0.99,5.11-0.173.0.0.0.0
consolidation/osnet/osnet-incorporation
```

```

    Publisher: solaris
    Version: 0.99
  Build Release: 5.11
    Branch: 0.173.0.0.0.1.0
Packaging Date: August 26, 2011 02:55:51 PM
    Size: 5.45 kB
    FMRI: pkg://solaris/developer/dtrace/
toolkit@0.99,5.11-0.173.0.0.0.1.0:20110826T145551Z
$ pkg list dtrace-toolkit
NAME (PUBLISHER)          VERSION          IFO
system/dtrace/dtrace-toolkit  0.99-0.175.2.0.0.34.0  i--
$ pkg install developer/dtrace/toolkit
No updates necessary for this image.

```

## Packages figés sur une version spécifique

Un "f" dans la colonne F indique que ce package est figé. Si un package est figé, vous pouvez uniquement installer ou mettre à jour vers des packages qui correspondent à la version figée. Reportez-vous à la section [“Verrouillage de packages en une version spécifiée” à la page 87](#) pour plus d'informations sur la manière de figer les packages.

```

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

```

## Affichage des descriptions ou des licences de package

La commande `pkg info` affiche des informations sur un package incluant le nom, l'état d'installation, la version, la date de création de package, la taille du package et le FMRI complet. Sans option ou opérande, la commande `pkg info` affiche des informations sur tous les packages installés dans l'image actuelle. Pour affiner les résultats de la recherche, vous devez fournir un ou plusieurs noms de package. Vous pouvez utiliser des caractères génériques dans les noms de package. Placez les caractères génériques entre apostrophes afin que l'argument soit directement transmis à la commande `pkg` et que le shell ne le développe pas.

Les deux sous-commandes, `info` et `list` montrent le nom du package, l'éditeur et les informations sur la version. 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, renommé ou figé. La commande `pkg list` peut également afficher le récapitulatif du package ainsi que le FMRI complet. La commande `pkg info` affiche le récapitulatif, la description, la catégorie et la taille du package, et vous pouvez afficher séparément les informations sur les licences.

## Affichage de la description du package, Taille, FMRI complet

Vous pouvez utiliser la commande `pkg list -s` pour afficher le récapitulatif du package.

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

La commande `pkg list -v` répertorie affiche le FMRI du package complet.

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

La commande `pkg info` affiche plus d'informations.

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

Comme indiqué dans [“Packages renommés et obsolètes” à la page 24](#), vous pouvez utiliser la commande `pkg info` pour rechercher le nouveau nom d'un package renommé.

## Affichage de licences de package

Utilisez l'option `--license` pour afficher les textes de licence des packages correspondants. Ces informations peuvent être assez conséquentes. Les autres informations affichées par la commande `pkg info` lorsque l'option `--license` n'est pas spécifiée ne sont pas affichées.

```
$ pkg info --license osnet-incorporation
You acknowledge that your use of this Oracle Solaris software product
is subject to (i) the license terms that you accepted when you obtained
the right to use Oracle Solaris software; or (ii) the license terms that
you agreed to when you placed your Oracle Solaris software order with
Oracle; or (iii) the Oracle Solaris software license terms included with
```

the hardware that you acquired from Oracle; or, if (i), (ii) or (iii) are not applicable, then, (iv) the OTN License Agreement for Oracle Solaris (which you acknowledge you have read and agree to) available at <http://www.oracle.com/technetwork/licenses/solaris-cluster-express-license-167852.html>.  
Note: Software downloaded for trial use or downloaded as replacement media may not be used to update any unsupported software.

## Affichage des informations à partir du fichier manifeste du package

La commande `pkg contents` affiche le contenu de système de fichiers des packages. Sans options ou opérandes, cette commande affiche les informations sur le chemin d'accès pour tous les packages installés sur l'image actuelle. Utilisez les options de commande pour indiquer le contenu d'un package particulier à afficher. Pour affiner les résultats de la recherche, vous devez fournir un ou plusieurs noms de package. Vous pouvez utiliser des caractères génériques dans les noms de package. Placez les caractères génériques entre apostrophes afin que l'argument soit directement transmis à la commande `pkg` et que le shell ne le développe pas.

Les sous-commandes `contents` et `search` interrogent le contenu des packages. La commande `pkg contents` affiche les actions et les attributs des packages. La commande `pkg search` répertorie les packages qui correspondent à la requête.

## Etablissement de la liste des fichiers installés par un package

L'exemple suivant illustre le comportement par défaut de la commande `pkg contents` : Afficher la valeur de l'attribut `path` pour chaque objet de système de fichiers pouvant être installé dans cette image.

```
$ pkg contents entire
pkg: This package delivers no filesystem content, but may contain metadata. Use
the -o option to specify fields other than 'path', or use the -m option to show
the raw package manifests.
$ pkg contents zip
PATH
usr/bin/zip
usr/bin/zipcloak
usr/bin/zipnote
usr/bin/zipsplit
usr/share/man/man1/zip.1
usr/share/man/man1/zipcloak.1
usr/share/man/man1/zipnote.1
usr/share/man/man1/zipsplit.1
```

La commande `pkg contents` affiche uniquement le contenu pouvant être installé dans cette image. Si vous affichez le manifeste du package (à l'aide de l'option `-m`), vous constatez que le package `zip` a 12 actions de fichier. Les quatre fichiers ne figurant pas dans la sortie ci-dessus sont des fichiers qui ne peuvent pas être installés dans cette image. Cette image est une architecture x86. Les fichiers qui ne sont pas indiqués sont les quatre fichiers `/usr/bin` de configuration de l'architecture SPARC. Pour plus d'informations sur les variantes et les facettes, reportez-vous à la section [“Contrôle de l'installation des composants optionnels” à la page 83](#).

## Attributs de l'affichage des fichiers installés par un package

Pour répertorier plus d'informations sur les fichiers d'installation de celui que le chemin d'accès, utilisez les options `-t` et `-o` de la commande `pkg contents`.

L'option `-t` spécifie le type d'actions à afficher. Vous pouvez spécifier plusieurs types dans une liste de valeurs séparées par une virgule, ou vous pouvez spécifier l'option `-t` plusieurs fois.

L'option `-o` indique les attributs des actions à afficher. Vous pouvez spécifier plusieurs attributs dans une liste de valeurs séparées par une virgule, ou vous pouvez spécifier l'option `-o` plusieurs fois. Pour obtenir une liste des actions et attributs de packages, reportez-vous à la page de manuel [pkg\(5\)](#).

```
$ pkg contents -t file -o owner,group,mode,pkg.size,path zip
OWNER GROUP MODE PKG.SIZE PATH
root bin 0555 231260 usr/bin/zip
root bin 0555 110852 usr/bin/zipcloak
root bin 0555 104960 usr/bin/zipnote
root bin 0555 109340 usr/bin/zipsplit
root bin 0444 86192 usr/share/man/man1/zip.1
root bin 0444 2705 usr/share/man/man1/zipcloak.1
root bin 0444 2396 usr/share/man/man1/zipnote.1
root bin 0444 1837 usr/share/man/man1/zipsplit.1
```

Par défaut, la sortie est triée suivant les chemins d'accès ou en fonction du premier attribut spécifié par l'option `-o`. Vous pouvez utiliser l'option `-s` pour spécifier un autre attribut en tant que clé de tri. L'option `-s` peut être spécifiée plusieurs fois.

## Affichage des autres attributs et objets du système de fichiers

L'exemple suivant affiche le chemin d'accès et cible des liens installés par les packages spécifiés. En plus des attributs affichés dans la page de manuel `pkg(5)`, plusieurs pseudo-attributs peuvent

être utilisés. Pour obtenir une liste des pseudo-attributs, reportez-vous à la page de manuel [pkg\(1\)](#).

Dans l'exemple suivant, le pseudo-attribut `pkg.name` indique le nom du package qui fournit l'action spécifiée. Dans cet exemple, les deux Python 2.6.8 et Python 2.7.3 sont installés et la commande montre le chemin à utiliser pour accéder à la version spécifique si vous ne souhaitez pas avoir recours au lien `/usr/bin/python`. Reportez-vous également à [“Spécification d'une implémentation d'application par défaut”](#) à la page 91 pour plus d'informations à propos des médiations de plusieurs versions.

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

## Etablissement de la liste de tous les packages installables dans un package de groupe

L'installation d'un système Oracle Solaris fournit plusieurs les packages de groupe. L'interface graphique d'installation Oracle Solaris 11 installe le package de groupe `solaris-desktop`. Le programme d'installation en mode texte et le manifeste AI par défaut dans une installation automatisée installent le package de groupe `solaris-large-server`. Le manifeste d'installation par défaut pour les zones non globales installe le package de groupe `solaris-small-server`. Le package de groupe `solaris-minimal-server` installe l'ensemble de packages minimal pris en charge requis pour exécuter Oracle Solaris.

Vous pouvez utiliser la commande suivante pour afficher l'ensemble de packages inclus dans chaque groupe :

```
$ pkg contents -Hro fmri -t depend -a type=group solaris-minimal-server
network/ping
service/network/ssh
shell/tcsh
shell/zsh
system/network
```

L'option `-t` correspond aux actions `depend` dans le package. L'option `-a` correspond aux actions `depend` de type `group`. L'option `-o` affiche uniquement l'attribut `fmri` de l'action de groupe `depend`. Notez que les packages de groupe ne précisent pas le contenu des packages tels que les fichiers, mais qu'ils spécifient les autres packages faisant partie du groupe. Reportez-vous à la section [“Packages de groupe”](#) à la page 13 pour plus d'informations sur les packages de groupe.

Pour afficher également le récapitulatif de chaque package, utilisez la commande `pkg list -s :`

```
$ pkg list -Has `pkg contents -Hro fmri -t depend -a type=group solaris-minimal-server`
```

network/ping	Ping command
service/network/ssh	Secure Shell protocol server
shell/tcsh	Tenex C-shell (tcsh)
shell/zsh	Z Shell (zsh)
system/network	Core Network Infrastructure

## Affichage des conditions d'octroi de licence

Cet exemple affiche tous les packages d'incorporation qui nécessitent que vous acceptiez la licence de package.

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

Reportez-vous à la section [“Affichage de licences de package” à la page 26](#) au niveau local pour plus d'informations sur la manière d'afficher le texte de la licence.

## Recherche de packages

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

## La comparaison des commandes `pkg search` et `pkg contents`

Tout comme la commande `pkg contents`, la commande `pkg search` examine le contenu des packages. Alors que la commande `pkg contents` renvoie le contenu, la commande `pkg search` renvoie les noms des packages qui correspondent à la requête. Le tableau ci-après comprend certaines des similitudes et différences entre ces deux commandes.

---

<code>pkg contents</code>	■ Examine les packages installés. Utilisez l'option <code>-r</code> pour examiner des packages dans les référentiels associés à tous les éditeurs configurés pour cette image.
---------------------------	--

---

	<ul style="list-style-type: none"> <li>■ Utilisez l'option -g pour spécifier l'URI d'un référentiel à examiner.</li> <li>■ Utilisez l'option -t pour spécifier une action.</li> <li>■ Utilisez l'option -a pour spécifier un attribut et une valeur d'attribut.</li> <li>■ Utilisez l'option -o pour spécifier les colonnes de résultats.</li> <li>■ Utilisez l'option -s pour trier les résultats.</li> </ul>
pkg search	<ul style="list-style-type: none"> <li>■ Les recherches des packages dans les référentiels associés à tous les éditeurs configurés pour cette image. Utilisez l'option -l pour ne rechercher que les packages installés.</li> <li>■ Utilisez l'option -s pour spécifier l'URI du référentiel dans lequel effectuer la recherche.</li> <li>■ Utilisez la requête de recherche pour indiquer une action.</li> <li>■ Utilisez la requête de recherche pour spécifier un attribut et la valeur d'attribut.</li> <li>■ Utilisez l'option -o pour spécifier les colonnes de résultats.</li> </ul>

**Astuce** - En général, la commande `pkg contents` est utilisée pour afficher le contenu d'un package spécifié, et la commande `pkg search` pour afficher les packages qui correspondent à une requête. Si vous savez quel package fournit le contenu qui vous intéresse, utilisez la commande `pkg contents`.

## Spécification de la requête de recherche

Par défaut, la requête de recherche est interprétée comme une série de termes à faire correspondre exactement, en ne tenant pas compte de la casse. Utilisez l'option `-I` pour spécifier une recherche qui respecte la casse.

Vous pouvez utiliser les caractères génériques `?` et `*` dans les termes de la recherche. Vous pouvez utiliser des délimiteurs tels que des apostrophes ou des guillemets pour rechercher des expressions. Veillez à tenir compte du shell utilisé lorsque vous saisissez des caractères génériques, des apostrophes ou des guillemets.

Vous pouvez spécifier plusieurs termes de recherche. Par défaut, plusieurs termes sont joints avec AND. Vous pouvez explicitement joindre deux termes avec OR.

Les requêtes peuvent s'exprimer sous la structure suivante :

*package* : *action* : *index* : *jeton*

<i>package</i>	Le nom du package pour effectuer une recherche ou le motif auquel plusieurs packages peuvent correspondre.
<i>action</i>	Le nom d'une action répertoriés dans la section "Actions" dans la page de manuel <a href="#">pkg(5)</a> .
<i>index</i>	Le nom d'un attribut d' <i>action</i> tel que répertorié dans la section "Actions" de la page de manuel (5) de <a href="#">pkg(5)</a> .

*token* La valeur d'*index* ou le motif auquel peut correspondre à la valeur de l'*index*.

Les champs manquants sont implicitement considérés comme des caractères génériques.

Certains attributs ne peuvent pas faire l'objet de recherches. Par exemple, *mode* est un attribut de l'action *file*, mais *mode* n'est pas une valeur valide pour *index*.

Certaines valeurs *index* sont des valeurs dérivées d'autres attributs. Par exemple, *index* peut être *basename*, qui est le dernier composant de l'attribut *path* (chemin) d'une action *file* ou *dir*. Constituent par exemple des valeurs utiles pour *index* les valeurs *basename* et *path* pour des actions *file* et actions *dir*, le type de dépendance (*require* ou *group* par exemple) pour les actions *depend* et *driver\_name* et *alias* pour les actions *driver* sur les pilotes.

En général, la valeur *token* est comparée à la valeur de l'attribut spécifié par *index*. Par exemple, dans l'action partielle *driver* suivante, *alias* est un nom d'attribut qui peut être spécifié pour *index* et *pci108e* peut être spécifié pour *token*.

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

La syntaxe d'une action *set* est légèrement différente. Les deux attributs d'une action *set* sont *name* et *value*. Dans ce cas, la valeur *index* est la valeur de l'attribut *name* et la valeur *token* est comparée à la valeur de l'attribut *value* correspondant. L'exemple suivant illustre une action d'ensemble partiel pour un package de pilote :

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

L'exemple suivant spécifie l'*set* pour l'*action*, *pkg.summary* pour *index* et *Broadcom* pour *token*. Les spécificateurs de colonnes *search.match* et *pkg.name* sont des pseudo-attributs. Reportez-vous à la page de manuel [pkg\(1\)](#).

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

*pkg.fmri*, *info.classification*, *pkg.description* et *pkg.summary* sont des exemples de valeurs d'attributs *name* de l'action *set* bien définies. Pour plus d'informations, reportez-vous à la rubrique "Set Actions" de la page de manuel [pkg\(5\)](#).

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.

Par défaut, les résultats sont affichés pour toutes les actions correspondantes, ce qui signifie qu'il peut y avoir plusieurs lignes de résultats pour un package. Utilisez l'option *-p* pour que chaque package correspondant ne soit affiché qu'une seule fois.

## Identification du package fournissant un fichier spécifique

Les exemples suivants montrent que la bibliothèque `libpower` provient du package `system/kernel/power`.

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

## Identification du package fournissant un service SMF spécifié

Pour afficher les packages qui fournissent un service SMF particulier, effectuez une recherche en indiquant le nom du service en tant que valeur de l'attribut `org.opensolaris.smf.fmri`.

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

Dans ce cas, chaque attribut a deux valeurs : le nom du service avec le nom d'instance, et le nom du service sans le nom d'instance. L'exemple suivant illustre la manière dont cet attribut est spécifié dans le manifeste du package :

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

L'exemple suivant illustre les mêmes informations avec chaque package figurant qu'une seule fois. Ignorez le caractère deux-points pour qu'il soit interprété comme partie du *token* et non pas comme un autre champ de requête de recherche.

