

Páginas del comando man de Image Packaging System

Copyright © 2007, 2012, Oracle y/o sus filiales. Todos los derechos reservados.

Este software y la documentación relacionada están sujetos a un contrato de licencia que incluye restricciones de uso y revelación, y se encuentran protegidos por la legislación sobre la propiedad intelectual. A menos que figure explícitamente en el contrato de licencia o esté permitido por la ley, no se podrá utilizar, copiar, reproducir, traducir, emitir, modificar, conceder licencias, transmitir, distribuir, exhibir, representar, publicar ni mostrar ninguna parte, de ninguna forma, por ningún medio. Queda prohibida la ingeniería inversa, desensamblaje o descompilación de este software, excepto en la medida en que sean necesarios para conseguir interoperabilidad según lo especificado por la legislación aplicable.

La información contenida en este documento puede someterse a modificaciones sin previo aviso y no se garantiza que se encuentre exenta de errores. Si detecta algún error, le agradeceremos que nos lo comuniqué por escrito.

Si este software o la documentación relacionada se entrega al Gobierno de EE.UU. o a cualquier entidad que adquiera licencias en nombre del Gobierno de EE.UU. se aplicará la siguiente disposición:

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.

Este software o hardware se ha desarrollado para uso general en diversas aplicaciones de gestión de la información. No se ha diseñado ni está destinado para utilizarse en aplicaciones de riesgo inherente, incluidas las aplicaciones que pueden causar daños personales. Si utiliza este software o hardware en aplicaciones de riesgo, usted será responsable de tomar todas las medidas apropiadas de prevención de fallos, copia de seguridad, redundancia o de cualquier otro tipo para garantizar la seguridad en el uso de este software o hardware. Oracle Corporation y sus subsidiarias declinan toda responsabilidad derivada de los daños causados por el uso de este software o hardware en aplicaciones de riesgo.

Oracle y Java son marcas comerciales registradas de Oracle y/o sus subsidiarias. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

Intel e Intel Xeon son marcas comerciales o marcas comerciales registradas de Intel Corporation. Todas las marcas comerciales de SPARC se utilizan con licencia y son marcas comerciales o marcas comerciales registradas de SPARC International, Inc. AMD, Opteron, el logotipo de AMD y el logotipo de AMD Opteron son marcas comerciales o marcas comerciales registradas de Advanced Micro Devices. UNIX es una marca comercial registrada de The Open Group.

Este software o hardware y la documentación pueden ofrecer acceso a contenidos, productos o servicios de terceros o información sobre los mismos. Ni Oracle Corporation ni sus subsidiarias serán responsables de ofrecer cualquier tipo de garantía sobre el contenido, los productos o los servicios de terceros y renuncian explícitamente a ello. Oracle Corporation y sus subsidiarias no se harán responsables de las pérdidas, los costos o los daños en los que se incurra como consecuencia del acceso o el uso de contenidos, productos o servicios de terceros.

Contenido

Prefacio	5
Comandos de usuario	9
packagemanager(1)	10
pkg(1)	13
pkgdepend(1)	44
pkgdiff(1)	49
pkgfmt(1)	51
pkglint(1)	52
pkgmerge(1)	57
pkgmogrify(1)	62
pkgrecv(1)	69
pkgrepo(1)	73
pkgsend(1)	81
pkgsign(1)	85
pm-updatemanager(1)	87
Comandos de administración del sistema	89
pkg.depotd(1m)	90
pkg.sysrepo(1m)	98
Estándares, entornos y macros	101
pkg(5)	102

Prefacio

En este documento, se proporcionan las páginas del comando man para las herramientas de instalación del sistema Oracle Solaris 11.

Visión general

A continuación, se incluye una breve descripción de cada sección de la página del comando man y la información a la que hace referencia:

- En la sección 1 se describen los comandos disponibles con el sistema operativo.
- En la sección 1M se describen los comandos que se utilizan principalmente para el mantenimiento y la administración del sistema.
- En la sección 5 se incluye otra documentación, como tablas de conjuntos de caracteres.

A continuación, se muestra un formato genérico de las páginas del comando man. Las páginas del comando man de cada sección del manual generalmente siguen este orden, pero sólo incluyen los encabezados necesarios. Por ejemplo, si no hay ningún error para informar, no se incluye la sección de errores. Consulte el comando man para obtener más información sobre las páginas del comando man en general.

NAME	En esta sección se proporcionan los nombres de los comandos o las funciones documentados, y una breve descripción de lo que hacen.
SYNOPSIS	<p>En esta sección se muestra la sintaxis de los comandos o las funciones. Cuando un comando o un archivo no existe en la ruta estándar, se muestra el nombre de la ruta completa. Las opciones y los argumentos se ordenan alfabéticamente, en primer lugar, los argumentos de una sola letra y, luego, las opciones con argumentos, a menos que se requiera un orden de argumentos diferente.</p> <p>En esta sección se utilizan los siguientes caracteres especiales:</p>

	[]	Corchetes. Las opciones o los argumento entre corchetes son opcionales. Si se omiten los corchetes, se debe especificar el argumento.
	...	Puntos suspensivos. Se pueden proporcionar diversos valores para el argumento anterior, o el argumento anterior se puede especificar varias veces, por ejemplo, <i>filename</i> ...
		Separador. Sólo uno de los argumentos separados por este carácter se puede especificar a la vez.
	{ }	Llaves. Las opciones y/o argumentos entre llaves son interdependientes, de manera que todo lo que está dentro se debe tratar como una unidad.
PROTOCOLO		Esta sección aparece sólo en la subsección 3R para indicar el archivo de descripción del protocolo.
DESCRIPTION		En esta sección se define la funcionalidad y el comportamiento del servicio. De esta manera, se describe de manera concisa lo que hace el comando. No se analizan las OPCIONES ni se citan EJEMPLOS. En USO se describen las funciones, las macros, las solicitudes, los subcomandos y los comandos interactivos.
OPTIONS		En esta sección se muestran las opciones de comandos y un resumen conciso de lo que hace cada opción. Las opciones se muestran literalmente y en el orden en que aparecen en la sección SINOPSIS. Los argumentos posibles para las opciones se analizan en la opción y, cuando corresponde, se proporcionan valores predeterminados.
OPERANDS		En esta sección se muestran los operandos del comando y se describe cómo éstos afectan las acciones del comando.
OUTPUT		En esta sección se describe la salida (salida estándar, error estándar o archivos de salida) generada por el comando.
VALORES DE RETORNO		Si la página del comando man documenta funciones que devuelven valores, en esta sección se muestran estos valores y se describen las condiciones en las que se devolverán. Si una función puede devolver sólo valores constantes, como 0 o -1, estos valores se muestran en párrafos etiquetados. De lo contrario, los valores de devolución de cada función se describen en un párrafo

	único. Las funciones declaradas nulas no devuelven valores, por lo que no se tratan en VALORES DE DEVOLUCIÓN.
ERRORES	En caso de fallo, la mayoría de las funciones colocan un código de error en la variable global <code>errno</code> que indica el motivo del fallo. En esta sección se muestran alfabéticamente todos los códigos de error que puede generar una función y se describen las condiciones que causan cada error. Cuando más de una condición puede causar el mismo error, cada condición se describe en un párrafo aparte en el código de error.
USAGE	En esta sección se muestran los comandos, las funciones y las reglas especiales que requieren explicaciones detalladas. Las subsecciones que se muestran aquí se utilizan para explicar las funcionalidades integradas: <ul style="list-style-type: none"> Comandos Modificadores Variables Expresiones Gramática de entrada
EXAMPLES	En esta sección se proporcionan ejemplos de uso o de cómo utilizar un comando o una función. Cuando es posible se muestra un ejemplo completo que incluye la entrada de la línea de comandos y la respuesta de la máquina. Cuando se proporciona un ejemplo, se muestra el indicador, como <code>example%</code> , o, si el usuario debe ser superusuario, <code>example#</code> . Los ejemplos incluyen explicaciones, reglas de sustitución de variables o valores devueltos. La mayoría de los ejemplos ilustran conceptos relacionados con las secciones SINOPSIS, DESCRIPCIÓN, OPCIONES y USO.
ENVIRONMENT VARIABLES	En esta sección se muestran todas las variables de entorno a las que afecta el comando o la función, y una breve descripción del efecto.
EXIT STATUS	En esta sección se muestran los valores que devuelve el comando al programa o el shell que realiza la llamada, y las condiciones que generan que se devuelvan estos valores. Normalmente, se devuelve cero cuando se finaliza

	correctamente y valores distintos de cero cuando hay diversas condiciones de error.
FILES	En esta sección se muestran todos los nombres de archivo a los que hace referencia la página del comando <code>man</code> , los archivos de interés y los archivos creados o requeridos por comandos. Cada uno está seguido por un resumen descriptivo o una explicación.
ATTRIBUTES	En esta sección se muestran las características de los comandos, las utilidades y los controladores de dispositivos mediante la definición del tipo de atributo y el valor correspondiente. Para obtener más información, consulte la página del comando <code>man attributes(5)</code> .
SEE ALSO	En esta sección se muestran las referencias a otras páginas del comando <code>man</code> , documentación interna y publicaciones externas.
DIAGNÓSTICO	En esta sección se muestran los mensajes de diagnóstico y una breve explicación de la condición que causó el error.
ADVERTENCIAS	En esta sección se muestran las advertencias sobre condiciones especiales que podrían afectar seriamente las condiciones de trabajo. Esto no es una lista de diagnósticos.
NOTES	En esta sección se muestra información adicional que no pertenece a ninguna otra sección de la página. Adopta la forma de un apartado para el usuario, y abarca puntos de especial interés. Aquí nunca se incluye información crítica.
ERRORES	En esta sección se describen errores conocidos y, cuando es posible, se sugieren soluciones.

REFERENCIA

Comandos de usuario

Nombre	packagemanager – interfaz gráfica de usuario para Image Packaging System
Sinopsis	<pre>/usr/bin/packagemanager [opciones] /usr/bin/packagemanager [-hiRu] [--help] [--info-install <i>archivo</i>] [--update-all] [--image-dir <i>directorio</i>] /usr/bin/packagemanager [<i>archivo</i>]</pre>
Descripción	<p>packagemanager es la interfaz gráfica de usuario para pkg(5), el software Image Packaging System.</p> <p>Package Manager permite efectuar las tareas siguientes:</p> <ul style="list-style-type: none">■ Buscar, instalar y eliminar paquetes.■ Agregar, eliminar y modificar editores.■ Crear, eliminar y gestionar entornos de inicio. <p>Si se especifica el operando <i>archivo</i> y su sufijo es .p5i, packagemanager se inicia en modo de instalación web, que agrega uno o varios publicadores y un número de paquetes para cada publicador.</p>
Opciones	<p>Se admiten las siguientes opciones:</p> <p>-h o --help Muestra un mensaje de uso.</p> <p>-i o --info-install <i>archivo</i> Permite especificar un archivo .p5i para ejecutar packagemanager en modo de instalación web. El <i>archivo</i> debe tener el sufijo .p5i.</p> <p>-R o --image-dir <i>directorio</i> Realiza operaciones en la imagen que tiene raíz en el <i>directorio</i>, en lugar de la imagen detectada automáticamente.</p> <p>-U o --update-all Actualiza todos los paquetes instalados que tienen actualizaciones disponibles.</p> <p>Nota – Si los paquetes package/pkg, package/pkg/package-manager o package/pkg/update-manager deben ser actualizados, packagemanager, primero, actualiza estos paquetes y, a continuación, reinicia para llevar a cabo las demás actualizaciones.</p>
Operandos	<p><i>archivo</i> Un archivo de instalación web. Este archivo debe tener el sufijo .p5i. Consulte la ayuda en pantalla de Package Manager, para obtener más información sobre la instalación web.</p>

Ejemplos EJEMPLO 1 Realización de operaciones en la imagen actual

Invoque packagemanager en la imagen actual.

```
$ packagemanager
```

EJEMPLO 2 Realización de operaciones en una imagen especificada

Invoque packagemanager en la imagen almacenada en /aux0/example_root.

```
$ packagemanager -R /aux0/example_root
```

EJEMPLO 3 Invocación en el modo de instalación web

Invoque packagemanager en el modo de instalación web.

```
$ packagemanager ~/test.p5i
```

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 Ejecución correcta.
- 1 Se produjo un error.
- 2 Se especificaron opciones de línea de comandos no válidas.

Archivos Dado que las imágenes pkg(5) se pueden ubicar arbitrariamente en un sistema de archivos más grande, se utiliza el token \$IMAGE_ROOT para distinguir las rutas relacionadas. Para una instalación de sistema típica, \$IMAGE_ROOT es equivalente a /.

\$IMAGE_ROOT/var/pkg Directorio de metadatos para una imagen completa o parcial.

\$IMAGE_ROOT/.org.opensolaris, pkg Directorio de metadatos para una imagen de usuario.

En los metadatos de una imagen en particular, ciertos archivos y directorios contienen información útil durante la reparación y la recuperación. El token \$IMAGE_META se utiliza para hacer referencia al directorio de nivel superior que contiene los metadatos. \$IMAGE_META generalmente es una de las dos rutas especificadas anteriormente.

\$IMAGE_META/gui-cache Ubicación para los metadatos almacenados en la antememoria mantenidos by packagemanager para acelerar el inicio del programa y cambiar de editor.

Otras rutas dentro de la jerarquía de directorios de \$IMAGE_META son privadas y están sujetas a cambios.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg/package-manager
Estabilidad de interfaz	Sin asignar.

Véase también [pm-updatemanager\(1\)](#), [pkg\(1\)](#), [pkg\(5\)](#)

Ayuda en pantalla de Package Manager

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

Notas `packagemanager` se debe invocar con privilegios suficientes para utilizar los archivos y directorios de una imagen.

Nombre pkg – cliente de recuperación de Image Packaging System

Sinopsis /usr/bin/pkg [*options*] *command* [*cmd_options*] [*operands*]

```

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

/usr/bin/pkg install [-nvq] [-g path_or_uri ...] [--accept]
    [--licenses] [--no-be-activate] [--no-index] [--no-refresh]
    [--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 ...

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

/usr/bin/pkg update [-fnvq] [-g path_or_uri ...] [--accept]
    [--licenses] [--no-be-activate] [--no-index] [--no-refresh]
    [--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 ...]

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

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

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

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

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

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

/usr/bin/pkg 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 ...)

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

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

/usr/bin/pkg unset-mediator [-nvIV] [--no-be-activate]
    [--no-backup-be | --require-backup-be] [--backup-be-name name]

```

```

    [--deny-new-be | --require-new-be] [--be-name name]
    mediator ...

/usr/bin/pkg variant [-H] [variant.variant_name ...]

/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_name=value ...

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

/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_name=[True|False|None] ...

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

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

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

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

/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 ...

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

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

```

```

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

/usr/bin/pkg unset-publisher publisher ...

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

/usr/bin/pkg purge-history

/usr/bin/pkg rebuild-index

/usr/bin/pkg update-format

/usr/bin/pkg version

/usr/bin/pkg help

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

```

Descripción pkg es el cliente de recuperación de Image Packaging System. Con una configuración válida, pkg se puede invocar para crear ubicaciones para los paquetes que se instalarán, llamar imágenes e instalar los paquetes en esas imágenes. Los paquetes son publicados por los editores, que pueden colocar sus paquetes en uno o varios depósitos, o en archivos de paquetes para que estén disponibles. pkg recupera los paquetes desde archivos de paquetes o depósitos del editor, e instala los paquetes en una imagen.

