

## **Ajout et mise à jour de packages logiciels Oracle® Solaris 11**

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

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

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

#### U.S. GOVERNMENT 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 des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

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

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

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

# Table des matières

---

<b>Préface .....</b>	<b>7</b>
<b>1 Introduction à Image Packaging System .....</b>	<b>11</b>
Image Packaging System .....	11
Privilèges d'installation .....	12
Concepts d'IPS .....	12
Packages IPS .....	12
Identificateurs de ressource de gestion des pannes .....	13
Editeurs, référentiels et archives de paquets .....	14
Sources et miroirs du référentiel .....	15
Images et environnements d'initialisation .....	15
Facettes et variantes de package .....	15
<b>2 Interfaces graphiques IPS .....</b>	<b>17</b>
Utilisation du Gestionnaire de packages .....	17
Options de ligne de commande du Gestionnaire de packages .....	18
Utilisation de l'Installation Web .....	18
Utilisation du Gestionnaire de mises à jour .....	20
Options de ligne de commande du Gestionnaire de mises à jour .....	21
<b>3 Obtention d'informations sur les packages logiciels .....</b>	<b>23</b>
Affichage des informations d'état d'installation du package .....	23
Affichage des descriptions ou des licences de package .....	25
Affichage des informations à partir du fichier manifeste du package .....	27
Liste des fichiers installés par un package .....	27
Liste de tous les packages installables dans un package de groupe .....	28
Affichage des conditions d'octroi de licence .....	28

Recherche de packages .....	29
Identification du package fournissant un fichier spécifique .....	29
Liste des packages par catégorie .....	30
Affichage des packages dépendants .....	30
Liste de tous les packages dans un package de groupe .....	31
<b>4 Installation et mise à jour des packages logiciels .....</b>	<b>33</b>
Aperçu d'une opération .....	34
Installation et mise à jour de packages .....	35
Options d'environnement d'initialisation .....	35
Installation d'un nouveau package .....	36
Installation d'un package dans un nouvel environnement d'initialisation .....	38
Rejet d'un package .....	40
Mise à jour d'un package .....	40
Résolution des problèmes de package .....	41
Vérification de l'installation des packages .....	42
Correction des erreurs de vérification .....	42
Restauration d'un fichier .....	43
Désinstallation de packages .....	43
<b>5 Configuration des images installées .....</b>	<b>45</b>
Configuration des éditeurs .....	45
Affichage des informations sur les éditeurs .....	45
Ajout, modification ou suppression des éditeurs de packages .....	46
Verrouillage de packages en une version spécifiée .....	48
Marquage des packages à éviter .....	49
Contrôle de l'installation des composants optionnels .....	50
Affichage d'une variante .....	52
Modification d'une variante .....	52
Affichage d'une facette .....	53
Modification d'une facette .....	53
Mise à jour d'une image .....	54
Configuration des propriétés d'image et d'éditeur .....	55
Propriétés d'image de la stratégie d'environnement d'initialisation .....	55
Propriétés de signature de packages .....	57

Propriétés de l'image supplémentaires .....	59
Définitions des propriétés d'image .....	60
Création d'une image .....	62
Affichage de l'historique des opérations .....	63



# Préface

---

*Ajout et mise à jour de packages logiciels Oracle Solaris 11* 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](#).

## 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).

- [Pages de manuel d'Image Packaging System](#)
- [Copie et création de référentiels de packages Oracle Solaris 11](#)
- [Création et administration d'environnements d'initialisation Oracle Solaris 11](#)
- [Installation des systèmes Oracle Solaris 11](#)

## Accès au support technique 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> adapté aux utilisateurs malentendants.

## 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 . login.  Utilisez <code>ls -a</code> pour afficher la liste de tous les fichiers.  <code>nom_machine%</code> Vous avez reçu du courrier.
<b>AaBbCc123</b>	Ce que vous entrez, par opposition à ce qui s'affiche à l'écran.	<code>nom_machine% su</code>  Mot de passe :
<i>aabbcc123</i>	Paramètre fictif : à remplacer par un nom ou une valeur réel(le).	La commande permettant de supprimer un fichier est <code>rm nom_fichier</code> .



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.  <b>Remarque</b> : en ligne, certains éléments mis en valeur s'affichent en gras.

# Invites de shell dans les exemples de commandes

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

TABLEAU P-2 Invites de shell

Shell	Invite
Bash shell, korn shell et bourne shell	\$
Bash shell, korn shell et bourne shell pour superutilisateur	#
C shell	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 de capacités accessibles via l'interface de ligne de commande IPS est disponible par le biais de l'interface utilisateur graphique du Gestionnaire de packages.