```
$ pkg search -o pkg.name,search.match 'org.opensolaris.smf.fmri:*network/http\:'
```

PKG.NAME	SEARCH.MATCH
web/server/apache-22	svc:/network/http:apache22
web/server/lighttpd-14	svc:/network/http:lighttpd14
web/proxy/privoxy	svc:/network/http:privoxy
web/proxy/squid	svc:/network/http:squid
web/java-servlet/tomcat	svc:/network/http:tomcat6

## Etablissement de la liste des packages par classification ou catégorie

L'exemple suivant identifie tous les packages ayant "Source Code Management" dans la valeur de leur attribut `info.classification`.

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

Dans l'exemple ci-dessous, qui affiche les métadonnées du package correspondantes dans cette recherche, procédez comme suit :

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

Ces informations sont affichées dans la ligne "Catégorie" de la sortie de la commande `pkg info`.

Reportez-vous à [“ Classification Values ”](#) du manuel [“ Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2 ”](#) pour connaître les autres classifications que vous pouvez rechercher.

Vous pouvez également utiliser la commande `pkg list` en devinant un des composants du nom de package, comme dans les exemples suivants :

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

## Affichage des packages dépendants

Ces exemples permettent d'afficher les packages qui présentent des dépendances vis-à-vis du package spécifié.

L'exemple suivant montre les packages qui ont une dépendance de type `require` par rapport au package `system/kernel/power`.

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

La commande `pkg contents` confirme les résultats de la recherche. La sortie requise `action.raw` est un pseudo-attribut qui affiche l'action exactement de la façon dont elle apparaît dans le manifeste de package.

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

L'exemple suivant montre que de nombreux packages ont une dépendance de type `exclude` par rapport à `pkg:/x11/server/xorg@1.14.99` :

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

## Etablissement de la liste des packages compris dans un package de groupe

L'interface graphique d'installation Oracle Solaris 11 installe le package de groupe `solaris-desktop`. Le programme d'installation en mode texte et le manifeste AI par défaut dans une installation automatisée installent le package de groupe `solaris-large-server`. Le manifeste d'installation par défaut pour les zones non globales installe le package de groupe `solaris-small-server`. Le package de groupe `solaris-minimal-server` installe l'ensemble de packages minimal pris en charge requis pour exécuter Oracle Solaris. Vous pouvez utiliser la recherche suivante pour afficher l'ensemble de packages inclus dans chaque groupe

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

Dans cet exemple, `-o pkg.name` renvoie uniquement le nom du package spécifié dans le champ *package* de la requête :

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

L'option `-o fmri` renvoie la valeur de l'attribut `fmri` des packages spécifiés dans le package `solaris-large-server` en tant que dépendances de type `group`.

La commande `pkg searchest` renvoyée correctement la valeur de l'attribut d'une action dans un package spécifié. Dans cet exemple, il se trouve que cette valeur d'attribut est un nom de package. Le nombre de résultats obtenu à l'aide de cette commande peut être plus important que celui obtenu avec la commande `pkg contents` montrée dans [“Etablissement de la liste de tous les packages installables dans un package de groupe”](#) à la page 29 car les résultats incluent les noms de tous les packages spécifiés dans les actions de type `groupdepend` dans le package spécifié, pas seulement ceux des packages installables. Par exemple, des variantes de package qui ne sont pas installables dans cette image peuvent être incluses. Pour voir cette différence, essayez les deux exemples avec le package `solaris-large-server`.

## Installation et mise à jour des packages logiciels

---

L'installation et la mise à jour d'un package sont affectées par la configuration de l'image, notamment la contrainte sur certains packages à une version particulière, la configuration de l'ordre de recherche des éditeurs, ainsi que la configuration des propriétés de signature du package. La configuration d'image est traitée dans le [Chapitre 5, Configuration des images installées](#).

Les procédures permettant de déterminer les packages déjà installés, les packages disponibles pour l'installation, et les packages avec des mises à jour disponibles sont traitées dans le [Chapitre 2, Obtention d'informations sur les packages logiciels](#).

Ce chapitre explique comment effectuer les tâches suivantes :

- Exécution d'une installation d'essai pour vérifier que l'installation va aboutir et connaître les éléments qui seraient installés
- Installation, mise à jour et désinstallation des packages
- Validation des packages
- Dépannage des paquets installés
- Restauration d'un fichier installé à son contenu d'origine
- Désinstallation des packages

La section "[Utilisation de zones non globales](#)" à la page 56 présente différents aspects d'opérations sur les packages spécifiques aux zones non globales.

L'installation, la mise à jour et la désinstallation des packages exigent des privilèges accrus. Reportez-vous à la section "[Privilèges d'installation](#)" à la page 18 pour plus d'informations.

Pour obtenir la liste complète de toutes les options pour les commandes abordées dans ce chapitre, reportez-vous à la page de manuel [pkg\(1\)](#).

### Aperçu d'une opération

De nombreuses commandes illustrées dans ce chapitre et dans le [Chapitre 5, Configuration des images installées](#) disposent d'une option `n` qui permet de voir le résultat de l'exécution de la commande sans effectuer aucune modification.

---

**Astuce** - Il est recommandé d'utiliser l'option `-n` à chaque fois qu'elle est disponible. Utilisez l'option `-n` avec une ou plusieurs options détaillées (`nv`, `-nvv`) et examinez les effets de la commande avant de l'exécuter sans l'option `-n`.

---

L'exemple suivant affiche des informations sur l'installation d'un package qui n'est pas réellement effectuée :

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

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

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

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

Cette sortie indique que cette installation fait alors automatiquement l'objet et dans la période comptable en cours dans un nouveau BE, pas une sauvegarde de BE ce rôle devienne le et BE en cours ne sera pas créé. Vous pouvez indiquer des options ou de propriétés de l'image de façon à exiger un BE nouveau ou une sauvegarde de BE. La section « Changed packages » (packages modifiés) indique que le package de groupe `amp` serait installée, et ses cinq des dépendances de groupe, seraient installées. La sortie affiche la version de chaque package serait installé. Le jeton `None` signifie que ces packages ne sont pas actuellement installés et qu'ils ne sont donc pas en cours de mise à jour.

La commande suivante produit une sortie de données de grand volume, car de nombreux packages seraient concernés. Définir cette facette installe tout le contenu traduit pour tous les

packages. Exécutez cette commande d'aperçu peut modifier votre décision de programmer cette opération ou d'ajouter moins de nouveaux environnements locaux. Notez qu'un nouvel environnement d'initialisation ne serait pas créé par défaut, mais qu'un nouvel environnement d'initialisation de secours serait créé.

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

## Installation et mise à jour de packages

Le tableau suivant montre les similarités et différences entre les commandes `pkg install` et `pkg update`.

<code>pkg install</code>	<ul style="list-style-type: none"> <li>■ Requiert un ou plusieurs noms de package en tant qu'opérandes.</li> <li>■ Installe les packages qui ne sont pas actuellement installés.</li> <li>■ Permet de mettre à jour des packages qui sont déjà installés.</li> <li>■ Ne réalise pas une mise à jour inférieure sur les packages. Si vous spécifiez un package installé à une version inférieure, le système n'installe pas ce package.</li> </ul>
<code>pkg update</code>	<ul style="list-style-type: none"> <li>■ Nécessite zéro ou plusieurs noms de packages qui sont déjà installés en tant qu'opérandes.</li> <li>■ Met à jour les packages installés.</li> <li>■ Si aucun nom de package n'est spécifié, ou si '*' est spécifié, tous les packages installés dans l'image sont mis à jour.</li> <li>■ Permet de revenir à la version antérieure packages installés vers la version indiquée dans le FMRI.</li> <li>■ N'installe pas les packages qui ne sont pas déjà installés. Si vous spécifiez un package qui n'est pas déjà installé, le système n'installe pas ce package.</li> </ul>

Reportez-vous aux attributs `preserve` et `overlay` du fichier action dans la page du manuel [pkg\(5\)](#) pour comprendre comment les fichiers dotés de ces attributs seront gérés lors de l'installation et de la mise à jour.

## Options d'installation communes

Cette section présente les options qui sont communes à plusieurs commandes liées aux opérations d'installation. Notez que le paramétrage ou l'annulation des paramètres d'un médiateur, la modification d'une variante ou d'une facette ou le rétablissement d'un fichier peut aussi impliquer l'installation, la mise à jour ou la désinstallation des packages.

## Options d'environnement d'initialisation

Un nouvel environnement d'initialisation ou un environnement d'initialisation de sauvegarde peut être créé lorsque vous installez, mettez à jour ou désinstallez un package. Dans le cadre des contraintes de la stratégie d'image concernant les environnements d'initialisation, vous pouvez contrôler la création d'environnements d'initialisation nouveaux et de sauvegarde en vous servant des options décrites ci-dessous. Reportez-vous à la section [“Propriétés d'image de la stratégie d'environnement d'initialisation” à la page 96](#) pour plus d'informations à propos des nouveaux environnements d'initialisation et des environnements d'initialisation de sauvegarde et sur la procédure de définition de stratégie d'image concernant les environnements d'initialisation.

Utilisez les options d'environnement d'initialisation (BE) pour forcer la création ou non d'un BE nouveau ou de sauvegarde, pour donner un nom personnalisé au BE, et pour spécifier si le nouveau BE ne doit pas être activé. Ces options sont disponibles pour les sous-commandes `install`, `exact-install`, `uninstall`, `update`, `revert`, `set-mediator`, `unset-mediator`, `change-variant` et `change-facet`.

`--no-be-activate`

Si un environnement d'initialisation est créé, ne le définit pas en tant qu'environnement d'initialisation actif lors de la prochaine initialisation.

Dans la sortie de la commande, soyez attentif à tout message indiquant qu'un nouvel environnement d'initialisation a été créé. Si un nouvel environnement d'initialisation a été créé et activé, cet environnement est initialisé par défaut à la réinitialisation suivante si vous ne spécifiez pas l'option `--no-be-activate`.

Vous pouvez utiliser la commande `beadm(1M)` et modifier le BE actif moyennant toute commande `pkg`.

`--no-backup-be`

Ne crée pas d'environnement d'initialisation de sauvegarde.

`--require-backup-be`

Crée un environnement d'initialisation de sauvegarde si un nouvel environnement d'initialisation n'est pas créé. Sans cette option, un environnement d'initialisation de sauvegarde est créé en fonction de la stratégie d'image. Reportez-vous à la section [“Propriétés d'image de la stratégie d'environnement d'initialisation” à la page 96](#) pour

obtenir des informations sur la création automatique des environnements d'initialisation de sauvegarde.

--backup-be-name *name*

Si un environnement d'initialisation de sauvegarde est créé, appelez-le *name* au lieu de lui attribuer le nom par défaut. Utilisation de --backup-be-name nécessite --require-backup-be.

--deny-new-be

Ne crée pas de nouvel environnement d'initialisation. L'installation, la mise à jour, la désinstallation ou le rétablissement ne sont pas effectués si un nouvel environnement d'initialisation est requis.

--require-new-be

Crée un nouvel environnement d'initialisation. Sans cette option, un environnement d'initialisation est créé en fonction de la stratégie d'image. Reportez-vous à la section [“Propriétés d'image de la stratégie d'environnement d'initialisation” à la page 96](#) pour obtenir des informations sur la création automatique des environnements d'initialisation. Cette option ne peut pas être combinée avec --require-backup-be.

--be-name *name*

Si un environnement d'initialisation est créé, appelez-le *name* au lieu de lui attribuer le nom par défaut. L'utilisation de --be-name implique --require-new-be. L'utilisation de cette option est la manière la plus sûre pour effectuer des opérations

## Les options agissant sur les zones non globales

Comme expliqué à la section [“Utilisation de zones non globales” à la page 56](#), seules certaines installations, suppressions et mises à jour de packages effectuées dans la zone globale influent automatiquement sur les zones non globales. L'option -r option effectue la même opération pkg dans les zones non globales que vous avez saisis dans la zone globale, ce qui peut affecter beaucoup plus de packages que si vous n'aviez pas utilisé -r. Ces options sont disponibles pour les sous-commandes install, uninstal, update, change-variant et change-facet.

-r

Exécutez l'opération suivante dans la zone globale et également dans toutes les zones non globales marquées solaris. L'effet sur la zone non globale est similaire à la connexion à chaque zone non globale et de l'exécution de la commande directement.

Si cette option n'est pas activée, lorsque vous exécutez des commandes pkg dans la zone globale, les zones non globales ne sont modifiées que dans la mesure où il s'agit de conservation des mots de passe compatibles avec la zone globale, comme décrit dans [“Utilisation de zones non globales” à la page 56](#). Avec cette option, l'opération pkg est

appliquée à toutes les zones non globales installées, à l'exception de ce qui est limité par les options `-z` et `-Z`. Les zones qui sont exclues par les options `-z` et `-Z` peuvent toujours être modifiées si des mises à jour sont requises pour les maintenir synchronisées avec la zone globale.

*-z zone*

Exécutez l'opération suivante que dans la zone non globale spécifiée. L'option `-z` peut être spécifiée plusieurs fois. L'option `-z` ne peut être utilisée qu'avec l'option `-r`. L'option `-z` ne peut pas être utilisée avec l'option `-Z`.

*-Z zone*

Exécutez l'opération suivante dans toutes les zones non globales à l'exception de la zone spécifiée. L'option `-Z` peut être spécifiée plusieurs fois. L'option `-Z` ne peut être utilisée qu'avec l'option `-r`. L'option `-Z` ne peut pas être utilisée avec l'option `-z`.

L'option suivante indique le nombre de zones non globales pour mettre à jour simultanément avec la zone globale. Cette option est disponible pour les sous-commandes `install`, `exact-install`, `uninstall`, `update`, `change-variant` et `change-facet`.

*-C n*

Mettez à jour tout au plus les zones marquées non globales `n` installées dans `solaris` en parallèle avec la zone globale. Si la valeur `n` est 0 ou un nombre négatif, toutes les zones non globales sont mises à jour parallèlement à la zone globale.

La variable d'environnement `PKG_CONCURRENCY` peut également être défini sur la valeur `n`. L'option `-C` remplace le paramètre `PKG_CONCURRENCY`. Si l'option `-C` est spécifiée, `PKG_CONCURRENCY` n'est pas pris en compte.

## Options d'action d'appel de service

SMF un package peut indiquer des actions en libre-service comme le redémarrage ou de la régénération d'un service spécifié lorsque le package est installé ou mis à jour. Si vous travaillez sur un grand nombre de packages, l'opération `pkg` risque se terminer avant que toutes les des actions en libre-service se terminent. Ensuite, vous ne pourrez peut-être pas utiliser le logiciel qui vient d'être installé car un service associé n'est pas encore disponible.

Pour éviter ce problème, utilisez une des options suivantes à exécuter de manière synchrone les actionneurs SMF avec la commande `pkg`. Ces options sont disponibles pour les sous-commandes `install`, `uninstall`, `update`, `change-variant` et `change-facet`.

*--sync-actuators*

Lorsque vous spécifiez cette option, la commande `pkg` ne s'interrompt pas avant que tous les actionneurs aient terminé dans la zone où `pkg` a été appelée (zone globale ou non globale).

--sync-actuators-timeout *timeout*

Lorsque vous spécifiez cette option, la commande `pkg` ne renvoie pas de résultat jusqu'à ce que tous les actionneurs aient terminée ou lorsque le délai *timeout* est atteint, le plus court étant retenu. Si les mécanismes de positionnement ne se terminent pas dans les secondes suivants le délai de *timeout*, la commande `pkg` se poursuit avec le renvoi du code d'opération et le quitte avec le code de retour 8.

## Options de licence

Vous pouvez être amené à accepter une licence pour pouvoir installer ou mettre à jour un package. Les options suivantes vous permettent de voir et d'accepter les licences requises. Ces options sont disponibles pour les sous-commandes `install`, `exact-install`, `update`, `fix`, `change-variant` et `change-facet`.

--licenses

Utilisez l'option `--licenses` pour afficher toutes les licences pour les packages installés ou mis à jour dans le cadre de cette opération. Licences pour tous les packages sont affichés, et non pas seulement les licences qui doit être acceptée pour activer cette opération pour continuer. Si une licence doit être acceptée pour continuer, de licence s'affiche même si vous ne spécifiez pas l'option `--licenses`. Pour visualiser la licence pour un package sans démarrer une autre opération, utilisez la commande `pkg list` comme illustré dans [“Affichage de licences de package” à la page 26](#). Pour afficher la liste de licences qui doivent être acceptées, utilisez la commande `pkg info` comme illustré dans [“Affichage des conditions d'octroi de licence” à la page 30](#)

--accept

Utilisez l'option `--accept` pour indiquer que vous acceptez les conditions de licence des packages qui sont mis à jour ou installés. Si vous ne spécifiez pas cette option alors que les licences de package exigent l'acceptation, l'opération d'installation échoue.

## D'autres options d'installation

--no-index

Par défaut, les index de recherche sont mis à jour quand vous installez, mettez à jour ou désinstallez des packages. Utilisez l'option `--no-index` pour ne pas mettre à jour les index de recherche une fois l'opération effectuée de ces opérations. La définition de cette option peut vous faire gagner du temps si vous installez un grand nombre de packages. Une fois les opérations d'installation, de mise à jour et de désinstallation terminées, vous pouvez utiliser `pkg refresh` pour mettre à jour la liste des packages disponibles et les métadonnées d'éditeur pour chaque éditeur spécifié. Si aucun éditeur n'est spécifié,

l'actualisation est effectuée pour tous les éditeurs. Cette option est disponible pour les sous-commandes `installexact-installuninstallupdate`.

`--no-refresh`

Lorsque vous spécifiez l'option `--no-refresh`, les référentiels pour les éditeurs de l'image ne sont pas contactés en vue de la récupération de la liste des packages disponibles la plus récente et autres métadonnées. Cette option est disponible pour les sous-commandes `installexact-installupdate`.

## Installation d'un nouveau package

Par défaut, la version la plus récente d'un package compatible avec le reste de l'image est installée à partir du premier éditeur dans l'ordre de recherche des éditeurs proposés par le package. Pour demander explicitement la version la plus récente, utilisez l'option `latest` pour la partie relative à la version du FMRI du package.

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 plus d'un package est spécifié, et si l'un des packages spécifiés ne peut pas être installé dans cette image, alors aucun des packages spécifiés ne sera installé.

Si un package est sur la liste à éviter, son installation supprime ce package de cette liste. Reportez-vous à la section [“Annulation de l'installation de certains packages compris dans un package de groupe” à la page 94](#) pour plus d'informations sur la liste "avoid".

## Identification et indication d'un package installable

Si l'image a plusieurs éditeurs activés, vous pouvez contrôler quel éditeur fournit un package en définissant le caractère résident/non permanent 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 14](#) pour obtenir la description d'un identificateur de ressource de gestion des pannes de package. Reportez-vous à la section [“Configuration des éditeurs” à la page 75](#) pour obtenir des informations sur la définition du caractère résident/non permanent et de l'ordre de recherche des éditeurs.

Si le nom du package n'indique pas l'éditeur, le premier éditeur qui fournit un package correspondant est utilisé en tant que source d'installation. Si cet éditeur ne fournit pas une version du package qui peut être installée sur cette image, l'opération d'installation échoue.

Utilisez la commande `pkg list -a` pour connaître les éditeurs qui fournissent une version du package qui peut être installée dans cette image.

Les commandes suivantes indiquent qu'une version installable du package `atool` est disponible à partir d'un éditeur configuré, mais que le premier éditeur dans l'ordre de recherche propose une version qui ne peut pas être installée dans cette image. Reportez-vous à [“Affichage des informations d'état d'installation du package” à la page 21](#) pour plus d'informations sur les options de la commande `pkg list`.

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

Dans ce cas, la commande d'installation suivante échoue. Le système d'empaquetage trouve une correspondance pour le nom de package `atool` chez le premier éditeur dans l'ordre de recherche, mais ce package ne peut pas être installé.

```
$ pkg install atool
```

Pour installer ce package, définissez de manière plus précise le nom du package, comme indiqué dans les exemples suivants :

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

Utilisez l'option `-nv` pour voir ce qui sera installé avant de procéder à l'installation proprement dite. Si vous recevez un message d'erreur, reportez-vous à [Annexe A, Installation d'un package et mettre à jour Dépannage](#) pour obtenir de l'aide.

## Définition de la source du package

Utilisez l'option `-g` pour ajouter temporairement le référentiel de packages spécifié ou l'archive de packages à la liste des sources dans l'image à partir de laquelle extraire les données de package. Les référentiels qui exigent un certificat SSL client ne peuvent pas être utilisés avec cette option. Cette option ne peut pas être utilisée dans les images ayant des images enfant (zones non globales). Si des zones non globales sont installées dans cette image, utilisez la commande `pkg set-publisher` pour ajouter l'éditeur et l'origine concernés. Cette option peut être spécifiée plusieurs fois.

Lorsque l'option `-g` est indiquée, les éditeurs activés dans l'image sont préférés lors de la récupération des packages.

- Si un package correspondant à ce nom de package spécifié ou si le modèle du nom de package est disponible auprès d'un éditeur activé dans l'image, et si ce même éditeur est

introuvable à l'emplacement spécifié par l'option `-g`, le système d'empaquetage tente d'installer le package à partir de l'éditeur activé dans l'image. Après `install` ou `update`, tout package fourni par les éditeurs n'ayant pas été trouvé dans l'image est ajouté à la configuration de l'image sans point d'origine.

- Si un package correspondant à ce nom de package spécifié ou si le modèle du nom de package est disponible auprès d'un éditeur activé dans l'image, et si ce même éditeur publie ce package à l'emplacement spécifié par l'option `-g`, le système d'empaquetage tente d'installer le package à partir de l'emplacement spécifié par l'option `-g`.

Dans l'exemple suivant, `btool` est disponible auprès de l'éditeur `solaris` configuré dans l'image. Le package `btool` est aussi disponible auprès de l'éditeur `devtool` avec l'origine de référentiel `http://pkg.example1.com/`, mais l'éditeur `devtool` n'est pas configuré dans l'image. La commande suivante tente d'installer le package à partir de l'éditeur `solaris`, car l'éditeur configuré dans l'image est préféré à la source `-g` lorsque le package est disponible auprès de l'éditeur configuré.

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

Pour installer le package à partir de l'éditeur `devtool`, indiquez le nom de l'éditeur dans le nom du package.

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

Dans l'exemple suivant, `isvpub` est un éditeur configuré dans l'image avec l'origine `/export/IPSpkgrepos/isvrepo`. L'éditeur `isvpub` publie également des packages dans un référentiel à l'adresse `http://pkg.example2.com/`, mais cette origine n'est pas indiquée pour l'éditeur configuré dans l'image. La commande suivante tente d'installer le package à partir de l'emplacement `http://pkg.example2.com/`, car le même éditeur fournit le package aux deux emplacements.

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

Reportez-vous également à la description du caractère résident/non permanent d'un éditeur dans la section [“Ajout, modification ou suppression des éditeurs de packages” à la page 77](#)

## Installation d'un package dans un nouvel environnement d'initialisation

---

**Astuce** - Spécifier explicitement un nouvel environnement d'initialisation est la manière la plus sûre pour installer ou mettre à jour. Reportez-vous à la section [“Propriétés d'image de la stratégie d'environnement d'initialisation” à la page 96](#) pour obtenir des informations sur la création des environnements d'initialisation.

---

Le nouvel environnement d'initialisation est un clone de l'environnement d'initialisation actuel avec l'installation, la désinstallation spécifiée ou les modifications de mise à jour appliquées. L'environnement d'initialisation actuel n'est pas modifié. Le système n'est pas automatiquement redémarré. Le nouvel environnement d'initialisation est l'environnement par défaut qui sera utilisé lors de la prochaine réinitialisation du système. L'environnement d'initialisation actuel est toujours disponible pour être l'initialisation.

Si vous spécifiez l'option `-no-be-activate`, le nouvel environnement d'initialisation n'est pas l'environnement utilisé par défaut lors de la prochaine réinitialisation.

Utilisez l'option `--be-name option` pour forcer la à créer un nouveau BE ou pour attribuer au nouveau BE un nom significatif si un nouvel environnement d'initialisation serait créé par défaut.

L'exemple dans [“Aperçu d'une opération” à la page 37](#) a montré qu'un nouveau BE ne serait pas créé par défaut lorsque vous installez le package `group/feature/amp`. Dans la sortie partielle suivante, un nouveau BE sera créé parce que l'option `--be-name` est spécifiée :

```
$ pkg install -v --be-name s11amp group/feature/amp
      Packages to install:      6
      Mediators to change:     1
      Estimated space available: 22.70 GB
Estimated space to be consumed: 751.08 MB
      Create boot environment:   Yes
      Activate boot environment: Yes
Create backup boot environment: No
      Rebuild boot archive:     No
```

Le message suivant s'affiche à la fin de l'opération d'installation, procédez comme suit :

```
A clone of s11 exists and has been updated and activated.
On the next boot the Boot Environment s11amp will be
mounted on '/'. Reboot when ready to switch to this updated BE.
```

La commande `pkg list` indique que le package `group/feature/amp` n'est pas installé parce que le package `group/feature/amp` n'est pas installé dans l'environnement d'initialisation actuel. Le package `group/feature/amp` est installé dans le nouvel environnement d'initialisation `s11amp`.

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

Utilisez la commande `beadm list` pour vérifier que le système est doté d'un nouvel environnement d'initialisation actif nommé `s11amp`. L'environnement d'initialisation "N" est actuellement démarré. L'environnement d'initialisation "R" est l'environnement d'initialisation par défaut après réinitialisation. Utilisez la commande `beadm activate` pour modifier l'environnement d'initialisation par défaut pour la réinitialisation.

```
$ beadm list
BE          Active Mountpoint Space  Policy Created
--          -
s11         N      /              30.92M static 2014-03-05 08:51
```

```
s11amp      R      -      25.75G  static 2014-03-26 10:45
```

Vérifiez que le package `group/feature/amp` est installé dans le nouvel environnement d'initialisation. Monter le nouveau BE et utilisez l'option `-R` à exécuter sur le nouveau BE monté. La lettre "i" dans la colonne I indique que le package `group/feature/amp` est installé.

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

N'oubliez pas de démonter l'environnement d'initialisation `s11amp`.

```
$ beadm unmount s11amp
```

## Rejet d'un package

Utilisez l'option `--reject` de la commande `pkg install` afin d'éviter que les packages spécifiés sont installés. Si des packages de même nom sont déjà installés, ils sont supprimés dans le cadre de cette opération.

Les packages qui sont des dépendances de groupe sont placées sur la liste à éviter. Reportez-vous à la section [“Annulation de l'installation de certains packages compris dans un package de groupe”](#) à la page 94 pour plus d'informations sur la liste "avoid".

L'exemple de commande ci-dessous de installe toutes les dépendances de groupe du package `developer-gnu`, à l'exception du package `cvs` :

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

## Mise à jour d'un package

Vous pouvez utiliser la sous-commande `install` ou `update` pour mettre à jour un package installé vers sa version la plus récente qui est compatible avec le reste de l'image de l'éditeur qui a fourni la version actuellement installée. Afin d'éviter toute installation involontaire d'un package qui n'était pas déjà installé, utilisez la commande `pkg update` pour mettre à jour les packages.

Si l'image a plusieurs éditeurs activés, vous pouvez contrôler quel éditeur fournit un package en définissant le caractère résident/non permanent 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. Pour demander explicitement la version la plus récente d'un package, utilisez le mot clé `latest` pour la partie du nom de package relative à la version. Reportez-vous à [“Identificateurs de ressource de gestion des pannes” à la page 14](#) pour obtenir la description d'un identificateur de ressource de gestion des pannes de package. Reportez-vous à la section [“Configuration des éditeurs” à la page 75](#) pour obtenir des informations sur la définition du caractère résident/non permanent et de l'ordre de recherche des éditeurs.

Tous les fichiers de configuration conservés qui font partie d'un package à mettre à jour sont installés, enregistrés ou renommés en fonction de la valeur de l'attribut `preserve` dans le fichier et du fait que le fichier a changé ou non. Pour plus d'informations sur la manière dont les fichiers sont conservés pendant les mises à jour de package, reportez-vous à l'attribut `preserve` dans la section "File Actions" de la page de manuel [pkg\(5\)](#).

Reportez-vous à [“Installation d'un nouveau package” à la page 44](#) plus d'informations sur la persistance de l'éditeur et l'ordre de recherche ainsi que sur l'utilisation de l'option `-g`.

Si vous tentez de mettre à jour un package qui n'est pas installé actuellement, l'opération `pkg update` quitterait sans mettre à jour les packages. Utilisez l'option `--ignore-missing` pour ne pas prendre en compte ne les packages qui ne sont pas installés et empêchent la mise à jour de `pkg update` si certains packages à mettre à jour ne sont pas installés actuellement.

Reportez-vous à [Chapitre 4, Mise à jour ou mise à niveau d'une image Oracle Solaris](#) pour plus d'informations sur le comportement particulier de la commande `pkg update` lorsqu'aucune valeur FMRI package ou de modèle n'est indiquée ou lorsque le modèle est spécifié à l'aide d'un astérisque (\*).

## Mise à niveau vers une version antérieure d'un package

Vous pouvez utiliser la commande `pkg update` pour mettre à niveau les packages à niveau un inférieur. Pour revenir à la version antérieure d'un package, spécifiez le package FMRI portant un numéro de version antérieure à celle qui est actuellement installé. Reportez-vous à [“Identificateurs de ressource de gestion des pannes” à la page 14](#) pour obtenir la description d'un identificateur de ressource de gestion des pannes de package. Utilisez la commande `pkg list` pour voir quelle version du package est installé et quelles versions sont disponibles à partir des éditeurs configurés.

Tous les fichiers de configuration conservés qui font partie d'un package installé ou à mettre à niveau à une version inférieure sont installés ou renommés en fonction de la valeur de l'attribut

préserve du fichier et selon si le fichier a changé. Pour plus d'informations sur la préservation des fichiers durant le processus de revenir à la version antérieure, reportez-vous à l'attribut `preserve` dans la section « Actions sur les fichiers » de la page de manuel [pkg\(5\)](#).

Reportez-vous à «[Installation d'un nouveau package](#)» à la page 44 pour des informations sur l'utilisation de l'option `-g`.

## Dépannage des packages installés

IPS fournit les opérations pour valider l'installation correcte d'un package installé, pour corriger les problèmes de validation et restaurer les fichiers installés à leur statut dans le package.

### Comparaison des commandes `pkg fix` et `pkg revert`

La commande `pkg fix` et la commande `pkg revert` permettent les deux de réinstaller des composants de packages installés. Le tableau ci-après comprend certaines des similitudes et différences entre ces deux commandes.

<code>pkg fix</code>	<ul style="list-style-type: none"><li>■ Effectue des opérations sur les packages. Accepte un ou plusieurs noms de package ou les modèles qui correspondent au nom d'un package en tant qu'opérandes.</li><li>■ Elle ne fonctionne que sur les packages où <code>pkg verify</code> échoue.</li><li>■ Corrige les éventuelles erreurs signalées par <code>pkg verify</code>. Ne relivre pas d'autre contenu ou de métadonnées à partir du package.</li></ul>
<code>pkg revert</code>	<ul style="list-style-type: none"><li>■ A un impact sur les fichiers. Accepte un ou plusieurs noms de fichiers ou les noms de balise en tant qu'opérandes.</li><li>■ Relivre les fichiers identifiés par les opérandes. Ne relivre pas d'autre contenu ou de métadonnées à partir du package.</li></ul>

### Vérification des packages et résolution d'erreurs de vérification

Utilisez la commande `pkg verify` pour valider l'installation des packages sur l'image actuelle. Si la stratégie de signature en cours pour les éditeurs concernés n'est pas ignorée, les signatures de chaque package sont validées en fonction de cette stratégie. Reportez-vous à «[Propriétés de](#)

[l'image pour les packages signés](#) à la page 98 pour une explication de la manière dont les stratégies de signature sont appliquées. La vérification du contenu des packages installés repose sur une analyse de contenu personnalisée dont les résultats peuvent différer de ceux des autres programmes.

Si vous ne fournissez pas un nom de package, tous les packages installés sont examinés. L'option `-v` fournit des messages d'information, au moins une ligne pour chaque package installé. L'exemple suivant présente uniquement un petit exemple de sortie. L'installation du package `pkg/depot` présente une erreur.

```
$ pkg verify -v
PACKAGE                                     STATUS
pkg://solaris/archiver/gnu-tar              OK
pkg://solaris/audio/audio-utilities         OK
pkg://solaris/benchmark/x1lperf            OK
...
pkg://solaris/package/pkg/depot             ERROR
  dir: var/cache/pkg/depot
      Group: 'pkg5srv (97)' should be 'bin (2)'
  file: var/log/pkg/depot/access_log
      editable file has been changed
  file: var/log/pkg/depot/error_log
      editable file has been changed
...
pkg://solaris/security/sudo                 OK
  file: etc/sudoers
      editable file has been changed
...
pkg://solaris/x11/xlock                      OK
pkg://solaris/x11/xmag                       OK
pkg://solaris/x11/xvidtune                   OK
```

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

La sortie de `pkg verify` indique que des composants du package `sudo` installé sont différents des composants du package, mais ces différences ne sont pas signalés comme des erreurs de validation. La commande `pkg fix` n'apporte pas de modifications. Le fichier `/etc/sudoers` n'est pas remplacé.

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

Si vous supprimez le fichier `/etc/sudoers`, la validation du package échoue, et le fichier est remplacé par `pkg fix`.

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

```
Repairing: pkg://solaris/security/sudo
Creating Plan (Evaluating mediators):
```

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

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

Le fichier manquant est remplacé uniquement, comme indiqué par le fichier téléchargé le fichier et l'exécution d'une seule action (file) a été modifiée. Le reste du contenu du package sudo reste intact. L'opération enregistre un instantané de l'installation en cours avant d'effectuer la réparation. Reportez-vous à la ligne "Instantané ZFS créée" dans la sortie pkg fix. La réparation a été effectuée sur l'image actuelle.

```
$ zfs list -r rpool/ROOT/s11
```

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

La sortie pkg verify affiche une erreur de propriété d'un répertoire dans le package pkg/depot installé. La sortie pkg fix affiche uniquement une erreur dans la section "Vérification". Les autres différences avec les composants en package ne sont pas affichées.

```
$ ls -ld /var/cache/pkg/depot
```

```
drwxr-xr-x 3 pkg5srv pkg5srv 3 Dec 2 19:47 /var/cache/pkg/depot/
```

```
$ pkg fix pkg://solaris/package/pkg/depot
```

```
Verifying: pkg://solaris/package/pkg/depot ERROR
dir: var/cache/pkg/depot
Group: 'pkg5srv (97)' should be 'bin (2)'
```

```
Created ZFS snapshot: 2014-03-13-22:18:52
Repairing: pkg://solaris/package/pkg/depot
Creating Plan (Evaluating mediators):
```

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

La sortie suivante indique que seul l'erreur a été corrigée. Les autres différences entre les composants en package et ceux installés subsistent.

