

Ajout et mise à jour de packages logiciels Oracle® Solaris 11.1

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

Préface	7
1 Introduction à Image Packaging System	11
Image Packaging System	11
Concepts d'IPS	12
Packages IPS	12
Identificateurs de ressource de gestion des pannes	13
Editeurs, référentiels et archives de packages	15
Sources et miroirs du référentiel	15
Images et environnements d'initialisation	16
Facettes et variantes de package	16
Privilèges d'installation	17
2 Interfaces graphiques IPS	19
Utilisation du Gestionnaire de packages	19
Options de ligne de commande du Gestionnaire de packages	20
Utilisation de l'Installation Web	20
Utilisation du Gestionnaire de mises à jour	22
Options de ligne de commande du Gestionnaire de mises à jour	24
3 Obtention d'informations sur les packages logiciels	25
Affichage des informations d'état d'installation du package	25
Affichage des descriptions ou des licences de package	28
Affichage des informations à partir du fichier manifeste du package	29
Etablissement de la liste des fichiers installés par un package	30
Etablissement de la liste de tous les packages installables dans un package de groupe	31
Affichage des conditions d'octroi de licence	31

Recherche de packages	32
Identification du package fournissant un fichier spécifique	33
Affichage des services SMF fournis par les packages	33
Etablissement de la liste des packages par catégorie	34
Affichage des packages dépendants	35
Etablissement de la liste des packages compris dans un package de groupe	35
4 Installation et mise à jour des packages logiciels	37
Aperçu d'une opération	38
Installation et mise à jour de packages	39
Options d'environnement d'initialisation	39
Installation d'un nouveau package	40
Installation d'un package dans un nouvel environnement d'initialisation	43
Rejet d'un package	45
Mise à jour d'un package	45
Résolution des problèmes de package	47
Vérification de l'installation des packages	47
Correction des erreurs de vérification	47
Restauration d'un fichier	48
Désinstallation de packages	49
Utilisation de zones non globales	49
Installation de packages dans des zones non globales	50
Mise à jour simultanée de plusieurs zones non globales	52
5 Configuration des images installées	55
Configuration des éditeurs	55
Affichage des informations sur les éditeurs	55
Ajout, modification ou suppression des éditeurs de packages	57
Contrôle de l'installation des composants optionnels	60
Affichage et modification des valeurs de variables	62
Affichage et modification des valeurs de facettes	63
Verrouillage de packages en une version spécifiée	64
Assouplissement des contraintes de version spécifiées par les incorporations	65
Spécification d'une implémentation d'application par défaut	68
Annulation de l'installation de certains packages compris dans un package de groupe	69

Mise à jour d'une image	72
Meilleures pratiques de mise à jour d'images	72
Spécification de la version à installer	74
Spécification d'une contrainte de version avant la mise à jour	74
Imposition de contraintes sur les packages disponibles	75
Retour à la version antérieure d'une image	75
Options supplémentaires de la commande de mise à jour	75
Configuration des propriétés d'image et d'éditeur	76
Propriétés d'image de la stratégie d'environnement d'initialisation	76
Propriétés de signature de packages	78
Propriétés de l'image supplémentaires	80
Définitions des propriétés d'image	82
Création d'une image	83
Affichage de l'historique des opérations	84

Préface

Ajout et mise à jour de packages logiciels Oracle Solaris 11.1 décrit les fonctionnalités d'installation logicielle de la fonction Oracle Solaris Image Packaging System (IPS). Les commandes IPS vous permettent de lister, rechercher, installer, mettre à jour et gérer des packages de logiciels pour le système d'exploitation Oracle Solaris 11. Une seule commande IPS peut mettre à jour votre image vers une nouvelle version du système d'exploitation. Les commandes IPS vous permettent de limiter les packages qui peuvent être installés ou les différentes versions de packages qui peuvent être installées.

Les commandes IPS vous permettent également de copier et de créer un référentiel de packages IPS et de créer des packages IPS. Reportez-vous à la section [“Documentation connexe”](#) à la page 8 pour plus d'informations sur ces outils.

Pour utiliser IPS, le SE Oracle Solaris 11 doit être en cours d'exécution. Pour installer le SE Oracle Solaris 11, reportez-vous à la section [Installation des systèmes Oracle Solaris 11.1](#).

Utilisateurs de ce manuel

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

Organisation de ce document

- [Chapitre 1, “Introduction à Image Packaging System”](#) décrit Image Packaging System et des composants tels que des packages, des éditeurs et des référentiels.
- [Chapitre 2, “Interfaces graphiques IPS”](#) explique comment utiliser le Gestionnaire de packages (Package Manager) et le Gestionnaire de mises à jour (Update Manager), y compris l'Installation Web (Web Install).
- [Chapitre 3, “Obtention d'informations sur les packages logiciels”](#) indique comment rechercher des packages et afficher les informations sur les packages.
- [Chapitre 4, “Installation et mise à jour des packages logiciels”](#) indique comment installer, mettre à jour et désinstaller les packages.

- [Chapitre 5, “Configuration des images installées”](#) indique comment configurer les caractéristiques qui s'appliquent à la totalité d'une image, comme les éditeurs de packages de configuration ou la restriction des packages pouvant être installés.

Documentation connexe

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

- [Copie et création de référentiels de packages Oracle Solaris 11.1](#)
- [Création et administration d'environnements d'initialisation Oracle Solaris 11.1](#)
- [Installation des systèmes Oracle Solaris 11.1](#)
- [Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.1](#)

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.

Conventions typographiques

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

TABLEAU P-1 Conventions typographiques

Type de caractères	Description	Exemple
AaBbCc123	Noms des commandes, fichiers et répertoires, ainsi que messages système.	Modifiez votre fichier <code>.login</code> . Utilisez <code>ls -a</code> pour afficher la liste de tous les fichiers. <code>nom_machine%</code> Vous avez reçu du courrier.
AaBbCc123	Ce que vous entrez, par opposition à ce qui s'affiche à l'écran.	<code>nom_machine%</code> su Mot de passe :
<i>aabbc123</i>	Paramètre fictif : à remplacer par un nom ou une valeur réel(le).	La commande permettant de supprimer un fichier est <code>rm nom_fichier</code> .

TABLEAU P-1 Conventions typographiques (Suite)

Type de caractères	Description	Exemple
<i>AaBbCc123</i>	Titres de manuel, nouveaux termes et termes importants.	Reportez-vous au chapitre 6 du <i>Guide de l'utilisateur</i> . Un <i>cache</i> est une copie des éléments stockés localement. <i>N'enregistrez pas</i> le fichier. Remarque : en ligne, certains éléments mis en valeur s'affichent en gras.

Invites de shell dans les exemples de commandes

Le tableau suivant présente les invites système UNIX et les invites superutilisateur pour les shells faisant partie du système d'exploitation Oracle Solaris. Dans les exemples de commandes, l'invite de shell indique si la commande doit être exécutée par un utilisateur standard ou un utilisateur doté des privilèges nécessaires.

TABLEAU P-2 Invites de shell

Shell	Invite
Shell Bash, shell Korn et shell Bourne	\$
Shell Bash, shell Korn et shell Bourne pour superutilisateur	#
C shell	nom_machine%
C shell pour superutilisateur	nom_machine#

Introduction à Image Packaging System

Oracle Solaris Image Packaging System (IPS) est une structure permettant de répertorier, de rechercher, d'installer, de mettre à jour et de supprimer des packages de logiciels pour le système d'exploitation Oracle Solaris 11. Une seule commande IPS peut mettre à jour votre image vers une nouvelle version du système d'exploitation.

Image Packaging System

Le logiciel Oracle Solaris 11 est distribué en 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 en images Oracle Solaris 11. Un sous-ensemble des fonctions accessibles via l'interface de ligne de commande d'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 jour une image vers une nouvelle version du système d'exploitation.
- Créer des copies des référentiels de package IPS existants. Créer des référentiels de packages.
- Créer et éditer des packages.
- Créer des environnements d'initialisation.

Pour utiliser IPS, le SE Oracle Solaris 11 doit être en cours d'exécution. Pour installer le SE Oracle Solaris 11, reportez-vous à la section [Installation des systèmes Oracle Solaris 11.1](#).

Concepts d'IPS

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

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 la page de manuel [pkg\(5\)](#) pour obtenir des descriptions d'actions de package.

Les packages de groupe et d'incorporation ne distribuent pas des contenus tels que des fichiers, mais ils vous aident à installer des ensembles de packages apparentés.

Une *incorporation* est un package qui contraint les versions d'un ensemble défini de packages. Par exemple, un package installé dans une incorporation installée est à la version 1.4.3, aucune version antérieure à 1.4.3 ou supérieure ou égale à 1.4.4 ne peut être installée. Cependant, les versions qui ne font qu'étendre la séquence, comme 1.4.3.7, peuvent être installées. Les incorporations forcent les packages incorporés à effectuer une mise à niveau simultanément. 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 65 pour plus d'informations.

Le package nommé `entire` est un package d'incorporation particulier qui restreint les versions d'autres packages d'incorporation.