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

- Lister, rechercher, installer, restreindre l'installation mettre à jour et supprimer des packages de logiciels.
- Lister, ajouter et supprimer des éditeurs de package. Modifier des attributs d'éditeur comme la priorité de recherche et la persistance. 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](#).

## 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 :

- Utilisez la commande `profiles` pour répertorier les profils de droits d'accès qui vous sont affectés. Si vous disposez du profil de droits d'installation des logiciels, vous pouvez utiliser la commande `pfexec` pour installer et mettre à jour les packages.

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

Les autres droits de profils fournissent également des privilèges d'installation, notamment le profil de droits d'administrateur.

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

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

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

## 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. Les actions appelées "set" définissent les métadonnées du package tels que la classification, la synthèse, et la 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.

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.

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.

## 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 :

*scheme://publisher/package\_name@version:dateTimeZ*

Exemple :

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

Méthode                      pkg

Editeur                      solaris

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

Nom du package            editor/vim

L'espace de noms de package est hiérarchique et de profondeur quelconque. Dans les commandes IPS, vous pouvez spécifier la plus petite partie du nom du package qui identifie de façon unique le package. 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      7.3.254

	Pour les composants étroitement liés au système d'exploitation, il s'agit généralement de la valeur de <code>uname -r</code> pour cette version du système d'exploitation.
Version de compilation	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	0.174.0.0.0.0.504
	La version de branche doit suivre un trait d'union (-). La version de branche fournit des informations spécifiques au fournisseur.
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 : <code>AAAAMMJJTHHMMSSZ</code> .

## Editeurs, référentiels et archives de paquets

Un *éditeur* identifie une personne ou une organisation qui fournit un ou plusieurs packages. Les éditeurs peuvent distribuer leurs packages en utilisant des référentiels de packages ou des 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 le package n'est pas trouvé, le deuxième éditeur dans l'ordre de recherche est recherché, et ainsi de suite jusqu'à ce que le package soit trouvé ou que tous les éditeurs aient été recherchés.

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 installées. Lorsque vous initialisez votre système, vous avez la possibilité de démarrer dans n'importe quel EI 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 55.

## 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 de l'image et ne peuvent pas être définies sur des packages individuels.

Les différentes actions incluses dans les manifestes de package peuvent posséder des repères de facettes et de variantes. Une seule action peut avoir plusieurs repères de facette et de variante.

L'interaction des valeurs des repères de facettes et de variantes associées à une action et des valeurs des facettes et des variantes définies dans l'image déterminent si l'action concernée peut être installée.

- Les actions sans repère de facette ou de variante sont toujours installées.
- Les actions avec repères de facettes sont installées, sauf si toutes les facettes ou tous les modèles de facettes correspondant aux repères 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 repères de variantes sont uniquement installées si les valeurs des repères de variantes sont les mêmes que celles définies dans l'image.
- Les actions comportant à la fois des repères de facettes et des repères 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 50.



# 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 :

<b>Barre d'outils</b>	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.
<b>Icône du bureau</b>	Cliquez deux fois sur l'icône du Gestionnaire de packages sur le bureau.
<b>Barre de menu</b>	Choisissez Système > Administration > Gestionnaire de packages.
<b>Ligne de commande</b>	# <b>packagemanager</b>

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

# Options de ligne de commande du Gestionnaire de packages

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

TABLEAU 2-1 Options de commande du Gestionnaire de packages

Option	Description
<code>--image-dir</code> ou <code>-R dir</code>	S'exécute sur l'image résidant au niveau de <i>dir</i> . Par défaut, la commande s'exécute sur l'image actuelle.  La commande suivante s'exécute sur l'image stockée au niveau de <code>/aux0/example_root</code> :  <b># packagemanager -R /aux0/example_root</b>
<code>--update-all</code> ou <code>-U</code>	Met à jour tous les packages installés pour lesquels une mise à jour est disponible. Spécifier cette option équivaut à choisir l'option Mises à jour dans l'interface graphique du Gestionnaire de packages. Pour plus d'informations sur la mise à jour de tous les packages, reportez-vous à la section “Utilisation du Gestionnaire de mises à jour” à la page 20.
<code>--info-install</code> ou <code>-i file.p5i</code>	Spécifie un fichier <code>.p5i</code> pour exécuter le Gestionnaire de packages en mode Installation Web. Le fichier spécifié doit avoir l'extension <code>.p5i</code> . Pour de plus amples informations, reportez-vous à la section “Utilisation de l'Installation Web” à la page 18.
<code>--help</code> ou <code>-h</code>	Affiche les informations d'utilisation de la commande.