```
$ ls -ld /var/cache/pkg/depot
```

```
drwxr-xr-x 3 pkg5srv bin 3 Dec 2 19:47 /var/cache/pkg/depot/
```

```
$ pkg verify -v pkg://solaris/package/pkg/depot
```

PACKAGE	STATUS
pkg://solaris/package/pkg/depot	OK

```
file: var/log/pkg/depot/access_log
    editable file has been changed
file: var/log/pkg/depot/error_log
    editable file has been changed
```

## Restauration d'un fichier

Utilisez la commande `pkg revert` pour restaurer les fichiers à leur condition lors de la livraison. La propriété des fichiers et les protections sont également restaurées.



**Attention** - Le fait de rétablir les valeurs par défaut de certains fichiers modifiables peut empêcher le système de s'initialiser, ou provoquer d'autres dysfonctionnements.

Utilisez l'option `--require-backup-be` lorsque vous restaurez le statut d'une option fichier clé modifiable.

## Rétablir les fichiers nommés

L'exemple ci-dessous spécifie l'un des deux fichiers installés dans le package `pkg/depot` qui diffère de leurs versions en package.

```
$ pkg revert -v /var/log/pkg/depot/access_log
    Packages to fix:      1
    Estimated space available: 21.08 GB
    Estimated space to be consumed: 460.87 MB
    Create boot environment: No
    Create backup boot environment: No
    Rebuild boot archive: No

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

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

Le fichier indiqué a été remplacé par la version en package. Aucun autre composant du package `pkg.depot` n'a été modifié.

## Rétablir les fichiers et répertoires balisés

Utilisez l'option `--tagged` pour réaliser les opérations suivantes :

- Rétablir tous les fichiers marqués avec le nom de balise.
- Supprimer les fichiers ou répertoires non packagés qui sont sous des répertoires avec le nom de balise spécifié et qui correspondent au motif spécifié.

Reportez-vous à la description de l'attribut `revert-tag` dans “ [File Actions](#) ” du manuel “ [Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2](#) ” et “ [Directory Actions](#) ” du manuel “ [Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2](#) ” pour plus d'informations.

L'exemple suivant illustre certains packages qui contiennent des fichiers de données, accessibles avec le nom de balise `dev-init` :

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

La commande suivante montre un aperçu d'une opération de rétablissement de tous les fichiers qui ont un identifiant de balise `dev-init`. Notez que l'archive d'initialisation serait reconstruite. À l'aide d'une option de créer d'environnement d'initialisation ou un environnement d'initialisation de sauvegarde constitue une bonne pratique.

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

Changed packages:

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

0.5.11,5.11-0.175.2.0.0.34.0:20140303T150218Z

## Désinstallation de packages

Utilisez la commande `pkg uninstall` pour désinstaller les packages existants.

Si vous souhaitez annuler une opération d'installation de package, vous pouvez utiliser la commande `pkg history` pour trouver exactement ce qui a été installé. Reportez-vous à [“Viewing Operation History” à la page 105](#) pour plus d'informations.

Si vous tentez de désinstaller un package qui n'est pas installé, l'opération `pkg uninstall` quitterait sans désinstaller un package. Utilisez l'option `--ignore-missing` pour ne pas prendre en compte les packages qui ne sont pas installés et empêcher l'échec de la commande `pkg uninstall` si certains packages à désinstaller ne sont pas actuellement installés.

Si un package est une dépendance de groupe, sa désinstallation le place dans la liste à éviter. Reportez-vous à la section [“Annulation de l'installation de certains packages compris dans un package de groupe” à la page 94](#) pour plus d'informations sur la liste "avoid".

## Réinstallation d'une image

Si vous savez exactement ce que vous le souhaitez, et que le résultat nécessite un grand nombre de modifications d'emballage comme la désinstallation d'un grand nombre de packages, vous pouvez utiliser la commande `pkg exact-install`. Le résultat de la commande `pkg exact-install` est une image avec uniquement les packages spécifiés et leurs dépendances installés. Les packages qui ne sont pas actuellement spécifiés sur la ligne de commande `pkg exact-install` et qui ne sont pas une dépendance des packages spécifiés sont supprimés.

La commande `pkg exact-install` ignore les restrictions de ne pas installer les packages de la liste à éviter. Si un package est sur la liste à éviter, son installation supprime ce package de cette liste. Reportez-vous à la section [“Annulation de l'installation de certains packages compris dans un package de groupe” à la page 94](#) pour plus d'informations sur la liste "avoid". La commande `pkg exact-install` ne prend pas en compte les restrictions de ne pas mettre à jour les packages qui se trouvent sur la liste figée. Reportez-vous à [“Verrouillage de packages en une version spécifiée” à la page 87](#) pour plus d'informations sur la manière de figer les packages.

Sinon, la sous-commande `exact-install` fonctionne de la même façon que la sous-commande `install`. Les paramètres de la variante d'image et de la facette, de la propriété d'image et

d'éditeur sont conservés. Si un package quelconque ne peut pas être installé dans cette image, alors aucun des packages spécifiés n'est installé. Les zones non globales sont soumises aux éventuelles mises à jour ou suppressions de package résultant, comme le décrit la section [“Utilisation de zones non globales” à la page 56](#). Notez que l'option `-r` n'est pas disponible pour `exact-install`.

Reportez-vous à [“Installation d'un nouveau package” à la page 44](#) plus d'informations sur la persistance de l'éditeur et l'ordre de recherche ainsi que sur l'utilisation de l'option `-g`.

Les pratiques suivantes sont recommandées pour l'utilisation de la commande `pkg exact-install` :

- Vérifiez les versions des packages disponibles à partir des éditeurs configurés à l'aide de `pkg list -a` comme indiqué dans [“Packages pour installation” à la page 22](#). Si vous souhaitez utiliser `exact-install` pour réinstaller les versions actuelles, et des versions plus récentes sont disponibles pour l'installation, vous devez indiquer le FMRI partie relative à la version du package dans la liste des packages à installer.
- Incluez le package d'incorporation `entire` dans la liste des packages à installer.
- Incluez l'un des packages comme le package `solaris-minimal-server` dans la liste des packages à installer.
- Exécutez la commande, d'abord avec les options `-nv` ou `-nvv` pour voir exactement ce qui sera installé et ce qui sera supprimé.
- Utilisez l'option `--be-name` pour procéder à l'installation dans un nouveau BE qui a un nom significatif

L'exemple suivant crée une nouvelle image offrant un minimum d'installation à partir de la même version que l'image actuelle, procédez comme suit :

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

## Utilisation de zones non globales

Vous pouvez utiliser la plupart des commandes IPS dans une zone non globale de la même manière que vous les utilisez dans la zone globale. Reportez-vous à [“Images et environnements d'initialisation” à la page 17](#) pour plus d'informations sur les zones.

En ce qui concerne l'installation d'un package, la zone globale et des zones non globales possèdent une relation parent-enfant comme décrit dans [“Relation entre la planification des zones globales et non globales” à la page 57](#) et [“Mise à jour simultanée de plusieurs zones non globales” à la page 60](#).

Une différence importante entre la zone globale et les zones non globales concerne l'utilisation des éditeurs de packages comme décrit dans [“Le référentiel système et de services proxy” à la page 58](#).

## Relation entre la planification des zones globales et non globales

Les zones non globales installées `solaris` peuvent être affectées par l'installation, la mise à jour et la désinstallation de packages dans la zone globale.

Modifier les facettes et les variantes peut provoquer l'installation et la suppression de packages et affecter les zones non globales.

Pour être mises à jour à partir de la zone globale, les zones non globales n'ont pas besoin d'être initialisées. Les zones non globales doivent simplement être installées pour être affectées par les modifications de package dans la zone globale.

Lorsque vous exécutez l'installation et des commandes de mise à jour dans la zone globale, par défaut, la zone globale et chaque zone non globale installée est mise à jour en série, et les zones non globales sont modifiées que dans la mesure nécessaire pour que la zone non globale concernée reste compatible avec la zone globale.

- Pour effectuer la même opération dans les zones non globales que vous effectuez dans la zone globale au lieu d'effectuer les mises à jour minimales requises uniquement dans les zones non globales, utilisez l'option `-r` comme décrit dans [“Les options agissant sur les zones non globales” à la page 41](#).
- Pour mettre à jour zones non globales en même temps que la zone globale, utilisez l'option `-C` comme décrit dans [“Les options agissant sur les zones non globales” à la page 41](#) et indiqué dans [“Mise à jour simultanée de plusieurs zones non globales” à la page 60](#).

---

**Astuce** - Utilisez les option `-nv` pour passer en revue les modifications qui seront apportées dans les zones non globales ainsi que dans la zone globale.

---

Si vous exécutez des commandes de package lorsque vous êtes connecté à une zone non globale, cette zone non globale est la seule affectée. Les zones non globales peuvent être différents de leurs parents en utilisant l'une des méthodes suivantes de la zone globale, par exemple :

- Différents packages peuvent être installés.
- Différentes versions du même package peuvent être installées si le résultat est compatible avec la zone globale.
- Des packages différents peut se trouver sur la liste "avoid".
- Différents packages et versions peuvent être figés.

- Les médiateurs peuvent être définis de façon à sélectionner différentes implémentations par défaut.
- Différentes facettes peuvent être définies.

Les versions des packages installées dans une zone non globale peuvent être restreintes par les versions installées dans la zone globale. Certains packages ne peuvent pas être mis à jour ou rétrogradés dans une zone non globale car ils doivent présenter le même niveau de version dans la zone non globale et dans la zone globale. Par exemple, le package nommé `entire` doit être identique dans la zone globale et dans chaque zone non globale.

Si un package installé dans une zone non globale contient une dépendance parent, la mise à jour de ce package dans la zone globale entraîne la mise à jour de ce package dans la zone non globale. Les packages qui dépendent des packages qui ont une dépendance parent sont également touchés.

Les packages qui ne sont pas affectées par les dépendances parent peuvent être installés dans une autre version dans une zone non globale que la version installée dans la zone globale. Pour installer une version différente dans la zone non globale, indiquez la version dans la commande `pkg install` ou figez la version à la version souhaitée.

Reportez-vous à [“Impossible d'installer le package lié de synchronisation”](#) à la page 128 et [“Impossible d'installer des zones non globales”](#) à la page 129 pour de l'aide sur les packages des zones non globales.

## Le référentiel système et de services proxy

Dans une zone non globale, le *référentiel système* fournit un accès aux référentiels de packages configurés dans la zone globale. Les modifications apportées à la configuration d'éditeurs dans la zone globale sont immédiatement visibles dans toutes les zones non globales par le biais du référentiel système.

Une origine d'éditeur ou d'un miroir qui est configuré dans une zone non globale doit être accessible à partir de la zone globale, même si cet emplacement n'est pas configuré dans la zone globale liste d'éditeurs. Par exemple, si vous avez configuré l'éditeur `localsw` dans une zone non globale, mais pas dans la zone globale, tous les origines et miroirs de l'éditeur `localsw` doivent rester accessibles à partir de la zone globale.

Le référentiel système fera office de proxy pour les référentiels de fichiers `http`, `https` et `v4` et les référentiels d'archives `.p5p`.

Le proxy de zones est un service qui permet à des commandes `pkg` s'exécutant au sein d'une zone de communiquer avec le référentiel système, qui s'exécute dans la zone globale. Le proxy de zones comprend deux parties. Le service suivant s'exécute dans la zone globale :

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

Le service suivant s'exécute dans la zone non globale :

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

Reportez-vous à la page de manuel [pkg.sysrepo\(1M\)](#) pour plus d'informations sur le référentiel système et sur les services du proxy de zones.

L'exemple suivant montre des éditeurs dans la zone globale :

```
global:~$ pkg publisher
PUBLISHER      TYPE      STATUS P LOCATION
solaris        origin   online F http://pkg.oracle.com/solaris/release/
solaris        origin   online F file:///export/IPSpkgrepos/Solaris/
devtool (disabled) origin   online F http://pkg.example1.com/
isvpub         origin   online F http://pkg.example2.com/
```

L'exemple suivant montre la manière dont ces mêmes éditeurs apparaissent lorsque vous êtes connecté à une zone non globale :

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

Dans la zone non globale, le référentiel système toujours apparaît sous la forme d'un proxy. Il s'agit du proxy que la zone non globale utilise pour communiquer avec le référentiel système dans la zone globale.

Vous ne pouvez pas reconfigurer le référentiel système à partir d'une zone non globale. Par exemple, vous ne pouvez pas modifier l'origine, les propriétés ou l'ordre de recherche d'éditeurs dont l'emplacement est `<system-repository>`. Si un éditeur est ajouté dans la zone globale ou de la reconfiguration, ces modifications sont immédiatement visibles dans les zones non globales. Si un éditeur n'est pas défini dans la zone globale, celui-ci n'est pas défini dans les zones non globales, sauf si la zone non globale possède un package installé à partir de cet éditeur.

---

**Astuce** - Avant d'annuler la définition d'un éditeur dans la zone globale, désinstaller les packages à partir de cet éditeur dans les zones non globales.

---

Si vous ne pouvez pas atteindre un éditeur, vous pouvez définir un proxy dans la zone globale, comme décrit dans [“Spécification d'un proxy”](#) à la page 81. Pour plus d'informations sur

la définition des proxy lorsque vous disposez de zones non globales et des instructions sur le moment et la manière d'utiliser les variables d'environnement `http_proxy` et `https_proxy`, reportez-vous à [“ Configuration du proxy sur un système comportant des zones installées ”](#) du manuel [“ Création et utilisation des zones Oracle Solaris ”](#).

Pour établir la liste des packages d'un éditeur spécifique qui est déjà configuré dans la zone globale, la commande `pkg list` donne le même résultat dans la zone globale et dans les zones non globales :

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

Les référentiels peuvent être réseau ou d'un système de fichiers accessibles à la zone non globale par même si ces référentiels ne sont pas configurés dans la zone globale. La zone non globale doit correspondre à la configuration de l'éditeur de la configuration de l'éditeur ou la zone globale doit être un surensemble de la zone globale par la configuration de l'éditeur. Par exemple, l'éditeur `localsw` peut être configuré dans une zone non globale avec l'origine `file:///export/IPSpkgrepos/localrepo`, car cet emplacement est accessible dans la zone globale même si l'éditeur `localsw` n'est pas configuré dans la zone globale.

## Mise à jour simultanée de plusieurs zones non globales

Par défaut, lorsque vous utilisez la commande `pkg update` dans la zone globale, le système d'empaquetage effectue une mise à jour en série de la zone globale et des zones non globales. Pour mettre à jour simultanément plusieurs zones non globales, utilisez l'option `C` ou définissez la variable d'environnement `PKG_CONCURRENCY` dans la zone globale. Reportez-vous à [“ Les options agissant sur les zones non globales ”](#) à la page 41 pour obtenir plus d'informations.

Dans l'exemple suivant, les zones non globales sont mises à jour en même temps que la zone globale. La sortie fait référence à toutes les zones non globales comme étant lié images car ils sont liés à leur parent d'une image de zone globale.

```
global:~$ pkg update -C 0 --be-name s11.2
Startup: Linked image publisher check ... Done
Startup: Refreshing catalog 'solaris' ... Done
Startup: Refreshing catalog 'isvpub' ... Done
Startup: Checking that pkg(5) is up to date ... Done
Planning: Solver setup ... Done
Planning: Running solver ... Done
Planning: Finding local manifests ... Done
Planning: Package planning ... Done
Planning: Merging actions ... Done
Planning: Checking for conflicting actions ... Done
```

```

Planning: Consolidating action changes ... Done
Planning: Evaluating mediators ... Done
Planning: Planning completed in 39.00 seconds
      Packages to remove: 2
      Packages to install: 1
      Packages to update: 640
      Create boot environment: Yes
      Create backup boot environment: No

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

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

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

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

```



# ◆◆◆ 4 CHAPITRE 4

## Mise à jour ou mise à niveau d'une image Oracle Solaris

---

[Chapitre 3, Installation et mise à jour des packages logiciels](#), discussed installing, updating, fixing, and uninstalling one or a few packages that are named on the command line. Ce chapitre décrit la mise à niveau d'une image à la prochaine Oracle Solaris ou vers le prochain la prise en charge la version mise à jour. La mise à niveau de votre système signifie la mise à jour d'une image amorçable. Un lieu physique ou une machine virtuelle peut avoir plusieurs images amorçables comme expliqué dans [“Images et environnements d'initialisation”](#) à la page 17.

La mise à jour nécessite des privilèges accrus. Reportez-vous à la section [“Privilèges d'installation”](#) à la page 18 pour plus d'informations.

Pour obtenir la liste complète des toutes les options de la commande `pkg update`, reportez-vous à la page de manuel [pkg\(1\)](#).

### Généralités sur l'image de mise à jour

Quand vous exécutez la commande `pkg update` sans spécifier un package FMRI ou un motif, ou en saisissant un astérisque (\*) à la place du motif, tous les packages installés pour lesquels des mises à jour sont disponibles pour la version la plus récente autorisée suivant les contraintes imposées sur le système par les dépendances de package installés et la configuration de l'éditeur seront mis à jour.

- Aucun nouveau package n'est installé, sauf s'il s'agit de nouvelles dépendances requises par un package installé.
- Les packages installés ne sont mis à jour que si un éditeur configuré fournit une version mise à jour répondant aux contraintes de l'image en cours de mise à jour. Les contraintes sont imposées par les dépendances de package et la configuration que vous pouvez contrôler comme décrit dans le [Chapitre 5, Configuration des images installées](#) :
  - Une version particulière packages et l'autre verrouillés avec la
  - Les facettes et des variantes définies sur l'image

- Les propriétés configurées de signature d'un package
- Ordre de recherche et le caractère résident / non permanent Publisher (disponible en anglais uniquement)
- Si le package ne peut pas être installé éventuellement requises, aucun package n'est mis à jour ou installés. Reportez-vous à [Annexe A, Installation d'un package et mettre à jour Dépannage](#).

Si les zones non globales sont configurées dans l'image actuelle, ces zones sont également mises à jour. Reportez-vous à ["Utilisation de zones non globales"](#) à la page 56.

## Meilleures pratiques de mise à jour d'images

Avant de mettre à jour, procédez comme suit :

- Lisez les notes de version.
- Vérifier le package versions disponibles à partir de l'origine d'éditeur configurée. Il peut être nécessaire d'exécuter la commande `pkg refresh` sur l'éditeur ou d'exécuter la commande sur le lieu d'origine `pkgrepo refresh`.
- Utilisez la commande `pkg update` avec l'option `-nv` pour afficher la liste des packages qui seront mis à jour sans effectuer réellement la mise à jour.

Lorsque vous mettez à jour, utilisez l'option `--be-name` ou `--require-new-be` pour rendre actives les modifications dans un nouvel environnement d'initialisation, pas dans l'environnement d'initialisation. Pour plus d'informations, reportez-vous à ["Options d'environnement d'initialisation"](#) à la page 40 et ["Propriétés d'image de la stratégie d'environnement d'initialisation"](#) à la page 96.

## Vérifiez les versions disponibles

Si vous souhaitez mettre à jour votre système d'exploitation, vérifiez les versions disponibles du package d'incorporation `pkg:/entire`. La commande suivante indique que la SRU 10 d'Oracle Solaris 11 11/11 est installée, que les SRU 11, 12 et 13 d'Oracle Solaris 11 11/11 sont disponibles et qu'Oracle Solaris 11.1 est disponible à partir de l'éditeur `solaris` actuellement configuré. Pour plus d'informations sur les champs du FMRI, reportez-vous à la section ["Identificateurs de ressource de gestion des pannes"](#) à la page 14.

```
$ pkg list -af entire
NAME (PUBLISHER)  VERSION                                IFO
entire            0.5.11,5.11-0.175.1.0.0.24.2         ---
entire            0.5.11,5.11-0.175.0.13.0.4.0         ---
entire            0.5.11,5.11-0.175.0.12.0.4.0         ---
entire            0.5.11,5.11-0.175.0.11.0.4.1         ---
entire            0.5.11,5.11-0.175.0.10.0.5.0         i--
```

Si aucune de ces versions ne correspond à la version souhaitée, vous devez définir l'origine de votre éditeur `solaris` sur un emplacement de référentiel de package différent.

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 le caractère résident/non permanent de l'éditeur et l'ordre de recherche. Reportez-vous à la section [“Ajout, modification ou suppression des éditeurs de packages”](#) à la page 77.

## Afficher un aperçu de l'opération de mise à jour

La commande suivante indique, le cas échéant, les packages qui seraient effectivement installés par une mise à jour. L'option `-v` étant spécifiée, cette commande affiche les FMRI complets, y compris les versions, de l'ensemble des 627 packages qui seraient mis à jour, des trois packages qui seraient supprimés et du nouveau package qui serait installé. Cet exemple omet la plus grande partie de la sortie et indique uniquement le package `entire`. Etant donné que l'option `n` est spécifiée, la mise à jour n'est pas réellement effectuée. Vérifiez la sortie avant d'effectuer une mise à jour sans l'option `n`.

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

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

L'exemple qui précède indique que le package d'incorporation `entire` pour Oracle Solaris 11.1 serait installé.

- Tous les packages installés appartenant à l'incorporation `entire` seraient donc mis à jour.
- Etant donné qu'aucun FMRI de package n'a été spécifié, tous les packages installés n'appartenant pas à l'incorporation `entire` seraient également mis à jour. Tous les packages installés seraient mis à jour vers la version la plus récente autorisée par les contraintes imposées au système par les dépendances des packages installés et l'éditeur configuré.
- Il est possible que des packages installés soient supprimés et de nouveaux packages installés si des packages installés mis à jour spécifient de nouvelles dépendances.

## Indiquez un nouvel environnement d'initialisation

L'exemple dans [“Afficher un aperçu de l'opération de mise à jour” à la page 65](#) indique qu'un nouvel environnement d'initialisation serait créé pour cette mise à jour si vous exécutez à nouveau la même commande sans l'option `-n`. Si vous exécutez cette commande sans l'option `n`, le message suivant s'affiche à la fin de la sortie de la mise à jour :

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

L'environnement d'initialisation actuel n'est pas modifié. Toutes les modifications sont apportées dans le nouvel environnement d'initialisation.

Spécifier explicitement un nouvel environnement d'initialisation est la manière la plus sûre pour installer ou mettre à jour. Reportez-vous à la section [“Propriétés d'image de la stratégie d'environnement d'initialisation” à la page 96](#) pour obtenir des informations sur la création des environnements d'initialisation. Vous pouvez utiliser l'option `--be-name` pour donner un nom transparent au nouvel environnement d'initialisation. Le nouvel environnement d'initialisation est activé et est donc initialisé par défaut lors de l'initialisation suivante du système. Si vous ne souhaitez pas que le nouveau BE en tant que valeur par défaut lors de la réinitialisation suivante, utilisez l'option `--no-be-activate` à l'aide de la commande `pkg update`. Vous pouvez modifier le BE d'initialisation par défaut à tout moment à l'aide de la commande `beadm activate`.

Si vous êtes satisfait de votre nouveau, vous pouvez détruire l'ancien BE.

---

**Astuce** - Conserver un BE préliminaire pour chaque version du système d'exploitation. Si nécessaire, vous pouvez revenir à la version la plus ancienne et utilisez-la pour BE mise à jour à une version entre cette version et la prochaine version plus récente que vous avez installée.

---

## Spécification de la version à installer

Si vous ne souhaitez pas mettre à jour vers la dernière version autorisée, vous pouvez préciser le nom du package souhaité avec la commande `pkg update`, en indiquant notamment une partie de la chaîne de version. L'exemple suivant indique comment mettre à jour la version de l'incorporation entière vers la SRU 13 d'Oracle Solaris 11 11/11, même si une version plus récente est autorisée. Assurez-vous d'utiliser à nouveau les options `nv` et vérifiez à nouveau la sortie.

```
$ pkg update -nv entire@0.5.11,5.11-0.175.0.13  
Packages to remove:      2  
Packages to install:     1  
Packages to update:      486  
Estimated space available: 48.39 GB
```

```

Estimated space to be consumed: 2.50 GB
  Create boot environment:      Yes
  Activate boot environment:    Yes
Create backup boot environment: No
  Rebuild boot archive:        Yes

```

```

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

```

Certains packages installés peuvent ne pas être dépendants d'un autre package qui est soumis aux limites de l'incorporation `entire`. Ces packages ne seront pas mis à jour en mettant à jour de l'incorporation `entire` seule. Vous pouvez ajouter ces packages à la même commande `pkg update`.

## Spécification d'une contrainte de version avant la mise à jour

Si vous souhaitez autoriser les mises à jour vers n'importe quelle version d'Oracle Solaris 11.11/11 mais ne pas autoriser de mise à jour vers Oracle Solaris 11.1, vous pouvez figer l'incorporation `entire` comme indiqué dans la commande suivante. L'indication `0.175.0` signifie que le package `entire` peut être mis à jour vers `0.175.0.13` par exemple, mais pas vers `0.175.1`.

```

$ pkg freeze -c "Keep this image at 11.11/11." entire@0.5.11,5.11-0.175.0
entire was frozen at 0.5.11,5.11-0.175.0
$ pkg freeze
NAME      VERSION          DATE                COMMENT
entire    0.5.11,5.11-0.175.0 30 Jan 2013 15:50:01 PST Keep this image at 11.11/11.
$ pkg list entire
NAME (PUBLISHER)  VERSION          IFO
entire            0.5.11,5.11-0.175.0.10.0.5.0 if-

```

Pour plus d'informations sur le blocage de packages, reportez-vous à la section [“Verrouillage de packages en une version spécifiée” à la page 87](#).

## Installation d'une incorporation personnalisée

De la même manière que pour l'utilisation de la commande `pkg freeze`, comme l'illustre la section précédente, vous pouvez utiliser votre propre package d'incorporation personnalisé

pour spécifier des contraintes que vous souhaitez. Reportez - vous à [“Les packages d'incorporation” à la page 12](#) pour plus d'informations sur les packages d'incorporation et leur utilisation dans Oracle Solaris.