Un nombre de editor identifica a una persona, un grupo de personas o una organización como el origen de uno o varios paquetes. Para evitar conflictos entre nombres de editores y ayudar a identificar al editor, se aconseja utilizar como nombre de editor un nombre de dominio que represente a la entidad que publica los paquetes.

Un depósito es una ubicación donde los clientes pueden publicar y recuperar el contenido de los paquetes (archivos incluidos en el paquete, como programas y documentos) y los metadatos (información sobre el paquete, como el nombre y la descripción). Por ejemplo, un editor llamado `example.org` podría tener su depósito ubicado en el URI `http://example.org/repository`.

`pkg` también puede desinstalar paquetes, refrescar metadatos del editor (como la lista de paquetes disponibles), validar la instalación de paquetes en una imagen y consultar distintos tokens en una imagen. Estas consultas también se puede crear a partir de depósitos `pkg(5)`.

Las imágenes pueden ser de tres tipos: imágenes completas, que pueden proporcionar un sistema completo; imágenes parciales, que están enlazadas a una imagen completa (imagen principal), pero no pueden proporcionar un sistema completo por sí solas; e imágenes de usuario.

Opciones Se admiten las siguientes opciones:

`-R directorio`

Realiza operaciones en la imagen con raíz en *directorio*. Si no se especificó ningún directorio, o si no se determinó ninguno en función del entorno, el valor predeterminado es `/`. Para obtener más información, consulte la sección "Variables del entorno".

`--help o -?`

Muestra un mensaje de uso.

Subcomandos Se admiten los subcomandos siguientes:

`refresh [--full] [editor ...]`

Actualiza la lista de paquetes disponibles del cliente y los metadatos del editor para cada editor especificado. Si no se especifica ningún editor, la operación se realiza para todos los editores.

Con `--full`, se puede forzar una recuperación completa de todos los metadatos del editor, en lugar de intentar una actualización incremental, y solicitar que los proxies utilizados durante la operación ignoren los datos almacenados en la antememoria. Esta opción existe para solucionar problemas y no se debe utilizar con regularidad.

`install [-nvq] [-g ruta_o_URI ...] [--accept] [--licenses] [--no-be-activate] [--no-index] [--no-refresh] [--no-backup-be | --require-backup-be] [--backup-be-name nombre] [--deny-new-be | --require-new-be] [--be-name nombre] [--reject mod_ID rec ges err_paq ...] mod_ID rec ges err_paq ...`

Instala los paquetes y los actualiza con la versión más reciente que coincide con el valor `mod_ID rec ges err_paq` permitido por los paquetes instalados en la imagen. Para solicitar de forma explícita la versión más reciente de un paquete, use `latest` para la parte de la versión del *patrón_fmri_paquete*. Por ejemplo, especifique `vim@latest`.

Es posible que durante el proceso de instalación se reemplacen algunos archivos de configuración o se les cambie el nombre. Para obtener más información sobre cómo el

sistema de paquetes determina los archivos que se van a preservar y cómo se preservan dichos archivos durante las operaciones de paquetes, consulte "Acciones de archivo" en la página del comando `man pkg(5)`.

Si un paquete está en la lista de paquetes que se deben evitar, una vez que se instala, se elimina de esa lista.

Con `-g`, puede agregar temporalmente el archivo o depósito de paquetes especificado a la lista de orígenes de la imagen desde donde se recuperarán los datos del paquete. Si los paquetes de los orígenes especificados también están disponibles en los editores configurados en la imagen, el cliente recuperará el contenido de esos paquetes sólo desde los orígenes especificados. Al decidir qué versión de un paquete se utilizará, los editores configurados en la imagen, pero no encontrados en los orígenes determinados, tienen prioridad. Después de la instalación o actualización, los paquetes proporcionados por los editores no se hayan encontrado en la imagen se agregarán a la configuración de la imagen sin un origen. Esta opción se puede especificar varias veces.

Con `-n`, puede llevar a cabo una ejecución de prueba de la operación sin realizar cambios en los paquetes.

Con `-q`, puede ocultar los mensajes de progreso durante la operación solicitada.

Con `-v`, puede enviar mensajes de progreso detallados durante la operación solicitada y mostrar información detallada de planificación (como el cambio de facetas, mediadores y variantes). Esta opción se puede especificar varias veces para aumentar la cantidad de información de planificación que se muestra.

Con `--accept`, puede indicar que está de acuerdo con las condiciones de las licencias de los paquetes que está actualizando o instalando y que las acepta. Si no proporciona esta opción, y alguna licencia de paquete requiere aceptación, la operación de instalación fallará.

Con `--licenses`, puede mostrar todas las licencias para los paquetes que se instalan o actualizan como parte de esta operación.

Con `--no-backup-be`, no puede crear un entorno de inicio de copia de seguridad.

Con `--no-be-activate`, si se crea un entorno de inicio, no lo establezca como el entorno de inicio activo en el inicio siguiente. Consulte `beadm` para obtener más información.

Con `--no-index`, no puede actualizar los índices de búsqueda una vez que la operación se ha completado correctamente.

Con `--no-refresh`, no intente contactar con los depósitos de los editores de la imagen para recuperar la lista más reciente de paquetes disponibles y otros metadatos.

Con `--backup-be-name`, puede nombrar el entorno de inicio de copia de seguridad creado utilizando el argumento proporcionado. El uso de `--backup-be-name` implica el uso de `--require-backup-be`. Consulte también `beadm(1M)`.

Con `--be-name`, puede cambiar el nombre del entorno de inicio recién creado para que sea el argumento especificado. El uso de `--be-name` implica `--require-new-be`. Consulte también `beadm(1M)`.

Con `--require-backup-be`, siempre cree un entorno de inicio de copia de seguridad si no creará un nuevo entorno de inicio. Sin esta opción, se crea un entorno de inicio de copia de seguridad basado en la política de imagen. Consulte `be-policy` en "Propiedades de la imagen" a continuación, para obtener una explicación de cuándo los entornos de inicio de copia de seguridad se crean automáticamente.

Con `--require-new-be`, siempre cree un entorno de inicio nuevo. Sin esta opción, se crea un entorno de inicio basado en la política de imagen. Consulte `be-policy` en "Propiedades de la imagen" a continuación, para obtener una explicación de cuándo los entornos de inicio se crean automáticamente. Esta opción no se puede combinar con `--require-backup-be`.

Con `--deny-new-be`, no puede crear un entorno de inicio nuevo. Esta operación no se realiza si se requiere un entorno de inicio nuevo.

Con `--reject`, puede impedir que se instalen los paquetes con nombres que coincidan con el modelo determinado. Si los paquetes coincidentes ya están instalados, se eliminan como parte de esta operación. Los paquetes rechazados que son el destino de las dependencias de grupo se colocan en la lista para evitar.

```
uninstall [-nvq] [--no-be-activate ] [--no-index] [--no-backup-be |
--require-backup-be] [--backup-be-name nombre] [--deny-new-be |
--require-new-be] [--be-name nombre] mod_ID rec ges err_paq ...
```

Elimina los paquetes instalados que coinciden con el *mod_ID rec ges err_paq*.

Si un paquete es el tema de una dependencia de grupo, su desinstalación lo coloca en la lista para evitar. Consulte el subcomando `avoid` a continuación.

Para todas las demás opciones, consulte el comando `install` anterior para ver el uso y sus efectos.

```
update [-fnvq] [-g ruta_o_URI ...] [--accept] [--licenses] [--no-be-activate]
[--no-index] [--no-refresh ] [--no-backup-be | --require-backup-be]
[--backup-be-name nombre] [--deny-new-be | --require-new-be] [--be-name
nombre] [--reject mod_ID rec ges err_paq ...] [mod_ID rec ges err_paq ...]
```

Sin argumentos, o si asterisco (*) es uno de los modelos proporcionados, actualice todos los paquetes instalados en la imagen actual a la versión más reciente permitida por las restricciones impuestas al sistema por los paquetes instalados y la configuración del editor. Para solicitar de manera explícita la versión más reciente de un paquete, use `latest` para la parte de la versión de *mod_ID rec ges err_paq*. Por ejemplo, especifique `vim@latest`.

Si se proporciona *mod_ID rec ges err_paq*, `update` sustituye los paquetes que están instalados, y que coinciden con *mod_ID rec ges err_paq*, con la versión más reciente permitida por los modelos y las restricciones impuestas al sistema por los paquetes

instalados y la configuración del editor. Se pueden especificar versiones más antiguas o más recientes que las que ya están instaladas para realizar cambios de versión de paquetes específicos en el lugar. No se admite la actualización de paquetes específicos con nombres cambiados o límites obsoletos.

A los archivos de configuración preservados que forman parte de los paquetes que se van a actualizar a una versión inferior mediante `update` y que han sido modificados desde la instalación de la versión original se les cambia el nombre mediante la adición de la extensión `.update`. Para obtener más información sobre cómo el sistema de paquetes determina los archivos que se van a preservar y cómo se preservan dichos archivos durante las actualizaciones de paquetes, consulte "Acciones de archivo" en la página del comando `man pkg(5)`.

Con la opción `-f` no ejecute la comprobación de actualizaciones de cliente al actualizar todos los paquetes instalados.

Para todas las demás opciones, consulte el comando `install` anterior para ver el uso y sus efectos.

`list [-Hafnsuv] [-g ruta_o_URI ...] [--no-refresh] [mod_ID rec ges err_paq ...]`

Sin argumentos, visualice una lista de los paquetes de la imagen actual, así como información sobre la versión y el estado de instalación. Con argumentos, visualice información sobre los paquetes especificados. De manera predeterminada, las variantes de paquete para un tipo de zona o una arquitectura diferente se excluyen. La salida habitual es en tres columnas:

NAME (PUBLISHER)	VERSION	IFO
system/core-os	0.5.11-0.169	i--
x11/wm/fvwm (fvwm.org)	2.6.1-3	i--

La primera columna contiene el nombre del paquete. Si el editor desde el que se instala el paquete (o donde está disponible el paquete, si aún no está instalado) no es el primero en el orden de búsqueda de editor, el nombre del editor aparece entre paréntesis a continuación del nombre del paquete. La segunda columna contiene la versión de lanzamiento y ramificación del paquete. Consulte la página del comando `man pkg(5)` para obtener información sobre las versiones de lanzamiento y ramificación, y sobre las variantes.

La última columna presenta un conjunto de indicadores que muestran el estado del paquete:

- Una `i` en la columna `I` indica que el paquete está instalado.
- Una `f` en la columna `F` indica que el paquete está congelado.
- Una `o` en la columna `O` indica que el paquete es obsoleto. Una `r` en la columna `O` indica que se ha cambiado el nombre del paquete (una forma de ser obsoleto).

Con `-H`, puede omitir los encabezados de la lista.

Con `-a`, puede mostrar los paquetes instalados y la versión más reciente de los paquetes que están disponibles para la instalación. Se considera que los paquetes están disponibles para la instalación si están permitidos por las incorporaciones instaladas y por las variantes de la imagen. Si se especifica uno o varios modelos, se muestra la versión más reciente que coincide con el modelo especificado y permitido por cualquiera de las incorporaciones instaladas y las variantes de la imagen. Sin `-a`, se muestran sólo los paquetes instalados.

Con `-f` y `-a`, se muestran todas las versiones de todos los paquetes para todas las variantes independientemente de las restricciones de incorporación o del estado de instalación. Para mostrar explícitamente la última versión de un paquete al utilizar estas opciones, utilice `latest` para la parte de la versión de *mod_ID rec ges err_paq*. Por ejemplo, especifique `vim@latest`.

Con `-g`, puede utilizar el archivo o el depósito de paquetes especificado como el origen de los datos del paquete para la operación. Esta opción se puede especificar varias veces. El uso de `-g` implica el uso de `-a` si `-n` no se ha especificado.

Con `-n`, puede mostrar las versiones más recientes de todos los paquetes conocidos, independientemente del estado de instalación.

Con `-s`, puede mostrar una forma abreviada de una línea con el nombre del paquete y el resumen. Esta opción se puede utilizar con `-a`, `-n`, `-u` o `-v`.

Con `-u`, puede mostrar sólo los paquetes con versiones más recientes disponibles. Esta opción no se puede utilizar con `-g`.

Con `-v`, puede mostrar identificadores de recursos de gestión de errores del paquete completo, incluidos el editor y la versión completa, todo en la primera columna (la columna de versión desaparece). Esta opción se puede utilizar con `-a`, `-n` o `-u`.

Con `--no-refresh`, no intente contactar con los depósitos de los editores de la imagen para recuperar la lista más reciente de paquetes disponibles.

`info [-lr] [-g ruta_o_uri ...] [--license] [mod_ID rec ges err_paq ...]`

Puede mostrar información sobre los paquetes en un formato legible. Se pueden especificar varios modelos de identificador de recurso de gestión de errores. Sin modelos, puede mostrar información sobre todos los paquetes instalados en la imagen.

Con `-g`, puede utilizar el archivo o el depósito de paquetes especificado como el origen de los datos del paquete para la operación. Esta opción se puede especificar varias veces. El uso de `-g` implica el uso de `-r`.

Con `-l`, sólo se muestra la información de los paquetes instalados. Ésta es la opción predeterminada.

Con `-r`, puede confrontar los paquetes según las versiones más recientes disponibles, y recuperar información para paquetes no instalados actualmente (si es necesario) de los

depósitos de los editores configurados de la imagen. Si se utiliza esta opción, se debe especificar al menos un paquete. Sin `-r`, sólo se muestran los paquetes instalados de manera predeterminada.

Con `--license`, puede mostrar los textos de las licencias de los paquetes. Esta opción se puede combinar con `-l` o `-r`.

```
contents [-Hmr] [-a atributo=modelo ...] [-g ruta_o_URI ...] [-o atributo,...]
[-s criterio_orden] [-t tipo_acción ...] [mod_ID rec ges err_paq ...]
```

Visualice el contenido (atributos de acción) de los paquetes. Sin operandos u opciones, visualice el valor del atributo `path` para las acciones instaladas en la imagen actual, ordenadas alfabéticamente por el valor del atributo. Para obtener información sobre las acciones y sus atributos, consulte "Acciones" en la página del comando `man pkg(5)`. Consulte también la lista de nombres de pseudoatributos que aparece a continuación.