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.

Un package de *groupe* spécifie l'ensemble de packages qui constituent une fonction ou un outil. Packages spécifiés 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 manifestes de packages de groupes indiquent les dépendances de type `group`. Les packages de groupe fournissent les packages cités dans ces dépendances `group`, 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 69 pour plus d'informations sur la liste "avoid" d'une image.

Par exemple, le package `group/feature/amp` fournit le serveur Web Apache, la base de données MySQL et PHP. Le package `group/system/solaris-desktop` fournit un ensemble de packages adapté à un système de bureau. Le package `group/system/solaris-large-server` ne fournit pas de package de bureau tels que des outils multimédia et des thèmes de fenêtres.

Reportez-vous à la section [“Etablissement de la liste de tous les packages installables dans un package de groupe”](#) à la page 31 pour un exemple illustrant la procédure d'établissement de la liste des packages fournis par un 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 du schéma, d'un éditeur, du nom du package et d'une chaîne de version dans le format suivant. Le schéma, l'éditeur et la chaîne de version sont facultatifs. Avec les commandes IPS, vous pouvez utiliser la plus petite partie du nom du package qui identifie de façon unique le package.

Format du FMRI :

scheme://publisher/package_name@version:dateTimeZ

Exemple de FMRI :

`pkg://solaris/driver/network/ethernet/bge@0.5.11.5.11-0.175.1.0.0.21.0:20120723T161616Z`

Méthode pkg

Editeur solaris

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

Nom du package driver/network/ethernet/bge

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 : 0.175.1.0.0.21.0

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

Les packages Oracle Solaris 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 : 1

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, 1 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

A l'heure actuelle, ce champ n'est pas utilisé pour les packages Oracle Solaris.

Numéro de version SRU : 21

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 : 20110921T002716Z

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 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.

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 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, telle que décrite dans la section [“Propriétés d'image de la stratégie d'environnement d'initialisation” à la page 76](#).

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. Dans IPS, les composants optionnels sont appelés des *facettes* et les composants incompatibles sont appelés des *variantes*.

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. La plupart des balises de variantes peuvent avoir plusieurs valeurs. Les balises de facettes définies sur une action peuvent uniquement avoir la valeur `true`. 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.

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 60.

Privilèges d'installation

L'utilisation des commandes abordées au [Chapitre 3, “Obtention d'informations sur les packages logiciels”](#) ne nécessite aucun privilège spécial. Les tâches telles que l'installation et la mise à jour des packages IPS, la définition des éditeurs et la modification d'images nécessitent davantage de privilèges. Utilisez l'une des méthodes suivantes pour obtenir davantage de privilèges.

Profils de droits	Utilisez la commande <code>profiles</code> 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 la commande <code>pfexec</code> pour installer et mettre à jour les packages et gérer les environnements d'initialisation.
	<pre>\$ pfexec pkg install editor/gnu-emacs \$ pfexec beadm activate solaris11_1-2</pre>
Rôles	Utilisez la commande <code>roles</code> pour répertorier les rôles qui vous sont affectés. Si vous disposez du rôle <code>root</code> , vous pouvez utiliser la commande <code>su</code> avec le mot de passe <code>root</code> pour prendre le rôle <code>root</code> .
commande sudo	En fonction de la stratégie de sécurité de votre site, vous pouvez être en mesure d'utiliser la commande <code>sudo</code> avec votre mot de passe utilisateur pour exécuter une commande privilégiée.

Interfaces graphiques 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 du SE Oracle Solaris et des technologies IPS, utilisez le Gestionnaire de packages pour identifier et installer rapidement des packages.
- Le Gestionnaire de 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 de packages fournit un sous-ensemble des tâches qui peuvent être effectuées à partir de la ligne de commande :

- Lister, rechercher, installer, mettre à jour et supprimer des packages
- Ajouter et configurer des sources de package
- Activer, renommer et supprimer des environnements d'initialisation

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

Barre d'outils	Cliquez sur l'icône du Gestionnaire de packages dans la barre d'outils. L'icône du Gestionnaire de packages représente une boîte entourée d'une flèche circulaire.
Ikône du bureau	Cliquez deux fois sur l'icône du Gestionnaire de packages sur le bureau.
Barre de menu	Choisissez System (Système) > Administration > Gestionnaire de packages.
Ligne de commande	\$ pfexec packagemanager &

Pour consulter la documentation complète du Gestionnaire de packages, choisissez Help (Aide) > Contents (Sommaire) dans la barre de menu du Gestionnaire de packages.

Options de ligne de commande du Gestionnaire de packages

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

TABLEAU 2-1 Options de commande du Gestionnaire de packages

Option	Description
- -image-dir ou -R dir	S'exécute sur l'image résidant au niveau de <i>dir</i> . Par défaut, la commande s'exécute sur l'image actuelle. La commande suivante s'exécute sur l'image stockée au niveau de <code>/aux0/example_root</code> : # packagemanager -R /aux0/example_root
- -update-all ou -U	Met à jour tous les packages installés pour lesquels une mise à jour est disponible. Spécifier cette option équivaut à choisir l'option Updates (Mises à jour) dans l'interface graphique du Gestionnaire de packages. Pour plus d'informations sur la mise à jour de tous les packages, reportez-vous à la section "Utilisation du Gestionnaire de mises à jour" à la page 22 .
- -info-install ou -i file.p5i	Spécifie un fichier <code>.p5i</code> pour exécuter le Gestionnaire de packages en mode Installation Web. Le fichier spécifié doit avoir l'extension <code>.p5i</code> . Pour plus d'informations, reportez-vous à la section "Utilisation de l'Installation Web" à la page 20 .
- -help ou -h	Affiche les informations d'utilisation de la commande.

Utilisation de l'Installation Web

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

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

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

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

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

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

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

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

1. Utilisez l'une des méthodes suivantes pour démarrer le Gestionnaire de packages en mode Installation Web :

- 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
```

- Allez à un emplacement d'URL contenant un lien vers un fichier `.p5i`.

Si le fichier `.p5i` réside sur un serveur Web où le type MIME est enregistré, il suffit de cliquer sur le lien vers le fichier `.p5i`.

Si le fichier `.p5i` réside sur un serveur Web où ce type MIME n'est pas enregistré, enregistrez le fichier `.p5i` sur votre bureau, puis sélectionnez ce dernier.

2. 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. Cliquez sur le bouton Continue (Continuer) pour poursuivre l'installation.
3. 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. Cliquez sur le bouton OK pour poursuivre l'installation.
4. Si un fichier .p5i contient des packages provenant d'un éditeur désactivé, Installation Web ouvre une boîte de dialogue Enable Publisher (Activer l'éditeur). Utilisez cette boîte de dialogue pour activer l'éditeur de sorte à pouvoir installer les packages.

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

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

Utilisation du Gestionnaire de mises à jour

Gestionnaire de mises 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 graphique du Gestionnaire de packages, choisissez le bouton Updates (Mises à jour) ou l'option de menu Package > Updates (Mises à jour).
- Utilisez la commande `packagemanager`.

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

- Utilisez la commande `pkg`.

```
$ pfexec pkg update
```

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

Barre d'état	Lorsque des mises à jour sont disponibles, une notification doit apparaître dans la barre d'état. Cliquez sur l'endroit indiqué dans la notification. L'icône du Gestionnaire de mises à jour représente une pile de trois boîtes.
Barre de menu	Choisissez System (Système) > Administration > Gestionnaire de mises à jour.
Ligne de commande	# <code>pm-updatemanager</code>

Automatisé

Le package Gestionnaire de mises à jour, `package/pkg/update-manager`, fournit le travail `cron` `/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 pour ouvrir l'interface graphique de Gestionnaire de mises à jour.

La fenêtre Updates (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.
3. Cliquez sur le bouton Continue (Continuer) pour poursuivre la mise à jour. Le système télécharge et installe toutes les mises à jour de package.

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

```
package/pkg
package/pkg/package-manager
package/pkg/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 fenêtre Manage Publishers (Organiser les éditeurs) du Gestionnaire de packages ou la commande `pkg set-publisher` pour définir un éditeur comme résident ou non permanent.

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 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.

4. Si le système a créé un environnement d'initialisation, vous pouvez modifier le nom de l'environnement d'initialisation par défaut. Lorsque vous êtes satisfait du nom de l'environnement d'initialisation, cliquez sur le bouton Restart Now (Redémarrer

maintenant) pour redémarrer votre système immédiatement. Cliquez sur le bouton Restart Later (Redémarrer ultérieurement) pour redémarrer votre système à une date ultérieure. 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.

Options de ligne de commande du Gestionnaire de mises à jour

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

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

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

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 peut être mis à jour.
- La description, la taille et la version du package.
- Quels packages font partie d'un package de groupe.
- Quels packages font partie d'une catégorie donnée.
- Quel package fournit un fichier spécifié.

Aucun privilège particulier n'est nécessaire pour exécuter ces commandes.

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 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. Les variantes de package pour un type d'architecture ou de zone qui ne correspondent pas à cette image ne sont pas répertoriées.