## Utilisation de l'Installation Web

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

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

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

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

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

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

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

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

1. Utilisez l'une des méthodes suivantes pour démarrer le Gestionnaire de packages en mode Installation Web :
  - 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 Installer/Mettre à jour s'affiche. L'étiquette en haut de la fenêtre est : "Package Manager Web Installer/The following will be added to your system." (Programme d'installation Web du Gestionnaire de packages/Les éléments suivants seront ajoutés à votre système.) Les éditeurs et les packages à installer apparaissent dans une liste. Cliquez sur le bouton Continuer pour poursuivre l'installation.
3. Si l'éditeur de package spécifié n'est pas encore configuré sur votre système, la fenêtre Ajouter un éditeur s'affiche. Le nom et l'URI de l'éditeur sont déjà spécifiés.

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

Si l'éditeur a bien été ajouté, la boîte de dialogue 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 Activer un éditeur. Utilisez cette boîte de dialogue pour activer l'éditeur de sorte à pouvoir installer les packages.

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

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

## Utilisation du Gestionnaire de mises à jour

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

- Dans l'interface graphique du Gestionnaire de packages, choisissez le bouton Mises à jour ou l'option de menu Package > Mises à jour.
- Utilisez la commande `packagemanager`.  
`# packagemanager --update-all`
- Utilisez la commande `pkg`.  
`# pkg update`

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

<b>Barre d'état</b>	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.
---------------------	--

<b>Barre de menu</b>	Choisissez Système > Administration > Gestionnaire de mises à jour.
----------------------	---

<b>Ligne de commande</b>	<code># pm-updatemanager</code>
--------------------------	---------------------------------

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

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

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

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

```
package/pkg
package/pkg/package manager
package/pkg/updatemanager
```

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

Un nouvel environnement d'initialisation peut être créé, 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 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 Redémarrer maintenant pour redémarrer votre système immédiatement. Cliquez sur le bouton Redémarrer plus tard pour redémarrer votre système à une date ultérieure. Vous devez redémarrer pour initialiser dans le nouvel 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	<p>S'exécute sur l'image résidant au niveau de <i>dir</i>. Par défaut, la commande s'exécute sur l'image actuelle.</p> <p>La commande suivante met à jour l'image au niveau de /aux0/example_root :</p> <p><b># pm-updatemanager -R /aux0/example_root</b></p>
- -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. 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 (isv.com)      1.0              i--
system/dtrace/dtrace-toolkit 0.99-0.174.0.0.0.17765 i--
```

Le nom de l'éditeur entre parenthèses indique que l'éditeur `isv.com` n'est pas le premier dans l'ordre de recherche d'éditeur 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   3.0.15-0.174.0.0.0.0  ---
isvtoolkit (isv.com)      1.0              i--
system/dtrace/dtrace-toolkit 0.99-0.174.0.0.0.0.17765 i--
```

Cette sortie indique que 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   3.0.15-0.174.0.0.0.0  ---
isvtoolkit (isv.com)      1.0              i--
system/dtrace/dtrace-toolkit 0.99-0.174.0.0.0.0.17765 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. Le package s'affiche sans utiliser l'option -f car le package web/amp a été renommé group/feature/amp, et group/feature/amp n'est pas installé. Si vous spécifiez la commande pour installer le package web/amp, le package group/feature/amp est installé automatiquement.

```
$ pkg list -a amp
NAME (PUBLISHER)          VERSION          IFO
group/feature/amp        0.5.11-0.174.0.0.0.0.2559 ---
web/amp                   0.5.11-0.174.0.0.0.0.0  --r
```

La commande pkg list ne permet pas d'obtenir le nouveau nom d'un package renommé. Pour obtenir ces informations, utilisez la commande pkg info comme illustré dans la section [“Affichage des descriptions ou des licences de package” à la page 25](#).

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        5.1.37-0.174.0.0.0.0.504 ---
```



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 48 pour plus d'informations sur la manière de figer les packages.

```
$ pkg list mercurial
NAME (PUBLISHER)          VERSION          IFO
developer/versioning/mercurial  1.8.4-0.174.0.0.0.504  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
pkg:///solaris/database/mysql-51@5.1.37,5.11-0.174.0.0.0.504:20110920T230125Z --- IFO
```

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

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'emballage, 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.

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

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: Fri Aug 26 14:55:51 2011
Size: 5.45 kB
FMRI: pkg://solaris/developer/dtrace/toolkit@0.99,5.11-0.173.0.0.0.1.0:20110826T145551Z
```

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

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.

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

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.

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.