Con `-a`, puede limitar la salida a las acciones que tengan un atributo que aparezca en el argumento de opción con un valor que coincida con el modelo (`glob`) en el argumento de opción (a continuación del nombre de atributo con un signo igual). Si se proporcionan varias opciones `-a`, se muestran las acciones que coincidan con cualquiera de ellas.

Con `-g`, visualice información de paquetes que se pueden instalar en esta imagen desde el archivo o el repositorio de paquetes especificado. Los paquetes que se pueden instalar incluyen paquetes que están instalados actualmente y otros paquetes que cumplen los criterios para la instalación en esta imagen, como restricciones de facetas y variantes. Esta opción se puede especificar varias veces. El uso de `-g` implica el uso de `-r`.

Con `-m`, visualice todos los atributos de todas las acciones de los paquetes especificados, incluidas las acciones que no se pueden instalar en esta imagen.

Con `-o`, visualice los atributos enumerados, ordenados en función de los valores del primer atributo de la lista. La opción `-o` se puede especificar varias veces, o se pueden especificar varios atributos como argumento para una opción `-o` separando los nombres de atributo con comas. Sólo se muestran las acciones que tienen los atributos solicitados.

Con `-r`, visualice información de las versiones disponibles más nuevas de los paquetes que se pueden instalar en esta imagen desde los repositorios de los editores configurados en esta imagen. Los paquetes que se pueden instalar incluyen paquetes que están instalados actualmente y otros paquetes que cumplen los criterios para la instalación en esta imagen, como restricciones de facetas y variantes. Si se utiliza esta opción, se debe especificar al menos un paquete.

Con `-s`, puede ordenar las acciones según el atributo de acción especificado. Si no se especifica, de manera predeterminada se ordena por `ruta` o por el primer atributo especificado por la opción `-o`. La opción `-s` se puede especificar varias veces.

Con `-t`, sólo se muestran las acciones del tipo especificado. Se pueden especificar varios tipos en una lista separada por comas. Esta opción se puede especificar varias veces.

Con `-H`, puede omitir los encabezados de la lista.

Con `pkg_fmri_pattern`, visualice información solamente para los paquetes especificados.

Hay diversos nombres de pseudoatributos especiales disponibles para su comodidad:

<code>action.hash</code>	El valor del hash de la acción si la acción lleva una carga útil.
<code>action.key</code>	El valor del atributo clave de la acción. Por ejemplo, para una acción <code>file</code> , el atributo clave es la ruta al archivo. Algunas acciones no tienen un atributo clave.
<code>action.name</code>	El nombre de la acción. Por ejemplo, para una acción de archivo, es <code>file</code> .
<code>action.raw</code>	Todos los atributos de acciones coincidentes.
<code>pkg_fmri</code>	El FMRI completo del paquete que contiene la acción, como <code>pkg://solaris/web/amp@0.5.11,5.11-0.169:20110705T153434Z</code> .
<code>pkg.name</code>	El nombre del paquete que contiene la acción, como <code>web/amp</code> .
<code>pkg.publisher</code>	El editor del paquete que contiene la acción, como <code>solaris</code> .
<code>pkg_short_fmri</code>	La forma abreviada del FMRI del paquete que contiene la acción, como <code>pkg://solaris/web/amp@0.5.11,5.11-0.169</code> .

Los subcomandos `contents` y `search` se relacionan: ambos consultan al sistema el contenido de los paquetes. El subcomando `contents` muestra acciones en uno o más paquetes instalables o instalados, y filtra la salida en función de las opciones especificadas. El subcomando `search` encara la consulta desde la dirección contraria, y muestra los nombres de todos los paquetes que contienen un token proporcionado por el usuario.

Cada subcomando puede formular algunas consultas que el otro subcomando puede realizar. El subcomando se debe seleccionar cuidadosamente, ya que una consulta determinada se puede formular con mayor naturalidad en un subcomando que en el otro.

`search [-HIaflpr] [-o atributo, ...] [-s dep_URI] consulta`

Permite buscar coincidencias con la *consulta* y mostrar los resultados. Los tokens que se indexan dependen de la acción, pero pueden incluir hashes de contenido y nombres de ruta. Para obtener información sobre las acciones y sus atributos, consulte "Acciones" en la página del comando `man pkg(5)`. También consulte la lista de nombres de pseudoatributos en `pkg contents` arriba y en `-o` abajo.

De manera predeterminada, las consultas se interpretan como una serie de términos con los que se debe coincidir con exactitud. Los caracteres `?` y `*` se pueden utilizar como comodines `glob(3C)`, lo que permite más flexibilidad en las coincidencias de las consultas.

Con `-H`, se pueden omitir los encabezados.

Con `-I`, se pueden realizar búsquedas con distinción entre mayúsculas y minúsculas.

De manera predeterminada, y con `-a`, se puede realizar una búsqueda y mostrar la información de las acciones coincidentes.

De manera predeterminada, `search` elimina los resultados de los paquetes de versiones anteriores a la versión instalada actualmente y de paquetes de versiones excluidas por las incorporaciones actuales. Utilice `-f` para mostrar todos los resultados, independientemente de la versión del paquete.

Con `-l`, puede buscar los paquetes instalados de la imagen.

Con `-o`, se pueden controlar las columnas de los resultados. La opción `-o` se puede especificar varias veces, o se pueden especificar varios atributos como argumento para una opción `-o` separando los nombres de atributo con comas. Además de los pseudoatributos mencionados anteriormente, se definen los siguientes atributos para los resultados de la búsqueda:

<code>search.match</code>	Corresponde a la cadena que coincide con la consulta de búsqueda.
<code>search.match_type</code>	Corresponde al atributo que contiene la cadena que coincide con la consulta de búsqueda.

Con `-p`, puede mostrar los paquetes que tienen algunas acciones que coinciden con cada término de consulta. Utilizar esta opción es equivalente a colocar paréntesis angulares (`<>`) alrededor de cada término en la consulta. Consulte la información que aparece a continuación para obtener una descripción más detallada del operador `<>`.

De manera predeterminada, y con `-r`, puede buscar los depósitos correspondientes a los editores de la imagen.

Con `-s`, puede buscar el depósito `pkg(5)` ubicado en el URI proporcionado. Esta opción se puede especificar varias veces. No se admiten archivos de paquetes.

Ambos `-l` y `-r` (o `-s`) se pueden especificar juntos, en cuyo caso se realizan búsquedas tanto locales como remotas.

Además de la confrontación simple de tokens y la búsqueda con caracteres comodín, se admite un lenguaje de consulta más complicado. Se pueden buscar frases mediante comillas simples o dobles (`'` o `"`). Asegúrese de tener en cuenta el shell para que `pkg` vea realmente `'` o `"`.

Se admite la búsqueda booleana con `AND` y `OR`. Se admiten las consultas de campo, o estructuradas. La sintaxis para ellas es *nombre_paquete: tipo_acción: clave: token*. Los campos vacíos implícitamente se llenan con caracteres comodín. Una búsqueda de `:basename:pkg` coincide con todos los tipos de acciones en todos los paquetes con una

clave de basename y que coinciden con el token pkg. En los campos pkg_name y token se admiten caracteres comodín explícitos. Los campos action_type y key deben coincidir exactamente.

Para convertir acciones para los paquetes que contienen esas acciones, utilice <>. Con la opción -a, la búsqueda de token genera información sobre las acciones que coinciden con token, mientras que la búsqueda de <token> genera una lista de paquetes que contienen las acciones que coinciden con token.

`verify [-Hqv] [mod_ID rec ges err_paq ...]`

Valida la instalación de paquetes en la imagen actual. Si la política de firmas actual para los editores relacionados no es ignore, las firmas de cada paquete se validan en función de la política. Consulte signature-policy en "Propiedades de la imagen" a continuación, para obtener una explicación de cómo se aplican las políticas de firmas.

Con -H, puede omitir los encabezados de la salida de la verificación.

Con -q, no puede imprimir nada, pero puede revertir el fallo si hay errores fatales.

Con -v, puede incluir mensajes informativos acerca de los paquetes.

`fix [--accept] [--licenses] [mod_ID rec ges err_paq ...]`

Corrija los errores informados por pkg verify. La verificación del contenido de paquetes instalados se basa en un análisis de contenido personalizado que podría devolver resultados distintos de los de otros programas.

Con --accept, puede indicar que está de acuerdo con las condiciones de las licencias de los paquetes que está actualizando o instalando y que las acepta. Si no proporciona esta opción, y alguna licencia de paquete requiere aceptación, la operación fallará.

Con --licenses, puede mostrar todas las licencias de los paquetes que se instalarán o actualizarán como parte de esta operación.

`revert [-nv] [--no-be-activate] [--no-backup-be | --require-backup-be]
[--backup-be-name nombre] [--deny-new-be | --require-new-be] [--be-name
nombre] [--tagged nombre_etiqueta ... | ruta_archivo ...]`

Permite revertir la condición de los archivos a la condición de entrega. Se pueden revertir todos los archivos etiquetados con un valor determinado o archivos individuales. También se restauran las protecciones y la propiedad de archivo.

Precaución – La restitución de los valores predeterminados de algunos archivos editables puede provocar que el sistema no se pueda iniciar u otras fallas.

Para todas las demás opciones, consulte el comando `install` anterior para ver el uso y sus efectos.

`mediator [-aH] [-F formato] [mediador ...]`

Permite mostrar la implementación y/o versión seleccionada actual de todos mediadores o, mediante argumentos, sólo los mediadores especificados.

Con `-a`, puede enumerar las mediaciones que se pueden definir para los paquetes instalados actualmente.

Con `-F`, puede especificar un formato de salida alternativo. Actualmente, sólo `tsv` (valores separados por tabulaciones) es válido.

Con `-H`, puede omitir los encabezados de la lista.

```
set-mediator [-nv] [-I implementación] [-V versión] [--no-be-activate]
[--no-backup-be | --require-backup-be] [--backup-be-name nombre]
[--deny-new-be | --require-new-be] [--be-name nombre] mediador ...
```

Permite establecer la versión y/o la implementación para los mediadores especificados en la imagen actual.

Con `-I`, puede establecer la implementación de la interfaz de mediación para utilizar. De manera predeterminada, si no se especifica ninguna versión, se permiten todas las versiones de implementación. Para especificar una implementación sin ninguna versión, agregue el símbolo arroba (`@`).

Con `-V`, puede establecer la versión de la interfaz de mediación para utilizar.

Si la implementación y/o la versión del mediador especificado no están disponibles actualmente, se eliminan todos los enlaces que utilizan los mediadores especificados.

Para todas las demás opciones, consulte el comando `install` anterior para ver el uso y sus efectos.

```
unset-mediator [-nvIV] [--no-be-activate] [--no-backup-be |
--require-backup-be] [--backup-be-name nombre] [--deny-new-be |
--require-new-be] [--be-name nombre] mediador ...
```

Permite revertir la versión y/o la implementación de los mediadores especificados a los valores predeterminados del sistema.

Con `-I`, puede revertir sólo la implementación la interfaz de mediación.

Con `-V`, puede revertir sólo la versión la interfaz de mediación.

Para todas las demás opciones, consulte el comando `install` anterior para ver el uso y sus efectos.

```
variant [-H] [variant.variant_name ...]
```

Sin argumentos, visualice los valores actuales de todas las variantes definidas en esta imagen. Con argumentos, visualice el valor de cada `variant.variant_name` especificada en esta imagen.

Con `-H`, puede omitir los encabezados de la lista.

Consulte "Facetas y variantes" en la página del comando `man pkg(5)` para obtener más información sobre las variantes.

```
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_name=value
...
```

Cambie los valores de las variantes especificadas en la imagen actual.

Para obtener información sobre el uso de la opción y sus efectos, consulte el comando `install` anterior.

El cambio del valor de una variante puede provocar que el contenido del paquete se elimine, actualice o instale. El cambio del valor de una variante también puede provocar que paquetes enteros se instalen, actualicen o eliminen para ajustarse a la configuración de la nueva imagen. Consulte "Facetas y variantes" en la página del comando `man pkg(5)` para obtener más información sobre las variantes.

```
facet [-H] [facet_name ...]
```

Sin argumentos, visualice los valores actuales de todas las facetas que se han establecido explícitamente en esta imagen mediante el comando `pkg change-facet`. Con argumentos, visualice el valor de cada *facet_name* especificada en esta imagen.

Con `-H`, puede omitir los encabezados de la lista.

Consulte "Facetas y variantes" en la página del comando `man pkg(5)` para obtener más información sobre las facetas.

```
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_name=[True|False|None] ...
```

Cambie los valores de las facetas especificadas en la imagen actual.

Las facetas se pueden establecer en `True` o `False`. Si se establece una faceta en `None`, se aplica el valor predeterminado `True` a esa faceta; por lo tanto, se instalará cualquier acción sujeta a la faceta. Consulte "Acciones" en la página del comando `man pkg(5)` para obtener información sobre las acciones.

Para obtener información sobre el uso de la opción y sus efectos, consulte el comando `install` anterior.

El cambio del valor de una faceta puede provocar que el contenido del paquete se elimine, actualice o instale. El cambio del valor de una faceta también puede provocar que paquetes enteros se instalen, actualicen o eliminen para ajustarse a la configuración de la nueva imagen. Consulte "Facetas y variantes" en la página del comando `man pkg(5)` para obtener más información sobre las facetas.

```
avoid [mod_ID rec ges err_paq ...]
```

Permite evitar los paquetes especificados si son el destino de una dependencia de grupo colocando los nombres de los paquetes que actualmente coinciden con los modelos

especificados en la lista para evitar. Sólo se pueden evitar los paquetes que no están instalados actualmente. Si un paquete actualmente es el destino de una dependencia de grupo, su desinstalación lo coloca en la lista para evitar.

Sin argumentos, permite mostrar cada paquete evitado junto con cualquier paquete que tenga una dependencia de grupo en ese paquete.

Los paquetes incluidos en la lista para evitar se instalan, si es necesario, para satisfacer una dependencia necesaria. Si se elimina esa dependencia, se desinstala el paquete.

`unavoid [mod_ID rec ges err_paq ...]`

Permite eliminar los paquetes especificados de la lista para evitar. Los paquetes incluidos en la lista para evitar que coinciden con una dependencia de grupo de un paquete instalado no se pueden eliminar mediante este subcomando. Para eliminar un paquete que coincide con una dependencia de grupo de la lista para evitar, instale el paquete.