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

La commande `pkg list` affiche une ligne d'informations sur chaque package.

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

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.

Le "i" dans la colonne I indique que ces packages sont installés dans cette image. Pour afficher la liste des paquets qui sont 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   3.0.15-0.175.1.0.0.14.0 ---
isvtoolkit (isvpub)       1.0             i--
system/dtrace/dtrace-toolkit 0.99-0.175.1.0.0.21.0 i--
```

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

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`. Afin de répertorier uniquement les versions les plus récentes de ces packages, spécifiez `@latest`.

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

Cette sortie indique que le package `developer/dtrace/toolkit` ne peut pas être installé dans cette image. Un `r` dans la colonne O indique que ce package a été renommé. Le package `developer/dtrace/toolkit` a été renommé en `system/dtrace/dtrace-toolkit` et `system/dtrace/dtrace-toolkit` est déjà installé.

Dans l'exemple suivant, le package `web/amp` a été renommé `group/feature/amp`. Si vous spécifiez la commande permettant d'installer le package `web/amp`, le package `group/feature/amp` est automatiquement installé.

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

La commande `pkg list` ne permet pas d'obtenir le nouveau nom d'un package renommé. Dans les exemples ci-dessus, le modèle saisi en tant qu'entrée pour la commande `pkg list` correspond au nouveau nom ainsi qu'à l'ancien nom, et la corrélation a pu être établie. En règle générale, pour afficher le nouveau nom d'un package renommé, utilisez la commande `pkg info` comme illustré à la section [“Affichage des descriptions ou des licences de package”](#) à la page 28.

L'option `-n` répertorie la version la plus récente de tous les packages connus. Un `o` dans la colonne O indique que ce package est obsolète. Vous ne pouvez pas installer un package obsolète.

```
$ pkg list -n '*mysql-5?'
```

NAME (PUBLISHER)	VERSION	IFO
database/mysql-50	5.0.91-0.171	--o
database/mysql-51	5.1.37-0.175.1.0.0.21.0	---

Cette sortie indique que le package database/mysql-50 ne peut pas être installé dans cette image. Ce package n'a pas été renommé. Si vous spécifiez la commande pour installer le package mysql-50, le package mysql-51 n'est pas installé. Aucun package n'est installé dans ce cas.

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 64](#) pour plus d'informations sur la manière de figer les packages.

```
$ pkg list mercurial
```

NAME (PUBLISHER)	VERSION	IFO
developer/versioning/mercurial	2.2.1-0.175.1.0.0.21.0	if-

L'option -s répertorie uniquement le nom et le récapitulatif du package.

```
$ pkg list -ns mysql-51 feature/amp
```

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

L'option -v répertorie l'ensemble du package FMRI.

```
$ pkg list -nv mysql-51
```

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

L'option -u permet d'afficher la liste de tous les packages installés qui ont de nouvelles versions disponibles.

```
$ pkg list -u 'compress/*'
```

NAME (PUBLISHER)	VERSION	IFO
compress/bzip2	1.0.6-0.175.1.0.0.19.0	i--
compress/gzip	1.4-0.175.1.0.0.19.0	i--
compress/p7zip	9.20.1-0.175.1.0.0.19.0	i--
compress/unzip	6.0-0.175.1.0.0.19.0	i--
compress/zip	3.0-0.175.1.0.0.19.0	i--

Remarque – Le nombre des packages pour lesquels il existe des versions plus récentes dans le référentiel de packages est peut-être plus grand que le nombre de packages pouvant être mis à jour dans cette image. Les packages peuvent uniquement être mis à jour avec les versions autorisées par les contraintes imposées par les dépendances de package installées et par la configuration de l'éditeur. Pour déterminer quels packages peuvent être mis à jour dans cette image, servez-vous de la commande `pkg update -nv`.

Utilisez l'option -g pour spécifier le référentiel ou l'archive de package à utiliser en tant que source de données de package pour l'opération.

Lorsque vous utilisez l'option `--no-refresh`, `pkg` ne tente pas de contacter les référentiels pour que les éditeurs de l'image récupèrent la liste des paquets disponibles la plus récente.

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 d'empaquetage, la taille du package et le FMRI complet. Sans option ou opérande, cette commande 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.

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

Les sous-commandes `info` et `list` affichent le nom et l'éditeur du package et les informations de 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 ou renommé. 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.

L'option `-r` affiche les versions les plus récentes disponibles, en récupérant les informations pour tous les packages qui ne sont pas actuellement installés auprès des référentiels des éditeurs configurés.

```
$ pkg info -r group/feature/amp
  Name: group/feature/amp
  Summary: AMP (Apache, MySQL, PHP) Deployment Kit for Oracle Solaris
  Description: Provides a set of components for deployment of an AMP (Apache,
  MySQL, PHP) stack on Oracle Solaris
  Category: Meta Packages/Group Packages (org.opensolaris.category.2008)
  Web Services/Application and Web Servers (org.opensolaris.category.2008)
  State: Not installed
  Publisher: solaris
  Version: 0.5.11
  Build Release: 5.11
  Branch: 0.175.1.0.0.21.0
  Packaging Date: July 23, 2012 06:20:57 PM
  Size: 5.46 kB
  FMRI: pkg://solaris/group/feature/amp@0.5.11,5.11-0.175.1.0.0.21.020120723T182057Z
```

Utilisez la commande `pkg info` pour trouver le nouveau nom d'un package renommé.

L'exemple suivant montre que le nouveau nom du package `developer/dtrace/toolkit` est `system/dtrace/dtrace-toolkit`.

```
$ pkg info -r developer/dtrace/toolkit
  Name: developer/dtrace/toolkit
  Summary:
```

```

State: Not installed (Renamed)
Renamed to: pkg:/system/dtrace/dtrace-toolkit@0.99,5.11-0.173.0.0.0.0.0
            consolidation/osnet/osnet-incorporation
Publisher: solaris
Version: 0.99
Build Release: 5.11
Branch: 0.173.0.0.0.1.0
Packaging Date: 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

```

L'option `--license` affiche les textes de licence pour les packages. Ces informations peuvent être assez conséquentes. Les informations ci-dessus (sans l'option `--license`) ne sont pas affichées.

```

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

```

Utilisez l'option `-g` pour spécifier le référentiel ou l'archive de package à utiliser en tant que source de données de package pour l'opération.

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.

```

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

```

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.

L'exemple suivant montre le comportement par défaut de `pkg contents`. Utilisez les options pour spécifier les actions et les attributs à afficher.

```

$ pkg contents zip
PATH
usr
usr/bin
usr/bin/zip

```

```
usr/bin/zipcloak
usr/bin/zipnote
usr/bin/zipsplit
usr/share
usr/share/man
usr/share/man/man1
usr/share/man/man1/zip.1
usr/share/man/man1/zipcloak.1
usr/share/man/man1/zipnote.1
usr/share/man/man1/zipsplit.1
```

L'option `-m` permet d'afficher l'intégralité du fichier manifeste du package.

L'option `-r` affiche les versions les plus récentes disponibles, en récupérant les informations pour tous les packages qui ne sont pas actuellement installés auprès des référentiels des éditeurs configurés.

Utilisez l'option `-g` pour spécifier le référentiel ou l'archive de package à utiliser en tant que source de données de package pour l'opération.

Etablissement de la liste des fichiers installés par un package

Utilisez l'option `-t` pour spécifier 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.

Utilisez l'option `-o` pour spécifier les attributs à afficher dans la sortie. 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\(5\)](#). Dans cet exemple, le pseudo-attribut `pkg.size` affiche la taille du fichier ; l'action `file` n'a pas d'attribut `size`. Pour obtenir une liste des pseudo-attributs, reportez-vous à la page de manuel [pkg\(1\)](#).

Utilisez l'option `-s` pour trier les actions suivant l'attribut d'action spécifié. Par défaut, la sortie est triée suivant les chemins d'accès ou en fonction du premier attribut spécifié par l'option `-o`. L'option `-s` peut être indiquée à plusieurs reprises.

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

Si vous affichez le manifeste du package, vous constatez que le package zip a douze 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 destinés à l'architecture SPARC ne sont pas affichés. 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 60](#).

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

L'interface graphique d'installation d'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`. Comme solution de rechange, vous pouvez également utiliser le package de groupe `solaris-small-server` pour installer un ensemble de packages plus petit sur un serveur. 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-large-server
archiver/gnu-tar
compress/bzip2
...
text/texinfo
web/wget
```

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 IPS” à la page 12](#) 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-large-server'
archiver/gnu-tar      GNU version of the tar archiving utility
compress/bzip2       high-quality block-sorting file compressor - utilities
compress/gzip        GNU Zip (gzip)
...
text/texinfo         Documentation system for on-line information and printed output
web/wget             wget - GTNU wget
```

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 must-display,license,pkg.name '*incorporation'
MUST-DISPLAY LICENSE                               PKG.NAME
true          usr/src/pkg.license_files/lic_OTN consolidation/osnet/osnet-incorporation
```

Recherche de packages

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

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

Tout comme la commande `pkg contents`, la commande `pkg search` examine le contenu des paquets. Alors que la commande `pkg contents` renvoie le contenu, la commande `pkg search` renvoie les noms des packages qui correspondent à la requête.