## 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`.

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

Si vous affichez le fichier manifeste du package, vous pouvez voir que le package `e1000g` a sept actions de fichiers. Les trois 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 et n'inclut pas de fichiers de débogage. Le fichier de débogage pour l'architecture x86 n'est pas indiqué ci-dessus, et les fichiers de débogage et de non-débogage ne sont pas indiqués pour l'architecture SPARC. Vous pouvez modifier la variante de débogage de l'image afin d'inclure ou non des fichiers de débogage dans l'image. Reportez-vous à la section [“Contrôle de l'installation des composants optionnels”](#) à la page 50.

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

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`.

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

## 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, les termes de requête de `pkg search` sont mis en correspondance exacte, excepté 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 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_type: key:token
```

Les champs manquants sont implicitement considérés comme des caractères génériques. Des caractères génériques explicites peuvent être utilisés dans les champs *pkg\_name* et *token*. Les champs *action\_type* et *key* doivent correspondre exactement. Pour obtenir une liste des actions et des clés de packages, reportez-vous à 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.

## Identification du package fournissant un fichier spécifique

L'exemple suivant montre que la bibliothèque `libpower` provient du package `system/kernel/power`.

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

## 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 ':set:info.classification:Source Code Management'
INDEX      ACTION VALUE      PACKAGE
info.classification set      Development/Source Code Management pkg:/developer/versioning/subversion@1.6.16
info.classification set      Development/Source Code Management pkg:/developer/versioning/git@1.7.3.2-0.174
info.classification set      Development/Source Code Management pkg:/developer/versioning/sccs@0.5.11-0.174
info.classification set      Development/Source Code Management pkg:/library/perl-5/subversion@1.6.16-0.174
info.classification set      Development/Source Code Management pkg:/library/java/subversion@1.6.16-0.174.0
info.classification set      Development/Source Code Management pkg:/library/python-2/subversion@1.6.16-0.1
info.classification set      Development/Source Code Management pkg:/developer/xopen/xcu4@0.5.11-0.174.0.0.
info.classification set      Development/Source Code Management pkg:/developer/quilt@0.47-0.174.0.0.0.504
info.classification set      Development/Source Code Management pkg:/developer/versioning/cvs@1.12.13-0.174
info.classification set      Development/Source Code Management pkg:/developer/versioning/mercurial@1.8.4-0
```

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 -o pkg.name -H ':set:info.classification:Source Code Management'
developer/versioning/subversion
developer/versioning/git
developer/versioning/sccs
library/perl-5/subversion
library/java/subversion
library/python-2/subversion
developer/xopen/xcu4
developer/quilt
developer/versioning/cvs
developer/versioning/mercurial
```

## Affichage des packages dépendants

Ces exemples montrent les packages qui dépendent du package spécifié.

L'exemple suivant montre les packages qui ont une dépendance de type `require` sur le package `system/kernel/power`. Si vous avez utilisé la commande `pkg contents` pour afficher les actions `depend` 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 -l -H -o pkg.name 'depend:require:system/kernel/power'
system/kernel/dynamic-reconfiguration/i86pc
system/hal
```

L'exemple suivant montre que de nombreux modules ont une dépendance de type `exclude` sur `pkg:/x11/server/xorg@1.10.99`.

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

## Liste de tous les packages 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`. 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 -o fmri -H '*/solaris-desktop:depend:group:'
archiver/gnu-tar
audio/audio-utilities
...
```

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-desktop` en tant que dépendance 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 provenant de cette commande est plus important que le nombre de résultats provenant de la commande `pkg contents` car ces résultats de recherche incluent les noms de tous les packages qui sont nommés dans les actions de groupe de type `depend` dans le package spécifié, et pas seulement dans les packages installables. Par exemple, des variantes de package qui ne sont pas installables dans cette image peuvent être incluses. Comparez la sortie de cette recherche à la sortie de la commande `pkg contents` figurant à la section [“Liste de tous les packages installables dans un package de groupe”](#) à la page 28.

---

**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 instructions et les résultats présentés dans ce chapitre supposent que la configuration de l'image est celle par défaut.

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

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 12](#) 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 :

```
# pkg install -nv group/feature/amp
  Packages to install: 8
    Estimated space available: 112.19 GB
  Estimated space to be consumed: 452.42 MB
    Create boot environment: No
  Create backup boot environment: No
    Services to change: 2
    Rebuild boot archive: No
Changed packages:
solaris
  database/mysql-51
    None -> 5.1.37,5.11-0.174.0.0.0.0.504:20110920T230125Z
  group/feature/amp
    None -> 0.5.11,5.11-0.174.0.0.0.0.2559:20110921T191255Z
  web/php-52
    None -> 5.2.17,5.11-0.174.0.0.0.0.504:20110921T041858Z
  web/php-52/extension/php-apc
    None -> 3.0.19,5.11-0.174.0.0.0.0.504:20110921T041245Z
  web/php-52/extension/php-mysql
    None -> 5.2.17,5.11-0.174.0.0.0.0.504:20110921T041411Z
  web/server/apache-22/module/apache-dtrace
    None -> 0.3.1,5.11-0.174.0.0.0.0.504:20110921T042357Z
  web/server/apache-22/module/apache-fcgid
    None -> 2.3.6,5.11-0.174.0.0.0.0.504:20110921T042430Z
  web/server/apache-22/module/apache-php5
    None -> 5.2.17,5.11-0.174.0.0.0.0.504:20110921T042738Z
Services:
  restart_fmri:
    svc:/system/manifest-import:default
    svc:/system/rbac:default
```

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