Créer le package d'incorporation et installer le package à partir d'un référentiel de packages IPS local ou un fichier d'archive de package. Pour changer les contraintes, modifier et livrer de nouveau le package d'incorporation personnalisé et utiliser `pkg update` pour installer la nouvelle incorporation.

Utiliser une incorporation personnalisée pour contrôler la version du logiciel qu'il est possible d'installer offre un moyen facile de maintenir plusieurs versions différentes d'Oracle Solaris sur différentes machines sans avoir à maintenir plusieurs référentiels de packages. Chaque image peut installer une version différente du package d'incorporation de contrôle de mise à jour personnalisée. Tous les systèmes partagent le même référentiel de packages qui contient toutes les versions de logiciel requises par l'un de ces systèmes.

## Créer un package d'incorporation personnalisé

Les versions des packages de base du système d'exploitation pouvant être installés dans une image sont contrôlées par le package d'incorporation `pkg:/entire`. Pour contrôler les mises à jour du système, créez un package qui spécifie la création d'une version donnée du package `pkg:/entire` en tant que dépendance `incorporate`.

### Créer le fichier de manifeste du package d'incorporation personnalisé

L'exemple suivant montre un manifeste nommé `upgradectl.p5m` pour un package d'incorporation personnalisé qui contrôle la version du package `pkg:/entire` pouvant être installé. Certains des paramètres dans ce manifeste sont décrits ci-dessous.

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

pkg.depend.install-hold
```

Si un utilisateur saisit la commande `pkg update upgradectl`, le package `pkg:/entire` sera automatiquement mis à jour.

`variant.opensolaris.zone`

Ce package peut être installé dans les zones globales et les zones non globales. Reportez-vous également à la description de la dépendance `parent`.

`variant.arch`

Ce package peut être installé sur les deux systèmes x86 et SPARC.

Dépendance `parent`

Ce package peut être installé dans une zone non globale que s'il a déjà été installé dans la zone globale.

dépendance `require`

Le package `upgradectl` peut être installé uniquement si le package `pkg://solaris/entire` est déjà installé ou peut être installé dans cette même opération.

Dépendance `incorporate`

Le package `pkg://solaris/entire` doit être installé avec la version spécifiée. Plusieurs versions peuvent satisfaire à une dépendance `incorporate`, en fonction du nombre de lieux de précision spécifiés. Dans cet exemple `0.175.1.0` indique Oracle Solaris 11.1 SRU 0. Cette mise à niveau de contrôle au package Oracle Solaris 11.1 maintenir leur système qui ne prennent pas en charge les mises à jour. Cette mise à niveau de contrôle autorisera cependant les packages non limités par l'incorporation `pkg:/entire` à mettre à jour.

## Publier le package de contrôle de mise à niveau

Publier le package `upgradectl` vers un référentiel local basé sur des fichiers. Ce référentiel est utilisé pour le développement et le test de ce nouveau package. Si vous créez un référentiel pour une utilisation générale, vous devez inclure des étapes supplémentaires, telles que la création d'un système de fichiers distinct pour le référentiel. Pour plus d'informations sur la création des référentiels de packages pour un usage général, reportez-vous à “ [Copie et création de référentiels de packages dans Oracle Solaris 11.2](#) ”.

Commande Créez un référentiel de développement de package sur votre système. Reportez-vous à la page de manuel [pkgrepo\(1\)](#) pour plus d'informations sur la commande `pkgrepo`.

```
$ pkgrepo create myrepo
```

Définir l'éditeur par défaut pour ce référentiel. L'éditeur par défaut est la valeur de la propriété `publisher/prefix` du référentiel.

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

Publie le package `upgradectl` vers le référentiel de développement.

```
$ pkgsend -s myrepo publish upgradectl.p5m
```

```
pkg://site/upgradectl@1.0,5.11:20131104T072336Z
PUBLISHED
```

Notez que le référentiel éditeur par défaut a été appliquée au FMRI de package.

Examiner le référentiel pour confirmer que le package a été publié.

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

Livrer le package à un référentiel local dans un système de fichiers ZFS séparé dans un emplacement partagé.

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

Vérifiez le package dans le référentiel et la version de pkg:/entire incorporé.

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

Reportez-vous à “ [Creating and Publishing a Package](#) ” du manuel “ [Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2](#) ” pour plus d'informations sur la création de packages IPS, puis en distribuant le contenu.

## Définir l'origine de l'éditeur

Définissez l'origine de l'éditeur de site. Le référentiel système est automatiquement mis à jour avec ces informations de manière à ce que les zones non globales puissent accéder aux packages à partir du site de l'éditeur.

```

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

```

## Installer le package de contrôle de mise à niveau

Installer les packages Dans ce cas, car certaines modifications doivent être faites parce que la version installée de `pkg:/entire` est identique à la version package incorporée par le package de contrôle de mise à niveau. Notez que ce package est également installé dans la zone non globale.

```

$ pkg list -v entire
FMRI                                                    IFO
pkg://solaris/entire@0.5.11,5.11-0.175.1.0.0.24.2:20120919T190135Z  i--
$ zoneadm list
global
z1
$ pkg install upgradectl
      Packages to install: 1
      Create boot environment: No
      Create backup boot environment: No

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

```

Les commandes suivantes indiquent que des versions de `pkg:/entire` plus récentes que la version installée sont disponibles à partir de l'éditeur `solaris` configuré, mais une tentative de mise à niveau est contrôlée par le package de contrôle de mise à niveau de contrôle que vous venez d'installer.

```

$ pkg list -af entire
NAME (PUBLISHER)      VERSION      IFO
entire                0.5.11-0.175.1.13.0.6.0  ---
entire                0.5.11-0.175.1.12.0.5.0  ---
entire                0.5.11-0.175.1.11.0.4.0  ---
entire                0.5.11-0.175.1.10.0.6.0  ---
entire                0.5.11-0.175.1.10.0.5.0  ---

```

```

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

```

## Mettre à jour le package de contrôle de mise à niveau

Lorsque vous êtes prêt à permettre aux utilisateurs de mettre à jour leurs systèmes vers une nouvelle version de, mettez à jour le manifeste `upgradectl.p5m` et publiez et livrez à nouveau le nouveau package de contrôle de mise à jour. Dans l'exemple ci-après, la version du package de contrôle de mise à jour et la version de l'incorporation `entire` sont mises à jour. Pour information pour les utilisateurs, la version du package de contrôle de mise à jour `1.10` est définie pour correspondre à la version mise à jour du package `pkg:/entire`, la mise à niveau est définie de façon à correspondre `1,10`, la version mise à jour du package, `0.175.1.10`

```

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

```

Les commandes suivantes contrôlent et republient le plan. Les intervenants peuvent mettre à jour et livrer de nouveau le package, procédez comme suit :

```

$ pkgsend -s myrepo publish upgradectl.p5m
pkg://site/upgradectl@1.10,5.11:20131120T021902Z
PUBLISHED
$ pkgrepo -s myrepo list
PUBLISHER NAME                                O VERSION
site          upgradectl                      1.10,5.11:20131120T021902Z
site          upgradectl                      1.0,5.11:20131120T010105Z
$ pkgrecv -s myrepo -d /export/IPSpkgrepos/Solaris upgradectl

```

```

Processing packages for publisher site ...
Retrieving and evaluating 1 package(s)...
PROCESS                ITEMS    GET (MB)  SEND (MB)
Completed              1/1      0.0/0.0   0.0/0.0
$ pkg refresh site
$ pkg list -af pkg://site/upgradectl
NAME (PUBLISHER)      VERSION                                IFO
upgradectl (site)    1.10                                  ---
upgradectl (site)    1.0                                    i--

```

## Mettre l'image à niveau

La commande suivante met à jour pkg update tous les packages vers les versions les plus récentes disponibles autorisées car aucun package n'est spécifié. La commande met à jour à la version la plus récente disponible du package avec la mise à niveau de l'image parce que le paramètre pkg.depend.install-hold du package upgradectl occasionne la mise à jour du package pkg:/entire quand le package upgradectl est mis à jour. L'image est mise à niveau vers la version d'incorporation pkg:/entire spécifiée dans la nouvelle incorporation upgradectl.

```

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

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

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

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

```

```
Building new search index                851/851
Executing linked: 0/1 done; 1 working: zone:z1
Executing linked: 1/1 done
```

A clone of s11u1\_0 exists and has been updated and activated.  
On the next boot the Boot Environment s11u1\_10 will be  
mounted on '/'. Reboot when ready to switch to this updated BE.

Vérifiez que le BE en cours n'est pas modifié et que le nouveau BE contient les packages mis à jour.

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

## Retour à la version antérieure d'une image

Pour revenir à une version antérieure de votre système d'exploitation, effectuez une initialisation dans un environnement d'initialisation antérieur à la version à laquelle vous souhaitez revenir, et effectuez la mise à niveau depuis cet environnement. Par exemple, si vous avez mis à jour à partir de la SRU 10 d'Oracle Solaris 11 11/11 vers la SRU 13 d'Oracle Solaris 11 11/11 et que vous vous êtes rendu compte que vous avez besoin d'une image SRU 12, réinitialisez vers votre environnement d'initialisation SRU 10 et mettez à jour vers la SRU 12 depuis cet environnement.

## Configuration des images installées

---

Ce chapitre indique comment configurer les caractéristiques qui s'appliquent à la totalité d'une image, comme la configuration des éditeurs de packages, la restriction des packages pouvant être installés, la définition de la stratégie de signature d'un package et la stratégie de configuration de l'environnement d'initialisation.

- La configuration des éditeurs, y compris la définition de l'ordre de recherche, les clés des origines et des certificats, et les proxys
- Installation des composants obligatoires et facultatifs contrôle en définissant les variantes et facettes
- Verrouillage de packages en une version spécifiée
- S'affranchir de contraintes de version spécifiées par les incorporations
- L'implémentation par défaut pour la spécification d'une application en utilisant une médiation
- Eviter l'installation de certains packages dans un package de groupe
- La configuration, y compris les propriétés d'image et d'éditeur et BE la stratégie de signature d'un package de stratégie de création
- Création d'une image
- Visionnement de l'historique d'une opération de package

Un grand nombre de ces opérations, vous avez besoin des privilèges accrus. Reportez-vous à la section [“Privilèges d'installation” à la page 18](#) pour plus d'informations.

Pour obtenir la liste complète de toutes les options pour les commandes abordées dans ce chapitre, reportez-vous à la page de manuel [pkg\(1\)](#).

### Configuration des éditeurs

Pour l'installation et la mise à jour de logiciels, le pkgdoit être en mesure de contacter un référentiel de packages.

## Affichage des informations sur les éditeurs

Utilisez la commande `pkg publisher` pour afficher des informations sur les éditeurs de packages configurés pour cette image. Les éditeurs sont répertoriés dans l'ordre dans lequel ils sont recherchés pour trouver les packages lorsque l'éditeur n'est pas spécifié dans le package FMRI.

Par défaut, l'éditeur `solaris` est configuré sur un système Oracle Solaris 11 récemment installé. Utilisez la commande `pkg publisher` pour connaître l'origine de votre éditeur.

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

La colonne `TYPE` indique si la valeur `LOCATION` est une origine ou un miroir. Reportez-vous à la section [“Sources et miroirs du référentiel” à la page 16](#) pour obtenir des descriptions.

Entre les colonnes `STATUS` et `LOCATION`, la colonne `P` indique si l'emplacement dispose d'un proxy. Les valeurs de cette colonne sont `true` (T) ou `false` (F). Les référentiels de fichiers ne sont pas pourvus d'un proxy. Les référentiels HTTP ayant la valeur `T` sont pourvus d'un proxy correspondant au proxy qui a été spécifié dans l'option `--proxy` lorsque l'origine a été ajoutée dans `pkg set-publisher`. Lorsque vous spécifiez l'option `-F tsv` pour `pkg publisher`, la colonne `PROXY` affiche le proxy définis pour l'emplacement.

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

Un `F` dans la colonne `P` ou `-` dans la colonne `PROXY` indique que l'emplacement n'a pas été spécifié proxy à l'aide de la commande `pkg set-publisher`. Si l'emplacement dispose d'un proxy en définissant une variable d'environnement `http_proxy`, la sortie `pkg publisher` affiche toujours `F` ou `-`. Voir [“Spécification d'un proxy” à la page 81](#) pour plus d'informations sur les différentes méthodes permettant de définir un proxy.

Spécifiez les éditeurs par nom pour afficher la configuration détaillée de ces éditeurs.

```
$ pkg publisher solaris
  Publisher: solaris
    Alias:
  Origin URI: http://pkg.oracle.com/solaris/release/
    SSL Key: None
    SSL Cert: None
  Client UUID: e15e3228-eada-11df-80ab-8023183d954b
```

```
Catalog Updated: March 4, 2014 11:48:02 PM
  Enabled: Yes
  Properties:
    proxied-urls = []
```

Utilisez l'option `-P` pour afficher uniquement le premier éditeur dans l'ordre de recherche des éditeurs.

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

Utilisez l'option `-n` pour afficher uniquement les éditeurs activés.

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

## Ajout, modification ou suppression des éditeurs de packages

Utilisez la commande `pkg set-publisher` pour effectuer les opérations suivantes :

- Configuration d'un nouvel éditeur
- Définition des origines et des miroirs d'un éditeur
- Définition de la persistance d'un éditeur
- Définition de l'ordre de recherche des éditeurs
- Définition et rétablissement de la propriété d'un éditeur, et ajout et suppression de la valeur de propriété d'un éditeur
- Spécification des certificats et clés SSL pour un éditeur
- Définir un éditeur proxy
- Activation ou désactivation d'un éditeur
- Supprimer un éditeur

La commande `pkg set-publisher` possède deux formes. Reportez-vous à la page de manuel [pkg\(1\)](#).

- Dans une forme, le nom de l'éditeur est obligatoire opérande.
- Dans l'autre forme, un référentiel URI est fourni comme l'argument de l'option `-p`, et les informations de l'éditeur sont obtenues à partir de ce référentiel spécifié. Le nom de l'éditeur est un opérande facultatif vous permettant de configurer uniquement l'éditeur nommé si plusieurs éditeurs publient des packages sur ce référentiel.

## Ajout d'éditeurs

Les exemples ci-après présentent les deux méthodes pour l'ajout d'un éditeur.

### EXEMPLE 5-1 Indiquez un nouvel éditeur

La commande suivante ajoute un nouvel éditeur nommé `devtool` avec un URI d'origine spécifié avec l'option `-g` et définit cet éditeur comme étant le premier dans l'ordre de recherche. Utilisez l'option `-P` ou l'option `--search-first` pour définir l'éditeur spécifié en premier dans l'ordre de recherche.

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

### EXEMPLE 5-2 Configuration d'importation Publisher

Utilisez l'option `-p` afin d'obtenir les informations de configuration d'éditeur auprès de l'URI du référentiel spécifié. Si un éditeur est spécifié, seul l'éditeur correspondant est ajouté ou mis à jour. Si aucun éditeur n'est spécifié, tous les éditeurs sont ajoutés ou mis à jour de façon adéquate.

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

## Origines d'un éditeur l'ajout et modification

Les commandes suivantes montrent l'ajout d'une origine à l'éditeur `solaris`. Si plusieurs origines sont configurées pour un éditeur donné dans une image, le client IPS tente de choisir la meilleure origine à partir de laquelle récupérer les données des packages.

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

Utilisez l'option `-G` pour supprimer un URI servant de miroir à l'éditeur spécifié.

Pour modifier l'URI d'origine d'un éditeur, ajoutez le nouvel URI et supprimez l'ancien.

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

## L'ajout et modification Publisher Mirrors

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 16](#) pour obtenir une explication sur la différence entre une source et un miroir. Vous ne pouvez pas accéder au contenu d'un référentiel miroir, sauf si la même version du même package existe également dans un référentiel d'origine pour ce même éditeur.

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

Utilisez l'option `-M` pour supprimer un URI servant de miroir à l'éditeur spécifié.

Pour modifier l'URI d'origine d'un éditeur, ajoutez le nouvel URI et supprimez l'ancien.

## L'option Affectation de l'ordre de recherche de la définition et Publisher (disponible en anglais uniquement)

Un éditeur récemment ajouté est persistant par défaut. Si un éditeur est non permanent, un package installé à partir de cet éditeur peut être mis à jour à partir d'un autre éditeur. Utilisez les options `--sticky` et `--non-sticky` pour définir la persistance de l'éditeur.

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 permanent. Utilisez les options `--search-before`, `--search-after` et `--search-first` pour modifier la recherche des éditeurs. L'option `-P` est un synonyme pour l'option `--search-first`.

Le premier éditeur qui fournit un package correspondant est utilisé en tant que source de l'installation. Si cet éditeur ne fournit pas une version du package qui peut être installée sur cette image, l'opération d'installation échoue. Pour installer à partir d'un éditeur qui se trouve plus bas dans l'ordre de recherche, fournissez des informations supplémentaires dans le package FMRI, comme par exemple le nom de l'éditeur ou la chaîne de version du package.

## Configuration de propriétés de l'éditeur

Les options suivantes vous permettent de définition et rétablissement et les propriétés de l'éditeur et enlever des valeurs d'pour ajouter les propriétés de l'éditeur, procédez comme suit :

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

Les propriétés `publisher-search-order` et `signature-required-names` peuvent avoir plusieurs valeurs.

Reportez-vous aux exemples `pkg set-publisher` de la section “[Configuration des propriétés de signature de package](#)” à la page 100.

## Configuration des clés et certificats de l'éditeur

**EXEMPLE 5-3** Spécifiez un certificat et une clé d'éditeur

Utilisez l'option `-k` pour spécifier la clé SSL client. Utilisez l'option `-c` pour spécifier le certificat SSL client. 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 commande `pkg publisher` pour cet éditeur. Reportez-vous à la section “[Affichage des informations sur les éditeurs](#)” à la page 76.

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

**EXEMPLE 5-4** Une clé et un certificat révoquer d'un éditeur

Utilisez l'option `--revoke-ca-cert` pour traiter le certificat spécifié comme révoqué. Les hachages des certificats de CA approuvés par l'utilisateur sont répertoriés dans la sortie de la commande `pkg publisher` pour cet éditeur.

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

## La configuration d'un éditeur proxy

Utilisez l'option `--proxy` pour spécifier un URI de proxy persistant à partir duquel récupérer du contenu pour l'origine ou le miroir indiqué. Le proxy a la valeur suivante `protocol://host[:port]`, où `protocol` est `http` ou `https` et `:port` est optionnel. Voir “[Spécification d'un proxy](#)” à la page 81 pour plus d'informations sur les différentes méthodes permettant de définir un proxy.

## Activation et désactivation des éditeurs.

Un éditeur récemment ajouté est activé par défaut. Un éditeur désactivé n'est pas utilisé lorsque la liste du package est renseignée, ou lors des opérations `install`, `uninstall` ou `update`. Il est toujours possible de définir et d'afficher les propriétés d'un éditeur désactivé. Si seul un éditeur est activé, celui-ci ne peut pas être désactivé.

La commande suivante vous permet d'activer l'éditeur `isvpub` et le place devant l'éditeur `devtool` dans l'ordre de recherche.

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

Utilisez l'option `--disable` pour la désactivation d'un éditeur. Il peut être utile de désactiver un éditeur si l'origine de l'éditeur est temporairement inaccessible, par exemple. Si aucun éditeur est n'est accessible, les opérations d'installation et de mise à jour des packages échouent.

## Suppression d'un éditeur

Utilisez la commande `pkg unset-publisher` pour supprimer un éditeur.

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

## Spécification d'un proxy

Les méthodes de définition d'un proxy ont divers effets et avantages. Par exemple, la commande `pkg set-publisher` stocke le proxy en tant que configuration de l'éditeur, tandis que les variables d'environnement `http_proxy` vous permettent de définir des proxy authentifiés.

## Utilisation de la commande `pkg set-publisher` pour définir un proxy

L'option `--proxy` de la commande `pkg set-publisher` définit un URI de proxy persistant pour l'origine d'éditeur spécifié et les URI miroirs. La valeur proxy est stockée en tant que configuration de l'éditeur. Le stockage de la valeur proxy en tant que configuration de l'éditeur met automatiquement à jour le référentiel système qui est utilisée par des images enfant. Le stockage de la valeur proxy en tant que configuration de l'éditeur signifie en outre que vous pouvez utiliser différents proxys pour différents éditeurs.

```
$ pkg publisher
PUBLISHER                                TYPE      STATUS P LOCATION
```

```

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

```

Si cette image a des zones non globales, le système est automatiquement mis à jour avec ce référentiel des informations concernant le serveur proxy ; il n'est pas nécessaire de configurer les propriétés dans le service `system-repository` `system-repository`. Si vous examinez le proxy de l'éditeur dans la zone non globale, vous ne voyez pas le même URI de proxy que celui de la zone globale. Dans la zone globale, l'URI utilise le proxy du référentiel de système. Dans la zone non globale, le référentiel système sert lui-même de proxy, ce qui permet à la zone non globale de communiquer avec le référentiel système dans la zone globale. [“Relation entre la planification des zones globales et non globales” à la page 57](#) indique un exemple de l'apparence de l'éditeur dans la zone non globale.

L'option `--proxy` de la commande `pkg set-publisher` ne peut pas être utilisée pour définir un proxy authentifié. La valeur de l'option `--proxy` ne peut pas avoir la forme suivante : `protocol://user:password@host`.

## Utilisation des variables d'environnement pour définir un proxy

Les valeurs de variable d'environnement proxy s'appliquent à tous les URI de ce protocole. Lors de l'exécution, les valeurs des variables d'environnement `http_proxy` définies à l'aide de

l'option `--proxy` de la commande `pkg set-publisher` sont ignorées. Reportez-vous à la section ENVIRONMENT de la page de manuel `curl(1)` pour plus d'informations sur les variables d'environnement proxy.

Si vous définissez les variables d'environnement `http_proxy` dans une image ayant des zones non globales, définissez les propriétés proxy dans le service SMF `svc:/application/pkg/system-repository` à ces mêmes valeurs et actualisez le service.

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

La commande `pkg publisher` n'affiche pas les proxys qui sont définis en définissant des variables d'environnement ou les propriétés du service SMF.

Si vous modifiez la valeur d'une variable d'environnement `http_proxy`, veillez à mettre à jour les propriétés de service `system-repository` et actualisez le service.

## Contrôle de l'installation des composants optionnels

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*.

Vous pouvez afficher les valeurs des variantes et des facettes qui sont définies sur l'image actuelle et vous pouvez modifier les variantes et facettes dans l'image actuelle. Pour afficher les valeurs actuelles des facettes et des variantes définies sur l'image, exécutez les commandes `pkg facet` et `pkg variant`. Pour modifier les valeurs des facettes et des variantes définies sur l'image, utilisez les commandes `pkg change-facet` et `pkg change-variant`. Reportez-vous à la page de manuel `pkg(1)` et aux exemples ci-dessous. La modification de variantes et de facettes peut entraîner la mise à jour d'un grand nombre de packages et peuvent nécessiter un nouvel environnement d'initialisation. Utilisez `-nv` pour passer en revue les modifications qui seront effectuées avant de les réaliser.

## Influence des valeurs de facettes et des variantes sur l'installation d'un package

Les facettes et les variantes sont des propriétés spéciales définies sur l'image et des balises définies sur des actions au sein d'un package. L'interaction des valeurs des balises de facettes et de variantes définies sur une action et des valeurs des facettes et des variantes définies sur l'image déterminent si l'action de package concernée peut être installée.

Chaque balise de facette et de variante possède un nom et une valeur. Une seule action peut avoir plusieurs balises de facettes et de variantes. Un fichier d'en-tête spécifique à une architecture utilisé par des développeurs ou un composant uniquement destiné à une zone globale SPARC sont des exemples de composants à balises de facettes et de variantes multiples.

La plupart des balises de variantes peuvent avoir plusieurs valeurs. Pour qu'un package utilisant une variante donnée puisse être installé, il faut que cette variante soit définie sur l'image. Les variantes `arch` et `zone` sont définies par le programme qui crée l'image et installe son contenu initial. Les variantes `debug.*` sont définies sur `false` dans l'image par défaut.

Les balises de facettes définies sur une action peuvent uniquement avoir la valeur `true`. Par exemple, si vous définissez sur `false` une facette d'environnement linguistique particulière, tous les fichiers ou les autres actions qui spécifient cette facette ne seront pas installés, et les fichiers actuellement installés qui spécifient cette facette sont désinstallés.

L'algorithme suivant décrit la manière dont les facettes et variantes définies sur une image déterminent si une action donnée est installée.

- Les actions sans balise de facette ou de variante sont toujours installées.
- Les actions avec balises de facettes sont installées, sauf si toutes les facettes ou tous les modèles de facettes correspondant aux balises sont définis sur `false` dans l'image. Si une seule facette est définie sur `true` ou n'est pas explicitement définie (`true` est la valeur par défaut), l'action est installée.
- Les actions avec balises de variantes sont uniquement installées si les valeurs des balises de variantes sont les mêmes que celles définies dans l'image.
- Les actions comportant à la fois des balises de facettes et des balises de variantes sont installées si les facettes et les variantes autorisent l'installation des actions concernées.

## Valeurs de facettes variante et exemple

La plupart des variantes peuvent avoir un nombre de valeurs quelconque. Par exemple, la variante `arch` peut être définie sur `i386`, `sparc`, `ppc`, `arm` ou n'importe quelle architecture prise en charge par la distribution. (Seules les valeurs `i386` et `sparc` sont utilisées dans Oracle Solaris.) Les variantes `debug` constituent une exception. Les variantes `debug` peuvent uniquement être définies sur `true` ou `false` ; les autres valeurs ont un comportement indéterminé. S'il existe pour une action `file` des versions non-débogage et débogage, les deux

versions doivent avoir la variante debug applicable explicitement définie, comme illustré dans l'exemple suivant :

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

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

Les balises suivantes sont couramment utilisées dans Oracle Solaris.

Nom de la variante	Valeurs valides
variant.arch	sparc, i386
variant.opensolaris.zone	global, nonglobal
variant.debug.*	true, false

Les facettes sont booléennes : elles peuvent uniquement être définies sur `true` (activé) ou `false` (désactivé). Par défaut, toutes les facettes sont considérées comme définies sur `true` dans une image. Un repère de facette sur une action doit uniquement avoir la valeur `true` ; les autres valeurs ont des comportements indéterminés.

Une facette définie sur l'image peut être une facette complète telle que `doc.man` ou un modèle tel que `locale.*`. L'intérêt de cette flexibilité est qu'elle vous permet de désactiver une partie de l'espace de noms d'une facette et de n'activer que des facettes individuelles en son sein. Par exemple, vous pouvez désactiver tous les environnements linguistiques, puis n'activer qu'un ou deux environnements linguistiques particuliers, comme illustré dans l'exemple suivant :

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

La liste suivante présente quelques exemples de repères de facettes utilisés dans Oracle Solaris :

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

## Affichage et modification des valeurs de variables

Utilisez la commande `pkg variant` pour afficher les valeurs des variantes définies.

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

L'option -v comprend les valeurs de la variante qui peuvent être définies sur les packages installés.