Par défaut, *query* est interprété 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 requête. 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 par AND. Vous pouvez explicitement joindre deux termes avec OR.

Les requêtes peuvent s'exprimer sous la structure suivante :

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

Les champs manquants sont implicitement considérés comme des caractères génériques. Les champs *pkg_name* et *token* peuvent contenir des caractères génériques explicites. Les valeurs *action_name* et *index* doivent correspondre exactement. La valeur *action_name* correspond au nom de l'action. La valeur *index* correspond au nom d'un attribut de l'action. Pour consulter la liste des actions de package et des attributs, reportez-vous à la rubrique “Actions” de la page de manuel [pkg\(5\)](#). 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* d'une action *file* ou *dir*. Constituent par exemple des valeurs utiles pour *index* les valeurs *basename* et *path* pour les actions *file* et *dir*, le type de dépendance (par exemple *require* ou *group*) pour les actions *depend* et *driver_name* et *alias* pour les actions de pilote.

En général, la valeur *token* est comparée à la valeur de l'attribut spécifié par *index*. Par exemple, dans l'action *driver* partielle 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. Par exemple, dans l'action `set` partielle suivante, `pkg.summary` peut être spécifié pour *index* et `Broadcom` peut être spécifié pour *token*.

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

`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, la recherche est effectuée sur les référentiels associés à tous les éditeurs configurés pour cette image. Utilisez l'option `-l` pour rechercher uniquement les packages installés dans cette image. Utilisez l'option `-s` pour spécifier l'URI du référentiel dans lequel effectuer la recherche.

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
```

Affichage des services SMF fournis par les packages

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

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
```

Etablissement de la liste des packages par catégorie

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

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

Cet exemple montre une grande quantité d'informations répétées qui "noient" les informations réellement recherchées.

L'exemple suivant utilise l'option `-o` pour afficher uniquement les noms des packages et utilise l'option `-H` pour omettre l'en-tête de colonne.

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

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`. Si vous avez utilisé la commande `pkg contents` pour afficher les actions dépend de type `require` pour les packages `i86pc` et `system/hal`, vous verrez que `system/kernel/power` figure dans la liste pour les deux packages.

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

L'exemple suivant montre que de nombreux packages ont une dépendance de type `exclude` par rapport à `pkg:/x11/server/xorg@1.12.99`.

```
$ pkg search -lo pkg.name,fmri 'depend:exclude:*xorg*'
PKG.NAME                               FMRI
x11/server/xvnc                         pkg:/x11/server/xorg@1.12.99
x11/server/xorg                         pkg:/x11/server/xorg@1.12.99
x11/server/xorg/driver/xorg-video-mga   pkg:/x11/server/xorg@1.12.99
x11/server/xorg/driver/xorg-video-vesa  pkg:/x11/server/xorg@1.12.99
x11/server/xorg/driver/xorg-input-vmouse pkg:/x11/server/xorg@1.12.99
...
```

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

L'interface graphique d'installation d'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`. Comme solution de rechange, vous pouvez également utiliser le package de groupe `solaris-small-server` pour installer un ensemble de packages plus petit sur un serveur. Vous pouvez utiliser la recherche suivante pour afficher l'ensemble de packages inclus dans chaque groupe.

```
$ pkg search -Hfo fmri '*/solaris-large-server:depend:group:*'
archiver/gnu-tar
compress/bzip2
...
text/texinfo
web/wget
```

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

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

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

Par défaut, la recherche renvoie uniquement les packages qui sont installables dans cette image. Dans cet exemple, la recherche ne renvoie pas les packages correspondants, mais renvoie plutôt 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 est plus important que celui obtenu avec la commande `pkg contents` car les résultats incluent les noms de tous les packages spécifiés dans les actions de type `group depend` 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. Comparez le résultat de cette recherche au résultat de la commande `pkg contents` illustré à la section [“Etablissement de la liste de tous les packages installables dans un package de groupe”](#) à la page 31.

Astuce – En général, la commande `pkg contents` s'utilise pour afficher le contenu d'un package spécifié, et la commande `pkg search` s'utilise 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`.

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 3, “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 49](#) présente différents aspects d'opérations sur les package 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 17](#) pour plus d'informations.

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 :

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

La commande suivante produit une sortie de données de grand volume, car de nombreux packages seraient concernés. Notez que la quantité d'espace supplémentaire qui serait utilisé est exprimée en Go et non en Mo. Cette opération peut nécessiter une grande quantité de temps et entraîner une grande quantité de trafic réseau entre cette image et le référentiel de packages. 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 76 pour obtenir des informations sur la création des environnements d'initialisation.

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

```

Estimated space available: 112.19 GB
Estimated space to be consumed: 2.96 GB
Create boot environment: No
Create backup boot environment: Yes
Rebuild boot archive: No
Changed variants/facets:
facet facet.locale.*: True
Changed packages:
solaris
...
```

Installation et mise à jour de packages

La commande `pkg install` installe les packages qui ne sont pas actuellement installés et met à jour les packages qui le sont déjà. La commande `pkg install` requiert un ou plusieurs noms de package.

La commande `pkg update` met à jour les packages installés. Si vous spécifiez un package qui n'est pas déjà installé à la commande `pkg update`, le système n'installe pas ce package. La commande `pkg update` prend zéro ou plusieurs noms de packages qui sont déjà installés. Si aucun nom de package n'est spécifié, tous les packages installés dans l'image sont mis à jour.

Reportez-vous aux attributs `preserve` et `overlay` de l'action `file` mentionnés dans la page de manuel [pkg\(1\)](#) pour comprendre comment les fichiers dotés de ces attributs seront gérés lors de l'installation et de la mise à jour.

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. 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. 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 76 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.

```
--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. Vous pouvez utiliser la commande <code>beadm(1M)</code> pour afficher et modifier l'environnement d'initialisation actif.
<code>--no-backup-be</code>	Ne crée pas d'environnement d'initialisation de sauvegarde.
<code>--require-backup-be</code>	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 76 pour obtenir des informations sur la création automatique des environnements d'initialisation de sauvegarde.
<code>--backup-be-name name</code>	Si un environnement d'initialisation de sauvegarde est créé, appelez-le <i>name</i> au lieu de lui attribuer le nom par défaut. L'utilisation de <code>--backup-be-name</code> implique <code>--require-backup-be</code> .
<code>--deny-new-be</code>	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.
<code>--require-new-be</code>	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 76 pour obtenir des informations sur la création automatique des environnements d'initialisation. Cette option ne peut pas être combinée avec <code>--require-backup-be</code> .
<code>--be-name name</code>	Si un environnement d'initialisation est créé, appelez-le <i>name</i> au lieu de lui attribuer le nom par défaut. L'utilisation de <code>--be-name</code> implique <code>--require-new-be</code> .

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 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 13](#) 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 55](#) pour obtenir des informations sur la définition du caractère résident/non permanent et de l'ordre de recherche des éditeurs.

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

Si la valeur *pkg_fmri_pattern* 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.

```
$ 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 la valeur de *pkg_fmri_pattern* “atool” chez le premier éditeur dans l'ordre de recherche, mais ce package ne peut pas être installé.

```
$ pfexec pkg install atool
```

Pour installer ce package, définissez de manière plus précise la valeur *pkg_fmri_pattern*, comme indiqué dans les exemples suivants :

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

Utilisez l'option `-nv` pour voir ce qui sera installé avant de procéder à l'installation proprement dite.

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 la valeur `pkg_fmri_pattern` spécifiée 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 des éditeurs qui n'ont pas été configurés dans l'image est ajouté à la configuration de l'image sans origine.
- Si un package correspondant à la valeur `pkg_fmri_pattern` spécifiée 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.

```
$ pfexec 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 la valeur `pkg_fmri_pattern`.

```
$ pfexec 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/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.

```
$ pfexec 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 57

Utilisez l'option `-C` pour installer des packages dans n zones non globales en même temps que dans la zone globale. Reportez-vous à la section [“Mise à jour simultanée de plusieurs zones non globales” à la page 52](#) pour voir un exemple d'utilisation de l'option `-C`.

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. Utilisez l'option `--licenses` pour afficher toutes les licences pour les packages installés ou mis à jour dans le cadre de cette opération.

Lorsque vous spécifiez l'option `--no-index`, les index de recherche ne sont pas mis à jour une fois l'opération terminée. La définition de cette option peut vous faire gagner du temps si vous installez un grand nombre de packages. Une fois les opérations `install`, `update` et `uninstall` 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.

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.

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`.

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 76](#) 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` pour forcer la création d'un nouvel environnement d'initialisation ou pour attribuer un nom significatif au nouvel environnement d'initialisation.