```
# 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` du fichier `file` mentionnés dans la page de manuel [pkg\(5\)](#) pour comprendre comment les fichiers avec 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 de sauvegarde peut être créé lorsque vous installez, mettez à jour ou désinstallez un package ou rétablissez un fichier. 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 55 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.

<code>--no-be-activate</code>	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 <a href="#">beadm(1M)</a> 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 <a href="#">“Propriétés d'image de la stratégie d'environnement d'initialisation”</a> à la page 55 pour obtenir des informations sur la création automatique des environnements d'initialisation de sauvegarde.
<code>--backup-be-name <i>name</i></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 <a href="#">“Propriétés d'image de la stratégie d'environnement d'initialisation”</a> à la page 55 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 <i>name</i></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.

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 la persistance et l'ordre de recherche des éditeurs ou en spécifiant l'éditeur dans l'identificateur de ressource de gestion des pannes de package. Vous pouvez également spécifier la version que vous souhaitez installer dans l'identificateur de ressource de gestion des pannes de package. Reportez-vous à [“Identificateurs de ressource de gestion des pannes” à la page 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 45](#) pour obtenir des informations sur la définition de la persistance et de l'ordre de recherche des éditeurs.

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

Pour installer un package d'un éditeur spécifique, indiquez le nom de l'éditeur dans la commande `pkg_fmri_pattern`. Dans l'exemple suivant, `isv.com` est le nom de l'éditeur.

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

Pour installer une version spécifique d'un package, spécifiez les informations de version dans la commande `pkg_fmri_pattern`.

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

Pour demander explicitement la dernière version d'un package, utilisez `latest` pour la portion de version de `pkg_fmri_pattern`.

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

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. Après `install` ou `update`, tout package fourni par les éditeurs n'ayant pas été trouvé dans l'image est ajouté à la configuration de l'image sans point d'origine.

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.

## 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 55](#) 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.

```
# pkg install --be-name s1lamp group/feature/amp
      Packages to install:  8
      Create boot environment: Yes
Create backup boot environment: No
```

DOWNLOAD	PKGS	FILES	XFER (MB)
Completed	8/8	640/640	70.9/70.9
PHASE	ACTIONS		
Install Phase	942/942		
PHASE	ITEMS		

```

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

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

```

A clone of solaris-174 exists and has been updated and activated.  
On the next boot the Boot Environment s1lamp 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 `s1lamp`.

Utilisez la commande `beadm list` pour vérifier que le système est doté d'un nouvel environnement d'initialisation actif nommé `s1lamp`. L'environnement d'initialisation "N" est en cours d'initialisation ; l'environnement d'initialisation "R" est la valeur par défaut à la 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
--      -
s1lamp  R      -                20.75G static 2011-09-23 13:58
solaris -      -                44.81M static 2010-11-07 17:45
solaris-151a -  -                158.12M static 2010-11-12 14:37
solaris-174 N    /                30.04M static 2011-09-02 12:38

```

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.

```

# beadm mount s1lamp /mnt
# pkg -R /mnt list group/feature/amp
NAME (PUBLISHER)      VERSION      IFO
group/feature/amp     0.5.11-0.174.0.0.0.2559  i--

```

N'oubliez pas de démonter l'environnement d'initialisation `s1lamp`.

```

# beadm list
BE      Active Mountpoint Space  Policy Created
--      -
s1lamp  R      /mnt        20.75G static 2011-09-23 13:58
solaris -      -                44.81M static 2010-11-07 17:45
solaris-151a -  -                158.12M static 2010-11-12 14:37
solaris-174 N    /                30.05M static 2011-09-02 12:38
# beadm unmount s1lamp
# beadm list
BE      Active Mountpoint Space  Policy Created
--      -

```

sllamp	R	-	20.75G	static	2011-09-23	13:58
solaris	-	-	44.81M	static	2010-11-07	17:45
solaris-151a	-	-	158.12M	static	2010-11-12	14:37
solaris-174	N	/	30.06M	static	2011-09-02	12:38

## 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 [“Marquage des packages à éviter” à la page 49](#) pour obtenir des informations sur la liste des packages à éviter.