```
$ pkg variant -v
VARIANT          VALUE
variant.arch     i386
variant.arch     sparc
variant.debug.osnet false
variant.debug.osnet true
variant.opensolaris.zone global
variant.opensolaris.zone nonglobal
```

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

La commande suivante produit une sortie de données de grand volume, car de nombreux packages seraient concernés. Notez qu'un nouvel environnement d'initialisation ne serait pas créé par défaut, mais qu'un nouvel environnement d'initialisation de secours serait créé. Reportez-vous à la section [“Propriétés d'image de la stratégie d'environnement d'initialisation” à la page 96](#) pour obtenir des informations sur la création des environnements d'initialisation. L'option -n montre ce qui changerait si vous exécutiez l'opération sans -n, mais la commande n'apporte pas des modifications réelles.

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

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

## Affichage et modification des valeurs de facettes

Utilisez la commande `pkg facet` pour afficher les valeurs en cours et la source de toutes les facettes d'origine définies au niveau local, à l'aide de la commande `pkg change-facet` ou

héritées d'une image parent. Par exemple, une zone non globale hérite des paramètres à partir de la zone globale facette.

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

Utilisez la commande `pkg change-facet` pour modifier la valeur d'une facette.

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

La commande suivante produit une sortie de données de grand volume, car de nombreux packages seraient concernés. Notez qu'un nouvel environnement d'initialisation ne serait pas créé par défaut, mais qu'un nouvel environnement d'initialisation de secours serait créé. Reportez-vous à la section [“Propriétés d'image de la stratégie d'environnement d'initialisation” à la page 96](#) pour obtenir des informations sur la création des environnements d'initialisation. L'option `-n` montre ce qui changerait si vous exécutiez l'opération sans `-n`, mais la commande n'apporte pas des modifications réelles.

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

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

## Verrouillage de packages en une version spécifiée

Utilisez la commande `pkg freeze` pour contraindre une version de package.

Si aucune version n'est fournie dans l'opérande du package, le package nommé doit être installé, et il est limité à la version installée sur le système. Si la version est fournie dans l'opérande du package, cette contrainte ou ce blocage agit comme si une dépendance `incorporate` avait été installée là où l'attribut `fmri` avait la valeur de la version du package fourni.

Lorsqu'un package figé est installé ou mis à jour, sa version doit correspondre à celle à laquelle il a été figé. Par exemple, si un package a été figé à la version 1.2, il peut être mis à jour vers

1.2.1, 1.2.9, 1.2.0.0.1, et ainsi de suite. Ce package ne peut pas avoir un numéro de version correspondant à 1.3 ou 1.1.

Un éditeur spécifié dans le package opérande est utilisé pour trouver des packages correspondants. Cependant, les informations de l'éditeur ne sont pas enregistrées dans le cadre de l'opération de blocage. Un package est figé par rapport à sa version uniquement, et non son éditeur.

Le blocage d'un package qui est déjà figé remplace la version figée par la nouvelle version spécifiée.

Si aucun package n'est spécifié, les informations suivantes relatives aux packages actuellement figés s'affichent : noms de package, de version figée, date du blocage du package et la raison pour laquelle le package a été figé.

Le fait de figer un package n'empêche pas la suppression du package. Aucun avertissement n'est affiché si le package est supprimé.

Dans l'exemple suivant, le package est figé à la version installée. L'argument d'option `-c` est la cause du blocage du package. La raison est affichée si un blocage empêche l'installation ou la mise à jour de réussir. La lettre "f" qui s'affiche dans la liste de packages indique que le package est figé.

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

Lorsque vous essayez d'installer une version différente du package figé, un message s'affiche rappelant le blocage.

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

Un blocage n'est jamais annulé automatiquement par le système d'empaquetage. Utilisez la commande `pkg unfreeze` permettant de supprimer les contraintes imposées par le blocage des packages spécifiés. Les versions éventuellement fournies sont ignorées.

## Assouplissement des contraintes de version spécifiées par les incorporations

Un package d'incorporation indique les différentes versions de packages qui peuvent être installés. Ces contraintes de version permettent de maintenir le système dans un état fonctionnel au fil des mises à jour. Reportez-vous à [“Les packages d'incorporation” à la page 12](#) pour plus d'informations sur les packages d'incorporation et les contraintes.

La rétrogradation ou la mise à niveau de certains packages incorporés vers une version différente de celle spécifiée par l'incorporation peut ne pas présenter de risque. La contrainte de version est exprimée par une facette `version-lock.package` spécifiée dans le package d'incorporation. La valeur par défaut de la facette `version-lock.packagepkg / readonly` est `true`. Pour assouplir la contrainte de version sur un package, définissez la valeur de sa facette `version-lock` sur `false`.

Dans l'exemple suivant, vous souhaitez effectuer une rétrogradation à une version antérieure du package. La commande `pkg update` permet de rétrograder et de mettre à niveau les packages.

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

La commande `pkg contents` illustre la manière dont cette contrainte de version est définie. Pour assouplir la contrainte de version sur ce package, définissez sa facette `version-lock` sur `false`. Puis retentez la rétrogradation. Notez qu'aucun nouvel environnement d'initialisation n'est créé, mais qu'un environnement d'initialisation de sauvegarde est créé. Reportez-vous à la section [“Propriétés d'image de la stratégie d'environnement d'initialisation” à la page 96](#) pour obtenir des informations sur la création des environnements d'initialisation.

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

PHASE          ITEMS
Removing old actions      1/1
```

```

Updating image state                               Done
Creating fast lookup database                      Done
Reading search index                              Done
Building new search index                         850/850
$ pkg update library/security/openssl@1.0.1.5-0.175.2.0.0.23.0
    Packages to update: 1
    Create boot environment: No
    Create backup boot environment: Yes

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

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

```

Pour éviter que ce package soit rétrogradé ou mis à niveau, figez le package à la version actuelle. La lettre "f" qui s'affiche dans la liste de packages indique que le package est figé.

```

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

```

Pour réactiver la rétrogradation ou la mise à niveau, utilisez la commande `pkg unfreeze` pour supprimer le blocage de la version. Si le package est installé à une version inférieure à celle spécifiée dans le package d'incorporation, le paramétrage de la facette `version-lock` de ce package sur `true` permet l'installation de la version spécifiée dans le package d'incorporation.

Si d'autres packages installés ont des relations de dépendance requise vis-à-vis du package que vous souhaitez rétrograder ou mettre à niveau, vous devrez peut-être également assouplir les contraintes de version de ces packages associés. Dans l'exemple suivant, les contraintes de version applicables au package `hexedit` ont été annulées, mais l'installation est rejetée en raison de contraintes de version applicables au package `system/library`.

```

$ pkg install editor/hexedit@1.2.12-0.175.2.0.0.25.0
Creating Plan (Solver setup): -
pkg install: No matching version of editor/hexedit can be installed:
  Reject: pkg://solaris/editor/hexedit@1.2.12-0.175.2.0.0.25.0:20131014T170634Z
  Reason: All versions matching 'require' dependency
  pkg://system/library@0.5.11,5.11-0.175.2.0.0.24.0 are rejected
  Reject: pkg://solaris/system/library@0.5.11,5.11-0.175.2.0.0.24.0:20131001T152820Z

```

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

Outre les contraintes de version des packages individuels, vous pouvez également assouplir les contraintes de version des incorporations. Dans ce cas, l'attribution de la facette `version-lock` sur `false` vous permet de déverrouiller l'incorporation et de l'isoler du reste du système. Bien que le package d'incorporation soit déverrouillé, les packages qu'elle contient continuent d'être synchronisés.

## Spécification d'une implémentation d'application par défaut

Plusieurs versions d'une application ou d'un outil seront peut-être intégrées dans la même image. Chaque version de l'application est disponible pour les utilisateurs en spécifiant le chemin d'accès complet. Une version, appelée la version préférée, est disponible à partir d'un répertoire commun comme `/usr/bin` pour une plus grande simplicité d'utilisation. Si toutes les versions participent à la même médiation, comme décrit ci-dessous, vous pouvez alors facilement réinitialiser la version. Il est préférable d'utiliser la version préférée. Cette sélection est conservée au fil des mises à jour de packages.

Une *médiation* est un ensemble de liens dans lequel le chemin du lien est identique pour chaque lien figurant dans le jeu, et la cible de chaque lien est différent. Par exemple, le chemin du lien peut être `/usr/bin/myapp` et les cibles du lien peuvent être `/usr/myapp/myapp1/bin/myapp` et `/usr/myapp/myapp2/bin/myapp`. Chaque lien figurant dans une médiation est appelé *participant* dans la médiation. Si `/usr/bin/myapp` appelle actuellement `myapp1`, vous pouvez facilement modifier la sélection afin que `/usr/bin/myapp` appelle `myapp2`. La version du logiciel qui est actuellement la cible du lien est la version *préférée*.

## Identification des participants à une médiation

Utilisez la commande `pkg mediator` pour afficher les informations d'identification et de versions de tous les liens de médiation dans l'image.

Dans la sortie suivante, `MEDIATOR` est le nom de l'ensemble de liens qui partagent le même chemin du lien. `VER.` `SRC.` et `IMPL.` `SRC.` indiquent si la version préférée a été sélectionnée par le système, a été sélectionnée en fonction d'une priorité attribuée (`vendor` ou `site`), ou si elle a été définie par un administrateur (`local`). `VERSION` est la version du participant de la médiation sélectionné qui doit être semblable à la version du logiciel que le lien représente. La `VERSION` est défini par le développeur du package. `IMPLEMENTATION` une chaîne pouvant être définie par le développeur du package en plus ou à la place de la chaîne de version.

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

L'option `-a` permet d'afficher tous les participants de la médiation. Utilisez-la pour afficher vos choix si vous souhaitez sélectionner une autre version préférée. L'exemple suivant indique tous les participants à la médiation `java`. Les mots clés système `system` indiquent que la version préférée dans cette médiation n'est pas spécifiée avec un paramètre de priorité dans le package et n'a pas été définie par un administrateur système : le système de packages sélectionné la version avec une valeur `VERSION` supérieure à la version préférée.

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

La sortie suivante permet de confirmer que deux versions différentes de cet Java Runtime Environment sont installées dans cette image, et la version `1.7.0_51` est actuellement la version préférée :

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

Les deux packages `jre-6` et `jre-7` définissent un lien symbolique dont le chemin est `/usr/bin/java`. Dans le package `jre-6`, la cible du lien est `jdk1.6.0`. Dans le package `jre-7`, la cible du lien est `jdk1.7.0`. Les commandes `pkg mediator` et `java -version` ci-dessus montrent que la version `1.7` est actuellement la version préférée, et que la cible du lien est `/usr/bin/java`.

## Modification de l'application préférée

Exécutez la commande `pkg set-mediator` pour réinitialiser la version d'un médiateur spécifié qui est la version par défaut ou préférée.

Reprenez la sortie de `pkg mediator -a` pour sélectionner une version de l'implémentation de l'argument `-V` ou une implémentation pour l'argument `-I`. Si vous faites une erreur typographique ou si vous indiquez une version ou une implémentation de médiateur qui n'est pas disponible actuellement, tous les liens qui utilisent le médiateur spécifié sont supprimés.

Utilisez l'option `-n` avec la sous-commande `set-mediator` pour vérifier si un BE de sauvegarde sera créé. Si aucun BE de sauvegarde n'est créé, vous pouvez spécifier l'option `--require-backup-be` avec la sous-commande `set-mediator`. Le changement de médiateur est effectué dans le BE actuel. Lorsque vous savez que votre BE en ne pose pas de problèmes une fois que le médiateur a changé, vous pouvez utiliser `beadm destroy` pour détruire le BE de sauvegarde.

La sortie précédente avait indiqué que la version 1.7 est actuellement sélectionnée comme la version de médiation `java` préférée. La commande suivante indique comment paramétrer la version 1.6 comme version préférée, ce qui signifie qu'appeler `/usr/bin/java` reviendra à appeler la version JRE 1.6. La version 1.7 de JRE est toujours disponible sur le système lorsque les utilisateurs spécifient le chemin d'accès complet à cette version. Comparez les sorties des deux commandes `pkg mediator`. Lorsque vous avez modifié la version préférée pour la médiation, `VER. SRC.` a également été changé en `local`, indiquant que la sélection a été spécifiée par un administrateur. Cette sélection sera conservée après les réinitialisations et mises à jour des packages.

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

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

$ pkg set-mediator -V 1.6 java
      Packages to update: 3
      Mediators to change: 1
      Create boot environment: No
      Create backup boot environment: No

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

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

Cette sélection est conservée au fil des mises à jour, même si l'implémentation sélectionnée n'est plus installée. Si l'implémentation sélectionnée n'est plus installée, la cible du lien de médiation n'existe pas. Utilisez l'une des méthodes suivantes pour réinitialiser l'implémentation préférée :

- Utilisez la commande `pkg set-mediator` à nouveau pour sélectionner une autre implémentation à partir de la liste mise à jour indiquée par `pkg mediator -a`.
- Utilisez la commande `pkg unset-mediator` pour permettre au système de sélectionner une nouvelle implémentation.

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

## Annulation de l'installation de certains packages compris dans un package de groupe

Utilisez la commande `pkg avoid` pour éviter d'installer certains packages donnés s'ils sont la cible d'une dépendance `group`. Vous avez toujours la possibilité d'installer explicitement un package compatible avec l'image, même si ce package est sur la liste "avoid" des packages à éviter. Lorsqu'un package figurant sur la liste "avoid" des packages à éviter est installé, il est supprimé de la liste "avoid". La commande `pkg avoid` vous permet de ne pas installer certains packages donnés d'un package de groupe lorsque vous installez ce package de groupe.

Sans argument, la commande `pkg avoid` affiche chaque package évité avec tout package ayant une dépendance de type `group` sur ce package.

Avec les packages spécifiés, la commande `pkg avoid` place les noms des packages qui correspondent actuellement aux schémas spécifiés sur la liste des packages à éviter. Seuls les packages qui ne sont pas installés peuvent être évités. Si un package est actuellement la cible d'une dépendance du groupe, sa désinstallation le place dans la liste `avoid`.

Les packages qui se trouvent sur la liste `avoid` sont installés si nécessaire pour satisfaire une dépendance `require` et par conséquent supprimés de cette liste `avoid`. Si cette dépendance `require` est supprimée, le package est désinstallé et remonté sur la liste "avoid".

Utilisez la commande `pkg unavoid` pour supprimer les packages spécifiés dans la liste des packages à éviter.

Les packages dans la liste des packages à éviter qui correspondent à la dépendance de type `group` d'un package installé ne peuvent pas être supprimés de cette liste avec la sous-commande `unavoid`. Pour supprimer de cette liste un package qui correspond à une dépendance de groupe, installez le package.

### EXEMPLE 5-5 Ajout et suppression de packages dans la liste Avoid

La sortie de la commande suivante indique que le package de groupe `group/feature/amp` n'est pas installé. Certains des packages qui font partie de ce package de groupe sont installés parce qu'ils ont été installés de manière explicite ou en tant que dépendances `require` d'autres packages. Utilisez l'option `-r` à l'aide de la commande `pkg contents` parce que le package spécifié n'est pas installé.

```
$ pkg list -a group/feature/amp
NAME (PUBLISHER)                VERSION                IFO
group/feature/amp                0.5.11-0.175.2.0.0.33.0  ---
$ pkg list -a `pkg contents -o fmri -Hrt depend -a type=group group/feature/amp`
NAME (PUBLISHER)                VERSION                IFO
database/mysql-51                5.1.37-0.175.2.0.0.34.0  ---
web/php-52                        5.2.17-0.175.2.0.0.34.0  i--
web/php-52/extension/php-apc      3.0.19-0.175.2.0.0.34.0  i--
```

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

La commande suivante place l'un des packages non installés et appartenant à ce package de groupe dans la liste "avoid". Le package de groupe n'est pas indiqué dans la liste "avoid" car il n'est pas installé.

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

Les commandes suivantes indiquent que le package évité n'est pas installé lorsque le package de groupe est installé. L'option -r n'est pas utilisée avec la commande pkg contents parce que le package spécifié est installé.

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

Après l'installation du package de groupe, le package de groupe est indiqué dans la liste "avoid".

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

La commande pkg unavoid n'entraîne pas le retrait d'un package de la liste "avoid" si ce package fait partie d'un package de groupe installé. Pour retirer un tel package de la liste "avoid", vous devez l'installer.

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

Vous ne pouvez pas placer un package sur la liste "avoid" s'il est déjà installé. Le package est placé sur la liste "avoid" si vous le désinstallez.

```
$ pkg avoid apache-fcgid
pkg avoid: The following packages are already installed in this image; use uninstall to avoid
these:
```

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

Si un package faisant partie d'un package de groupe est désinstallé, il est automatiquement placé sur la liste "avoid".

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

Si le package de groupe est désinstallé, les packages évités restent sur la liste "avoid", mais la liste "avoid" n'indique plus leur association avec le package de groupe.

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

## Configuration des propriétés d'image et d'éditeur

Pour mettre en oeuvre des stratégies d'image, définissez des propriétés d'image. Cette section décrit les propriétés d'image et d'éditeur et explique comment les définir. Pour consulter la description des propriétés d'image, reportez-vous à la page de manuel [pkg\(1\)](#).

### Propriétés d'image de la stratégie d'environnement d'initialisation

Une image est un emplacement dans lequel des packages IPS peuvent être installés et où d'autres opérations IPS peuvent être effectuées. Un environnement d'initialisation est une instance amorçable d'une image. Vous pouvez disposer de plusieurs environnements d'initialisation sur votre système, chacun pouvant avoir différentes versions de logiciels installés. Lorsque vous initialisez votre système, vous avez la possibilité de démarrer dans n'importe quel environnement d'initialisation sur votre système.

Un nouvel environnement d'initialisation peut être créé automatiquement à la suite des opérations de packages. Vous pouvez également créer explicitement un nouvel environnement d'initialisation. La création d'un nouvel environnement d'initialisation dépend de la stratégie d'image, comme détaillé dans cette section.

Par défaut, un nouvel environnement d'initialisation est automatiquement créé lorsque vous exécutez l'une des opérations suivantes :

- Mise à jour de packages de système clé particuliers, tels que certains pilotes et autres composants de noyau. Composants système clés peuvent être mis à jour lorsque vous modifiez une variante ou d'une facette ainsi que lorsque vous installez, désinstallez, et mettre à jour les packages.
- Spécifier une des options suivantes : `--be-name`, `--require-new-be`, `--backup-be-name`, `--require-backup-be`.
- Définissez la stratégie d'image `be-policy` sur `always-new`. Dans le cadre de cette stratégie, toutes les opérations sont effectuées dans un nouvel environnement d'initialisation défini comme actif lors de l'initialisation suivante.

Lorsqu'un environnement d'initialisation est créé, le système effectue les opérations suivantes :

1. Il crée un clone de l'environnement d'initialisation actuel.  
Un clone de l'environnement d'initialisation contient l'intégralité du contenu, ordonné dans une arborescence sous le jeu de données root principal de l'environnement d'initialisation 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, le nouvel environnement d'initialisation accède aux systèmes de fichiers partagés d'origine.
2. Il met à jour les packages dans le clone BE. Ne met pas à jour BE les packages dans l'en cours.  
Si des zones non globales sont configurées dans l'environnement d'initialisation actuel, ces zones existantes sont configurées dans le nouvel environnement d'initialisation.
3. Définit le nouvel environnement d'initialisation qui sera utilisé comme environnement par défaut lors de la prochaine initialisation du système, sauf si l'option `-no-be-activate` est spécifiée. L'environnement d'initialisation actuel figure toujours parmi les choix d'initialisation possibles.

Lorsqu'un environnement d'initialisation est créé, le système effectue les opérations suivantes :

1. Il crée un clone de l'environnement d'initialisation actuel.
2. Mise à jour des packages dans le BE en cours. Ne met pas à jour les packages dans le clone BE.

Si un nouvel environnement d'initialisation est requis mais que l'espace est insuffisant pour le créer, vous pouvez peut-être supprimer des environnements d'initialisation existants inutiles. Pour plus d'informations sur les environnements d'initialisation, reportez-vous à la section [“Création et administration d'environnements d'initialisation Oracle Solaris 11.2”](#).

Reportez-vous à [“Définitions des propriétés d'image”](#) à la page 102 pour savoir comment définir les propriétés de l'image suivantes.

#### `be-policy`

Indique à quel moment un environnement d'initialisation est créé lors des opérations d'empaquetage. Les valeurs suivantes sont autorisées :

#### default

Applique la stratégie de création d'IE par défaut : create-backup.

#### always-new

Requiert une réinitialisation de toutes les opérations en les effectuant dans un nouvel environnement d'initialisation défini comme actif lors de l'initialisation suivante. Aucun environnement d'initialisation de sauvegarde n'est créé sauf explicitement demandé.

Cette stratégie est la plus sûre, mais elle est plus stricte que ce qui est demandé dans la plupart des sites, dans la mesure où aucun package ne peut être ajouté sans qu'il faille réinitialiser le système.

#### create-backup

Pour les opérations sur les packages qui nécessitent une réinitialisation, cette politique crée un nouvel environnement d'initialisation défini comme actif lors de la prochaine initialisation. Si les packages sont modifiés ou si du contenu pouvant avoir des répercussions sur le noyau est installé, et que l'opération a une incidence sur l'environnement d'initialisation actif, un environnement de sauvegarde est créé, mais il n'est pas défini comme actif. Un environnement d'initialisation de sauvegarde peut également être explicitement demandé.

Cette stratégie est potentiellement dangereuse uniquement si les logiciels récemment installés entraînent une instabilité du système, ce qui est possible, mais relativement rare.

#### when-required

Pour les opérations sur les packages qui nécessitent une réinitialisation, cette politique crée un nouvel environnement d'initialisation défini comme actif lors de la prochaine initialisation. Aucun environnement d'initialisation de sauvegarde n'est créé sauf explicitement demandé.

Cette stratégie comporte les plus grands risques, car si un changement d'empaquetage dans l'environnement d'initialisation opérationnel empêche toute modification supplémentaire, un environnement d'initialisation de secours récent risque de ne pas exister.

## Propriétés de signature de packages

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 manifestes lors de l'installation, de la mise à jour, de la modification ou de la vérification 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. Par défaut, le client du package ne vérifie pas si les certificats ont été révoqués. Pour activer ces vérifications, qui peuvent nécessiter que le client contacte des sites web externes, définissez la propriété d'image `check-certificate-revocation` sur `true`. Les valeurs suivantes sont autorisées :

`ignore`

Ignore les signatures pour tous les manifestes.

`verify`

Vérifie que tous les fichiers manifestes avec signatures sont valablement signés, mais ne nécessite pas que tous les packages installés soient signés.

Il s'agit de la valeur par défaut.

`require-signatures`

Demande à ce que tous les nouveaux packages installés disposent au moins d'une signature valide. Les commandes `pkg fix` et `pkg verify` avertissent également lorsqu'un package installé ne possède pas de signature valide.

`require-names`

Suit les mêmes exigences que `require-signatures` mais nécessite aussi que les chaînes répertoriées dans la propriété d'image `signature-required-names` 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.

## 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.

L'exemple suivant configure cette image pour exiger que tous les packages soient signés. Il 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
```

L'exemple suivant configure cette image pour exiger que tous les packages signés soient vérifiés.

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

L'exemple suivant configure 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
```

L'exemple suivant ajoute 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-required-names trustedname
```

L'exemple suivant supprime 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-required-names trustedname
```

L'exemple suivant ajoute 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-required-names=trustedname example.com`

## Propriétés de l'image supplémentaires

### ca-path

Spécifie un nom de chemin d'accès qui pointe vers un répertoire dans lequel les certificats de CA sont conservés pour les opérations SSL. Le format de ce répertoire est spécifique à l'implémentation SSL sous-jacente. Pour utiliser un autre emplacement pour les certificats de CA de confiance, modifiez cette valeur de manière à ce qu'elle pointe vers un autre répertoire. Reportez-vous aux paragraphes `CPath` de `SSL_CTX_load_verify_locations(3openssl)` pour connaître les exigences concernant le répertoire des certificats de CA.

La valeur par défaut est `/etc/openssl/certs`.

### check-certificate-revocation

Si cette propriété est définie sur `True`, le client de package tente de contacter les points de distribution CRL dans les certificats utilisés pour la vérification des signatures pour déterminer si le certificat a été révoqué depuis son émission.