Sin argumentos, permite mostrar la lista de paquetes evitados.

`freeze [-n] [-c motivo] [mod_ID rec ges err_paq] ...`

Permite congelar los paquetes especificados para las versiones especificadas. Si no se proporciona ninguna versión, se debe instalar el paquete, y éste se congela en la versión instalada. Cuando se instala o actualiza un paquete que está congelado, éste debe terminar en una versión que coincida con la versión en la que fue congelado. Por ejemplo, si un paquete se congeló en 1.2, se podría actualizar a 1.2.1, 1.2.9, 1.2.0.0.1, etc. Ese paquete no podría terminar en 1.3 ni 1.1. Un editor especificado en *pkg_fmri_pattern* se utiliza para buscar paquetes coincidentes. Sin embargo, la información del editor no se registra como parte de la congelación. Un paquete se inmoviliza con respecto a su versión únicamente, no a su editor. La congelación de un paquete que ya está congelado sustituye la versión congelada por la nueva versión especificada.

Si no se proporcionan paquetes, se muestra la información sobre los paquetes congelados actualmente: nombres de paquete, versiones, cuándo se congeló el paquete y los motivos relacionados.

La congelación de un paquete no impide la eliminación del paquete. Si se elimina el paquete, no aparece ningún mensaje de advertencia.

Con `-c`, puede registrar el *motivo* en los paquetes congelados. El motivo se muestra si una congelación impide que se lleve a cabo una instalación o actualización.

Con `-n`, puede efectuar una ejecución de prueba de la operación, y mostrar la lista de paquetes que se congelarían sin congelar ningún paquete.

`unfreeze [-n] [mod_ID rec ges err_paq] ...`

Permite eliminar las limitaciones que impone la congelación de los paquetes especificados. Se ignoran las versiones proporcionadas.

Con `-n`, puede efectuar una ejecución de prueba de la descongelación, y mostrar la lista de paquetes que se descongelarían sin descongelar ningún paquete.

`property [-H] [nombre_prop ...]`

Permite mostrar la información de propiedad de la imagen. Sin argumentos, permite mostrar los nombres y los valores de todas las propiedades de la imagen. Si se solicita una lista específica de los nombres de propiedades, permite mostrar los nombres y los valores de dichas propiedades. Consulte "Propiedades de la imagen" que aparece a continuación para obtener descripciones de las propiedades de una imagen.

Con -H, puede omitir los encabezados de la lista.

`set-property nombre_prop valor_prop`

Permite actualizar una propiedad de imagen existente o agregar una propiedad de imagen nueva.

`add-property-value nombre_prop valor_prop`

Permite agregar un valor a una propiedad de imagen existente o agregar una propiedad de imagen nueva.

`remove-property-value nombre_prop valor_prop`

Permite eliminar un valor de una propiedad de imagen existente.

`unset-property nombre_prop ...`

Permite eliminar una propiedad de imagen existente, o varias.

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

Permite mostrar información del editor. Sin argumentos, permite mostrar la lista de todos los editores, los reflejos y los URI de origen en orden de preferencia de búsqueda. Si se solicitan editores específicos, permite mostrar la configuración detallada de esos editores.

Con -H, puede omitir los encabezados de la lista.

Con -P, puede mostrar sólo el primer editor en el orden de búsqueda de editor.

Con -n, puede mostrar sólo los editores habilitados.

`set-publisher [-Ped] [-k ssl_key] [-c ssl_cert] [-g origin_to_add |
--add-origin origin_to_add ...] [-G origin_to_remove | --remove-origin
origin_to_remove ...] [-m mirror_to_add | - --add-mirror mirror_to_add ...] [-M
mirror_to_remove | --remove-mirror mirror_to_remove ...] [--enable] [--disable]
[--no-refresh] [--reset-uuid] [--non-sticky] [--sticky]
[--search-after publisher] [--search-before publisher] [--search-first]
[--approve-ca-cert path_to_CA] [--revoke-ca-cert hash_of_CA_to_remove]
[--unset-ca-cert hash_of_CA_to_remove] [--set-property name_of_property=value]
[--add-property-value name_of_property=value_to_add] [--remove-property-value
name_of_property= value_to_remove] [--unset-property name_of_property_to_delete]
publisher`

Permite actualizar un editor existente o agregar un editor de paquete. Si no se especifican opciones que afectan el orden de búsqueda, se agregan editores nuevos al orden de búsqueda y, por tanto, se buscan en último lugar.

Con `-Po --search-first`, puede establecer el editor especificado en primer lugar en el orden de búsqueda. Cuando se instalen paquetes nuevos, este editor se buscará en primer lugar. Las actualizaciones de los paquetes ya instalados provienen del mismo editor que suministró el paquete originalmente, siempre que este editor sea permanente. Cuando se utiliza `-Po --search-first` con `-p`, únicamente los editores agregados se colocan en primer lugar en el orden de búsqueda.

Con `--non-sticky`, puede especificar que los editores con clasificación superior a la del editor actual pueden proporcionar actualizaciones para los paquetes originalmente instalados desde el editor actual.

Con `--sticky`, puede especificar que las actualizaciones de los paquetes que se instalaron desde este editor también deben provenir de este editor. Éste es el comportamiento predeterminado.

Con `--search-before`, puede modificar el orden de búsqueda de editor para que el editor que se va a modificar se busque antes que el editor especificado. Cuando se utiliza con `-p`, `--search-before` sólo se aplica a los editores agregados.

Con `--search-after`, puede modificar el orden de búsqueda de editor para que el editor que se va a modificar se busque después que el editor especificado. Cuando se utiliza con `-p`, `--search-after` sólo se aplica a los editores agregados.

Con `--approve-ca-cert`, puede agregar el certificado proporcionado como un certificado de CA que es de confianza. Los hashes de la representación PEM de los certificados de CA aprobados por el usuario aparecen en la salida detallada del comando `pkg publisher`.

Con `--revoke-ca-cert`, puede tratar el certificado con un hash determinado de su representación PEM como revocado. Los hashes de los certificados de CA revocados por el usuario aparecen en la salida detallada del comando `pkg publisher`.

Con `--unset-ca-cert`, puede eliminar el certificado con el hash determinado de la lista de certificados aprobados y de la lista de certificados revocados.

Con `--set-property`, puede actualizar una propiedad de editor existente o agregar una propiedad de editor nueva.

Con `--add-property-value`, puede agregar un valor a una propiedad de editor existente o agregar una propiedad de editor nueva.

Con `--remove-property-value`, puede eliminar un valor de una propiedad de editor existente.

Con `--unset-property`, puede eliminar una propiedad de editor existente.

Con `-c y -k`, puede especificar un certificado y una clave SSL de cliente respectivamente.

Con `-g (--add-origin)`, puede agregar la ruta o el URI especificado como origen del editor determinado. Esta debe ser la ubicación de un archivo o depósito de paquetes.

Con -G (--remove-origin), puede eliminar la ruta o el URI de la lista de orígenes del editor determinado. El valor especial * se puede utilizar para eliminar todos los orígenes.

Con --no-refresh, no intente contactar con los depósitos de los editores de la imagen para recuperar la lista más reciente de paquetes disponibles y otros metadatos.

Con --reset-uuid, puede seleccionar un nuevo identificador único que identifique esta imagen con su editor.

Con -m (--add-mirror), puede agregar el URI como un reflejo del editor determinado.

Con -M (--remove-mirror), puede eliminar el URI de la lista de reflejos de un editor determinado. El valor especial * se puede utilizar para eliminar todos los reflejos.

Con -p, puede recuperar la información de configuración del editor desde el URI del depósito especificado. Si se especifica un editor, sólo se agrega o actualiza el editor coincidente. Si no se especifica ningún editor, se agregan o actualizan todos los editores según corresponda. Esta opción no se puede combinar con las opciones -g, --add-origin, --G, --remove-origin, -m, --add-mirror, -M, --remove-mirror, --disable, --enable, --no-refresh, ni --reset-uuid.

Con -e (--enable), puede habilitar el editor. Con -d (--disable), puede deshabilitar el editor. Un editor deshabilitado no se utiliza para completar la lista de paquetes o ni para determinadas operaciones de paquetes (instalación, desinstalación y actualización). Sin embargo, las propiedades de un editor deshabilitado se pueden configurar y ver. Si existe sólo un editor, éste no se puede deshabilitar.

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

Con -p, puede recuperar la información de configuración del editor desde el URI del depósito especificado. Si se especifica un editor, sólo se agrega o actualiza el editor coincidente. Si no se especifica ningún editor, se agregan o actualizan todos los editores según corresponda. Consulte pkg set-publisher arriba para obtener descripciones de otras opciones que se pueden utilizar con la opción -p. La opción -p no se puede combinar con las opciones -g, --add-origin, -G, --remove-origin, -m, --add-mirror, -M, --remove-mirror, --disable, --enable, --no-refresh o --reset-uuid.

```
unset-publisher editor ...
```

Permite eliminar la configuración asociada al editor determinado.

```
history [-Hl] [-t [tiempo | tiempo-tiempo],...] [-o columna,...] [-n número]
```

Permite mostrar el historial de comandos. de la imagen aplicable.

Con -H, puede omitir los encabezados de la lista.

Con `-t`, puede mostrar registros para una lista separada por comas de indicadores de fecha y hora, con el formato `%Y-%m-%dT%H:%M:%S` (consulte `strftime(3C)`). Para especificar un rango de fecha y hora, utilice un guión (`-`) entre el indicador de fecha y hora de inicio y el de finalización. La palabra clave `now` se puede utilizar como un alias para la fecha y hora actual. Si los indicadores de fecha y hora especificados contienen indicadores de fecha y hora duplicados o rangos de fechas superpuestos, sólo se imprimirá una única instancia de cada evento de historial duplicado.

Con `-l`, puede mostrar registros en formato largo, que, además de lo incluido en el formato estándar, incluyen el resultado del comando, la hora en la que se completó el comando, la versión y el nombre del cliente utilizado, el nombre del usuario que realizó la operación y los errores detectados durante la ejecución del comando.

Con `-n`, puede mostrar sólo el número especificado de entradas más recientes.

Con `-o`, puede mostrar la salida utilizando la lista especificada de nombres de columna separados por comas. Los nombres de columna válidos son:

<code>be</code>	El nombre del entorno de inicio en el que se inició esta operación.
<code>be_uuid</code>	El uuid del entorno de inicio en el que se inició esta operación.
<code>client</code>	El nombre del cliente.
<code>client_ver</code>	La versión del cliente.
<code>comando</code>	La línea de comandos utilizada para esta operación.
<code>fin</code>	La fecha y hora en que finalizó esta operación.
<code>id</code>	El identificador de usuario que inició esta operación.
<code>new_be</code>	El entorno de inicio nuevo creado por esta operación.
<code>new_be_uuid</code>	El uuid del entorno de inicio nuevo creado por esta operación.
<code>operation</code>	El nombre de la operación.
<code>outcome</code>	Un resumen del resultado de esta operación.
<code>reason</code>	Información adicional sobre el resultado de esta operación.
<code>instantánea</code>	La instantánea tomada durante esta operación. Ésta se registra solamente si la instantánea no se eliminó automáticamente después de la finalización correcta de la operación.
<code>start</code>	La fecha y hora en que se inició esta operación.
<code>hora</code>	Tiempo total que llevó realizar esta operación. Para las operaciones que llevan menos de un segundo, se muestra 0:00:00.
<code>user</code>	El nombre del usuario que inició esta operación.

Si se especifican las columnas `command` o `reason`, deben ser el último elemento de la lista de `-o`, a fin de conservar separación de campo de salida. Estas dos columnas no se pueden mostrar en el mismo comando `history`.

Aparece un asterisco (*) después los valores para `be` o `new_be`, si el entorno de inicio ya no está presente en el sistema.

Los valores de `be` y `new_be` se obtienen consultando el nombre del entorno de inicio actual, utilizando los campos `be_uuid` o `new_be_uuid`. Si posteriormente el entorno de inicio cambia de nombre y, después, se suprime, los valores que aparecerán para `be` y `new_be` serán los valores registrados en el momento de la operación `pkg`.

`purge-history`

Permite suprimir la información de todo el historial existente.

`rebuild-index`

Permite volver a generar el índice utilizado por `pkg search`. Se trata de una operación de recuperación que no está pensada para uso general.

`update-format`

Permite actualizar el formato de la imagen a la versión actual. Una vez finalizada esta operación, la imagen ya no se podrá utilizar con versiones anteriores del sistema `pkg(5)`.

versión

Permite mostrar una cadena única que identifica la versión de `pkg(1)`. No se garantiza que esta cadena sea comparable de ninguna manera entre versiones.

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

Permite crear una imagen adecuada para operaciones de paquetes en la ubicación determinada por *directorio*. El tipo de imagen predeterminado es usuario, según lo establecido por la opción `-U` (`- -user`). El tipo de imagen se puede establecer en una imagen completa (`- -F` o `- -full`) o en una imagen parcial (`-P` o `- -partial`) enlazada a la imagen completa que abarca la ruta de *directorio*. Se pueden especificar orígenes adicionales mediante `-g` u `- -origin`. Se pueden especificar reflejos adicionales mediante `- -m` o `- -mirror`.

Se debe proporcionar un URI de depósito de paquetes mediante la opción `-p` o `- -publisher`. Si también se proporciona un nombre de editor, al crear la imagen sólo se agregará dicho editor. Si no se proporciona un nombre de editor, todos los editores conocidos por el depósito especificado se agregarán a la imagen. Después de las operaciones iniciales de creación, se realiza un intento para recuperar el catálogo asociado a este editor.

Para los editores que utilizan la autenticación SSL de cliente, se puede registrar una clave de cliente y un certificado de cliente mediante las opciones -c y -k. Esta clave y este certificado se utilizan para todos los editores agregados durante la creación de la imagen.

Si la imagen se ejecutará en un contexto de zona no global, se puede utilizar la opción -z (- - zone) para definir una variante adecuada.

Con -f (- - force), puede forzar la creación de una imagen respecto de una imagen existente. Esta opción se debe utilizar con cuidado.

Con - -no-refresh, no intente contactar con los depósitos de los editores de la imagen para recuperar la lista más reciente de paquetes disponibles y otros metadatos.

Con - -variant, puede establecer la variante especificada en el valor indicado. Consulte "Facetas y variantes" en la página del comando `man pkg(5)` para obtener más información sobre las variantes.

Con - -facet, puede establecer la faceta especificada en el valor indicado. Consulte "Facetas y variantes" en la página del comando `man pkg(5)` para obtener más información sobre las facetas.

Con - -set-property, establezca la propiedad de la imagen especificada en el valor indicado. Consulte "Propiedades de la imagen" a continuación para obtener descripciones de las propiedades de una imagen.