```
$ pfexec pkg install --be-name s11amp group/feature/amp
      Packages to install: 5
      Create boot environment: Yes
      Create backup boot environment: No

DOWNLOAD                                PKGS      FILES    XFER (MB)   SPEED
Completed                              5/5       271/271    52.3/52.3   0B/s

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

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

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

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`.

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

Vérifiez que le package `group/feature/amp` est installé dans le nouvel environnement d'initialisation. Le "i" dans la colonne I indique que le package `group/feature/amp` est installé.

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

```
group/feature/amp      0.5.11-0.175.1.0.0.21.0    i--
```

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

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

Rejet d'un package

Utilisez l'option `--reject` de la commande `pkg install` pour empêcher les packages avec des noms correspondant à la valeur `pkg_fmri_pattern` spécifiée d'être installés. Si des packages de même nom sont déjà installés, ils sont supprimés dans le cadre de cette opération. Les packages rejetés qui sont la cible des dépendances de groupe sont placés 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 69](#) pour plus d'informations sur la liste `avoid`.

```
$ pfexec pkg install -nv --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. Reportez-vous à [“Identificateurs de ressource de gestion des pannes” à la page 13](#) 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 55](#) pour obtenir des informations sur la définition du caractère résident/non permanent et de l'ordre de recherche des éditeurs.

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

```
[--reject pkg_fmri_pattern ...] [pkg_fmri_pattern ...]
```

Pour demander explicitement la version la plus récente d'un package, utilisez `latest` pour la partie de la valeur `pkg_fmri_pattern` relative à la version.

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

Vous pouvez spécifier une version de module antérieure à celle qui est actuellement installée pour effectuer une mise à niveau inférieur sur place. Tous les fichiers de configuration faisant partie des packages à mettre à niveau inférieur et qui ont été modifiés depuis que la version originale a été installée sont renommés avec l'extension `.update`. Pour plus d'informations sur la méthode utilisée par le système de package pour déterminer quels fichiers conserver pendant les mises à niveau de package, consultez la section relative aux actions de fichiers dans la page de manuel [pkg\(5\)](#).

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. Reportez-vous à la section [“Installation d'un nouveau package” à la page 40](#) pour une description détaillée et des exemples illustrant les effets de l'option `-g`.

Utilisez l'option `-C` pour mettre à jour des packages dans *n* zones non globales en même temps que dans la zone globale. Reportez-vous à la section [“Mise à jour simultanée de plusieurs zones non globales” à la page 52](#) pour obtenir un exemple.

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`.

Utilisez l'option `-accept` pour indiquer que vous acceptez les conditions de licence des packages qui sont mis à jour. Si vous ne spécifiez pas cette option alors que les licences de package exigent l'acceptation, l'opération de mise à jour échoue. Utilisez l'option `--licenses` pour afficher toutes les licences pour les packages mis à jour dans le cadre de cette opération.

Lorsque vous spécifiez l'option `--no-index`, les index de recherche ne sont pas mis à jour une fois l'opération terminée. La définition de cette option peut vous faire gagner du temps si vous installez un grand nombre de packages. Une fois les opérations `install`, `update` et `uninstall` 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.

Reportez-vous à la section [“Mise à jour d'une image” à la page 72](#) pour plus d'informations sur le comportement particulier de la commande `pkg update` lorsqu'aucune valeur `pkg_fmri` n'est indiquée ou lorsque la valeur `pkg_fmri` est spécifiée à l'aide d'un astérisque (*).

Résolution des problèmes de package

Un exemple de problème qui pourrait se produire après l'installation d'un package est l'altération d'un fichier fourni par le package. Dans l'exemple illustré dans cette section, le fichier `/usr/share/auto_install/manifest/default.xml` a été supprimé.

Utilisez la commande `pkg search` pour déterminer quel package a livré le fichier manquant :

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

Vérification de l'installation des packages

Utilisez la commande `pkg verify` pour valider l'installation des packages sur l'image actuelle.

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

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 la partie relative à signature-policy de la section “[Propriétés de signature de packages](#)” à la page 78 pour plus d'informations sur la manière dont les stratégies de signature sont appliquées.

Utilisez l'option `-H` pour omettre les en-têtes de la sortie de vérification. Utilisez l'option `-q` pour ne rien imprimer mais renvoyer un échec en cas de détection d'erreurs fatales. Utilisez l'option `-v` pour inclure les messages d'information concernant les packages.

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

Correction des erreurs de vérification

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

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

La vérification du contenu des packages installés repose sur une analyse de contenu personnalisée dont les résultats peuvent différer de ceux des autres programmes.

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 de correction échoue. Utilisez l'option `--licenses` pour afficher toutes les licences pour les packages mis à jour dans le cadre de cette opération.

```
$ pfexec pkg fix --accept system/install/auto-install/auto-install-common
Verifying: pkg://solaris/system/install/auto-install/auto-install-common      ERROR
      file: usr/share/auto_install/manifest/default.xml
      Missing: regular file does not exist
Created ZFS snapshot: 2012-08-06-23:32:03
Repairing: pkg://solaris/system/install/auto-install/auto-install-common
Creating Plan (Evaluating mediators):

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

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

Restauration d'un fichier

Utilisez la commande `pkg revert` pour restaurer les fichiers à leur condition lors de la livraison.

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

Tous les fichiers marqués avec l'indicateur `tag-name` ou des fichiers individuels peuvent être rétablis. 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.

L'exemple suivant illustre la restauration du contenu initial d'un fichier de configuration système qui a été modifié.

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

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

PHASE                                ITEMS
Updating modified actions              1/1
Updating image state                   Done
```

Creating fast lookup database

Done

Désinstallation de packages

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

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

Si un package est actuellement l'objet d'une dépendance de groupe, la désinstallation du package le place dans la liste des packages à éviter. Reportez-vous à la section [“Annulation de l'installation de certains packages compris dans un package de groupe”](#) à la page 69 pour plus d'informations sur la liste "avoid".

Lorsque vous spécifiez l'option `--no-index`, les index de recherche ne sont pas mis à jour une fois l'opération terminée. La définition de cette option peut vous faire gagner du temps si vous installez un grand nombre de packages. Une fois les opérations `install`, `update` et `uninstall` 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.

Utilisez l'option `-C` pour désinstaller des packages dans `n` zones non globales en même temps que dans la zone globale. Reportez-vous à la section [“Mise à jour simultanée de plusieurs zones non globales”](#) à la page 52 pour voir un exemple d'utilisation de l'option `-C`.

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. Notez que les zones marquées Oracle Solaris 10 sont différentes des zones non globales Oracle Solaris 11. Les commandes IPS ignorent les zones marquées Oracle Solaris 10. Dans ce manuel, "zone non globale" signifie zone non globale Oracle Solaris 11.

Une différence importante entre la zone globale et les zones non globales concerne l'utilisation des éditeurs de packages. 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.

Installation de packages dans des zones non globales

Les zones non globales peuvent être affectées par l'installation, la mise à jour et la désinstallation de packages dans la zone globale. Lorsque vous exécutez la commande `pkg update` sans argument dans la zone globale, la zone globale et chaque zone non globale est mise à jour. Reportez-vous à la section [“Mise à jour simultanée de plusieurs zones non globales” à la page 52](#) pour obtenir un exemple. Lorsque vous indiquez des noms de packages avec les commandes d'installation, de mise à jour ou de désinstallation dans la zone globale, IPS vérifie chaque zone non globale et effectue uniquement les modifications requises pour que la zone non globale concernée reste compatible avec la zone globale. La modification de facettes et de variantes dans la zone globale peut également avoir un impact sur des zones non globales.

Astuce – Utilisez l'option `-n` 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. Vous pouvez installer des packages différents et différentes versions du même package si le résultat est compatible avec la zone globale. Vous pouvez éviter divers packages, figer des packages à différentes versions, définir des médiateurs pour sélectionner différentes implémentations par défaut et définir différentes facettes dans l'image de zone non globale.

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. 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.

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. 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-proxyd: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/repoSolaris11/
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>
solaris (syspub)   origin   online F <system-repository>
isvpub (syspub)    origin   online F <system-repository>
```

La lettre T dans la colonne P signifie que cette origine dispose d'un proxy. Exécutez l'une des commandes suivantes pour obtenir plus d'informations :

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

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 vous ne pouvez pas atteindre un éditeur, vous pouvez définir un proxy dans la zone globale en paramétrant la variable d'environnement `http_proxy` ou en spécifiant l'option `--proxy` avec la commande `pkg set -publisher`. Reportez-vous à la page de manuel [pkg\(1\)](#) et à la section “Ajout, modification ou suppression des éditeurs de packages” à la page 57 pour plus

d'informations sur l'option `--proxy`. Reportez-vous à la section “[Configuration du proxy sur un système comportant des zones installées](#)” du manuel *Administration d'Oracle Solaris 11.1 : Oracle Solaris Zones, Oracle Solaris 10 Zones et gestion des ressources* pour plus d'instructions sur le paramétrage des variables d'environnement `http_proxy` et `https_proxy`. Reportez-vous à la section ENVIRONMENT de la page de manuel `curl(1)` pour plus d'informations sur les variables d'environnement proxy.