La valeur par défaut est `False`.

### content-update-policy

Indiquez la date à laquelle le système de package mettra à jour les fichiers non modifiables au cours des opérations de création de package. Les valeurs suivantes sont autorisées :

#### default

Toujours appliquer la stratégie de mise à jour du contenu par défaut.

#### always

Toujours télécharger et mettre à jour les fichiers non modifiables qui ont changé.

#### when-required

Téléchargez et mettez à jour les fichiers qui ont changé uniquement si le package système a déterminé qu'une mise à jour est requise.

La valeur par défaut est `always`.

### flush-content-cache-on-success

Si cette propriété est paramétrée sur `true`, le client du package supprime les fichiers de son image-cache de contenu une fois la modification des opérations réalisée. Pour les opérations qui créent un BE, le contenu est supprimé des BE source et cible.

Cette propriété peut être utilisée pour limiter la taille du cache de contenu sur les systèmes dotés d'un espace disque réduit. Cette propriété peut ralentir les opérations.

La valeur par défaut est `True`.

#### `mirror-discovery`

Cette propriété demande au client de découvrir les miroirs de contenu link-local à l'aide de mDNS et DNS-SD. Si cette propriété est définie sur `true`, le client tente de télécharger le contenu du package à partir de miroirs qu'il détecte de manière dynamique. Pour exécuter un miroir qui publie son contenu via mDNS, reportez-vous à la page de manuel [pkg.depotd\(1M\)](#).

La valeur par défaut est `False`.

#### `send-uuid`

Envoie l'identificateur unique universel (UUID) de l'image pendant les opérations sur le réseau. Bien que les utilisateurs puissent désactiver cette option, certains référentiels de réseau peuvent refuser de communiquer avec des clients qui ne fournissent pas un UUID.

La valeur par défaut est `True`.

#### `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.

La valeur par défaut est `ignore`.

#### `use-system-repo`

Cette propriété indique si l'image doit utiliser le référentiel du système comme une source pour l'image et la configuration de l'éditeur et comme un serveur proxy pour la communication avec les éditeurs fournis. Reportez-vous à la page de manuel [pkg.sysrepo\(1M\)](#) pour plus d'informations sur les référentiels système.

La valeur par défaut est `ignore`.

## Définitions des propriétés d'image

Utilisez la commande `pkg property` pour consulter les paramètres de propriété de l'image. Utilisez les sous-commandes `set-property`, `add-property-value`, `remove-property-value` et `unset-property` pour configurer les propriétés de l'image.

## Affichage des valeurs des propriétés de l'image

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

```
$ pkg property
PROPERTY                               VALUE
```

```

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

```

Vous pouvez avoir intérêt à utiliser les options de l'ordre de recherche de la commande `pkg set-publisher` afin de définir la propriété `publisher-search-order`. Reportez-vous à [“L'option Affectation de l'ordre de recherche de la définition et Publisher \(disponible en anglais uniquement\)” à la page 79](#).

## 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.

L'exemple suivant définit la valeur de la propriété `mirror-discovery`.

```

$ pkg set-property mirror-discovery true
$ pkg property -H mirror-discovery
mirror-discovery True

```

## Réinitialisation de la valeur d'une propriété d'image

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

```

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

```

# Création d'une image

Une image est un emplacement dans lequel des packages IPS et les fichiers répertoires, liens et dépendances qui leur sont associés peuvent être installés, et où d'autres opérations IPS peuvent être effectuées.

Les images créées à l'aide de la commande `pkg image-createne` sont pas amorçables. Pour créer des images amorçables, utilisez l'option `--be-name` ou `--require-new-be` avec les commandes `pkgou` utilisez les commandes `beadm` ou `zonecfg` et `zoneadm`. La commande `pkg image-create` est utilisée pour des tâches comme la maintenance des packages et de la distribution du système d'exploitation.

La commande `pkg image-create` exige un opérande qui est le répertoire dans lequel l'image sera créée. Le type par défaut de l'image créée sera une image d'utilisateur. Vous pouvez spécifier un des types d'images suivants :

- |         |   |
|---------|---|
| Full    | Les images complètes peuvent fournir un système complet. Dans une image complète, toutes les dépendances sont résolues au sein de l'image elle-même et IPS gère les dépendances de manière cohérente. Après l'installation d'Oracle Solaris OS, une image complète contient le système de fichiers racine et son contenu. Utilisez l'option <code>-F</code> ou <code>--full</code> pour spécifier une image complète.   |
| Partial | <p>Les images partielles sont liées à l'image complète délimitant le chemin <i>dir</i> donné (image parent). Les images partielles ne fournissent pas un système complet par elles-mêmes. Utilisez l'option <code>-P</code> ou <code>--partial</code> pour spécifier une image partielle.</p> <p>Une zone non globale est une image partielle. Pour utiliser l'image dans un contexte de zone non globale, spécifiez l'option <code>-z</code> or <code>--zone</code> pour définir une variante appropriée. Dans une image de zone, IPS assure la cohérence de la zone non globale avec la zone globale correspondante, selon les dépendances définies dans les packages. Reportez-vous à <a href="#">“Images et environnements d'initialisation” à la page 17</a> pour plus d'informations sur les zones.</p> |
| User    | Les images d'utilisateur contiennent uniquement des packages réadressables. Il s'agit du type de l'image par défaut qui est créée si vous ne spécifiez pas de type d'image. Utilisez l'option <code>-U</code> ou <code>--user</code> pour spécifier une image d'utilisateur.  |

Utilisez l'option `-p` ou `--publisher` pour fournir un référentiel de packages. Si un nom d'éditeur est également fourni, seul cet éditeur est ajouté lorsque l'image est créée. Si un nom de l'éditeur n'est pas fourni, tous les éditeurs connus par le référentiel spécifié sont ajoutés à l'image. Une tentative de récupération du catalogue associé à cet éditeur est effectuée après les opérations de création initiales.

Pour les éditeurs utilisant l'authentification de client SSL, utilisez les options `-c` et `-k` pour inscrire une clé de client et un certificat de client. Ce certificat et cette clé sont utilisés pour tous les éditeurs ajoutés pendant la création de l'image.

Utilisez les options `--variant`, `--facet` et `--set-property` pour définir les valeurs variante et les valeurs de propriété d'image facette valeurs.

## Viewing Operation History

Utilisez la commande `pkg history` pour visualiser l'historique des commandes `pkg`. Par défaut, les informations suivantes sont affichées :

- L'heure de début de l'opération
- Le nom de l'opération a été effectuée, par exemple, `install`
- Le client, par exemple, `pkg`
- Le résultat de l'opération : `Succeeded` ou `Failed`

Utilisez les options afin d'afficher des informations supplémentaires ou des informations plus précises.

`-l`

Afficher les informations suivantes, en plus de l'ensemble par défaut des informations sont les suivants :

- Version du client
- Nom de l'utilisateur qui a effectué cette tâche.
- Création d'admission d'un nouveau BE
- Heure de fin de l'opération
- La commande complète qui a été émise
- Les éventuelles erreurs survenues lors de l'exécution de la commande
- Les FMRI de complets des packages modifiés pour des opérations telles `update`

`-n number`

Afficher uniquement le nombre spécifié d'opérations les plus récentes.

`$ pkg history -n4`

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

`-o colonne[,colonne]...`

Affichage de la sortie sous la forme d'une liste de noms de colonne séparés par des virgules. Consultez la liste des noms de colonne dans la page de manuel [pkg\(1\)](#).

`$ pkg history -o start,time,operation,outcome -n4`

START	TIME	OPERATION	OUTCOME
2013-08-06T16:32:03	0:00:27	fix	Succeeded
2013-08-06T16:41:47	0:00:43	revert	Succeeded
2013-08-06T17:56:22	0:00:00	set-property	Succeeded
2013-08-06T17:56:53	0:00:00	unset-property	Succeeded

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

Des enregistrements de journal concernant une liste d'horodatage formaté avec %Y-%m-%dT%H:%M:%S (reportez-vous à la page de manuel `strftime(3C)`). Pour spécifier une plage de temps, utilisez un trait d'union (-) entre les horodatages de début et de fin. Le mot-clé `now` est un alias pour l'heure actuelle. Si les horodatages spécifiés contiennent des doublons ou des fourchettes de dates qui se chevauchent horodatages, une seule instance de chaque événement historique en double s'affiche.

-N

Utilisez l'option `N` pour afficher d'éventuels textes de note de version relatifs à l'opération. L'option `-N` ne peut pas être utilisée avec l'option `-o`. Si vous spécifiez l'option `-v` dans l'installation ou opération de mise à jour de packages contenant des notes de version installées, la sortie de l'opération affiche les notes sur la version. Si l'opération effectue une installation dans un nouveau BE, la sortie de l'opération fournit un chemin vers le fichier de notes sur la version dans `/tmp` du BE en cours dans. Lorsque vous devez initialiser dans le nouveau BE, les notes sur la version se trouvent dans `/usr/share/doc/release-notes` ou vous pouvez utiliser l'option `-N` pour visualiser les notes sur la version comme indiqué dans la commande suivante :

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

Si l'opération d'installation des notes de la version n'est pas la dernière opération `pkg` effectuée dans ce BE, utilisez une valeur plus élevée pour l'argument `-n` ou l'option `-t` pour identifier l'opération `pkg` qui a installé les notes de version comme indiqué dans la commande suivante :

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

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

```
$ pkg purge-history
```



## Installation d'un package et mettre à jour Dépannage

---

Cette annexe montre la procédure de gérer des erreurs que vous êtes susceptible de plus de détails sur la mise à installer ou mettre à jour des packages. Cette annexe propose aussi et en minimisant fournit des conseils portant sur des métadonnées stockées de manière à optimiser les performances.

### Initiale Troubleshooting Etapes

Vérifiez si le package que vous voulez installer est disponible à partir des éditeurs configurés et peut être installé dans cette image. La version installée de `pkg:/entire` ainsi que des origines configurés dans celle-ci sont des informations obligatoires. Il faudra éventuellement mettre à jour le package `pkg:/entire`. Il peut s'avérer nécessaire de modifier les origines de l'éditeur.

La création de deux contrôles devant être réalisée en premier pour quasiment tout problème d'installation de package, procédez comme suit :

- Vérifiez quelle version du package d'incorporation `pkg:/entire` est installée.
- Vérifiez l'origine de l'éditeur de package.

Lorsque vous avez déterminé que les packages dont vous avez besoin sont disponibles à partir des éditeurs configurés, suivez les étapes ci-dessous pour poursuivre l'installation :

- Utilisez les options `-nv` chaque fois que vous effectuez une installation ou une mise à jour pour voir quelles modifications seront effectuées, par exemple les versions de package qui seront installées ou mises à jour et si un nouvel environnement d'initialisation est créé. L'option `-v` présente également les notes de version qui s'appliquent à cette opération d'installation ou de mise à jour.
- Pour recevoir des messages d'erreur plus détaillés, spécifiez davantage de FMRI du package que vous souhaitez installer, y compris la version et l'éditeur.

## Vérifiez la version installée de pkg:/entire

Utilisez la commande `pkg list` pour vérifier la version du / package d'incorporation `pkg:/entire` actuellement installée.

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

Le package d'incorporation `pkg:/entire` limite les versions de nombreux autres packages pour maintenir une image fonctionnelle et qui peut être prise en charge. Reportez-vous à [“Les packages d'incorporation” à la page 12](#) pour plus d'informations sur les packages qui sont limités par des packages d'incorporation.

Vous ne pouvez pas installer ou mettre à jour directement un package qui est soumis aux limites du package d'incorporation `pkg:/entire`. Pour installer ou mettre à jour des packages qui sont limités par `pkg:/entire`, vous devez mettre à jour le package `pkg:/entire`. Pour plus d'informations, reportez-vous à [“Impossible de répondre aux contraintes” à la page 117](#). Dans certains cas, vous pouvez supprimer les contraintes comme indiqué dans [“Mise à jour d'un package contraint par une incorporation” à la page 118](#).

## Vérifiez les origines d'un éditeur de votre serveur de contenu configuré

Utilisez la commande `pkg publisher` vérifier la des origines de l'éditeur de package.

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

Pour garantir un URI sécurisé, la clé pertinente et le certificat doivent être correctement installés, et vous devez utiliser les options `-k` et `-c` lorsque vous configurez l'éditeur.

Si votre site nécessite un proxy pour lieux externes, option, utilisez l'option `--proxy` de la commande `pkg set-publisher` permettant de définir le proxy. Reportez-vous à [“Spécification d'un proxy” à la page 81](#) pour obtenir des instructions.

Utilisez la commande `pkg publisher publisher` pour visualiser des informations supplémentaires sur l'éditeur, comme la clé, les certificats et le proxy.

Le cas échéant de chacune des URI d'origine les éditeurs activés ne sont pas accessibles, l'installation ou de la mise à jour échoue, même si les emplacements dont vous avez besoin

sont accessibles. Si vous ne parvenez pas à résoudre le problème qui rend l'emplacement inaccessible, vous pouvez supprimer ou désactiver l'origine inaccessible à l'aide de la commande `pkg set-publisher -G` ou désactiver l'éditeur avec la commande `pkg set-publisher --disable`. Si vous n'avez plus besoin cet éditeur, utilisez la commande `pkg unset-publisher` pour supprimer l'éditeur.

Vérifiez si l'origine de l'éditeur de package contient les packages dont vous avez besoin. Par exemple, si l'origine de votre éditeur `solaris` est définie sur la version publique du référentiel, vous ne pouvez pas mettre à jour un package vers une version uniquement disponible à partir d'un référentiel de prise en charge.

## Vérifier si les packages installés requis sont disponibles

Pour mettre à jour les packages installés, installer des packages qui dépendent de packages installés, ou installer une zone non globale, le référentiel que vous avez défini comme origine de l'éditeur doit contenir au moins les mêmes logiciels déjà installés dans l'image. Le référentiel peut également contenir des versions plus anciennes ou plus récentes du logiciel, mais il doit contenir le même logiciel installé dans l'image.

Utilisez la commande `pkgrepo list`, et non pas la commande `pkg list` lors de la vérification des packages installés. La commande `pkg list` fait toujours apparaître les packages installés, même si le package n'est disponible à partir d'une origine d'aucun éditeur configuré.

La commande suivante indique que le référentiel indiqué n'est pas une origine de l'éditeur appropriée pour cette image, car la version installée de `pkg:/entire` n'est pas disponible à partir de cette origine.

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

Si le package dont vous avez besoin n'est pas répertorié, essayez de réexécuter la commande `pkgrepo refresh`, puis relancez la commande `pkgrepo list`.

La commande suivante indique que la version installée de `pkg:/entire` est disponible depuis le référentiel :

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

Si un packages requis n'est pas disponible à partir d'un éditeur configuré est disponible à partir d'un autre mais l'origine de référentiel, effectuez l'une des opérations suivantes :

- Utilisez l'option `-g` de la commande `pkg set-publisher` afin d'ajouter cette origine de l'éditeur `solaris`.
- Utilisez les options `-g` et `-G` de la commande `pkg set-publisher` pour modifier l'origine de l'éditeur `solaris`.
- Si le package dont vous avez besoin est fourni par un autre éditeur, utilisez la commande `pkg set-publisher` pour ajouter cet éditeur.
- Utilisez l'option `-g` de la commande d'installation (`install`, `uninstall`, `update`, `change-variant`, et `change-facet`) pour ajouter temporairement un référentiel à la fin de la liste des référentiels dans lesquels rechercher.

## Vérifiez si les packages que vous souhaitez installer sont disponibles

Exécutez la commande suivante pour vérifier si le package que vous souhaitez installer est disponible chez les éditeurs configurés. Si vous mettez à jour un package, la version du package actuellement installée et celle vers laquelle vous souhaitez mettre à jour doivent être disponibles.

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

Si le package dont vous avez besoin n'est pas répertorié, essayez d'exécuter la commande `pkg refresh`, puis relancez la commande `pkg list`.

Si le package dont vous avez besoin n'est pas répertorié, ajoutez un nouvel éditeur ou une nouvelle origine d'éditeur.

## Vérifiez si les packages que vous souhaitez installer peuvent l'être dans cette image

Si la version du package de votre choix est répertoriée lorsque vous utilisez les options `-af`, utilisez à nouveau la même commande sans l'option `-f` :

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

Si la version de votre choix est toujours répertoriée, ce package n'est pas limité et vous devez être en mesure de l'installer sans installer ou mettre à jour d'autres packages.

Si la version de votre choix n'est pas répertoriée, cette version est disponible auprès des éditeurs confirmés mais non installable dans cette image. Parmi les raisons pour lesquelles un package ne peut pas être installé figurent les suivantes :

- Le package est limité par des paramètres de variante ou de facette.

- La version du package est limitée par un package d'incorporation. Vous pouvez mettre à jour le package à l'origine de la contrainte et, dans certains cas, alléger la contrainte. Pour plus d'informations, reportez-vous à [“Impossible de répondre aux contraintes” à la page 117](#).
- La version du package est contrainte par une opération consistant à figer. Exécutez la commande `pkg freeze`. Le blocage peut se trouver sur un package qui a une dépendance de type `require` sur une version différente du package à installer et les deux versions ne peuvent pas être installées en même temps.

## Réessayez l'installation

Si vous spécifiez un package pour installer ou mettre à jour et que vous n'indiquez pas de l'éditeur, le premier éditeur dans l'ordre de recherche des éditeurs qui fournit un package correspondant à ce package FMRI ou un motif est utilisé en tant que source d'installation. Si cet éditeur ne fournit pas une version du package qui peut être installée sur cette image, l'opération d'installation échoue, et ce même si une autre activité éditeur fournit une version du package qui peut être installé dans cette image. Effectuez l'une des opérations suivantes pour résoudre ce problème, procédez comme suit :

- Indiquez l'éditeur dans le package FMRI. Par exemple, spécifiez `pkg://solaris/` en face du nom complet du package.
- Utilisez l'option `-P` de la commande `pkg set-publisher` permettant de définir l'éditeur qui fournit la version du package que vous voulez ajouter en tant que premier éditeur dans l'ordre de recherche.

Utilisez les options `-nv` chaque fois que vous effectuez une installation ou une mise à jour pour voir quelles modifications seront effectuées, par exemple les versions de package qui seront installées ou mises à jour et si un nouvel environnement d'initialisation est créé. L'option `-v` présente également les notes de version qui s'appliquent à cette opération d'installation ou de mise à jour.

- Si vous n'obtenez pas de messages d'erreur lorsque vous utilisez les options `-nv`, exécutez à nouveau la commande sans l'option `-n` pour effectuer l'installation ou la mise à jour. Vous devez vous demander si vous devez indiquer les éléments suivants : options pour résoudre BE l'installation dans un nouveau BE ou pour créer un nouvel environnement d'initialisation de sauvegarde ou si aucun environnement d'initialisation de secours seront créés par défaut.
- Si vous ne recevez des messages d'erreur, appliquez les actions suivantes :
  - Apportez des précisions concernant la version de votre choix dans le FMRI de package pour obtenir des informations supplémentaires pour vous permettre de diagnostiquer et de corriger le problème.
  - Spécifiez plus d'options `-v` (par exemple, `-nvv`).
  - Utilisez la commande `pkg history`. L'option `-l` comporte une gamme complète de FMRI de packages modifiés. Reportez-vous à [“Viewing Operation History” à la page 105](#).

Si vous indiquez plusieurs packages à installer ou mettre à jour ou si vous omettez la spécification du package pour une opération de mise à jour, l'installation ou opération de mise à jour échoue si l'un des packages ne peuvent pas être installé dans cette image. Si un package ne peut pas être installé, aucun package n'est installé. Pour plus d'informations, appelez la commande à nouveau en indiquant uniquement le package qui ne peut pas être installé, indiquant le FMRI complet de ce package et en fournissant une plusieurs options -v.

## Impossible d'accéder à l'éditeur ou au référentiel

Les erreurs abordées dans cette section sont liées à une incapacité à accéder à l'URI de l'éditeur.

### Impossible d'accéder au référentiel de packages

Messages d'erreur :

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

Utilisez la commande `pkg publisher` pour afficher les URI de vos éditeurs. Reportez-vous à la ligne ou à la colonne URI LOCATION dans les exemples suivants :

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

Si des éditeurs que vous n'utilisez plus sont répertoriés, vous devez les désactiver ou les supprimer :

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

Pour les éditeurs que vous utilisez, assurez-vous que les URI sont corrects.

- Essayez de consulter l'emplacement d'origine de chaque éditeur dans un navigateur ou envoyez une commande ping à cet emplacement.
- Essayez de répertorier les packages à cette origine à l'aide de la commande `pkgrepo list`.

Si un lieu d'origine n'est pas correct, utilisez les options -G et -g de la commande `pkg set-publisher` pour modifier l'URI en même temps.

Si l'éditeur a plusieurs origines, tous les emplacements d'origine doivent être accessibles. Si l'emplacement d'origine n'est pas accessible, utilisez l'option `-G` de la commande `pkg set-publisher` pour supprimer cette origine.

Quand un éditeur est configuré dans une zone non globale, tous les emplacements pour cet éditeur doivent être accessibles à partir de la zone globale, même si cet éditeur n'est pas configuré dans la zone globale.

## Problème de certificat SSL

Message d'erreur : `SSL certificate problem, verify that the CA cert is OK`

Effectuez l'une ou plusieurs des opérations suivantes si la commande `pkg` affiche un message concernant un problème de certificat SSL :

- Assurez-vous que la clé et le certificat requis sont installés. Les informations sur l'obtention de la clé et du certificat doivent être incluses avec d'autres informations relatives au référentiel sécurisé.
- Utilisez les options `-k` et `-c` à l'aide de la commande `pkg set-publisher` permettant d'indiquer l'emplacement où la clé et le certificat sont installés. La commande `pkg publisher` de l'éditeur affiche l'emplacement où l'éditeur est censé trouver la clé et le certificat. Si la clé et le certificat sont installés ailleurs, reconfigurez l'éditeur en précisant les valeurs correctes pour les options `-k` et `-c`.

Dans l'exemple suivant, la sortie présente les chemins de clé et de certificat et indique que le certificat pour l'origine de l'éditeur `abc` configuré est arrivé à expiration :

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

- Vérifiez que la date et l'heure sont correctes sur le système.
- Actualisez le service SMF `ca-certificates` :

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

- Si l'URI d'origine contient `oc-mgmt`, contactez votre représentant du support technique du centre d'opérations.

---

**Remarque** - Un référentiel qui nécessite un certificat client SSL ne peut pas être spécifié à l'aide d'une option `-g` avec la commande d'installation `pkg`.

---

## Emplacement introuvable

Message d'erreur : `http protocol error: code: 404 reason: Not Found`

Vérifiez vos URI éditeur comme décrit dans [“Impossible d'accéder au référentiel de packages” à la page 112](#). Si vous pouvez consulter l'emplacement ou lui envoyer une commande ping avec succès, utilisez la commande `pkgrepo list` pour essayer d'afficher l'un des packages dans le référentiel.

Si l'URI est un référentiel de type fichier, assurez-vous, via la commande `pkg5srv`, que les fichiers et répertoires peuvent être lus par l'utilisateur. Vous pouvez utiliser la commande `pkgrepo verify` pour vérifier si le référentiel `pkg5srv` est lisible par l'utilisateur.

Vérifier la configuration du serveur Web. Reportez-vous au [Chapitre 5, “Exécution du serveur de dépôt derrière un proxy Web”](#) du manuel [“Copie et création de référentiels de packages dans Oracle Solaris 11.2”](#) pour plus d'informations.

- Si vous exécutez le serveur de dépôt de packages derrière une instance du serveur Web Apache, incluez le paramètre suivant figurant dans votre fichier `httpd.conf` pour ne pas décoder les barres obliques :

```
AllowEncodedSlashes NoDecode
```

- Définissez le serveur de dépôt `pkg/proxy_base` à l'URL du référentiel du serveur Apache :
- ```
$ svccfg -s pkg/server:repo setprop pkg/proxy_base = astring: http://pkg.example.com/myrepo
$ svcadm refresh pkg/server:repo
```

Si le problème se produit dans une zone non globale, effectuez les opérations de dépannage suivantes. N'oubliez pas que les zones non globales utilisent un référentiel système spécial appelé référentiel système. Reportez-vous à la page de manuel [pkg.sysrepo\(1M\)](#) pour plus d'informations sur le référentiel système.

- Vous ne pouvez pas utiliser l'option `-g` dans une image ayant des zones non globales. Au lieu de cela, utilisez la commande `pkg set-publisher` permettant d'ajouter explicitement cet éditeur et l'origine concernés.
- Assurez-vous que les fichiers et répertoires dans les référentiels basés sur un fichier la base de données peuvent être lus par l'utilisateur `pkg5srv`. L'utilisateur `pkg5srv` exécute l'instance Apache `system-repository`. Reportez-vous à la section [“Relation entre la planification des zones globales et non globales” à la page 57](#) pour obtenir des exemples qui montrent comment trouver l'emplacement d'un référentiel système.
- Si votre site nécessite un proxy pour accéder aux emplacements externes, assurez-vous que le proxy a été indiqué correctement pour les éditeurs dans la zone globale. Utilisez l'option `--proxy` de la commande `pkg set-publisher` pour indiquer le proxy. Reportez-vous à [“Spécification d'un proxy” à la page 81](#) pour obtenir des instructions. L'une des méthodes

permettant de vérifier votre proxy est de vous assurer que vous ne recevez pas des messages d'erreur provenant de la commande `pkg refresh --full`.