Propiedades de la imagen

Las siguientes propiedades definen las características de la imagen. Estas propiedades almacenan información sobre la finalidad, el contenido y el comportamiento de la imagen. Para ver los valores actuales de estas propiedades en la imagen, use el comando `pkg property`. Para modificar los valores de estas propiedades, use los comandos `pkg set-property` y `pkg unset-property`.

be-policy

(Cadena). Permite especificar cuándo se crea un entorno de inicio durante las operaciones de empaquetado. Se admiten los siguientes valores:

`default` Permite aplicar la política de creación de entorno de inicio predeterminado, `create-backup`.

`always-new` Requiere reiniciar todas las operaciones de paquetes en un nuevo entorno de inicio establecido como habilitado en el siguiente inicio. La copia de seguridad del entorno de inicio no se crea a menos que se solicite de manera explícita.

Esta política es la más segura, pero es más estricta de lo que necesitan la mayoría de sitios, ya que no se pueden agregar paquetes sin reiniciar.

create-backup

Para las operaciones de paquetes que requieren un reinicio, se crea un entorno de inicio nuevo que se define como activo en el inicio siguiente. Si se modifican los paquetes o se instala contenido que puede afectar el núcleo y la operación afecta el entorno de inicio activo, se crea una copia de seguridad del entorno de inicio pero no se establece como activa. La copia de seguridad de un entorno de inicio también se puede solicitar de manera explícita.

Esta política puede ser riesgosa sólo si el software recién instalado provoca inestabilidad del sistema, lo cual es posible, pero relativamente raro.

when-required

Para las operaciones de paquetes que requieren un reinicio, se crea un entorno de inicio nuevo que se define como activo en el inicio siguiente. La copia de seguridad del entorno de inicio no se crea a menos que se solicite de manera explícita.

Esta política es la más riesgosa, ya que si un cambio de empaquetado en el entorno de inicio activo imposibilita la realización de otros cambios, es posible que no haya ningún entorno de inicio reciente con el que se pueda contar.

ca-path

(Cadena). Un nombre de ruta que apunta a un directorio donde se conservan los certificados de CA para operaciones SSL. El formato de este directorio es específico de la implementación SSL subyacente. Para utilizar una ubicación alternativa para los certificados de CA de confianza, cambie este valor para apuntar a un directorio diferente. Consulte las partes `CPath` de `SSL_CTX_load_verify_locations(3openssl)` para los requisitos del directorio de CA.

Valor predeterminado: `/etc/openssl/certs`

check-certificate-revocation

(Booleano). Si se define en `True`, el cliente del paquete intenta contactar a alguno de los puntos de distribución de CRL en los certificados utilizados para la verificación de firmas con el fin de determinar si el certificado se ha revocado desde que se emitió.

Valor predeterminado: `False`

flush-content-cache-on-success

(Booleano). Si se define en `True`, el cliente del paquete elimina los archivos de la antememoria del contenido cuando las operaciones de instalación o actualización se completan. En las operaciones de actualización, el contenido se elimina sólo del entorno de inicio de origen. Cuando se produce la siguiente operación de empaquetado en el entorno de inicio de destino, el cliente de paquete vacía su antememoria de contenido si esta opción no se ha cambiado.

Esta propiedad se puede utilizar para mantener una antememoria de contenido pequeña en los sistemas con espacio limitado en disco. Esta propiedad puede provocar que se demore la finalización de las operaciones.

Valor predeterminado: True

mirror-discovery

(Booleano). Esta propiedad indica al cliente que detecte reflejos de contenido local enlazado mediante mDNS y DNS-SD. Si esta propiedad se define en True, el cliente intenta descargar contenido de paquetes de reflejos que detecta de forma dinámica. Para ejecutar un reflejo que anuncia su contenido por medio de mDNS, consulte `pkg.depotd(1M)`.

Valor predeterminado: False

send-uuid

(Booleano). Permite enviar el identificador único universal (UUID) de la imagen cuando se realizan operaciones de red. Si bien los usuarios pueden deshabilitar esta opción, es posible que algunos depósitos de red rechacen la comunicación con los clientes que no proporcionan un UUID.

Valor predeterminado: True

signature-policy

(Cadena). Permite determinar qué controles se realizarán en los manifiestos al instalar, actualizar, modificar o verificar los paquetes de la imagen. La política final que se aplica a un paquete depende de la combinación de política de imagen y política de editor. La combinación será, como mínimo, tan estricta como la más estricta de las dos políticas tomadas individualmente. De manera predeterminada, el cliente de paquete no comprueba si los certificados han sido revocados. Para activar dichas comprobaciones, que pueden requerir que el cliente contacte sitios web externos, establezca la propiedad de la imagen `check-certificate-revocation` en True. Se admiten los siguientes valores:

<code>ignore</code>	Omite firmas para todos los manifiestos.
<code>verify</code>	Verifica que todos los manifiestos con firmas estén firmados de manera válida, pero no requiere que todos los paquetes instalados estén firmados. Éste es el valor predeterminado.
<code>require-signatures</code>	Requiere que todos los paquetes recién instalados tengan al menos una firma válida. Los comandos <code>pkg fix</code> y <code>pkg verify</code> también advierten si un paquete instalado no tiene una firma válida.
<code>require-names</code>	Tiene los mismos requisitos que <code>require-signatures</code> pero también requiere que las cadenas que aparecen en la propiedad <code>signature-required-names</code> aparezcan como un nombre común de los certificados utilizados para verificar las cadenas de confianza de las firmas.

signature-required-names

(Lista de cadenas). Una lista de nombres que se deben ver como nombres comunes de certificados durante la validación de firmas de un paquete.

`trust-anchor-directory`

(Cadena). El nombre de la ruta del directorio que contiene los anclajes de confianza de la imagen. Esta ruta es relativa a la imagen. El valor predeterminado es `ignore`.

`use-system-repo`

(Booleano). Esta propiedad indica si la imagen debe utilizar el depósito del sistema como origen para la configuración de la imagen y el editor, y como proxy para comunicarse con los editores proporcionados. El valor predeterminado es `False`. Consulte `pkg.sysrepo(1M)` para obtener información sobre los depósitos del sistema.

Propiedades del editor

Las siguientes propiedades definen la política de firmas de un editor particular. Las propiedades de la imagen del mismo nombre definen la política de firmas de la imagen. Para ver los valores actuales de las propiedades de un editor particular, use el comando `pkg publisher publisher_name`. Para modificar los valores de las propiedades de la política de firmas del editor, use las opciones `--set-property` y `--unset-property` del comando `pkg set-publisher`.

`signature-policy`

(Cadena). Esta propiedad funciona de la misma manera que la propiedad de imagen del mismo nombre, con la excepción de que sólo se aplica a los paquetes del editor determinado.

`signature-required-names`

(Lista de cadenas). Esta propiedad funciona de la misma manera que la propiedad de imagen del mismo nombre, con la excepción de que sólo se aplica a los paquetes del editor determinado.

Ejemplos EJEMPLO 1 Creación de una imagen con un editor configurado

Cree una imagen completa nueva con el editor `example.com`, almacenado en `/aux0/example_root`.

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

EJEMPLO 2 Creación de una imagen especificando reflejos y orígenes adicionales

Cree una imagen completa nueva con el editor `example.com` que también tiene un reflejo adicional y dos orígenes adicionales, y está almacenado en `/aux0/example_root`.

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

EJEMPLO 3 Creación de una imagen sin un editor configurado

Cree una imagen completa nueva sin editores configurados en `/aux0/example_root`.

```
$ pkg image-create -F /aux0/example_root
```

EJEMPLO 4 Instalar un paquete

Instale la última versión del paquete `widget` en la imagen actual.

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

EJEMPLO 5 Visualización del contenido especificado de un paquete

Visualice el contenido del paquete `system/file-system/zfs`. Muestre el nombre de la acción, el modo del archivo (si se ha definido), el tamaño (si se ha definido), la ruta y el destino (si es un enlace). Limite la acción a los tipos `dir`, `file`, `link` y `hardlink`, ya que al especificar el atributo `action.name`, que está disponible para todas las acciones, se muestra una línea para todas las acciones, y no es lo que se desea aquí.

```
$ pkg contents -t dir,file,link,hardlink \
-o action.name,mode,pkg.size,path,target system/file-system/zfs
ACTION.NAME MODE PKG.SIZE PATH TARGET
dir          0755          etc
dir          0755        etc/fs
dir          0755      etc/fs/zfs
link         0755      etc/fs/zfs/mount  ../../usr/sbin/zfs
link         0755      etc/fs/zfs/umount  ../../usr/sbin/zfs
dir          0755          etc/zfs
dir          0755          kernel
dir          0755        kernel/drv
dir          0755      kernel/drv/amd64
file         0755  1706744 kernel/drv/amd64/zfs
file         0644    980 kernel/drv/zfs.conf
dir          0755          kernel/fs
dir          0755        kernel/fs/amd64
hardlink     0755      kernel/fs/amd64/zfs  ../../kernel/drv/amd64/zfs
...
```

EJEMPLO 6 Visualización del contenido especificado de dos paquetes

Visualice el contenido de `web/browser/firefox` y `mail/thunderbird`, y limite la visualización sólo a los atributos de ruta y nombre de paquete de las acciones cuyo atributo `path` termine en `.desktop` o `.png`.

```
$ pkg contents -o pkg.name,path -a path=*.desktop \
-a path=*.png web/browser/firefox mail/thunderbird
PKG.NAME PATH
web/browser/firefox usr/share/applications/firefox.desktop
mail/thunderbird usr/share/applications/thunderbird.desktop
web/browser/firefox usr/share/pixmaps/firefox-icon.png
mail/thunderbird usr/share/pixmaps/thunderbird-icon.png
...
```

EJEMPLO 7 Buscar un paquete

Busque la base de datos de paquetes para el token `bge`.

EJEMPLO 7 Buscar un paquete *(Continuación)*

```
$ pkg search bge
INDEX      ACTION VALUE                                PACKAGE
driver_name driver bge                        pkg:/driver/network/bge@0.5.11-0.169
basename   file   kernel/drv/sparcv9/bge      pkg:/driver/network/bge@0.5.11-0.169
basename   file   kernel/drv/amd64/bge        pkg:/driver/network/bge@0.5.11-0.169
pkg.fmri    set    solaris/driver/network/bge  pkg:/driver/network/bge@0.5.11-0.169
```

El token está en el paquete `driver/network/bge` como el nombre de base de la acción de archivo que representa `/kernel/drv, 10x, / arch/bge` y como un nombre de controlador.

EJEMPLO 8 Búsqueda de paquetes que dependen del paquete especificado

Busque los paquetes instalados que dependen de `package/pkg`.

```
$ pkg search -l 'depend::package/pkg'
INDEX      ACTION VALUE                                PACKAGE
incorporate depend package/pkg@0.5.11-0.169  pkg:/consolidation/ips/ips-incorporation@0.5.11-0.169
require    depend package/pkg@0.5.11-0.169  pkg:/system/install@0.5.11-0.169
require    depend package/pkg@0.5.11-0.169  pkg:/package/pkg/system-repository@0.5.11-0.169
```

EJEMPLO 9 Búsqueda de dependencias

Busque todas las dependencias de `incorporate` en los paquetes instalados.

```
$ pkg search -l 'depend:incorporate:'
INDEX      ACTION VALUE                                PACKAGE
incorporate depend pkg:/BRcMbnx@0.5.11,5.11-0.133  pkg:/consolidation/osnet/osnet-incorporation@0.5.11-0.133
incorporate depend pkg:/BRcMbnxe@0.5.11,5.11-0.133  pkg:/consolidation/osnet/osnet-incorporation@0.5.11-0.133
...
```

EJEMPLO 10 Adición de un editor

Agregue un nuevo editor `example.com`, con un depósito ubicado en `Http://www.example.com/repo`.

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

EJEMPLO 11 Adición de un editor con clave y certificado

Agregue un nuevo editor `example.com`, con un depósito seguro ubicado en `https://secure.example.com/repo`, y una clave y un certificado almacenados en el directorio `/root/creds`.

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

EJEMPLO 12 Adición y configuración automática de un editor

Agregue un nuevo editor con un depósito ubicado en `/export/repo` mediante la configuración automática.

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

EJEMPLO 13 Adición y configuración manual de un editor

Agregue un nuevo editor `example.com` con un depósito ubicado en `/export/repo/example.com` mediante la configuración manual.

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

EJEMPLO 14 Verificación de todos los paquetes firmados

Configure una imagen para verificar todos paquetes firmados.

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

EJEMPLO 15 Requerimiento de que todos los paquetes estén firmados

Configure una imagen para requerir que todos los paquetes estén firmados y solicite que la cadena `example.com` se vea como un nombre común para uno de los certificados de la cadena de confianza.

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

EJEMPLO 16 Requerimiento de que todos los paquetes de un editor especificado estén firmados

Configure una imagen para que todos los paquetes instalados desde el editor `example.com` deban estar firmados.

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

EJEMPLO 17 Requerimiento de una cadena especificada en la cadena de confianza

Agregue la cadena `foo` a la lista de nombres comunes de la imagen que se deben ver en una cadena de confianza de firma para que se considere válida.

```
$ pkg add-property-value signature-require-names foo
```

EJEMPLO 18 Eliminación de una cadena de la cadena de confianza para un editor especificado

Elimine la cadena `foo` de la lista de nombres comunes que se deben ver para validar una firma para el editor `example.com`.

```
$ pkg set-publisher --remove-property-value signature-require-names=foo \
example.com
```

EJEMPLO 19 Adición de un certificado de CA de confianza

Agregue el certificado almacenado en `/tmp/example_file.pem` como certificado de CA de confianza para el editor `example.com`.

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

EJEMPLO 20 Revocación de un certificado

Revoque el certificado con el hash `a12345` del editor `example.com` e impida que el certificado valide firmas de los paquetes de `example.com`.

```
$ pkg set-publisher --revoke-ca-cert a12345 example.com
```

EJEMPLO 21 Olvido de un certificado

Haga que `pkg` olvide que el certificado `a12345` ha sido agregado o revocado por el usuario.

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

EJEMPLO 22 Actualización de un paquete a una versión anterior

Actualice el paquete instalado `foo@1.1` a una versión anterior.