```
# pkg install -nv --reject cvs 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 la persistance et l'ordre de recherche des éditeurs ou en spécifiant l'éditeur dans l'identificateur de ressource de gestion des pannes de package. Vous pouvez également spécifier la version que vous souhaitez installer dans l'identificateur de ressource de gestion des pannes de package. Reportez-vous à [“Identificateurs de ressource de gestion des pannes” à la page 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 45](#) pour obtenir des informations sur la définition de la persistance et de l'ordre de recherche des éditeurs.

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

Pour demander explicitement la dernière version d'un package, utilisez `latest` pour la portion de version de `pkg_fmri_pattern`.

```
# 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. Après `install` ou `update`, tout package fourni par les éditeurs n'ayant pas été trouvé dans l'image est ajouté à la configuration de l'image sans point d'origine.

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 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.

Si vous utilisez la commande `pkg update` sans *pkg-fmri* spécifié ou si la valeur *pkg-fmri* spécifiée est un astérisque (\*), tous les packages installés avec des mises à jour disponibles sont mis à jour. Utilisez la commande `pkg list -u` pour afficher la liste des packages pour lesquels des mises à jour sont disponibles. 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.

## Résolution des problèmes de package

Un exemple de problème qui pourrait se produire après l'installation d'un package est un fichier fourni par le package qui devient corrompu. 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 -l -Ho 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 ignore, 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 57 pour une explication de l'application des stratégies de signature.

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.

```
# 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.

```
# pkg fix --accept system/install/auto-install/auto-install-common
Verifying: pkg://solaris/system/install/auto-install/auto-install-common  ERROR
    file: usr/share/auto_install/manifest/default.xml
    Missing: regular file does not exist
Created ZFS snapshot: 2011-09-28-05:34:02
Repairing: pkg://solaris/system/install/auto-install/auto-install-common

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

PHASE                                    ACTIONS
```

Update Phase	1/1
PHASE	ITEMS
Image State Update Phase	2/2

## 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.

## Désinstallation de packages

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

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

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 [“Marquage des packages à éviter” à la page 49](#) pour obtenir des informations sur la liste des packages à éviter.

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.



# 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 du package et la stratégie de configuration d'environnement d'initialisation.

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

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/solaris11/release/
SSL Key: None
SSL Cert: None
Client UUID: 00000000-3db4-fcc2-0111-000000000000
Catalog Updated: Thu Sep 22 21:06:03 2011
Enabled: Yes
Signature Policy: verify
```

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.

## 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 de la persistance d'un éditeur. Un éditeur récemment ajouté est persistant par défaut. Si un éditeur n'est pas persistant, un package installé à partir de cet éditeur peut être mis à jour à partir d'un autre éditeur.
- Définition de l'ordre de recherche des éditeurs. Un éditeur récemment ajouté est affiché par défaut en dernier dans l'ordre de recherche. L'ordre de recherche des éditeurs sert à trouver des packages à installer. L'ordre de recherche des éditeurs est utilisé pour trouver des packages à mettre à jour si l'éditeur à partir duquel le package a été installé initialement est non persistant.
- 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 58.

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

```
[--approve-ca-cert path_to_CA]
[--revoke-ca-cert hash_of_CA_to_remove]
[--unset-ca-cert hash_of_CA_to_remove]
[--set-property name_of_property=value]
[--add-property-value name_of_property=value_to_add]
[--remove-property-value name_of_property=value_to_remove]
[--unset-property name_of_property_to_delete]
[publisher]
```

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

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

La commande suivante vous permet d'activer l'éditeur `example.com` et le met devant l'éditeur `isv.com` dans l'ordre de recherche.

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

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 supprimer l'ancien. Utilisez l'option `-g` pour ajouter un nouvel URI d'origine. Utilisez l'option `-G` pour supprimer l'ancien URI d'origine.

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

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.

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

Utilisez l'option `-k` pour spécifier la clé SSL client. Utilisez l'option `-c` pour spécifier le certificat SSL client. Utilisez l'option `--approve-ca-cert` pour ajouter le certificat spécifié comme certificat d'AC fiable. Les hachages des certificats 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 45](#).

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

Utilisez l'option `--revoked-ca-cert` pour traiter le certificat spécifié comme révoqué. Les hachages des certificats 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 la commande `pkg unset-publisher` pour supprimer un éditeur.

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

## Verrouillage de packages en une version spécifiée

Utilisez la commande `pkg freeze` pour contraindre une version de package. Vous pouvez par exemple geler 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 gel) agit comme si une dépendance `incorporate` avait été installée là où l'attribut `fmri` avait la valeur de la version du package fourni.

Lorsqu'un package figé est installé ou mis à jour, sa version doit correspondre à celle à laquelle il a été figé. Par exemple, si un package a été figé à la version 1.2, il peut être mis à jour vers 1.2.1, 1.2.9, 1.2.0.0.1, et ainsi de suite. Ce package ne peut pas avoir un numéro de version correspondant à 1.3 ou 1.1.

Un éditeur spécifié dans `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 gel d'un package qui est déjà gelé remplace la version gelée par la nouvelle version spécifiée.

Si aucun package n'est spécifié, les informations relatives aux packages actuellement gelés s'affichent : noms de package, versions, date du gel 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 gelé. 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 gelés sans réellement geler aucun package.

Un gel n'est jamais annulé automatiquement par le système d'empaquetage. 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 gel et afficher la liste des paquets qui seraient libérés sans réellement libérer aucun package.

## Marquage des packages à éviter

Utilisez la commande `pkg avoid` pour éviter les packages spécifiés s'ils sont la cible d'une dépendance `group`.