- Assurez-vous que le service `svc:/application/pkg/system-repository:default` est en ligne dans la zone globale.
- Assurez-vous que le service `svc:/application/pkg/zones-proxy:default` est en ligne dans la zone globale et que le service `svc:/application/pkg/zones-proxy-client:default` est en ligne dans la zone non globale.
- Dans la zone globale, vérifiez si les fichiers de journal `/var/log/pkg/sysrepo/*` annoncent des erreurs d'autorisation lors d'une tentative de lecture de fichiers ayant fait l'objet d'un rapport. Vérifiez s'il y a des erreurs 404 ou 503 signalées dans `/var/log/pkg/sysrepo/access_log`. Recherchez les erreurs signalées dans `/var/log/pkg/sysrepo/error_log`.
- Vérifiez que `localhost` est défini sur `127.0.0.1` dans le fichier `/etc/hosts`. Vérifiez que `Listen` est défini sur `127.0.0.1:1008` et que `ServerName` est défini sur `127.0.0.1` dans le fichier `/system/volatile/pkg/sysrepo/sysrepo_httpd.conf`.
- Dans la zone globale, vérifiez les lignes si le fichier `/system/volatile/pkg/sysrepo/sysrepo_httpd.conf` contient des lignes d'alias de la forme suivante :

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

Si le fichier `sysrepo_httpd.conf` n'a pas de lignes d'alias, redémarrez le service `sysrepo` :

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

## Service indisponible.

Message d'erreur : `http protocol error: code: 503 reason: Service Unavailable`

Utilisez la commande `pkg publisher` pour rechercher l'emplacement du référentiel de packages que vous tentez d'utiliser et examinez les services SMF. Utilisez la commande suivante SMF pour identifier les instances de service de référentiel de packages qui sont activées, mais qui ne sont pas en cours d'exécution et qui empêchent ainsi que toutes les instances qui empêchent une instance activée de s'exécuter :

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

Si un service signale un problème, consultez le fichier journal répertorié dans la sortie `svcs` pour déterminer le problème en question.

Assurez-vous que les propriétés `inst_root` et `port` ainsi que d'autres propriétés sont correctement définies.

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

Si nécessaire, utilisez la commande `svccfg` pour réinitialiser les valeurs de propriété, comme indiqué dans l'exemple suivant :

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

Utilisez la commande `svcadm` pour effacer, actualiser, redémarrer et activer l'instance de service selon vos besoins.

## Aucune mise à jour disponible

Message d'erreur : No updates available for this image

Si vous mettez à jour un package donné, utilisez la commande suivante pour obtenir une estimation de la version du même package installée dans cette image. Si vous mettez à jour tous les packages installés ou (`pkg update` sans packages spécifiés ou avec '\*' spécifié pour le nom du package), utilisez le nom `pkg:/entire` pour le *package* dans ces commandes.

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

Si aucune version n'est actuellement installée, utilisez la commande `pkg install` au lieu de `pkg update`.

Si une version de *package* est déjà installée, la commande suivante permet d'afficher les différentes versions de *package* disponibles à partir des éditeurs configurés :

```
$ pkg list -afv package
```

Si le package comportant le numéro de version le plus élevé est déjà installé, il est possible qu'il n'existe pas de version plus récente.

Si une nouvelle version est présente, déterminez l'emplacement du référentiel du package où la version la plus récente est disponible et utilisez la commande `pkg set-publisher` pour réinitialiser l'URI d'origine URI ou pour ajouter une origine appropriée de l'éditeur. Si nécessaire, installez la clé et le certificat requis et utilisez les options `-k` et `-c` permettant de les spécifier. Utilisez la commande `pkgrepo list` pour vérifier que la version actuellement installée du package est également disponible à partir des éditeurs configurés.

Réexécutez la commande `pkg update` en indiquant les options `-nv`, y compris la dernière version (ou le mot-clé `latest`) dans le FMRI des packages que vous souhaitez installer, comme indiqué dans l'exemple suivant. En fournissant davantage d'informations dans les noms de packages, vous pouvez obtenir plus d'informations sur les erreurs.

```
$ pkg update -nv package@latest
```

## Les packages ne peuvent pas être installés

Message d'erreur: No matching version of *package* can be installed

La commande suivante permet d'afficher les différentes versions de *package* disponibles à partir des packages configuré des éditeurs :

```
$ pkg list -afv package
```

Indiquez une plus grande partie du FMRI du package que vous souhaitez installer. La première correspondance trouvée n'est peut-être pas installable dans cette image, mais il est possible que la version que vous souhaitez installer puisse l'être. Si le FMRI plus spécifique ne peut toujours pas être installé, indiquer plus d'informations sur le FMRI permet d'obtenir plus d'informations sur la raison pour laquelle le package ne peut pas être installé.

Vérifiez que la version du package que vous souhaitez installer ou mettre à jour n'est pas figée. Utilisez la commande `pkg freeze` sans arguments pour afficher la liste de tous les packages dont les versions sont figées.

La commande suivante permet d'afficher la version du package `pkg:/entire` installé :

```
$ pkg list -v entire
```

Vous ne pouvez pas installer ou mettre à jour un package qui est limité par le package d'incorporation `pkg:/entire` ou par un autre package d'incorporation. Vous devez mettre à jour le package d'incorporation. Pour plus d'informations, reportez-vous à [“Impossible de répondre aux contraintes”](#) à la page 117.

## Impossible de répondre aux contraintes

Message d'erreur: No solution was found to satisfy constraints

Ce message indique que vous avez tenté d'installer une version d'un package qui ne correspond pas à la version dans lequel le package est contraint par un package d'incorporation. Reportez-vous à [“Les packages d'incorporation”](#) à la page 12 pour plus d'informations sur les packages d'incorporation et les contraintes.

Les incorporations contraignent un ensemble de packages à des versions qui travaillent ensemble pour garantir une image fonctionnelle et qui peut être prise en charge. C'est la raison pour laquelle vous ne devez pas mettre à jour le package qui est contraint par une incorporation.

Au lieu de cela, vous devez mettre à jour le package d'incorporation, ce qui entraîne la mise à jour de tous les packages dans un nouvel ensemble de versions testées de manière groupée.

Lorsque les packages sont en cours de modification (par exemple, en cours de mise à jour), le clientpkg et examine les packages liés et leurs dépendances. Si un package dépendant ne peut pas être installé ou mis à jour, un message d'erreur séparé est généré pour chaque package qui a une dépendance sur le package et qui ne peut pas être installé ou mis à jour. Le moyen le plus efficace de façon à pouvoir gérer un grand nombre de messages d'erreur consiste à examiner les messages d'erreur qui sont les plus en retrait en premier.

## Mise à jour d'un package contraint par une incorporation

Messages d'erreur :

- Aucune version appropriée de *package* installé n'a été trouvée
- Toutes les versions correspondant à la dépendance 'incorporate' *package* sont rejetés
- Cette version est exclue par la version d'installation spécifiée
- Cette version est exclue par l'incorporation installée

Pour les packages qui sont limités par une incorporation, la meilleure pratique consiste à mettre à jour le package d'incorporation, en conservant tous les packages contraints dans un ensemble testé de manière groupée.

Si vous voulez quand même mettre à jour un seul package à partir d'une incorporation, vérifiez si le package a une facette `version-lock` définie sur `true`. Si un package est associé à une facette `version-lock`, vous devez le déverrouiller à partir de son incorporation. Définissez la facette `version-lock` sur `false` pour supprimer la contrainte, puis réessayez d'installer ou de mettre à jour le package. Spécifiez les options `-nv` et la version du package souhaitée dans le FMRI du nom d'un package. Voir [“Assouplissement des contraintes de version spécifiées par les incorporations” à la page 89](#).

### EXEMPLE A-1 Déverrouillage et mise à jour de l'environnement Java Runtime

L'exemple suivant montre comment mettre à jour le package `runtime/java/jre-7`. Le package `jre-7` est limité par le package `consolidation/java/java-incorporation`, et le package `java-incorporation` est à son tour sous contraintes par le package d'incorporation `pkg:/entire`.

La commande suivante indique que la version `0.175.2.0.0.9.0` de `jre-7` est actuellement installée et montre qu'une version plus récente est disponible à partir des référentiels de packages configurés :

```
$ pkg list -af runtime/java/jre-7
NAME (PUBLISHER)                VERSION                IFO
runtime/java/jre-7              1.7.0.21-0.175.2.0.0.13.0  ---
runtime/java/jre-7              1.7.0.17-0.175.2.0.0.9.0   i--
```

La suppression de l'option -f permet de voir quelles versions sont disponibles pour mise à jour. La sortie `pkg list` suivante indique qu'aucune version plus récente ne peut être installée dans cette image, et la sortie de la commande `pkg update` confirme cet état. L'option -n indique quelles sont les éventuelles modifications faites mais n'ont aucune incidence.

```
$ pkg list -a runtime/java/jre-7
NAME (PUBLISHER)                VERSION                IFO
runtime/java/jre-7              1.7.0.17-0.175.2.0.0.9.0   i--
$ pkg update -nv runtime/java/jre-7
No updates available for this image.
```

Pour afficher davantage d'informations sur les raisons pour lesquelles ce package ne peut pas être mis à jour, indiquez la version vers laquelle vous souhaitez mettre à jour. La sortie illustrée dans l'exemple suivant indique que le package `java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0` installé ne permet pas l'installation du package `jre-7@1.7.0.21-0.175.2.0.0.13.0`. Le package `java-incorporation@0.5.11,5.11-0.175.2.0.0.13.0` permettrait l'installation du package `jre-7@1.7.0.21-0.175.2.0.0.13.0`, mais le package d'incorporation installé `entire@0.5.11,5.11-0.175.2.0.0.12.0` ne permet pas l'installation du package `jre-7@1.7.0.21-0.175.2.0.0.13.0`.

```
$ pkg update -nv runtime/java/jre-7@1.7.0.21-0.175.2.0.0.13.0
pkg update: No solution was found to satisfy constraints
```

```
maintained incorporations:
[output omitted]
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z
```

```
Plan Creation: dependency error(s) in proposed packages:
[output omitted]
```

```
No suitable version of required package pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z found:
  Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z
  Reason: All versions matching 'incorporate' dependency pkg://runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0 are rejected
  Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z
  Reason: This version excluded by specified installation version
  Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z
  Reason: This version excluded by specified installation version
  Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145534Z
  Reason: This version is excluded by installed incorporation pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z
```

```
Plan Creation: Errors in installed packages due to proposed changes:
[output omitted]
No suitable version of installed package pkg://solaris/consolidation/java/java
-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z found
  Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.1
75.2.0.0.9.0:20130304T213946Z
  Reason: All versions matching 'incorporate' dependency pkg://runtime/java/jr
e-7@1.7.0.17,5.11-0.175.2.0.0.9.0 are rejected
  Reject: pkg://solaris/runtime/java/jre-7@1.7.0.17,5.11-0.175.2.0.0.9.0:20
130304T214022Z
  Reason: This version excluded by specified installation version
  Reject: pkg://solaris/consolidation/java/java-incorporation@0.5.11,5.11-0.1
75.2.0.0.13.0:20130429T145534Z
  Reason: This version is excluded by installed incorporation pkg://solaris/e
ntire@0.5.11,5.11-0.175.2.0.0.12.0:20130415T172730Z
```

La meilleure pratique consiste à mettre à jour le package `entire`. La mise à jour du package `entire` mettrait à jour de la mise à jour le package `java-incorporation`, ce qui permettrait de mettre à jour le package `jre-7`. Dans cet exemple, vous devez mettre à jour les packages Java et vous ne pouvez pas transférer votre image vers une version mise à jour de `entire`.

La version du logiciel possible d'installer est soumise aux contraintes de la définition de la facette `version-lock` pour le package d'incorporation Java. Pour mettre à jour le logiciel sans mettre à jour d'autres logiciels Java, déverrouillez la facette `version-lock`, puis mettez à jour le package d'incorporation Java. Pour plus d'informations sur les facettes `version-lock`, reportez-vous [“Assouplissement des contraintes de version spécifiées par les incorporations” à la page 89](#).

La commande suivante remplace la valeur de la facette `version-lock` du package `java-incorporation` installé sur `false`. Le nombre de packages à mettre à jour est le nombre de packages installés dans cette image, car chaque package de la facette est coché pour cette image.

```
$ pkg change-facet \  
facet.version-lock.consolidation/java/java-incorporation=false  
  Packages to update: 856  
  Variants/Facets to change: 1  
  Create boot environment: No  
  Create backup boot environment: Yes  
  
Planning linked: 1/1 done  
PHASE                                ITEMS  
Removing old actions                  1/1  
Updating image state                  Done  
Creating fast lookup database         Done  
Reading search index                  Done  
Building new search index             856/856
```

La commande suivante indique que la valeur de facette a été modifiée :

```
$ pkg facet
```

```
FACETS                                VALUE
facet.version-lock.consolidation/java/java-incorporation False
```

Vu que la commande suivante indique l'option -n, cette commande affiche ce qui changerait, mais n'apporte pas réellement de modifications à votre image.

```
$ pkg update -nv java-incorporation
    Packages to update:      2
    Estimated space available: 80.91 GB
    Estimated space to be consumed: 687.28 MB
    Create boot environment: No
    Create backup boot environment: Yes
    Rebuild boot archive:   No

Changed packages:
solaris
  consolidation/java/java-incorporation
    0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z -> 0.5.11,5.11-0.175.2.0.0.13.0
:20130429T145534Z
  runtime/java/jre-7
    1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z -> 1.7.0.21,5.11-0.175.2.0.0.
13.0:20130429T145626Z
```

La commande suivante permet de lancer la mise à jour. Cette commande effectue la mise à jour dans l'image actuelle. Vous pouvez être amené à utiliser l'option --be-name pour effectuer la mise à jour dans un nouvel environnement d'initialisation.

```
$ pkg update -v java-incorporation
    Packages to update:      2
    Estimated space available: 80.91 GB
    Estimated space to be consumed: 687.28 MB
    Create boot environment: No
    Create backup boot environment: Yes
    Rebuild boot archive:   No

Changed packages:
solaris
  consolidation/java/java-incorporation
    0.5.11,5.11-0.175.2.0.0.9.0:20130304T213946Z -> 0.5.11,5.11-0.175.2.0.0.13.0
:20130429T145534Z
  runtime/java/jre-7
    1.7.0.17,5.11-0.175.2.0.0.9.0:20130304T214022Z -> 1.7.0.21,5.11-0.175.2.0.0.
13.0:20130429T145626Z

DOWNLOAD                                PKGS      FILES    XFER (MB)   SPEED
Completed                                2/2       171/171    61.9/61.9   0B/s
```

```
PHASE                                ITEMS
Removing old actions                   7/7
Installing new actions                  6/6
Updating modified actions               170/170
Updating package state database         Done
Updating package cache                  2/2
Updating image state                    Done
Creating fast lookup database           Done
```

```
Reading search index          Done
Updating search index         2/2
```

La commande suivante permet de vérifier que le package `jre-7` est mis à jour dans cette image. Si vous effectuez la mise à jour dans un nouvel environnement d'initialisation, utilisez les commandes `beadm mount` et `pkg -R` pour effectuer la vérification dans ce nouvel environnement d'initialisation.

```
$ pkg list jre-7
NAME (PUBLISHER)          VERSION          IFO
runtime/java/jre-7      1.7.0.21-0.175.2.0.0.13.0  i--
```

## Mise à jour d'une incorporation quand une dépendance appropriée ne peut pas être trouvée

Message d'erreur : A version for 'incorporate' dependency cannot be found

Reportez-vous à [“Les packages d'incorporation” à la page 12](#) pour plus d'informations sur les packages d'incorporation et leurs dépendances `incorporate`.

Les exemples de raisons pour lesquels un package d'incorporation peut échouer à être mis à jour comprend les états d'installation suivants d'une dépendance `incorporate` du package d'incorporation :

- Le package dépendant est figé à une autre version.
- Le package dépendant est déjà installé dans une version supérieure.
- Le package dépendant est installé à partir d'un autre éditeur, et cet éditeur est résident ("sticky").

**EXEMPLE A-2** Mise à jour de `pkg:/entire` : quand une dépendance est déverrouillée et mise à jour séparément.

L'exemple suivant tente de mettre à jour tous les packages installés car aucun nom de package n'est indiqué. L'un des packages installés que cette opération tente de mettre à jour est le package d'incorporation `pkg:/entire`. Cet exemple montre une dépendance `incorporate` de `pkg:/entire` qui est déjà installé dans une version supérieure.

```
$ pkg update --be-name s11.2
Creating Plan (Solver setup): /
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to latest available versions.
This may indicate an overly constrained set of packages are installed.
```

```
latest incorporations:
[output omitted]
```

```
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
```

The following indicates why the system cannot update to the latest version:

```
No suitable version of required package pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z found:
Reject: pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
Reason: A version for 'incorporate' dependency on pkg://consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0 cannot be found
```

Ce message précise que la version du package d'incorporation `pkg:/entire` que le système a tenté de mettre spécifie une version du package `ub_javavm-incorporation` impossible à installer. Un package ne peut pas être installé car aucun package n'est installé et la mise à jour échoue.

Les techniques suivantes peuvent fournir davantage d'informations concernant les raisons de l'échec de l'installation du package `ub_javavm-incorporation` :

- Utilisez les options `-v`. Par exemple, utilisez `-v` ou `-vv` pour obtenir une sortie verbose plus détaillée.
- Indiquer le package à mettre à jour. En général, en indiquant plus d'informations, on obtient des messages plus détaillés. Par exemple, outre le nom de la version, incluez le package dans le package FMRI.

La commande suivante inclut une option `-v` et spécifie de mettre à jour le package `entire@0.5.11,5.11-0.175.2.0.0.10.0` copié depuis le message ci-dessus. Cette commande spécifie également l'option `-n` au lieu de l'option `--be-name`. L'option `-n` montre ce qui serait fait, sans pour autant apporter des modifications à cette image.

```
$ pkg update -nv entire@0.5.11,5.11-0.175.2.0.0.10.0
Creating Plan (Solver setup): /
pkg update: No matching version of entire can be installed:
Reject: pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.10.0:20130318T181506Z
Reason: All versions matching 'require' dependency pkg://consolidation/ub_javavm/ub_javavm-incorporation are rejected
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.151.0.1:20101105T053418Z
pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0.0.0.2.0:20111019T144756Z
pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0.10.1.0.0:20120920T143020Z
Reason: Excluded by proposed incorporation 'entire'
Newer version pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z is already installed
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0:20130304T213739Z
Reason: Newer version pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z is already installed
Reject: pkg://solaris/consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.13.0:20130429T145201Z
Reason: Excluded by proposed incorporation 'entire'
```

Ces messages indiquent que la version du package `ub_javavm-incorporation` actuellement installé dans cette image est plus récente que celle spécifiée par le package d'incorporation `ub_javavm-incorporation` qui serait installé par l'opération de mise à jour.

La commande suivante affiche la version du package d'incorporation `ub_javavm-incorporation` actuellement installé.

```
$ pkg list ub_javavm-incorporation
NAME (PUBLISHER)                                VERSION                                IFO
consolidation/ub_javavm/ub_javavm-incorporation 0.5.11-0.175.2.0.0.13.0             i--
```

La commande suivante montre la version du package `ub_javavm-incorporation` spécifié par le package d'incorporation `pkg:/entire` que le système a tenté de mettre à jour. La version du package d'incorporation `pkg:/entire` que le système a tenté de mettre à jour est copiée à partir du premier message "Rejeter" dans la sortie ci-dessus. L'option `-r` cherche ce package dans les référentiels de packages configuré, et non pas dans les référentiels de packages de l'image préinstallée.

```
$ pkg contents -Hrt depend \
-a facet.version-lock.consolidation/ub_javavm/ub_javavm-incorporation=true \
-o fmri entire@0.5.11,5.11-0.175.2.0.0.10.0
consolidation/ub_javavm/ub_javavm-incorporation@0.5.11-0.175.2.0.0.9.0
```

Pour résoudre ce problème, vous pouvez également demander à l'opération de mise à jour de mettre à jour tous les packages, à l'exception de packages spécifiés spécialement. Utiliser une ou plusieurs options `--reject` de la commande `pkg update` pour effectuer la mise à jour sans tentative de mise à jour des packages spécifiés dans les options `--reject`. Il est possible d'utiliser des caractères génériques dans les arguments `--reject`. Dans la commande suivante, les packages à rejeter sont copiés à partir des messages "Reject" qui précèdent le message "Reason: Newer version is already installed".

```
$ pkg update -v --be-name s11.2 \
--reject 'consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.151*' \
--reject 'consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.0*' \
--reject 'consolidation/ub_javavm/ub_javavm-incorporation@0.5.11,5.11-0.175.2.0.0.9.0*'
```

---

**Astuce** - Soyez vigilant lorsque vous spécifiez des arguments `--reject` : tous les packages qui correspondent à l'argument `--reject` et qui sont déjà installés seront désinstallés.

---

## Mise à jour d'une incorporation lorsqu'une dépendance installée n'est pas autorisée

Messages d'erreur :

- The installed package *package* is not permissible

- Excluded by proposed incorporation

Si vous recevez ces messages d'erreur, le *package* est vraisemblablement un package clé du système d'exploitation qui doit rester synchronisé avec d'autres packages du système d'exploitation. Utilisez la commande `pkg facet` pour contrôler la valeur de la facette `facet.version-lock.package`. Si la valeur de cette facette `version-lock` est `false`, utilisez la commande `pkg change-facet` pour modifier la valeur de cette facette sur `true` et réexécutez la commande de mise à jour à nouveau.

## Package requis introuvable

Message d'erreur: A version for 'require' dependency cannot be found

Si vous recevez un message comme le suivant indiquant qu'un package obligatoire est introuvable, vérifiez si le package est disponible à partir des éditeurs configurés.

```
pkg update: No solution was found to satisfy constraints
Plan Creation: Package solver has not found a solution to update to
latest available versions.
This may indicate an overly constrained set of packages are installed.
[output omitted]
No suitable version of required package package1 found:
Reject: package1
Reason: A version for 'require' dependency package2 cannot be found
```

Utilisez la commande suivante afin d'indiquer si *package2* est disponible à partir de vos éditeurs de packages configurés :

```
$ pkg list -afv package2
```

Utilisez la commande `pkg publisher` pour vérifier l'origine de votre éditeur. L'éditeur de ce package est indiqué après `pkg://` en mode Full FMRI. Vous pouvez être amené à modifier par l'emplacement d'origine de l'éditeur. Si l'emplacement est un référentiel de packages local, il vous faudra peut-être mettre à jour le référentiel.

## Package requis rejeté

Messages d'erreur :

- No solution was found to satisfy constraints
- All versions matching 'require' dependency *package* are rejected

Dans la sortie suivante à partir de la commande `pkg update`, le message d'erreur initial précise qu'une version appropriée du package `desktop-incorporation` requis est introuvable. La raison pour laquelle le package `desktop-incorporation` ne convient pas est que l'un de ses

packages dépendants est introuvable. Dans la mesure où aucune version appropriée n'est trouvée, le package desktop-incorporation est rejeté. pkg update échoue car le package desktop-incorporation est requis par l'incorporation pkg:/entire. La commande suivante indique que le package desktop-incorporation est requis par l'incorporation pkg:/entire.

```
$ pkg search -Hlo pkg.name require:consolidation/desktop/desktop-incorporation
```

Le message d'erreur suivant précise qu'il est impossible de trouver une version appropriée du package requis python-extra-26. La raison pour laquelle le package python-extra-26 ne convient pas est que le package python-extra-26 nécessite le package desktop-incorporation, et qu'aucune version appropriée desktop-incorporation n'a été trouvée.

Cette information indique que la solution à cet échec de la mise à jour est de trouver une procédure d'installation de la version requise du package desktop-incorporation.

```
pkg update: No solution was found to satisfy constraints
```

```
maintained incorporations:  
[output omitted]
```

```
Plan Creation: dependency error(s) in proposed packages:  
[output omitted]
```

```
No suitable version of required package pkg://solaris/consolidation/desktop  
/desktop-incorporation@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z found:  
Reject: pkg://solaris/consolidation/desktop/desktop-incorporation  
@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z  
Reason: A version for 'incorporate' dependency on pkg://library/python-2  
/python-sexy-26@0.1.9-0.175.0.0.0.1.0 cannot be found  
No suitable version of required package pkg://solaris/library/python-2  
/python-extra-26@2.6.4-0.175.1.0.0.15.0:201205014T200156Z found:  
Reject: pkg://solaris/library/python-2/python-extra-26@  
2.6.4-0.175.1.0.0.15.0:201205014T200156Z  
Reason: All versions matching 'require' dependency pkg://consolidation  
/desktop/desktop-incorporation are rejected  
Reject: pkg://solaris/consolidation/desktop/desktop-incorporation  
@0.5.11,5.11-0.175.0.0.0.2.0:20111019T132128Z  
[output omitted]  
pkg://solaris/consolidation/desktop/desktop-incorporation  
@0.5.11,5.11-0.175.2.0.0.26.0:20131028T145233Z
```

Les commandes suivantes indiquent que le package ne peut pas être trouvé pour le package desktop-incorporation requis n'est pas obligatoire. Le package python-sexy-26 est incorporé par le package desktop-incorporation, mais cela n'est pas obligatoire.

```
$ pkg search -Hlo pkg.name require:library/python-2/python-sexy-26  
$ pkg search -Hlo pkg.name incorporate:library/python-2/python-sexy-26  
consolidation/desktop/desktop-incorporation
```

La dépendance incorporate précise que si le package python-sexy-26 est installé, il doit être installé dans la version spécifiée. Toutefois, vu qu'aucun package n'exige le package python-sexy-26, le package python-sexy-26 n'a pas besoin d'être installé. Par conséquent, l'échec de la

mise à jour l'une des solutions à ce problème consiste à désinstaller le package `python-sexy-26`. Une autre version de ce package est actuellement installée, la mise à jour n'a pas pu localiser la version spécifiée par l'incorporation `desktop-incorporation`. Si vous le désinstallez le package, le processus de mise à jour n'a pas besoin de trouver le package mis à jour.