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

EJEMPLO 23 Cambio de instalación de paquete en conflicto

En el caso de dos paquetes en conflicto, cambie uno de los paquetes que se están instalando. Supongamos que el paquete `A` depende del paquete `B` o el paquete `C`, y `B` y `C` se excluyen mutuamente. Si instala `A` y `B`, utilice el siguiente comando para pasar a usar `C` en lugar de `B` sin desinstalar `A`:

```
$ pkg install --reject B C
```

EJEMPLO 24 Visualización de paquetes de un archivo de paquetes

Visualice todas las versiones de todos los paquetes de un archivo de paquetes.

```
$ pkg list -f -g /my/archive.p5p
```

EJEMPLO 25 Visualización de paquetes de un depósito de paquetes

Visualice todas las versiones de todos los paquetes de un depósito.

```
$ pkg list -f -g http://example.com:10000
```

EJEMPLO 26 Visualización de información sobre un paquete de un archivo de paquetes

Visualice la información del paquete para ver la versión más reciente de un paquete de un archivo de paquetes. El paquete puede estar instalado actualmente o no.

```
$ pkg info -g /my/archive.p5p pkg_name
```


EJEMPLO 27 Visualización del contenido de un paquete de un archivo de paquetes

Visualice el contenido de un paquete de un archivo de paquetes. El paquete no está instalado actualmente.

```
$ pkg contents -g /my/archive.p5p pkg_name
```

EJEMPLO 28 Eliminación de todos los reflejos y orígenes de un editor

Elimine todos los reflejos y orígenes de un editor y agregue un nuevo origen.

```
$ pkg set-publisher -G '*' -M '*' -g http://example.com:10000 \
example.com
```

Variables del entorno

PKG_IMAGE

El directorio que contiene la imagen que se va a utilizar para las operaciones de paquetes. Si se especifica -R, se omite.

PKG_CLIENT_CONNECT_TIMEOUT

Los segundos que se debe esperar para la conexión durante las operaciones de transporte (para cada intento) antes de que el cliente anule la operación. Un valor de 0 significa que espere de manera indefinida.

Valor predeterminado: 60

PKG_CLIENT_LOWSPEED_TIMEOUT

La cantidad de segundos inferior al límite lowspeed (1024 bytes/s) que deben transcurrir durante las operaciones de transporte antes de que el cliente anule la operación. Un valor de 0 significa que no cancele la operación.

Valor predeterminado: 30

PKG_CLIENT_MAX_CONSECUTIVE_ERROR

El número máximo de errores temporales de transporte que se pueden producir antes de que el cliente anule la operación. Un valor de 0 significa que no cancele la operación.

Valor predeterminado: 4

PKG_CLIENT_MAX_REDIRECT

El número máximo de redireccionamientos de HTTPS o HTTP permitidos durante las operaciones de transporte antes de que se anule una conexión. Un valor de 0 significa que no cancele la operación.

Valor predeterminado: 5

PKG_CLIENT_MAX_TIMEOUT

El número máximo de intentos de transporte por host antes de que el cliente anule la operación. Un valor de 0 significa que no cancele la operación.

Valor predeterminado: 4

http_proxy, https_proxy
Servidor proxy HTTP o HTTPS.

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 El comando se ejecutó correctamente.
- 1 Se produjo un error.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 3 Se solicitaron varias operaciones, pero sólo algunas se realizaron correctamente.
- 4 No se realizaron cambios, no se requiere ninguna acción.
- 5 La operación solicitada no se puede realizar en una imagen activa.
- 6 La operación solicitada no se puede finalizar debido a que las licencias de los paquetes que se están instalando o actualizando no han sido aceptadas.
- 7 La imagen actualmente se está utilizando en otro proceso y no se puede modificar.
- 99 Se produjo una excepción no anticipada.

Archivos Se puede ubicar una imagen pkg(5) arbitrariamente en un sistema de archivos más grande. En las siguientes descripciones de archivo, el token `$IMAGE_ROOT` se utiliza para distinguir rutas relacionadas. Para una instalación de sistema típica, `$IMAGE_ROOT` es equivalente a /

<code>\$IMAGE_ROOT/var/pkg</code>	Directorio de metadatos para una imagen completa o parcial.
<code>\$IMAGE_ROOT/.org.opensolaris/pkg</code>	Directorio de metadatos para una imagen de usuario.

Dentro de los metadatos de una imagen determinada, algunos archivos y directorios pueden contener información útil durante la reparación y la recuperación. El token `$IMAGE_META` hace referencia al directorio de nivel superior que contiene los metadatos. `$IMAGE_META` generalmente es una de las dos rutas especificadas anteriormente.

<code>\$IMAGE_META/lost+found</code>	Ubicación de los archivos y directorios conflictivos movidos durante una operación de paquetes. Ubicación del contenido desempaquetado de un directorio eliminado.
<code>\$IMAGE_META/publisher</code>	Contiene un directorio para cada editor. Cada directorio almacena los metadatos específicos de un editor.

Otras rutas dentro de la jerarquía de directorios de `$IMAGE_META` son privadas y están sujetas a cambios.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkgsend\(1\)](#), [pkg.depotd\(1m\)](#), [glob\(3C\)](#), [pkg\(5\)](#), [beadm\(1M\)](#)

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

Nombre pkgdepend – analizador de dependencia de Image Packaging System

Sinopsis /usr/bin/pkgdepend [*options*] *command* [*cmd_options*] [*operands*]

```
/usr/bin/pkgdepend generate [-IMm] -d dir [-d dir]  
                        [-D name=value] [-k path] manifest_file
```

```
/usr/bin/pkgdepend resolve [-moSv] [-d output_dir]  
                        [-s suffix] manifest_file ...
```

Descripción El comando pkgdepend se utiliza para generar y resolver dependencias de paquetes. Un paquete puede depender de archivos de otros paquetes. El comando pkgdepend generalmente se utiliza en dos etapas: la generación de dependencia de archivos y la resolución de archivo a paquete.

E subcomando generate analiza el contenido de un paquete y determina qué archivos externos necesita el paquete.

El subcomando resolve toma la lista de archivos del paso generate y busca un conjunto de paquetes de referencia para determinar los nombres de los paquetes que contienen dichos archivos dependientes. El conjunto de paquetes de referencia en los que se buscan los archivos dependientes son los paquetes que están actualmente instalados en el sistema del editor.

Diversos componentes de los archivos entregados se utilizan como fuentes de información de dependencia:

ELF Los encabezados ELF de los archivos entregados se analizan en busca de información de dependencia, y las opciones -k y -D modifican la información obtenida. Para obtener más detalles sobre las dependencias ELF, consulte `ldd(1)` y la [Linker and Libraries Guide](#).

Secuencias de comandos Las secuencias de comandos de shell que contienen líneas `#!` que hacen referencia a un intérprete generan una dependencia del paquete que ofrece dicho intérprete.

Python Las secuencias de comandos de Python en primer lugar se analizan como secuencias de comandos. Además, cualquier importación que declare la secuencia de comandos también puede servir como fuente de información de dependencia.

Enlaces físicos Los enlaces físicos de los manifiestos generan una dependencia del paquete que ofrece el destino de enlace.

SMF Los manifiestos de servicios SMF entregados que incluyen dependencias `require_all` generan dependencias de los paquetes que ofrecen los manifiestos SMF que proporcionan los FMRI.

Opciones Se admiten las siguientes opciones:

- R *directorio* Realiza operaciones en la imagen con raíz en *directorio*. Si no se especificó ningún directorio, o si no se determinó ninguno en función del entorno, el valor predeterminado es /. Para obtener más información, consulte la sección "Variables del entorno".
- help o -? Muestra un mensaje de uso.

Subcomandos Se admiten los subcomandos siguientes:

generate [-Imm] [-d *dir* [-d *dir*] [-D *name=value*] [-k *path*] *manifest_file*

Produce las dependencias de los archivos del manifiesto especificado por *manifest_file*.

Con -I, muestre las dependencias que se satisfacen en *manifest_file*. No utilice el resultado de -I con pkgdepend resolve.

Con -M, puede mostrar una lista de los tipos de archivo que no se pudieron analizar.

Con -m, puede repetir el manifiesto original con cualquier dependencia detectada agregada después.

Con -d, puede agregar *directorio* a una lista de directorios en los que se buscarán los archivos del manifiesto.

Para cada -D, agregue el *valor* como una manera de expandir el token *nombre* en rutas de ejecución para las dependencias de archivos ELF.

Para cada -k, agregue la *ruta* a la lista de rutas de ejecución para la búsqueda de módulos de núcleo. Mediante el argumento -k se eliminan las rutas predeterminadas, que son /kernel y /usr/kernel.

Las rutas de ejecución como las especificadas por la opción -k también se pueden especificar por acción o por manifiesto mediante el atributo de acción o manifiesto pkg.depend.runpath. El valor del atributo pkg.depend.runpath es una cadena separada por dos puntos de las rutas para utilizar.

El uso de -k se sustituye por cualquier atributo pkg.depend.runpath establecido en el manifiesto o la acción.

El token especial \$PKGDEPEND_RUNPATH se puede utilizar como un componente del valor de atributo pkg.depend.runpath para incluir la ruta de ejecución del sistema estándar para el archivo que se está analizando.

En algunos casos, es posible que desee impedir la generación automática de dependencias. Por ejemplo, si un paquete ofrece una secuencia de comandos de Python de ejemplo que importa un conjunto de módulos, los módulos importados por la secuencia de comandos de ejemplo no son dependencias del paquete que ofrece la secuencia de comandos de ejemplo. Utilice el atributo de acción o manifiesto pkg.depend.bypass-generate para evitar generar dependencias con los archivos especificados.

Los valores de `pkg.depend.bypass-generate` son expresiones regulares de Python que coinciden con los nombres de archivo. Las expresiones regulares están implícitamente ancladas al inicio y al final de la ruta del archivo. El valor del ejemplo siguiente coincide con `this/that`, pero no coincide con `something/this/that/the/other`.

```
pkg.depend.bypass-generate=this/that
```

Para obtener más información sobre la sintaxis de la expresión regular de Python, utilice el comando `pydoc re` o vea documentación más completa en <http://docs.python.org/dev/howto/regex.html>.

Cuando los manifiestos de entrada `pkgdepend generate` contienen archivos de manifiestos SMF, los servicios o las instancias SMF declarados por esos archivos de manifiestos SMF se incluyen en la salida de `pkgdepend`. Estas instancias o servicios SMF se incluyen en forma de una acción set con el nombre `org.opensolaris.smf.fmri`.

```
resolve [-moSv] [-d output_dir] [-s suffix] manifest_file ...
```

Permite transformar dependencias de archivos en dependencias de los paquetes que ofrecen esos archivos. Las dependencias se resuelven, en primer lugar, según los manifiestos de la línea de comandos y, después, según los paquetes instalados en el sistema. De manera predeterminada, las dependencias de cada manifiesto se colocan en un archivo llamado `manifest_file.res`.

Con `-m`, puede repetir el manifiesto, con cualquier dependencia producida por el paso `generate` eliminada, antes de agregar las dependencias resueltas.

Con `-o`, puede escribir los resultados en la salida estándar. Esta opción está diseñada para uso humano. Es posible que al agregar esta salida a un archivo se produzca un manifiesto no válido. Se recomienda utilizar las opciones `-d o -s` en lugar de `-o` en una canalización para el procesamiento de manifiestos.

Con `-d`, puede escribir las dependencias resueltas de cada manifiesto proporcionado en un archivo independiente en `dir_salida`. De manera predeterminada, cada archivo tiene el mismo nombre de base que el manifiesto que fue el origen de las dependencias escritas en ese archivo.

Con `-s`, para cada archivo de salida, puede adjuntar *sufijo* al nombre de base del archivo que fue el origen de las dependencias resueltas. A “.” se antepone a *sufijo* si no se proporciona.

Con `-S`, sólo puede resolver según los manifiestos proporcionados en la línea de comandos y no según los manifiestos instalados en el sistema.

Con `-v`, puede incluir metadatos de depuración de dependencia de paquete adicional.

Ejemplos EJEMPLO 1 Generación de dependencias

Genere las dependencias para el manifiesto escrito en `foo`, cuyo directorio de contenido se encuentra en `./bar/baz`, y almacene los resultados en `foo.fdeps`.

```
$ pkgdepend generate -d ./bar/baz foo > foo.fdeps
```

EJEMPLO 2 Resolución de dependencias

Resuelva las dependencias de archivos en `foo.fdeps` y `bar.fdeps` según el uno y el otro, y según los paquetes instalados actualmente en el sistema.

```
$ pkgdepend resolve foo.fdeps bar.fdeps
$ ls *.res
foo.fdeps.res    bar.fdeps.res
```

EJEMPLO 3 Generación y resolución de dependencias para dos manifiestos

Genere las dependencias de archivos para dos manifiestos (`foo` y `bar`) y retenga toda la información en los manifiestos originales. A continuación, resuelva las dependencias de archivos y coloque los manifiestos resultantes en `./res`. Estos manifiestos resultantes se pueden utilizar con `pkgsend publish`.

```
$ pkgdepend generate -d /proto/foo -m foo > ./deps/foo
$ pkgdepend generate -d /proto/bar -m bar > ./deps/bar
$ pkgdepend resolve -m -d ./res ./deps/foo ./deps/bar
$ ls ./res
foo    bar
```

EJEMPLO 4 Adición de valores a tokens para dependencias de archivos ELF

Reemplace todos los tokens `PLATFORM` de las rutas de ejecución de los archivos ELF por `sun4v` y `sun4u` al generar las dependencias para el manifiesto escritos en `foo`, cuyo directorio de contenido se encuentra en `/`.

```
$ pkgdepend generate -d / -D 'PLATFORM=sun4v' -D 'PLATFORM=sun4u' foo
```

EJEMPLO 5 Especificación de un directorio de módulo de núcleo

Especifique `/kmod` como el directorio en el que se buscarán los módulos de núcleo al generar las dependencias para el manifiesto escrito en `foo`, cuyo directorio de contenido se encuentra en `/`.

```
$ pkgdepend generate -d / -k /kmod foo
```

EJEMPLO 6 Omisión de la generación de dependencia

Agregue `opt/python` a la ruta de ejecución de Python estándar para una secuencia de comandos de Python determinada y omita la generación de dependencia con respecto a todos los módulos Python llamados `test` para un archivo entregado como `opt/python/foo/file.py`.