```
/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 qui se trouvent sur la liste `avoid` sont installés si nécessaire pour satisfaire une dépendance obligatoire. 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.

## 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 de l'image et ne peuvent pas être définies sur des packages individuels. Pour afficher les valeurs actuelles des facettes et des variantes définies sur l'image, utilisez les commandes `pkg facet` et `pkg variant`. Pour modifier les valeurs des facettes et des variantes définies sur l'image, utilisez les commandes `pkg change-facet` et `pkg change-variant`. Reportez-vous à la page de manuel [pkg\(1\)](#) et aux exemples ci-dessous.

Les facettes et les variantes sont spécifiées sous forme de repères sur les actions de package. Chaque repère de facette et de variante possède un nom et une valeur. Une seule action peut avoir plusieurs repères de facette et de variante. Un fichier d'en-tête spécifique à une architecture utilisé par des développeurs ou un composant uniquement destiné à une zone globale SPARC sont des exemples de composants à repères de facettes et de variantes multiples.

`variant.arch=sparc` est un exemple de repère de variante. `facet.devel=true` est un exemple de repère de facette. 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. Un repère de facette sur une action doit uniquement avoir la valeur `true` ; les autres valeurs ont des comportements indéterminés. Une facette définie sur l'image peut être une facette complète telle que `doc.man` ou un modèle tel que `locale.*`. L'intérêt de cette caractéristique est qu'elle vous permet de désactiver une partie de l'espace de noms d'une facette et de n'activer que des facettes individuelles en son sein. Par exemple, vous pouvez désactiver tous les environnements linguistiques, puis n'activer qu'un ou deux environnements linguistiques particuliers, comme illustré dans l'exemple suivant :

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

La plupart des variantes peuvent avoir un nombre de valeurs quelconque. Par exemple, la variante `arch` peut être définie sur `i386`, `sparc`, `ppc`, `arm` ou n'importe quelle architecture prise en charge par la distribution. (Seuls `i386` et `sparc` sont utilisés dans Oracle Solaris.) Les variantes `debug` constituent une exception. Les variantes `debug` peuvent uniquement être définies sur `true` ou `false` ; les autres valeurs ont un comportement indéterminé. S'il existe une version non-débogage et une version débogage d'un même fichier, la variante `debug` applicable doit être explicitement définie sur ces deux fichiers, 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. Par défaut, les variantes `debug.*` sont définies sur `false` dans l'image.

Les facettes et les variantes définies sur l'image déterminent si une action particulière est installée ou non.

- Les actions sans repère de facette ou de variante sont toujours installées.
- Les actions avec repères de facettes sont installées, sauf si toutes les facettes ou tous les modèles de facettes correspondant aux repères 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 repères de variantes sont uniquement installées si les valeurs des repères de variantes sont les mêmes que celles définies dans l'image.
- Les actions comportant à la fois des repères de facettes et des repères de variantes sont installées si les facettes et les variantes autorisent l'installation des actions concernées.

Vous pouvez créer vos propres repères de facettes et de variantes. Les repères suivants sont couramment utilisés 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 repères de facettes utilisés dans Oracle Solaris :

<code>facet.devel</code>	<code>facet.doc</code>
<code>facet.doc.html</code>	<code>facet.doc.info</code>
<code>facet.doc.man</code>	<code>facet.doc.pdf</code>
<code>facet.locale.de</code>	<code>facet.locale.en_GB</code>
<code>facet.locale.en_US</code>	<code>facet.locale.fr</code>
<code>facet.locale.ja_JP</code>	<code>facet.locale.zh_CN</code>

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 d'une variante

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

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

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

## Modification d'une variante

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

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

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

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

```
# pkg change-variant -nv --accept variant.debug.*=true
      Packages to update:      831
      Variants/Facets to change: 4
      Estimated space available: 112.19 GB
      Estimated space to be consumed: 220.76 MB
      Create boot environment: No
      Create backup boot environment: Yes
      Rebuild boot archive: No
Changed variants/facets:
  variant variant.debug.*: true
    facet facet.locale.en_US: None
    facet facet.locale.en: None
    facet facet.locale.*: None
Changed packages:
solaris
...
```

## Affichage d'une facette

Utilisez la commande `pkg facet` pour afficher les valeurs des facettes qui sont définies.