Si vous avez besoin du package `python-sexy-26`, et que vous ne souhaitez pas le désinstaller, trouvez un référentiel de packages qui fournit `pkg:/library/python-2 /python-sexy-26@0.1.9-0.175.0.0.0.1.0`. Utilisez la commande `pkg set-publisher` afin d'ajouter à votre origine de l'éditeur ce référentiel ou à l'aide de la commande `pkgrecv` pour ajouter ce package jeu à votre origine de l'éditeur actuellement configurée.

## Packages non mis à jour comme prévu

Message d'erreur: `pkg update: The installed package package is not permissible.`

Si vous utilisez des caractères génériques avec une opération de mise à jour, vous risquez de ne pas voir les messages d'erreur, même si certaines de livraison qui auraient dû être mis à jour les packages n'ont pas été mises à jour. Vous pourrez être en présence d'un message d'erreur si vous avez indiqué un nom de package sans caractères génériques.

Par exemple, l'opération suivante peut se terminer sans erreur sans que les packages soient mis à jour pour autant, même si vous avez vérifié que de nouveaux packages sont disponibles auprès de vos éditeurs configurés :

```
$ pkg update '*'
```

Pour obtenir des informations supplémentaires, au lieu d'utiliser le caractère générique, spécifiez la version de `pkg:/entire` que vous voulez mettre à jour :

```
$ pkg list -Hafv entire
pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.9.0:20130304T214506Z ---
$ pkg update -nv pkg://solaris/entire@0.5.11,5.11-0.175.2.0.0.9.0:20130304T214506Z
Creating Plan (Solver setup): -
pkg update: The installed package compress/zip is not permissible.
  Reject: pkg://solaris/compress/zip@3.0,5.11-0.175.2.0.0.7.0:20121119T070339Z
  Reason: Excluded by proposed incorporation 'consolidation/userland/userland-incorporation'
```

Dans cet exemple, le package `compress/zip` empêche la mise à jour de package d'incorporation `userland-incorporation` qui empêche la mise à jour de l'incorporation `pkg:/entire`. La commande suivante vous permet d'afficher plus d'informations sur le package `compress/zip` :

```
$ pkg list compress/zip
NAME (PUBLISHER) VERSION          IFO
compress/zip    3.0-5.11-0.175.2.0.0.7.0 if-
```

Cette sortie indique que le package `compress/zip` est figé. Ce package ne peut pas être mis à jour car il est figé.

```
$ pkg unfreeze compress/zip
compress/zip was unfrozen.
```

Avec le package `compress/zip` débloqué, l'opération initiale `pkg update '*'` doit mettre à jour tous les packages dans l'image pour lesquels des mises à jour sont disponibles. Reportez-vous à [“Verrouillage de packages en une version spécifiée” à la page 87](#) pour plus d'informations sur les commandes `pkg freeze` et `pkg unfreeze`.

L'erreur `pkg sync-linked` affichée dans [“Impossible d'installer le package lié de synchronisation” à la page 128](#) est similaire, mais indique qu'il s'agit d'un package dans une zone non globale qui empêche la mise à jour.

## Impossible d'installer le package lié de synchronisation

```
Message d'erreur: pkg sync-linked: The installed package package is not
permissible.
```

Un package `sync-linked` est un package dans une zone non globale. Un package peut être rejeté au motif d'une incohérence de version avec l'image parent.

```
Linked progress: -pkg: update failed (linked image exception(s)):
```

```
A 'sync-linked' operation failed for child 'zone:z1' with an unexpected
return value of 1 and generated the following output:
```

```
pkg sync-linked: The installed package package is not permissible.
Reject: package
Reason: Parent image has a incompatible newer version: package
```

Ce message d'incompatibilité peut s'afficher pour les raisons suivantes :

- Les noyaux de système d'exploitation doivent présenter le même niveau de version dans les zones non globales comme dans la zone globale. Vous ne pouvez pas mettre à jour ces packages séparément dans une zone non globale. De même, figer un de ces packages dans une zone non globale entraîne l'échec de la mise à jour pour la zone globale et toutes les zones non globales.

La commande suivante vous permet d'afficher la liste des packages devant rester en synchronisation entre la zone globale et zones non globales :

```
$ pkg search -o pkg.name :depend:parent:
```

- Si vous mettez à jour un BE dont des zones sont configurées BE, il est impossible de monter ce BE en question, et vous devez utiliser la commande `pkg -R` de mise à jour de ce BE de remplacement (ABE), dans la mesure où la configuration de l'éditeur dans l'environnement ABE est différente de la configuration du BE actuel. Des zones non globales dans la configuration de l'éditeur l'ABE utiliser le BE à partir de celui qui est actif.

## Impossible d'installer des zones non globales

Message d'erreur: The following pattern(s) did not match any allowable packages.  
Try using a different matching pattern, or refreshing publisher information

Les noyaux de système d'exploitation doivent présenter le même niveau de version dans les zones non globales comme dans la zone globale. Si l'origine de l'éditeur solaris dans cette image est définie sur un référentiel de packages qui ne contient pas les mêmes versions des packages système qui sont installés dans la zone globale, essayer d'installer une zone non globale génère l'erreur suivante :

```
$ zoneadm -z myzone install
The following ZFS file system(s) have been created:
  rpool/export/zones/myzone
Progress being logged to /var/log/zones/zoneadm.20131104T181301Z.myzone.install
Image: Preparing at /export/zones/myzone/root.

AI Manifest: /tmp/manifest.xml.9daq.i
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
  Zonename: myzone
Installation: Starting ...

      Creating IPS image
Startup linked: 1/1 done
Installing packages from:
  solaris
      origin: http://pkg.oracle.com/solaris/release/
Error occurred during execution of 'generated-transfer-4606-1' checkpoint.
Failed Checkpoints:

Checkpoint execution error:

      The following pattern(s) did not match any allowable packages. Try
      using a different matching pattern, or refreshing publisher information:

Installation: Failed. See install log at /system/volatile/install.4606/install_log
ERROR: auto-install failed.
```

Pour installer une zone non globale, le référentiel que vous avez défini comme éditeur solaris d'origine doit contenir au moins le même système logiciel installé dans la zone globale sur lequel vous installez la zone non globale. Le référentiel peut également contenir des versions plus anciennes ou plus récentes du logiciel, mais il doit contenir le même logiciel que celui de la zone globale. La commande suivante indique que le référentiel /export/IPSpkgrepos/Solaris est une origine de l'éditeur appropriée pour cette zone globale, car ce référentiel contient la même version du package pkg:/entire que celui installé dans la zone globale :

```
$ pkg list entire
NAME (PUBLISHER)      VERSION                               IFO
entire                0.5.11-0.175.2.0.0.26.0             i--
$ pkgrepo list -H -s /export/IPSpkgrepos/Solaris \
```

```
entire@0.5.11-0.175.2.0.0.26.0
solaris      entire      0.5.11-0.175.2.0.0.26.0:20131028T190148Z
$ pkg set-publisher -G '*' -M '*' -g /export/IPSpkgrepos/Solaris/ solaris
```

## Impossible de modifier l'image

Message d'erreur : pkg: The image cannot be modified as it is currently in use by another package client

Le message d'erreur doit inclure le nom et le pid du client possédant l'image verrouillé, comme indiqué dans l'exemple suivant :

```
pkg: The image cannot be modified as it is currently in use by another package
client: pkg on cbusl0406l, pid 26604.
```

Relancez la commande pkg dès que le processus en cours (26604 dans cet exemple) s'arrête. Si le processus ne se termine pas rapidement, utilisez, par exemple, `pt ree -a` pour commencer à examiner le processus.

Le processus de package en cours sur cette image peut être le Gestionnaire de mises à jour qui vérifie si des mises à jour sont disponibles. Reportez-vous à [“Utilisation du Gestionnaire de mise à jour”](#) à la page 136 pour plus d'informations sur la tâche de Gestionnaire de mise à jour cron et le service pkg/update.

## Les fichiers ont été récupérés

Le message d'information suivant est suivi du chemin d'accès des fichiers qui ont été récupérés et l'emplacement temporaire dans lequel les fichiers ont été déplacés :

```
The following unexpected or editable files and directories were
salvaged while executing the requested package operation; they
have been moved to the displayed location in the image:
```

Il s'agit d'un message d'information et il peut être ignoré ou faire l'objet d'une action si nécessaire.

Les répertoires sont des références prises en compte dans IPS. Lorsque le dernier package installé dans l'image faisant explicitement ou implicitement référence à un répertoire ne le fait plus, ce répertoire est supprimé. Si ce répertoire contient des objets système de fichiers non empaquetés, ces éléments sont déplacés vers `$IMAGE_META/lost+found`. Les objets de système de fichiers non empaquetés qui sont des fichiers et répertoires IPS n'ont pas été livrés par un package. La valeur de `IMAGE_META` est en général `/var/pkg`. Reportez-vous à la section Fichiers de la page de manuel pkg (5) pour plus d'informations sur le répertoire `IMAGE_META`.

## Minimiser les métadonnées d'images stockées

Le répertoire `/var/pkg` regroupe l'ensemble de métadonnées pour l'image. Ce répertoire peut être assez volumineux. Ne supprimez rien manuellement dans le répertoire `/var/pkg`.

Assurez-vous que la valeur de la propriété de l'image `flush-content-cache-on-success` est paramétrée sur `true`. La valeur de la propriété `flush-content-cache-on-success` est définie, par défaut sur `true`. Lorsque la valeur de `flush-content-cache-on-success` est `true`, les fichiers mis en cache sont enlevés lorsque des opérations `pkg install` et `pkg update` terminent avec succès. Si la propriété `flush-content-cache-on-success` est définie sur `false`, vous pouvez utiliser la commande illustre l'exemple suivant pour réinitialiser la valeur à `true` :

```
$ pkg property flush-content-cache-on-success
PROPERTY          VALUE
flush-content-cache-on-success False
$ pkg set-property flush-content-cache-on-success true
$ pkg property -H flush-content-cache-on-success
flush-content-cache-on-success True
```

Suite au vidage de la mémoire cache du contenu (définir `flush-content-cache-on-success` sur `true`) certaines opérations `pkg` peuvent prendre plus de temps.

Les zones non globales disposent un autre cache que vous pourrez définir avec l'option `-c` de `/usr/lib/pkg.sysrepo`. Vous pouvez définir la taille maximale du cache avec l'option `-s`. Consultez la page de manuel [pkg.sysrepo\(1M\)](#).

## Augmenter les performances d'installation d'un package

Les étapes ci-après peuvent vous aider à améliorer les performances, à savoir : package d'installation et de mise à jour

- Assurez-vous que votre capacité du pool de stockage ZFS est inférieure à 80%.

```
$ zpool list
NAME  SIZE ALLOC FREE CAP DEDUP HEALTH ALROOT
rpool 186G 75.2G 111G 40% 1.00x ONLINE -
```

- Utilisez un référentiel de packages local. Reportez-vous à la section “[Copie et création de référentiels de packages dans Oracle Solaris 11.2](#)”.
- Si `http_proxy` est défini, consultez si les performances du proxy.



## Interfaces graphiques utilisateur IPS

---

IPS inclut deux outils d'interface graphique.

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

### Utilisation du Gestionnaire de packages

Le Gestionnaire des 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 environnements d'initialisation

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

|                   |                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Barre d'outils    | Cliquez sur l'icône dans la barre d'outils Gestionnaire de packages. L'icône du représente une boîte Gestionnaire de packages entourée d'une flèche circulaire. |
| Icône Bureau      | Cliquez deux fois sur l'icône du sur le bureau Gestionnaire de packages.                                                                                        |
| Barre de menus    | Sélectionnez Système -> Administration -> Gestionnaire de packages                                                                                              |
| Ligne de commande | <b>\$ packagemanager &amp;</b>                                                                                                                                  |

Pour consulter la documentation complète du Gestionnaire de packages, sélectionnez Aide->Contenu dans la barre de menus du Gestionnaire de packages.

## Options de ligne de commande Gestionnaire de packages

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

**TABLEAU B-1** Options de la commande 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.<br><br>La commande suivante s'exécute sur l'image stockée au niveau de <code>/aux0/example_root</code> :<br><br><b>\$ packagemanager -R /aux0/example_root</b>                                                                                       |
| <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 mise à jour" à la page 136. |
| <code>--info-install</code> ou <code>-i file.p5i</code> | Indiquez un fichier <code>.p5i</code> pour exécuter le gestionnaire de packages en mode d'installation Web. Le fichier spécifié doit avoir l'extension <code>.p5i</code> . Pour plus d'informations, reportez-vous à la section "Utilisation de l'installation Web" à la page 134.                                                                                    |
| <code>--help</code> ou <code>-h</code>                  | Afficher les informations d'utilisation de la commande.                                                                                                                                                                                                                                                                                                               |

## Utilisation de l'installation Web

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

Prend en charge l'installation de packages Gestionnaire de packages simple en un clic à l'aide d'un processus d'installation Web. 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 installation Web.

### ▼ Création d'un fichier d'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.

**1. Sélectionnez l'éditeur.**

A partir du menu déroulant Gestionnaire de packages, sélectionnez l'éditeur dont vous souhaitez inclure les packages dans le fichier .p5i.

**2. Sélectionner le package.**

Dans le volet de la liste des packages gestionnaire des packages sélectionnez le package dont vous souhaitez distribuer les instructions d'installation.

**3. Exporter vos sélections.**

Sélectionnez Fichier -> Exporter les sélections pour afficher la fenêtre Confirmation des sélections d'exportation.

**4. Confirmez les sélections.**

Cliquez sur le bouton OK pour confirmer les sélections. La fenêtre Export Selections (Exporter les sélections) s'affiche.

**5. (Facultatif) Modifiez le nom du fichier.**

Un nom par défaut est attribué au fichier .p5i. Vous pouvez modifier le nom du fichier.

Ne modifiez pas l'extension .p5i.

**6. (Facultatif) Modifiez l'emplacement du fichier.**

Un emplacement par défaut est attribué au fichier .p5i. Vous pouvez modifier l'emplacement.

**7. Enregistrer le fichier d'installation Web.**

Cliquez sur le bouton Save (Enregistrer) pour enregistrer le nom de fichier et l'emplacement.

**▼ Procédure d'utilisation et d'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. Démarrez le Gestionnaire de packages en mode d'installation Web.**

- **Sélectionnez un fichier .p5i sur votre bureau.**

- **Démarrez le Gestionnaire de packages à partir de la ligne de commande et spécifiez un fichier .p5i.**

```
$ packagemanager ./wifile.p5i
```

- **Accédez 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 sélectionnez ce dernier.

La fenêtre Install/Update (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 vont être ajoutés à votre système.) Les éditeurs et les packages à installer apparaissent dans une liste.

**2. Cliquez sur le bouton Continue (Continuer) pour poursuivre l'installation.**

- Si nécessaire, ajoutez des éditeurs.
- Si nécessaire,

**3. Si nécessaire, ajoutez des éditeurs.**

Si l'éditeur de packages spécifié n'est pas encore configuré sur votre système, la fenêtre Add Publisher (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.

Si l'éditeur a bien été ajouté, la boîte de dialogue Adding Publisher Complete (Ajout de l'éditeur terminé) s'affiche.

**4. Cliquez sur le bouton OK pour poursuivre l'installation.**

**5. Activer un éditeur désactivé.**

Si un fichier .p5i contient des packages provenant d'un éditeur désactivé, Web Install ouvre une boîte de dialogue Activer l'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 Gestionnaire de packages.

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

## Utilisation du Gestionnaire de mise à jour

Le Gestionnaire de mise à jour met à jour tous les packages installés vers la version la plus récente autorisée par les contraintes imposées au système par les dépendances des packages installés et la configuration de l'éditeur. Cette fonction est la même que les fonctions suivantes :

- Dans l'interface du Gestionnaire de packages, sélectionnez le bouton Updates (Mises à jour) ou l'option de menu Package->Updates (Package-> Mises à jour).

- Utilisez la commande `packagemanager`.

```
$ 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. Le Gestionnaire des mises à jour icône représente une pile de trois boîtes.

Barre de menus                Choisissez Système -> Administration -> Update Manager.

Ligne de commande            \$ **pm-updatemanager**

Automatisé                    Le package du gestionnaire de mise à jour, `package/pkg/update-manager`, fournit la tâche cron, `/usr/lib/update-manager/update-refresh.sh`.

```
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
```

Lorsque le service SMF `svc:/application/pkg/update` est en ligne, ce travail cron vérifie régulièrement les packages mis à jour disponibles à partir des éditeurs configurés (les deux premières étapes du processus suivant). Si des packages peuvent être mis à jour, une notification s'affiche dans la barre d'outils de votre bureau. Sélectionnez l'icône de notification GUI pour ouvrir le Gestionnaire des mises à jour.

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 sont 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 Cancel (Annuler) pour annuler la mise à jour.

Cliquez sur le bouton Continue (Continuer) pour poursuivre la mise à jour.

3. 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/update-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 permanent, une version plus récente du package compatible avec cette image peut être installée à partir d'un autre éditeur. Utilisez la Package Manager Manage Publishers ou la commande `pkg set-publisher` pour définir un éditeur comme résident ou non permanent.

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

- Un nouvel environnement d'initialisation peut être créé, en fonction des packages qui sont mis à jour et en fonction de votre stratégie d'image.

Si le système a créé un environnement d'initialisation, vous pouvez modifier le nom de l'environnement d'initialisation par défaut.

Vous devez redémarrer pour initialiser dans le nouvel environnement d'initialisation. Le nouvel environnement d'initialisation sera votre choix d'initialisation par défaut. Votre environnement d'initialisation actuel sera disponible en tant que choix d'initialisation alternatif.

- Cliquez sur le bouton Redémarrer maintenant pour redémarrer votre système immédiatement.
- Cliquez sur le bouton Restart Later (Redémarrer ultérieurement) pour redémarrer votre système à une date ultérieure.

## Options le Gestionnaire de mises à jour de ligne de commande

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

**TABLEAU B-2** Le Gestionnaire de mises à jour d'options de commande

| 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.<br><br>La commande suivante met à jour l'image au niveau de <code>/aux0/example_root</code> :<br><br>\$ <code>pm-updatemanager -R /aux0/example_root</code> |

| <b>Option</b> | <b>Description</b>                                      |
|---------------|---------------------------------------------------------|
| - -help ou -h | Afficher les informations d'utilisation de la commande. |



# Index

---

## A

aperçupkg commandes, 37  
archive de package, 16, 45

## B

BE, 17  
activer, 40  
création pendant l'installation du package, 46  
définition de la propriété de la stratégie de l'image, 96  
exiger, 40  
nommer, 40  
pkg options de commandes, 40

## C

commande pkg revert , 53  
commande beadm , 47  
commande pkg change-facet, 89  
commande pkg contents, 27  
    comparer avec pkg search, 30  
commande pkg freeze , 87  
commande pkg info, 25  
commande pkg mediator , 91  
commande pkg refresh , 43  
commande pkg search, 30  
    comparer avec pkg contents, 30  
commande pkg set-mediator , 92  
commande pkg set-publisher , 81  
commande pkg unfreeze, 87  
commande pkg uninstall , 55  
commande pkg unset-mediator , 92  
composants de chaîne de version, 14

contenu du  
    système de fichiers du package, 27  
contraintes, 12, 117  
    s'affranchir, 89  
contraintes d'installation, 12, 117  
contraintes de mise à jour, 117  
    package d'incorporation ent ire, 12  
contraintes de version, 12, 117  
    s'affranchir, 89

## D

description  
de packages, 25

## E

éditeur, 16, 75  
    configuration, 77  
    origine, 76  
    propriétés, 96  
    utilisation de packages signés, 99  
    proxy, 81  
éditeur de package Voir éditeur  
ent ire package d'incorporation, 13  
environnement d'initialisation (BE), 17  
éviter liste, 94

## F

facette, 18, 83, 89  
facette pkg commande, 86  
Fault Management Resource Identifier (FMRI), 14  
fichiers  
    préserver ou écraser éditables, 39  
FMRI, 14

## G

Gestionnaire de package, 133  
groupe, 13, 29, 35, 94

## H

historique pkg commande, 105

## I

image, 17  
  création, 103  
  mise à niveau, 63  
  propriétés, 96  
    ajouter et supprimer, 100  
    stratégie de création BE, 96  
    utilisation de packages signés, 98  
  réinstallation, 55  
  stratégies, 96  
image enfant, 17  
image liée, 17  
image parent, 17  
image reliée, 41, 60  
  pkg uninstall, 55  
  pkg update, 49  
incorporation, 12, 89, 117  
index de recherche, 21, 43  
installation Web, 134

## L

la commande pkg info  
  est comparée avec pkg list, 25  
la commande pkg list  
  est comparée avec pkg info, 25  
liste à éviter  
  ajouter à, 48, 55  
  supprimer de, 44, 55

## M

médiateur, 91  
médiation, 91  
médiation de la version d'application, 91  
mise à jour des informations relatives aux packages, 21

mise à jour des informations sur les packages, 43  
Mise à jour du gestionnaire, 133

## O

Ordre de  
  recherche de l'éditeur, 22

## P

p5i fichiers, 134  
package d'incorporation, 12, 89, 108, 117  
  personnalisé, 67  
package de groupe, 13, 29, 35, 94  
packages, 12  
  actionneurs de services SMF, 42  
  bloqués, 87  
  composants facultatifs, 18, 83  
  composants mutuellement exclusifs, 18, 83  
  contraint, 117  
  contrainte, 12  
  contrainte de version, 87  
  contrainte par version, 12, 127  
  contraintes par la version, 117  
  création, 11  
  dépannage de fichiers installés, 53  
  dépannage des packages installés, 50  
  dépendances, 34  
  désinstaller, 55  
  éditeur, 16, 44  
  éviter liste, 94  
  facettes, 18, 83  
  fichiers livrés, 33  
  figés, 25, 55, 127  
  FMRI, 14, 44  
  groupe, 13, 29, 35, 94  
  identifiant, 14  
  ignorer packages non installés, 49, 55  
  incorporation, 12, 117  
  installable, 44  
  installation, 44  
  installés, 22  
  licences, 26, 30, 43  
  liste à éviter  
    ajouter à, 48, 55  
    supprimer de, 44, 55

- listing, 21
  - mise à jour de tous, 63
  - mise à jour disponible, 23
  - mise à jour ou mise à niveau, 48
  - mise à niveau à une version inférieure, 49
  - nettoyage de l'image, 55
  - nom, 14
  - obsolètes, 24
  - package d'incorporation personnalisé , 67
  - parent dépendances, 57
  - pkg:/entire, 108, 117
  - pour installation, 22
  - publication, 11
  - récents, 22
  - rechercher, 30
  - référentiel, 16
  - refus durant l'installation, 124
  - rejet pendant installation, 48
  - renommés, 24
  - services livrés, 33
  - signer propriétés, 98
  - source d'installation, 45
  - supprimer, 55
  - variantes, 18, 83
  - vérification de packages installés, 50
  - version-lock. facettes, 89
  - Packages
    - FMRI complet, 26
  - pkg commande
    - aperçu, 37
    - visualiser l'historique, 105
  - pkg avoid commande, 94
  - pkg change-facet commande, 86
  - pkg change-variant commande, 85
  - pkg éditeur commande, 76
  - pkg exact-install commande, 55
  - pkg fix commande, 50
    - comparaison avec pkg revert, 50
  - pkg freeze commande, 127
  - pkg image-create commande, 103
  - pkg install commande, 44
    - be-name option, 46
    - reject option, 48, 124
    - comparer avec pkg update, 39
    - pkg list command, 21
    - pkg list commande, 44
    - pkg purge-history commande, 105
    - pkg refresh, commande, 21
    - pkg revert commande
      - comparaison avec pkg fix, 50
    - pkg set-publisher commande, 77
      - ajouter et supprimer des propriétés, 100
    - pkg unavoid commande, 94
    - pkg unfreeze commande, 127
    - pkg update commande
      - comparer avec pkg install, 39
      - mise à jour ou mise à niveau de packages, 48
      - mise à niveau à une version inférieure des packages, 49
      - mise à niveau d'une image, 63
    - pkg verify commande, 50, 50
    - pkg:/entire package, 108, 117
    - pkg.sysrepo commande, 58
    - pkgrepo, 69
    - propriétés, 96
      - affichage, 102
      - ajouter et supprimer, 100
      - définition, 103
      - stratégie de création BE, 96
      - utilisation de packages signés, 98
  - proxy
    - éditeur package, 81
    - propriétés de service SMF, 82
    - variables d'environnement, 82
  - proxy services, 58
  - publisher
    - propriétés
      - ajouter et supprimer, 100
- R**
- référentiel, 11, 16, 45
    - miroir, 16
    - mise à jour de contenu, 21
    - origine, 16
  - référentiel d'origine, 16
  - référentiel de logiciel, 11
  - référentiel de package *Voir* référentiel
  - référentiel miroir, 16

référentiel système, 58

mise à jour, 60

## S

services de proxy de zones

proxy de zone, 58

zones-proxy-client, 58

services de proxy SMF, 58

services SMF, 42

solaris-kz zone marquée, 17

solariszone marquée, 17

surface, 12, 117

## T

tous les

packages disponibles, 23

## U

Universal Resource Identifier (URI), 16

## V

variante, 18, 83

variante pkg commande, 85

## Z

zone, 41, 56

zone globale, 17

zone kernel, 17

zone non globale, 17

zone Oracle Solaris 10, 17

zone globale, 17, 41, 56

zone kernel, 17

zone marquée solaris10zone marquée, 17

zone non globale, 17, 41, 56

éditeurs de package , 58

éditeurs de packages, 113

installation de packages, 57

zone Oracle Solaris 10, 17

zones non globales