EJEMPLO 6 Omisión de la generación de dependencia (Continuación)

Evite generar dependencias respecto de cualquier archivo entregado en `usr/lib/python2.6/vendor-packages/xdg`.

```
$ cat manifest.py
set name=pkg.fmri value=pkg:/mypackage@1.0,1.0
set name=pkg.summary value="My test package"
dir path=opt mode=0755 group=sys owner=root
dir path=opt/python mode=0755 group=sys owner=root
dir path=opt/python/foo mode=0755 group=sys owner=root
file NOHASH path=opt/python/__init__.py mode=0644 group=sys owner=root
file NOHASH path=opt/python/foo/__init__.py mode=0644 group=sys owner=root
#
# Add runpath and bypass-generate attributes:
#
file NOHASH path=opt/python/foo/file.py mode=0644 group=sys owner=root \
  pkg.depend.bypass-generate=.*test.py.* \
  pkg.depend.bypass-generate=.*testmodule.so \
  pkg.depend.bypass-generate=.*test.so \
  pkg.depend.bypass-generate=usr/lib/python2.6/vendor-packages/xdg/. * \
  pkg.depend.runpath=$PKGDEPEND_RUNPATH:/opt/python

$ pkgdepend generate -d proto manifest.py
```

- Variables del entorno
- PKG_IMAGE
- Especifica el directorio que contiene la imagen que se va a utilizar para operaciones de paquetes. Este valor se ignora, si se especifica -R.
- Estado de salida
- Se devolvieron los siguientes valores de salida:
- 0
- Ejecución correcta.
- 1
- Se produjo un error.
- 2
- Se especificaron opciones de línea de comandos no válidas.
- 99
- Se produjo una excepción no anticipada.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkg\(5\)](#)

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

Nombre pkgdiff – comparación de manifiestos de paquete

Sinopsis /usr/bin/pkgdiff [-i *atributo* ...] [-o *atributo*]
[-v *nombre=valor* ...] *archivo1* *archivo2*

Descripción pkgdiff compara dos manifiestos de paquete y diferencias de informes. pkgdiff ordena cada manifiesto y acción de manera coherente antes de la comparación.

La salida tiene el formato siguiente:

+ *acción_completa* Esta acción está en *archivo2* pero no en *archivo1*.

- *acción_completa* Esta acción está en *archivo1* pero no en *archivo2*.

actionname keyvalue [*variant values, if any*]

- *atributo1=valor1* Este *atributo,valor* está en *archivo1* pero no en *archivo2*.

+ *atributo2=valor2* Este *atributo,valor* está en *archivo2* pero no en *archivo1*.

Las acciones con variantes diferentes pero con el mismo valor de atributo de tipo y clave se tratan como acciones separadas para la comparación. Por lo tanto, las acciones que cambian los atributos aparecen en su formato completo en lugar de aparecer como cambios de atributos.

Opciones Se admiten las siguientes opciones:

- i *atributo* Ignora el *atributo* si está presente durante las comparaciones. Los valores hash de archivo se pueden ignorar con -i *hash*. Esta opción no se puede utilizar con la opción -o. Esta opción se puede repetir.
- o *atributo* Sólo informa las diferencias en *atributo*. Esta opción no se puede utilizar con la opción -i. Esta opción borra cualquier cambio de acción que no afecte el *atributo* en una acción.
- v *nombre= valor* Sólo calcula las diferencias para este valor de variante. Por ejemplo, sólo calcula las diferencias para arch=sparc. Esta etiqueta de variante se elimina de todas las acciones antes de la comparación. Sólo se puede especificar un valor por variante. Esta opción se puede repetir para distintas variantes.

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 No se encontró ninguna diferencia.
- 1 Se encontraron diferencias.
- >1 Se produjo un error.
- 99 Se produjo una excepción no anticipada.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkg\(5\)](#)

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

Nombre pkgfmt – formateo un manifiesto de paquete

Sinopsis /usr/bin/pkgfmt [-c|-d|-u] [*archivo_manifiesto_paquete*]

Descripción pkgfmt sin las opciones -c o -d formatea un manifiesto de paquete de manera coherente, por ejemplo, mediante el ajuste de líneas a 80 caracteres, la clasificación de acciones por tipo y la clasificación de atributos. Líneas que no se analizan en las acciones (como macros, comentarios o transformaciones) no aparecerán en orden.

Si no se proporcionan argumentos, pkgfmt lee stdin hasta EOF y, a continuación, escribe el manifiesto formateado en stdout. Los manifiestos especificados en la línea de comandos se formatean en su lugar.

pkgfmt con la opción -c comprueba si los manifiestos se formatearon en el estilo pkgfmt. La opción -d muestra las diferencias si el archivo no está formateado correctamente.

Opciones Se admiten las siguientes opciones:

- c Comprueba si el manifiesto está formateado en el estilo pkgfmt.
- d Muestra las diferencias del manifiesto con respecto a la versión formateada de forma unificada.
- u No ajusta las líneas a 80 caracteres. Esta opción es útil para aplicar herramientas de procesamiento de texto tradicionales a los manifiestos de paquete.

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 El comando se ejecutó correctamente.
- 1 Se especificaron las opciones -c o -d y uno o varios manifiestos no están en el formato normal pkgfmt, o se produjo un error.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 99 Se produjo una excepción no anticipada.

Atributos Consulte attributes(5) para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también pkg(5)

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

Nombre pkglint – lint de paquete de Image Packaging System

Sinopsis /usr/bin/pkglint [-c directorio] [-r uri] [-p exp_reg]
[-f archivo_origen] [-b núm_gen] [-v]
[-l uri] | manifiesto ...

/usr/bin/pkglint -L [-v]

Descripción pkglint realiza una serie de comprobaciones en uno o varios manifiestos de paquete y, opcionalmente, hace referencia a otro depósito.

pkglint se debe utilizar durante el proceso de creación de paquetes, antes de la publicación de paquetes. pkglint realiza pruebas exhaustivas en los manifiestos que pueden ser demasiado costosas para realizar durante el funcionamiento normal de pkgsend(1) o pkg.depotd(1M). pkglint realiza pruebas de inclusión para detectar acciones duplicadas, atributos faltantes y permisos de archivo inusuales.

Los manifiestos para hacer lint se pueden pasar como una lista de archivos locales separados por espacios en la línea de comandos, o los manifiestos se pueden recuperar desde un depósito.

Al recuperar manifiestos desde depósitos, en la primera ejecución pkglint crea y rellena las imágenes de usuario de pkg(5) en el directorio de antememoria especificado. Si se suministra la opción -r, se crea una imagen de usuario llamada *dir_antem/ref_image* para el depósito de referencia. Si se suministra la opción -l se crea una imagen de usuario llamada *dir_antem/lint_image* para el depósito de lint. No se instala ningún contenido en estas imágenes. Estas imágenes sólo son utilizadas por pkglint para recuperar los manifiestos de los depósitos.

Las invocaciones posteriores de pkglint pueden volver a utilizar el directorio de antememoria y pueden omitir cualquier argumento -r o -l.

pkglint provee asistencia limitada para configurar editores en el directorio de antememoria. Use pkg(1) para realizar una configuración de editor más compleja en estas imágenes.

pkglint permite a los autores de paquetes omitir comprobaciones para una acción o un manifiesto determinado. Un manifiesto o una acción que contiene el atributo pkg.linted establecido en True no produce ninguna salida de lint para dicho manifiesto o dicha acción.

Se pueden realizar más ajustes granulares de pkg.linted mediante subcadenas de comprobación de nombres pkglint. Por ejemplo, pkg.linted.comprobar.id establecido en True omite todas las comprobaciones con el nombre *comprobar.id* para la acción o el manifiesto determinado.

El comportamiento de pkglint se puede configurar especificando un archivo pkglintrc. De manera predeterminada, pkglint busca opciones de configuración en /usr/share/lib/pkg/pkglintrc y \$HOME/.pkglintrc. Utilice la opción -f para especificar un archivo de configuración diferente.

Durante la ejecución de lint, los errores o advertencias detectados se imprimen en `stderr`.

Opciones Se admiten las siguientes opciones:

- b *núm_gen* Especifica un número de generación utilizado para restringir la lista de paquetes utilizados durante la ejecución de lint desde depósitos de lint y de referencia. Si no se especifica ninguna opción -b, se utilizan las últimas versiones de los paquetes. Consulte también la propiedad de configuración `version.pattern`.
- c *dir_antem* Especifica un directorio local utilizado para almacenar en la antememoria los metadatos del paquete de los depósitos de lint y de referencia.
- l *uri_lint* Especifica un URI que representa la ubicación del depósito de lint. Se admite la publicación basada en HTTP y en sistema de archivos. Si especifica -l, también debe especificar -c.
- L Muestra las comprobaciones lint conocidas y excluidas, y finaliza. Muestra el nombre abreviado y la descripción de cada comprobación. Cuando se combina con el indicador -v, muestra el método que implementa la comprobación en lugar de la descripción.
- f *archivo_config* Configura la sesión de `pkglint` mediante el archivo de configuración *archivo_config*.
- p *exp_reg* Especifica una expresión regular utilizada para restringir la lista de paquetes que se deben comprobar del depósito de lint. Se cargan todos los manifiestos del depósito de referencia (siempre que coincidan con el valor para -b, si se proporciona), ignorando este modelo.
- r *uri_dep* Especifica un URI que representa la ubicación del depósito de referencia. Si especifica -r, también debe especificar -c.
- v Ejecuta `pkglint` en un modo detallado, lo que sustituye cualquier configuración de `log_level` en el archivo de configuración.
- help o -? Muestra un mensaje de uso.

Archivos El archivo de configuración `pkglint.rc` requiere los siguientes argumentos de clave/valor:

- `log_level` El nivel mínimo en el que se deben emitir mensajes lint. Los mensajes lint inferiores a este nivel se descartan. El valor predeterminado es `INFO`.

Los niveles de registro en orden de severidad ascendente son `DEBUG`, `INFO`, `WARNING`, `ERROR` y `CRITICAL`.

<code>do_pub_checks</code>	Si es <code>True</code> , realiza comprobaciones que quizás sólo tengan sentido para los paquetes publicados. El valor predeterminado es <code>True</code> .
<code>pkglint.ext.*</code>	El mecanismo de complemento de <code>pkglint</code> permite agregar módulos lint adicionales en tiempo de ejecución. Cualquier clave que empieza con <code>pkglint.ext</code> lleva un valor que debe ser un módulo Python completamente especificado. Para obtener más información, consulte la sección “Desarrolladores”.
<code>pkglint.exclude</code>	Una lista de nombres de función, clases o módulos Python completamente especificados separados por espacios, que se deben omitir del conjunto de comprobaciones que se realizarán.
<code>use_progress_tracker</code>	Si es <code>True</code> , utiliza un indicador de progresos al iterar los manifiestos durante las ejecuciones de lint. El valor predeterminado es <code>True</code> .
<code>version.pattern</code>	Un modelo de versión, utilizado al especificar un número de generación para realizar el lint (<code>-b</code>). Si no se especifica en el archivo de configuración, la opción <code>-b</code> utiliza el modelo <code>*,5.11-0.</code> , que coincide con todos los componentes de la versión 5.11, con el prefijo de ramificación 0.

Desarrolladores Para ampliar el conjunto de comprobaciones realizadas por `pkglint`, subclase `pkg.lint.base.Checker` y sus subclases, `ManifestChecker`, `ActionChecker` y `ContentChecker`. Agregue el nombre del módulo Python que contiene esas clases a una nueva clave `pkglint.ext` en el archivo de configuración.

Las instancias de esas nuevas subclases son creadas por `pkglint` en el inicio. Durante el transcurso de la sesión lint se invocan métodos dentro de cada subclase con el argumento de palabra clave especial `pkglint_id`. Esos métodos deben tener la misma firma que el método `check()` correspondiente en la superclase. A los métodos también se les debe asignar un atributo `pkglint_desc`, que se utiliza como la descripción impresa por `pkglint -L`.

Los parámetros están disponibles para las subclases `Checker`, lo que les permite ajustar su comportamiento. La convención de denominación de parámetro recomendada es *pkglint_id.nombre*. Los valores de los parámetros se pueden almacenar en el archivo de configuración, o se puede acceder a ellos mediante las acciones o los manifiestos recuperados con el método `LintEngine.get_param()`. Al acceder a los parámetros desde el manifiesto, se antepone el prefijo `pkg.lint` al nombre de la clave para asegurarse de que los parámetros `pkglint` no se superpongan con los valores de las acciones o los manifiestos existentes.

Ejemplos EJEMPLO 1 Primera ejecución en un depósito determinado

Ejecución de una sesión pkglint por primera vez en un depósito determinado.

```
$ pkglint -c /space/cache -r http://localhost:10000 mymanifest.mf
```

EJEMPLO 2 Ejecución subsiguiente en el mismo depósito

Una ejecución subsiguiente en el mismo depósito utilizado en el Ejemplo 1.

```
$ pkglint -c /space/cache mymanifest-fixed.mf
```

EJEMPLO 3 Uso de un depósito lint con un conjunto de manifiestos limitado

Ejecución de una sesión pkglint con un depósito lint y especificación de un subconjunto de manifiestos para comprobar.

```
$ pkglint -c /space/othercache -l http://localhost:10000 \
-p '*.firefox.*'
```

EJEMPLO 4 Especificación de una versión

Ejecución de una sesión pkglint en una versión determinada, en modo detallado.

```
$ pkglint -c /space/cache -r http://localhost:10000 \
-l http://localhost:12000 -b 147 -v
```

EJEMPLO 5 Modificación de un archivo de configuración

Un archivo de configuración con un módulo lint nuevo, con algunas comprobaciones excluidas.

```
$ cat ~/.pkglintrc
[pkglint]
```

```
log_level = DEBUG
```

```
# log_level = INFO
```

```
pkglint.ext.mycheck = org.timf.mychecks
```

```
pkglint.ext.opensolaris = pkg.lint.opensolaris
```

```
pkglint.exclude: pkg.lint.opensolaris.OpenSolarisActionChecker
```

```
pkg.lint.pkglint.PkgActionChecker.unusual_perms pkg.lint.pkglint.PkgManifestChecker
```

```
pkg.lint.opensolaris.OpenSolarisManifestChecker
```

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 El comando se ejecutó correctamente.
- 1 Una o varias comprobaciones lint emitieron una salida.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 99 Se produjo una excepción no anticipada.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkg\(1\)](#), [pkg.depotd\(1m\)](#), [pkgsend\(1\)](#), [pkg\(5\)](#)
<http://hub.opensolaris.org/bin/view/Project+pkg/>

Nombre pkgmerge – utilidad de fusión de paquetes de Image Packaging System

Sinopsis `/usr/bin/pkgmerge [-n] -d dep_dest
 -s variante=valor[,...],dep_origen ...
 [mod_ID rec ges err_paq ...]`

Descripción pkgmerge es una herramienta de publicación de paquetes para crear paquetes de variantes múltiples. Esto se realiza mediante la fusión de paquetes con versiones y nombres idénticos (excluido el indicador de fecha y hora), mediante el etiquetado de las acciones que son únicas en las versiones que se van a fusionar con el nombre y valor de la variante especificada para el origen determinado y, por último, mediante la publicación de paquetes nuevos en el depósito de destino. Sólo se utiliza la versión más reciente de cada paquete de cada origen.