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

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

## Modification d'une facette

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

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

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

```
# 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
  ...
```

## Mise à jour d'une image

Utilisez la commande `pkg update` sans *pkg-fmri* spécifié ou avec un astérisque (\*) à la place de *pkg-fmri* pour mettre à jour tous les packages installés qui disposent de mises à jour disponibles pour la version la plus récente autorisée suivant les contraintes imposées sur le système par les packages installés et la configuration d'éditeur. Si les zones non globales sont configurées dans l'image actuelle, ces zones sont également mises à jour. Reportez-vous à la section [“A propos des packages et des zones”](#) du manuel *Administration Oracle Solaris : Oracle Solaris Zones, Oracle Solaris 10 Zones et gestion des ressources*.

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

Utilisez la commande `pkg list -u` pour afficher la liste des packages pour lesquels des mises à jour sont disponibles. L'opération `pkg update` entraîne souvent la création automatique d'un nouvel environnement d'initialisation. Utilisez `pkg update -nv` pour passer en revue les modifications qui seraient apportées à l'image actuelle ou dans un nouvel environnement d'initialisation. 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.

---

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

---

Par défaut, chaque package est mis à jour depuis l'éditeur qui a fourni la version actuellement installée. Vous pouvez contrôler l'éditeur qui fournit les packages en spécifiant la persistance de l'éditeur et l'ordre de recherche. Reportez-vous à la section [“Ajout, modification ou suppression des éditeurs de packages”](#) à la page 46.

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. Après l'opération `update`, tout package fourni par les éditeurs n'ayant pas été trouvée dans l'image est ajouté à la configuration de l'image sans point d'origine.

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 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 installés. Lorsque vous initialisez votre système, vous avez la possibilité de démarrer dans n'importe quel EI 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 abordé 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](#).

Reportez-vous à la section “[Définitions des propriétés d'image](#)” à la page 60 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 :

<code>default</code>	Applique la stratégie de création d'IE par défaut : <code>create-backup</code> .
<code>always-new</code>	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>	<p>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.</p>
<code>trust-anchor-directory</code>	<p>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.</p> <p>La valeur par défaut est ignore.</p>

## Propriétés de l'éditeur pour les packages signés

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

### `signature-policy`

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

### `signature-required-names`

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

## Configuration des propriétés de signature de package

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

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

L'exemple suivant configure cette image pour exiger que tous les packages soient signés. Il nécessite également que la chaîne `oracle.com` soit considérée comme un nom commun pour l'un des certificats dans la chaîne de confiance.

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

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

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

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

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

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

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

```
# 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.

```
# 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`.

#### `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 obtenir des 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
display-copyrights                       True
flush-content-cache-on-success            False
mirror-discovery                          False
preferred-authority                       solaris
publisher-search-order                   ['solaris', 'opensolaris.org', 'extra']
pursue-latest                             True
send-uuid                                 True
signature-policy                         verify
signature-required-names                  []
trust-anchor-directory                    etc/certs/CA
use-system-repo                           False

```

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 46.

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

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

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

```

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

```

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

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

```

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

```

## Création d'une image

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

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_spec=instance ...]
  [-g path_or_uri | --origin path_or_uri ...]
  [-m uri | --mirror uri ...]
  [--facet facet_spec=(True|False) ...]
  [(-p | --publisher) [name=]repo_uri] dir
```

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 [-Hl] [-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
2011-09-07T12:15:52  update                  pkg                  Succeeded
2011-09-26T18:53:12  refresh-publishers      pkg                  Succeeded
2011-09-26T18:53:50  rebuild-image-catalogs  pkg                  Succeeded
2011-09-27T09:05:34  update                  pkg                  Succeeded
```

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
2011-09-07T12:15:52  0:13:56            update                  Succeeded
2011-09-26T18:53:12  0:01:22            refresh-publishers      Succeeded
2011-09-26T18:53:50  0:00:44            rebuild-image-catalogs  Succeeded
2011-09-27T09:05:34  0:20:08            update                  Succeeded
```

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

**# pkg purge-history**