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

```
z1:~$ pkg list -a '//isvpub/*'
NAME (PUBLISHER)    VERSION    IFO
isvtool (isvpub)    1.0-0      ---
```

Pour les référentiels qui ne sont pas configurés dans la zone globale mais qui sont accessibles à la zone non globale par l'intermédiaire du réseau ou du système de fichiers, les deux commandes suivantes établissent la même liste de packages si `file:///export/myrepo` est un emplacement de référentiel accessible à la zone non globale :

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

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. L'option `-C n` et la variable d'environnement `PKG_CONCURRENCY=n` indiquent de mettre à jour en parallèle *n* images au plus pour une valeur *n* supérieure ou égale à 1. La valeur par défaut de *n* est 1. Si la valeur *n* est 0 ou un chiffre négatif, toutes les zones non globales sont mises à jour parallèlement à la zone globale.

La variable d'environnement `PKG_CONCURRENCY` est ignorée si l'option `-C` est spécifiée. L'option `-C` et la variable d'environnement `PKG_CONCURRENCY` peuvent être utilisées avec les commandes `pkg install`, `pkg uninstall`, `pkg change-variant` et `pkg change-facet` ainsi qu'avec `pkg update`.

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 uniquement être montées.

Dans l'exemple suivant, les zones non globales sont mises à jour en même temps que la zone globale :

```
global:~$ pfexec pkg update -C 0 --be-name s11u1
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.

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.

- “Configuration des éditeurs” à la page 55
- “Contrôle de l'installation des composants optionnels” à la page 60
- “Verrouillage de packages en une version spécifiée” à la page 64
- “Assouplissement des contraintes de version spécifiées par les incorporations” à la page 65
- “Spécification d'une implémentation d'application par défaut” à la page 68
- “Annulation de l'installation de certains packages compris dans un package de groupe” à la page 69
- “Mise à jour d'une image” à la page 72
- “Configuration des propriétés d'image et d'éditeur” à la page 76
- “Création d'une image” à la page 83
- “Affichage de l'historique des opérations” à la page 84

Configuration des éditeurs

Pour l'installation et la mise à jour de logiciels, vous devez 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.

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

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/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 15](#) 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 jamais pourvus d'un proxy. Les référentiels HTTP présentant la valeur `F` ne sont pas pourvus d'un proxy, sauf si une variable d'environnement `$http_proxy` est actuellement définie (la sortie `pkg publisher` affiche toujours `F`). 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` sur `pkg publisher`, la colonne `P` contient le proxy défini pour ce miroir ou cette origine. Reportez-vous à la section [“Installation de packages dans des zones non globales” à la page 50](#) pour obtenir un exemple.

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

Utilisez l'option `-P` pour afficher uniquement le premier éditeur dans l'ordre de recherche des éditeurs. Utilisez l'option `-n` pour afficher uniquement les éditeurs activés. L'option `-H` omet les en-têtes dans la sortie.

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

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.
- Activation ou désactivation d'un éditeur. 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 d'installation, de désinstallation ou de mise à jour de package. 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.
- Définition du caractère résident/non permanent d'un éditeur. Un éditeur récemment ajouté est résident 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.
- Définition de l'ordre de recherche des éditeurs. 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.

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.

- Spécification des certificats et clés SSL pour un éditeur.
- Définition et rétablissement de la propriété d'un éditeur, et ajout et suppression de la valeur de propriété d'un éditeur. Reportez-vous à la section [“Configuration des propriétés de signature de package”](#) à la page 79.

La commande `pkg set-publisher` possède deux formes. Quand la commande est sous la forme suivante, le nom de l'éditeur est requis :

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

```
[--unset-ca-cert hash_of_CA_to_remove]
[--set-property name_of_property=value]
[--add-property-value name_of_property=value_to_add]
[--remove-property-value name_of_property=value_to_remove]
[--unset-property name_of_property_to_delete]
[--proxy proxy_to_use] publisher
```

Quand la commande est sous la forme suivante, le nom de l'éditeur est facultatif car vous avez spécifié l'URI du référentiel :

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

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.

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

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

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

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. L'option `-p` ne peut pas être associée aux options `-g`, `--add-origin`, `-G`, `--remove-origin`, `-m`, `--add-mirror`, `-M`, `--remove-mirror`, `--disable`, `-enable`, `--no-refresh` ou `--reset-uuid`.

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

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

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

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 15](#) pour obtenir une explication sur la différence entre une source et un miroir. Utilisez l'option `-M` pour supprimer un URI servant de miroir à l'éditeur spécifié.

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

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 de CA. Les hashages des certificats de CA 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 55](#).

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

Utilisez l'option `--revoked-ca-cert` pour traiter le certificat spécifié comme révoqué. Les hashages des certificats de CA révoqués 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.

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.

Utilisez l'option `--reset-uuid` pour choisir un nouvel identifiant unique qui identifie cette image auprès de son éditeur.

Utilisez l'option `--proxy` pour spécifier un URI de proxy Web persistant à partir duquel récupérer du contenu pour l'origine (`-g`) ou le miroir (`-m`) indiqué. La valeur proxy est stockée en tant que configuration de l'éditeur. Au moment de l'exécution, `$http_proxy` ou des variables d'environnement connexes remplacent ce paramétrage de proxy. Reportez-vous à la section ENVIRONMENT de la page de manuel `curl(1)` pour plus d'informations sur les variables d'environnement de proxy.

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

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

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*.

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. La plupart des balises de variantes peuvent avoir plusieurs valeurs. Les balises de facettes définies sur une action peuvent uniquement avoir la valeur `true`. 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 sur `false` une facette d'environnement linguistique particulière, tous les fichiers ou autres actions qui spécifient cette facette ne seront pas installés, et les fichiers actuellement installés qui spécifient cette facette sont désinstallés.

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 variantes définies sur l'image, exécutez les commandes `pkg change-facet` et `pkg change-variant`. Reportez-vous à la page de manuel [pkg\(1\)](#) et aux exemples ci-dessous.

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.

`variant.arch=sparc` est un exemple de balise de variante. Voici un exemple de balise de facette : `facet.devel=true`. Il est souvent fait référence aux facettes et variantes sans les faire précéder de `facet.` et `variant.`.

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. Une balise 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 :

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

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

La plupart des variantes peuvent avoir un nombre de valeurs quelconque. Par exemple, il est possible de définir la variante `arch` sur `i386`, `sparc`, `ppc`, `arm` ou une autre architecture prise en charge par la distribution. (Seules les valeurs `i386` et `sparc` sont utilisées dans Oracle Solaris.) Les variantes `debug` constituent une exception. Il est possible de définir les variantes `debug` uniquement sur `true` ou `false`. Toute autre valeur correspond à un comportement indéfini. Si une action sur un fichier présente à la fois des versions avec et sans débogage, il faut définir explicitement la variante `debug` pour les deux versions, 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
```

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.

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.

Vous pouvez créer vos propres balises de facettes et de variantes. Les balises suivantes sont couramment utilisées dans Oracle Solaris.

Nom de la variante	Valeurs possibles
<code>variant.arch</code>	<code>sparc</code> , <code>i386</code>
<code>variant.opensolaris.zone</code>	<code>global</code> , <code>nonglobal</code>
<code>variant.debug.*</code>	<code>true</code> , <code>false</code>

La liste suivante présente quelques exemples de balises de facettes utilisées 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

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. 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.

Affichage et modification des valeurs de variables

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

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

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

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

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

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 76](#) pour obtenir des informations sur la création des environnements d'initialisation.

Utilisez l'option `-C` pour modifier les variantes dans les zones non globales *n* simultanément avec la zone globale. Reportez-vous à la section [“Mise à jour simultanée de plusieurs zones non globales” à la page 52](#) pour voir un exemple d'utilisation de l'option `-C`.

Utilisez l'option `-n` pour voir ce qui serait modifié si vous aviez effectué l'opération sans `-n`, mais sans effectuer de modification réelle.

```
$ pfexec 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 des facettes qui sont définies.

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

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

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

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

Utilisez l'option `-C` pour modifier les facettes dans les zones non globales *n* simultanément avec la zone globale. Reportez-vous à la section [“Mise à jour simultanée de plusieurs zones non globales” à la page 52](#) pour voir un exemple d'utilisation de l'option `-C`.

Utilisez l'option `-n` pour voir ce qui serait modifié si vous aviez effectué l'opération sans `-n`, mais sans effectuer de modification réelle.

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. Cette opération peut nécessiter une grande quantité de temps et entraîner une grande quantité de trafic réseau entre cette image et le référentiel de packages. 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 76](#) pour obtenir des informations sur la création des environnements d'initialisation.

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

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

Verrouillage de packages en une version spécifiée

Utilisez la commande `pkg freeze` pour contraindre une version de package. Vous pouvez par exemple figer un paquet lorsque vous ne souhaitez pas que le package dans une zone non globale soit mis à jour lorsque la zone globale est mise à jour.

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

Si aucune version n'est fournie dans `pkg_fmri_pattern`, le package nommé doit être installé et il est limité à la version installée sur le système. Si la version est fournie dans `pkg_fmri_pattern`, cette contrainte (également appelée figement) agit comme si une dépendance incorporée 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 `pkg_fmri_pattern` est utilisé pour trouver des packages correspondants. Cependant, les informations de l'éditeur ne sont pas enregistrées dans le cadre de l'opération de figement. Un package est figé par rapport à sa version uniquement, et non son éditeur.

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

Si aucun package n'est spécifié, les informations relatives aux packages actuellement figés s'affichent : noms de package, versions, date du figement du package et toute raison associée le cas échéant.

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

Utilisez l'option `-c` pour enregistrer la raison pour laquelle le package est figé. La raison est affichée si un figement empêche l'installation ou la mise à jour de réussir.

Utilisez l'option `-n` pour effectuer un essai de l'opération et afficher la liste des packages qui seraient figés sans réellement figer aucun package.

Dans l'exemple suivant, le package est figé à la version installée. La lettre "f" qui s'affiche dans la liste de packages indique que le package est figé.

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

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

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

Un figement n'est jamais annulé automatiquement par le système d'emballage. Pour relâcher une contrainte, utilisez la commande `pkg unfreeze`.

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

Supprime des packages spécifiés les contraintes imposées par le figement. Les versions éventuellement fournies sont ignorées.

Utilisez l'option `-n` pour effectuer un essai de l'annulation du figement et afficher la liste des paquets qui seraient libérés sans réellement libérer aucun package.

Assouplissement des contraintes de version spécifiées par les incorporations

Chaque package qui fait partie du SE Oracle Solaris 11 possède une dépendance à un package d'incorporation. Les packages d'incorporation limitent les versions de leurs packages incorporés afin de maintenir le système dans un état pris en charge au fur et à mesure des mises à jour. 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. Pour ces packages incorporés, un attribut de facette `version-lock.pkg_name` est spécifié dans le package d'incorporation. La valeur de la facette `version-lock.pkg_name` est `true`. Pour assouplir la contrainte de version sur un package, définissez la valeur de sa facette `version-lock.pkg_name` 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  1.0.0.10-0.175.1.0.0.19.0  i--
library/security/openssl  1.0.0.10-0.175.1.0.0.18.0  ---
$ pfexec pkg update library/security/openssl@1.0.0.10-0.175.1.0.0.18.0
Creating Plan (Solver setup): |
pkg update: No matching version of library/security/openssl can be installed:
Reject: pkg:///solaris/library/security/openssl@1.0.0.10,5.11-0.175.1.0.0.18.0:20120611T201116Z
Reason: This version is excluded by installed incorporation
pkg:///solaris/consolidation/userland/userland-incorporation@0.5.11,5.11-0.175.1.0.0.19.0:20120625T163952Z
```

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 76 pour obtenir des informations sur la création des environnements d'initialisation.

```
$ pfexec pkg change-facet facet.version-lock.library/security/openssl=false
Packages to update: 850
Variants/Facets to change: 1
Create boot environment: No
Create backup boot environment: Yes

PHASE                                ITEMS
Removing old actions                  1/1
Updating image state                  Done
Creating fast lookup database          Done
Reading search index                  Done
Building new search index              850/850
$ pfexec pkg update library/security/openssl@1.0.0.10-0.175.1.0.0.18.0
Packages to update: 1
Create boot environment: No
Create backup boot environment: Yes

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

PHASE                                ITEMS
Removing old actions                  3/3
Installing new actions                3/3
Updating modified actions              14/14
Updating package state database        Done
Updating package cache                 1/1
Updating image state                  Done
Creating fast lookup database          Done
```

```

Reading search index                               Done
Updating search index                             1/1
$ pkg list library/security/openssl
NAME (PUBLISHER)                                VERSION                                IFO
library/security/openssl                        1.0.0.10-0.175.1.0.0.18.0  i--

```

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.

```

$ pfexec pkg freeze -c "Downgrade to avoid bug" library/security/openssl
library/security/openssl was frozen at 1.0.0.10-0.175.1.0.0.18.0:20120611T201116Z
$ pkg list library/security/openssl
NAME (PUBLISHER)                                VERSION                                IFO
library/security/openssl                        1.0.0.10-0.175.1.0.0.18.0  if-

```

Pour réactiver la rétrogradation ou la mise à niveau, utilisez la commande `pkg unfreeze` pour supprimer le figement 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 requi re 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`.

```

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

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

```

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

Spécification d'une implémentation d'application par défaut

Il peut arriver que vous souhaitiez fournir plusieurs versions d'une application ou d'un outil dans la même image. Si les différentes versions sont distribuées dans le cadre de la même médiation, vous pouvez alors facilement réinitialiser la version par défaut ou préférée. Une *médiation* est un ensemble de liens vers différentes implémentations d'une application, où chaque lien possède le même nom de médiateur et le même chemin de lien mais pas les mêmes chemins de lien cibles.

Exécutez la commande `pkg mediator` pour afficher tous les médiateurs dans l'image ou afficher la version sélectionnée actuelle du médiateur spécifié.

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

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

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

Dans l'exemple suivant, deux versions différentes de Java Runtime Environment sont installées.

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

La commande suivante montre qu'une médiation java est définie et que la version 1.7 est la version par défaut.

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

La commande suivante indique toutes les médiations java qui peuvent être définies. `jre-6` et `jre-7` définissent tous les deux un lien symbolique à partir de `/usr/bin/java`. Dans le package `jre-6`, la cible du lien `/usr/bin/java` est `jdk1.6`. Dans le package `jre-7`, la cible du lien `/usr/bin/java` est `jdk1.7`. La commande précédente avait indiqué que la version 1.7 est actuellement la cible du lien `/usr/bin/java`. Cette médiation n'indique pas l'implémentation préférée et le système de package a sélectionné l'implémentation avec la version la plus récente comme l'implémentation préférée.

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

La commande suivante indique comment paramétrer la version 1.6 comme implémentation préférée, auquel cas appeler `/usr/bin/java` reviendra à appeler JRE version 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`.

```
$ pkg mediator java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      system 1.7      system
$ pfexec pkg set-mediator -V 1.6 java
    Packages to update: 3
    Mediators to change: 1
    Create boot environment: No
    Create backup boot environment: No

PHASE                                ITEMS
Removing old actions                 2/2
Updating modified actions            3/3
Updating image state                 Done
Creating fast lookup database        Done
Reading search index                 Done
Updating search index                3/3
$ pkg mediator java
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
java      local  1.6      system
```

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.

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

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

Avec `pkg_fmri_pattern` spécifié, 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.

Si un package est sur la liste à éviter, son installation supprime ce package de cette liste. Les packages placés sur la liste "avoid" sont installés s'ils sont nécessaires pour satisfaire une dépendance de type requi re. Si cette dépendance est supprimée, le package est désinstallé.

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

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

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 cette sous-commande. Pour supprimer de cette liste un package qui correspond à une dépendance de groupe, installez le package.

La sortie de la commande suivante indique que le package de groupe n'est pas installé group/feature/amp et qu'aucun des packages qui font partie de ce package de groupe ne sont installés. Certains de ces packages auraient pu être installés de manière explicite ou en tant que dépendances de type requi re d'autres packages.

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

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

```
$ pfexec pkg avoid web/server/apache-22/module/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. Après l'installation du package de groupe, le package de groupe est indiqué dans la liste "avoid".

```
$ pfexec pkg install group/feature/amp
...
$ pkg list -a 'pkg contents -o fmri -H -rt depend -a type=group group/feature/amp'
NAME (PUBLISHER)      VERSION      IFO
database/mysql-51     5.1.37-0.175.1.0.0.21.0  i--
web/php-52            5.2.17-0.175.1.0.0.21.0  i--
web/php-52/extension/php-apc      3.0.19-0.175.1.0.0.21.0  i--
```

```

web/php-52/extension/php-mysql          5.2.17-0.175.1.0.0.21.0  i--
web/server/apache-22                    2.2.22-0.175.1.0.0.21.0  i--
web/server/apache-22/module/apache-dtrace 0.3.1-0.175.1.0.0.21.0  i--
web/server/apache-22/module/apache-fcgid  2.3.6-0.175.1.0.0.21.0  ---
web/server/apache-22/module/apache-php5   5.2.17-0.175.1.0.0.18   i-r
$ pkg avoid
    web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

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.

```

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

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.

```

$ pfexec pkg avoid web/server/apache-22/module/apache-fcgid
pkg avoid: The following packages are already installed in this image; use uninstall to avoid these:
    web/server/apache-22/module/apache-fcgid
$ pfexec pkg uninstall web/server/apache-22/module/apache-fcgid
...
$ pkg avoid
    web/server/apache-22/module/apache-fcgid (group dependency of 'group/feature/amp')
```

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

```

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

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.

```

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

Mise à jour d'une image

Exécutez la commande `pkg update` sans spécifier de valeur *pkg-fmri* ou en saisissant un astérisque (*) à la place de la valeur *pkg-fmri* pour mettre à jour 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. Si des zones non globales sont montées dans l'image actuelle, ces zones sont également mises à jour. Reportez-vous à la section [“Mise à jour simultanée de plusieurs zones non globales”](#) à la page 52.

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

Meilleures pratiques de mise à jour d'images

Avant d'exécuter la commande `pkg update`, vérifiez les versions disponibles à partir de l'origine d'éditeur configurée, et utilisez les options `-nv` pour afficher la liste des packages qui seront mis à jour, sans pour autant effectuer la mise à jour.

Si vous souhaitez mettre à jour votre système d'exploitation, vérifiez les versions disponibles du package d'incorporation *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 13.

```
$ 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 57.

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`.

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

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

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.

L'exemple précédent indique qu'un nouvel environnement d'initialisation serait créé pour cette mise à jour si vous exécutiez 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 76](#) 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 ce n'est

pas ce que vous souhaitez, indiquez l'option `--no-be-activate` avec la commande `pkg update`. Lorsque vous êtes prêt à utiliser le nouvel environnement, exécutez la commande `beadm activate` pour activer le nouvel environnement d'initialisation.

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 11.1/11, même si une version plus récente est autorisée. Assurez-vous d'utiliser à nouveau les options `-nv` et vérifiez à nouveau la sortie.

```
$ pfexec pkg update -nv entire@0.5.11,5.11-0.175.0.13
  Packages to remove:      2
  Packages to install:     1
  Packages to update:     486
  Estimated space available: 48.39 GB
  Estimated space to be consumed: 2.50 GB
  Create boot environment: Yes
  Activate boot environment: Yes
  Create backup boot environment: No
  Rebuild boot archive:    Yes

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

Certains packages installés peuvent ne pas appartenir à l'incorporation entière. Ces packages peuvent avoir été installés séparément et ils ne seront pas mis à jour lors de la mise à jour de l'incorporation entière 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 11.1/11 mais ne pas autoriser de mise à jour vers Oracle Solaris 11.1, vous pouvez figer l'incorporation entière comme indiqué dans la commande suivante. L'indication `0.175.0` signifie que le package entier peut être mis à jour vers `0.175.0.13` par exemple, mais pas vers `0.175.1`.

```
$ pfexec pkg freeze -c "Keep this image at 11.1/11." 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 figement de packages, reportez-vous à la section [“Verrouillage de packages en une version spécifiée” à la page 64.](#)

Imposition de contraintes sur les packages disponibles

Vous pouvez également contrôler les versions pouvant être installées ou mises à jour en indiquant votre propre référentiel de packages IPS local et en contrôlant son contenu. Par exemple, votre référentiel peut contenir tous les mises à jour de support pour Oracle Solaris 11 11/11, mais ne pas contenir de packages Oracle Solaris 11.1.

Si vous créez votre propre référentiel, assurez-vous de mettre à jour votre origine d'éditeur `solaris`.

Pour plus d'informations, reportez-vous à la section [Copie et création de référentiels de packages Oracle Solaris 11.1.](#)

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.

Options supplémentaires de la commande de mise à jour

Lorsque vous spécifiez l'option `-f` lors de la mise à jour de tous les packages installés, il n'est pas vérifié si le client est à jour.

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. Reportez-vous à la section [“Installation d'un nouveau package” à la page 40](#) pour une description détaillée et des exemples illustrant les effets de l'option `-g`.

Utilisez l'option `-C` pour mettre à jour n zones non globales en même temps que la zone globale. Reportez-vous à la section “[Mise à jour simultanée de plusieurs zones non globales](#)” à la page 52 pour obtenir un exemple.

Utilisez l'option `-accept` pour indiquer que vous acceptez les conditions de licence des packages qui sont mis à jour. Si vous ne spécifiez pas cette option alors que les licences de package exigent l'acceptation, l'opération de mise à jour échoue. Utilisez l'option `--licenses` pour afficher toutes les licences pour les packages mis à jour dans le cadre de cette opération.

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.

Lorsque vous spécifiez l'option `--no-index`, les index de recherche ne sont pas mis à jour une fois l'opération terminée. La définition de cette option peut vous faire gagner du temps si vous installez un grand nombre de packages. Lorsque l'opération de mise à jour est terminée, 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. Si un nouvel environnement d'initialisation a été créé, actualisez l'éditeur dans ce nouvel environnement d'initialisation.

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 (BE) 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 installées. 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. Ceci peut se produire lorsque vous installez, désinstallez, mettez à jour, ou modifiez une variante ou une facette.
Il arrive souvent qu'un environnement d'initialisation soit créé lorsque vous exécutez la commande `pkg update` pour mettre à jour tous les packages pour lesquels des mises à jour sont disponibles.
- Indication de l'une des options suivantes : `--be-name`, `--require-new-be`, `--backup-be-name`, `--require-backup-be`.
- Définition de 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 l'environnement d'initialisation clone sans mettre à jour les packages dans l'environnement d'initialisation actuel.
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. Il 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.

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 11.1](#)*.

Reportez-vous à la section “[Définitions des propriétés d'image](#)” à la page 82 pour obtenir des instructions sur la définition des propriétés de l'image ci-dessous.

`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`.

a lways - 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'emballage 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 :

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

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

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

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

```
$ pfexec 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.

```
$ pfexec 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.

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

```
$ pfexec pkg remove-property-value signature-require-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.

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

`flush-content-cache-on-success`

Si cette propriété est définie sur `True`, le client de package supprime les fichiers de son cache de contenu une fois les opérations d'installation ou de mise à jour terminées. Pour les opérations de mise à jour, le contenu n'est supprimé que dans l'environnement d'initialisation source. Lorsqu'une opération sur les packages se produit ensuite sur l'environnement d'initialisation de destination, le client du package vide son cache de contenu si cette option n'a pas été modifiée.

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

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

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
```

Les propriétés `preferred-authority` et `publisher-search-order` peuvent être définies à l'aide des options de commande `pkg set-publisher`. Reportez-vous à la section [“Ajout, modification ou suppression des éditeurs de packages”](#) à la page 57.

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`.

```
$ pfexec 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.

```
$ pfexec 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.

Une image peut être de trois types différents :

- Les images complètes peuvent fournir un système complet. Dans une image entière, toutes les dépendances sont résolues au sein de l'image elle-même et IPS gère les dépendances de manière cohérente. Après l'installation du SE Oracle Solaris, une image complète contient le système de fichiers root et son contenu.
- Les images partielles sont liées à une image complète (image parent) mais ne suffisent pas pour fournir un système complet. Une zone non globale est une image partielle. Utilisez l'option `-z` ou `--zone` 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 la section [Partie II, “Oracle Solaris Zones” du manuel *Administration Oracle Solaris : Oracle Solaris Zones, Oracle Solaris 10 Zones et gestion des ressources*](#) pour obtenir des informations sur les zones non globales.
- Les images d'utilisateur contiennent uniquement des packages réadressables.

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

A l'emplacement donné par *dir*, créez une image appropriée aux opérations sur les packages. L'image par défaut est de type user (`-U` ou `--user`). Le type d'image peut être défini pour une image complète (`-F` ou `--full`) ou une image partielle (`-P` ou `--partial`) liée à l'image complète englobant le chemin d'accès *dir*.

Pour exécuter la nouvelle image dans un contexte de zone non globale, utilisez l'option `-z` ou `--zone` pour définir une variante adéquate.

Un référentiel de packages URI doit être fourni à l'aide de l'option `-p` ou `--publisher`. 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.

Utilisez l'option `-g` pour spécifier des origines supplémentaires. Utilisez l'option `-m` pour spécifier les miroirs.

Pour les éditeurs utilisant l'authentification de client SSL, utilisez les options `-c` ou `-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 l'option `-f` pour forcer la création d'une image sur une image existante. Utilisez cette option avec prudence.

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.

Utilisez l'option `--variant` pour définir la variante spécifiée sur la valeur indiquée. Utilisez l'option `--facet` pour définir la facette spécifiée sur la valeur indiquée.

Affichage de l'historique des opérations

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

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

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

Utilisez l'option `-n` pour afficher uniquement le nombre spécifié d'opérations les plus récentes.

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

Utilisez l'option `-o` pour afficher la sortie à l'aide de la liste de noms de colonne séparés par des virgules spécifiée. 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
2012-08-06T16:32:03                 0:00:27                 fix                      Succeeded
2012-08-06T16:41:47                 0:00:43                 revert                  Succeeded
2012-08-06T17:56:22                 0:00:00                 set-property            Succeeded
2012-08-06T17:56:53                 0:00:00                 unset-property          Succeeded
```

Utilisez l'option `-t` pour consigner les enregistrements pour une liste d'horodatages séparés par des virgules, formatés avec `%Y-%m-%dT%H:%M:%S` (voir `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` peut être utilisé en tant qu'alias pour l'heure actuelle. Si les horodatages spécifiés contiennent des doublons ou des plages de dates qui se chevauchent, une seule instance de chaque événement historique en double est imprimée.

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`.

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

\$ pfexec pkg purge-history