Si una acción tiene el atributo `pkg.merge.blend` definido en el nombre de la variante que se va a fusionar, esa acción se copia en los demás manifiestos antes de la fusión, para que la acción aparezca sin etiquetas de variante agregadas en la salida final. Tenga en cuenta que el atributo `pkg.merge.blend` en sí se elimina de todas las acciones en el manifiesto de salida. Este atributo se puede repetir con distintos valores para fusiones de transferencias múltiples.

Las acciones no idénticas que realizan la entrega en la misma ruta en un manifiesto de entrada generan un error en la salida de pkgmerge.

Opciones Se admiten las siguientes opciones:

`-d dep_dest`

El URI o la ruta del sistema de archivos del depósito de destino donde se publican los paquetes fusionados. El depósito de destino debe existir previamente. Se pueden crear nuevos depósitos mediante `pkgrepo(1)`.

`-n`

Realiza una ejecución de prueba sin que se realicen cambios en el depósito de destino.

`-s variante=valor[,...],dep_origen`

El nombre y el valor de la variante que se va a utilizar para los paquetes de este origen, seguidos por el URI o la ruta del sistema de archivos del depósito de origen o el archivo de paquetes de donde se recuperarán los paquetes. Se pueden especificar múltiples variantes separadas por comas. Se deben nombrar las mismas variantes para todos los orígenes. Esta opción se puede especificar varias veces.

`--help o -?`

Muestra un mensaje de uso.

Variables del entorno Se admite la siguiente variable de entorno:

TMPDIR La ruta absoluta del directorio donde se deben almacenar los datos temporales durante la ejecución del programa. Si no se establece, de manera predeterminada los datos temporales se almacenan en `/var/tmp`.

Ejemplos EJEMPLO 1 Especificación del nombre y el valor de la variante

Etiquete cada paquete encontrado en el origen especificado con el nombre y el valor de la variante determinada especificada para el origen del cual se recuperó:

```
$ pkgmerge -s arch=sparc,http://src.example.com \  
-d http://dest.example.com
```

Paquete de ejemplo:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z  
dir group=sys mode=0755 owner=root path=usr
```

Paquete de ejemplo después de la operación:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z  
set name=variant.arch value=sparc  
dir group=sys mode=0755 owner=root path=usr
```

EJEMPLO 2 Fusión y publicación de paquetes

Fusione la versión más reciente de cada uno de los paquetes de los orígenes determinados y publique los paquetes nuevos en el depósito de destino:

```
$ pkgmerge -s arch=sparc,http://src1.example.com \  
-s arch=i386,http://src2.example.com \  
-d /path/to/target/repository
```

Paquete de ejemplo del origen 1 (SPARC):

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T121410Z  
file id mode=0555 owner=root group=bin path=usr/bin/foo  
dir group=sys mode=0755 owner=root path=usr
```

Paquete de ejemplo del origen 2 (i386):

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z  
file id mode=0555 owner=root group=bin path=usr/bin/foo  
dir group=sys mode=0755 owner=root path=usr
```

Paquete fusionado:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z  
set name=variant.arch value=sparc value=i386  
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=sparc  
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=i386  
dir group=sys mode=0755 owner=root path=usr
```

EJEMPLO 3 Fusión de paquetes depurados y no depurados para sistemas i386 y SPARC

Fusione la versión más reciente de cada uno de los paquetes de un conjunto de depósitos depurados y no depurados para sistemas i386 y SPARC:

EJEMPLO 3 Fusión de paquetes depurados y no depurados para sistemas i386 y SPARC (Continuación)

```
$ pkgmerge -s arch=sparc,debug=false,/repo/sparc-nondebug \
-s arch=sparc,debug=true,/repo/sparc-debug \
-s arch=i386,debug=false,/repo/i386-nondebug \
-s arch=i386,debug=true,/repo/i386-debug \
-d /path/to/target/repository
```

Paquete de ejemplo del origen 1 (SPARC, no depurado):

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T121410Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

Paquete de ejemplo del origen 2 (SPARC, depurado):

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T121411Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

Paquete de ejemplo del origen 3 (i386, no depurado):

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

Paquete de ejemplo del origen 4 (i386, depurado):

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163428Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

Paquete fusionado:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163428Z
set name=variant.arch value=sparc value=i386
set name=variant.debug value=false value=true
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=sparc variant.debug=false
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=sparc variant.debug=true
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=i386 variant.debug=false
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=i386 variant.debug=true
dir group=sys mode=0755 owner=root path=usr
```

EJEMPLO 4 Fusión mediante pkg.merge.blend

Fusione paquetes para dos arquitecturas que no chocan, mediante el atributo `pkg.merge.blend`.

EJEMPLO 4 Fusión mediante pkg.merge.blend (Continuación)

```
$ pkgmerge -s arch=sparc,http://src1.example.com \
-s arch=i386,http://src2.example.com \
-d /path/to/target/repository
```

Paquete de ejemplo del origen 1 (SPARC):

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T121410Z
file 1d5eac1aab628317f9c088d21e4afda9c754bb76 mode=0555 owner=root \
    group=bin path=usr/bin/sparc/foo pkg.merge.blend=arch
file d285ada5f3cae14ea00e97a8d99bd3e357caadc0 mode=0555 owner=root \
    group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

Paquete de ejemplo del origen 2 (i386):

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
file a285ada5f3cae14ea00e97a8d99bd3e357cb0dca mode=0555 owner=root \
    group=bin path=usr/bin/i386/foo pkg.merge.blend=arch
file d285ada5f3cae14ea00e97a8d99bd3e357caadc0 mode=0555 owner=root \
    group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

Paquete fusionado:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
set name=variant.arch value=sparc value=i386
file d285ada5f3cae14ea00e97a8d99bd3e357caadc0 mode=0555 owner=root \
    group=bin path=usr/bin/foo
file a285ada5f3cae14ea00e97a8d99bd3e357cb0dca mode=0555 owner=root \
    group=bin path=usr/bin/i386/foo
file 1d5eac1aab628317f9c088d21e4afda9c754bb76 mode=0555 owner=root \
    group=bin path=usr/bin/sparc/foo
dir group=sys mode=0755 owner=root path=usr
```

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 El comando se ejecutó correctamente.
- 1 Se produjo un error.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 99 Se produjo una excepción no anticipada.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkgrepo\(1\)](#), [pkg\(5\)](#)

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

Nombre pkgmogrify – transmogrifier de manifiesto de Image Packaging System

Sinopsis /usr/bin/pkgmogrify [-vi] [-I *incluir_dir* ...]
[-D *macro=valor* ...] [-O *archivo_salida*]
[-P *impr_archivo*] [*archivo_entrada* ...]

Descripción pkgmogrify permite la edición programática de los manifiestos de paquetes para simplificar las transformaciones típicas necesarias para automatizar la generación de software y la publicación de paquetes.

pkgmogrify proporciona lo siguiente:

- Sustitución de macro para facilitar el intercambio de un único manifiesto en diversas arquitecturas y plataformas.
- Inclusión de otros manifiestos o fragmentos de manifiestos, como transformaciones y componentes estándar.
- Transformación de acciones de paquetes, incluidas la modificación, la supresión o la adición de atributos de acción.

Opciones Se admiten las siguientes opciones:

-D *nombre=valor*

Define el *nombre* como una macro, con el valor *valor*. En el archivo de entrada las macros aparecen como \$ (*macro*). La sustitución se repite hasta que no se encuentren más traducciones. Los idiomas comunes incluyen:

- Eliminación de líneas en un manifiesto en otras arquitecturas mediante una etiqueta específica de arquitectura al principio de la línea:

```
$(sparc_ONLY)file ...
```

Al procesar la arquitectura SPARC, esta macro se establecería en la cadena vacía. Al procesar otras arquitecturas, esta macro se establecería en # en la línea de comandos y, de esta manera, eliminaría esta acción del manifiesto en la arquitectura actual.

- Definición de partes de nombres de rutas específicas de la plataforma, como el nombre del directorio de la arquitectura de 64 bits para archivos ejecutables y bibliotecas:

```
file NOHASH path=usr/bin/$(ARCH64)/cputrack ...
```

Estas macros se debe definir en el valor deseado en la línea de comandos. No hay valores de macro predefinidos.

-I *incluir_directorio*

Agrega el directorio especificado a la ruta de búsqueda para archivos especificados en la línea de comandos y directivas de inclusión incrustadas.

-O *archivo_salida*

Escribe la salida del manifiesto en el archivo especificado. Si se produce un error o si una directiva de transformación provoca una operación de cancelación, el archivo no se escribe. De manera predeterminada, la salida del manifiesto se escribe en stdout.

-P *impr_archivo*

Escribe la salida que resulta de las operaciones de impresión de la directiva de transformación en el archivo especificado. Si se produce un error o si una directiva de transformación provoca una operación de cancelación, el archivo no se escribe. De manera predeterminada, la salida de impresión se escribe en `stdout`.

-i

Ignora las directivas de inclusión en los archivos. Sólo se procesan los archivos especificados en la línea de comandos (o `stdin`).

-v

Escribe comentarios en el manifiesto de salida que muestra el efecto de las transformaciones. Esta información puede facilitar la depuración.

--help o -?

Muestra un mensaje de uso.

Directivas incrustadas

Se admiten dos tipos de directivas en los archivos de manifiesto: directivas de inclusión y directivas de transformación.

Las directivas de inclusión tienen el formato siguiente:

```
<include file>
```

Esta directiva hace que `pkgmogrify` busque un archivo llamado `file`, primero, en el directorio actual y, luego, en los directorios especificados mediante la opción `-I`. Si se encuentra este archivo, su contenido se inserta en el manifiesto en el punto en el que se detecta la directiva. Si no se encuentra, `pkgmogrify` se cierra con un error.

Las directivas de transformación tienen el formato siguiente:

```
<transform matching-criteria -> operation>
```

Estas directivas se acumulan hasta que todas las entradas se hayan leído en la memoria y, luego, aplicado a las acciones en el orden en el que se detectaron.

Los criterios de coincidencia tienen el formato siguiente:

```
[action-type ... ] [attribute=<value-regexp> ...]
```

Uno de los *tipos_acción* especificados debe coincidir. Todos los *atributos* especificados deben coincidir. La sintaxis de la expresión regular utilizada es la de Python. Para obtener más información sobre la sintaxis de la expresión regular de Python, utilice el comando `pydoc re` o vea documentación más completa en <http://docs.python.org/dev/howto/regex.html>. La expresión regular se ancla al principio, no al final. Por lo tanto, una expresión regular para confrontar archivos según sus extensiones debe incluir `*` al principio y `$` al final.

Se pueden especificar varios criterios separados por espacios.

Las siguientes operaciones están disponibles:

<code>add</code>	Agrega un valor a un atributo. Esta operación utiliza dos argumentos. El primer argumento es el nombre del atributo y el segundo es el valor.
<code>default</code>	Establece el valor de un atributo si no existe previamente. Esta operación utiliza los mismos dos argumentos que la operación <code>add</code> .
<code>delete</code>	Elimina los valores de atributos. Esta operación utiliza dos argumentos. El primer argumento es el nombre del atributo. El segundo argumento es una expresión regular para confrontar los valores de atributos suprimidos. A diferencia de la expresión regular que se utiliza para confrontar acciones, esta expresión no se ancla.
<code>drop</code>	Desecha la acción.
<code>edit</code>	Modifica un atributo de la acción. Esta operación utiliza tres argumentos. El primer argumento es el nombre del atributo y el segundo es una expresión regular para confrontar el valor de atributo. El tercer argumento es la cadena de sustitución reemplazada por la parte del valor de coincidencia detectado por la expresión regular. A diferencia de la expresión regular que se utiliza para confrontar acciones, esta expresión no se ancla. Si se definen grupos en la expresión regular, hay retroreferencias de expresión regular normal, con el formato <code>\1</code> , <code>\2</code> , etcétera, disponibles en la cadena de sustitución.
<code>emit</code>	Emite una línea a la secuencia de salida del manifiesto. Se debe tratar de una cadena de acción válida, vacía (que genera una línea en blanco), o un comentario (un <code>#</code> seguido de un texto arbitrario).
<code>exit</code>	Finaliza el procesamiento del manifiesto. No se genera la salida de ningún manifiesto ni se aplican operaciones <code>print</code> . Si se proporciona un argumento, debe ser un valor entero, y se utiliza como código de salida. El valor predeterminado es 0. Si se especifican dos argumentos, el primero es el código de salida y el segundo es un mensaje para imprimir en <code>stderr</code> .
<code>print</code>	Imprime un mensaje en el archivo de salida especificado mediante <code>-P</code> .
<code>set</code>	Establece el valor de un atributo. Esta operación utiliza los mismos dos argumentos que la operación <code>add</code> .

Todas las operaciones, excepto `delete` y `drop`, utilizan (posiblemente opcional) argumentos cuyo contenido se dirige a la secuencia de salida. Estas cadenas pueden contener tres tipos diferentes de tokens especiales que permiten que la salida incluya información que no está basada en una transformación fija de cada acción.

El primer tipo de sustitución permite a la operación hacer referencia a los valores de los atributos de la acción actual, mediante la colocación del nombre del atributo entre paréntesis a continuación de un signo de porcentaje. Por ejemplo, `%(alias)` hace referencia al valor del atributo `alias` de la acción.

Existen varios atributos sintéticos. Dos son exclusivos de pkgmogrify:

- `pkg.manifest.filename` hace referencia al nombre del archivo en el que se encontró la acción.
- `pkg.manifesto.lineno` hace referencia a la línea en la que se encontró la acción.

Tres atributos sintéticos son similares a los utilizados en pkg(1):

- `action.hash` hace referencia al valor hash de la acción si la acción lleva una carga útil. Para acciones con cargas útiles, la operación `set` puede cambiar el hash de la acción realizando la operación en el atributo `action.hash`.
- `action.key` hace referencia al valor del atributo clave.
- `action.name` hace referencia al nombre del tipo de acción.

Si el atributo cuyo valor se solicita no existe, pkgmogrify se cierra con un error. Para evitar un cierre con error, a continuación del nombre del atributo escriba `;notfound=` y un valor para sustituir el valor del atributo. Por ejemplo, `%(alias;notfound='no alias')` imprime el valor del atributo `alias` si existe, y de lo contrario imprime `no alias`.

Si el atributo cuyo valor se solicita tiene múltiples valores, se imprimen todos los valores separados por espacios. De la misma manera que el token `notfound`, los tokens `prefix`, `suffix` y `sep` se pueden utilizar para cambiar este comportamiento. La cadena indicada mediante `prefix` se antepone a cada valor, la cadena indicada mediante `suffix` se agrega a cada valor, y `sep` se coloca entre el sufijo de un valor y el prefijo del valor siguiente.

De la misma manera que los atributos de acción, las directivas pkgmogrify pueden hacer referencia a los atributos del paquete mediante llaves en lugar de paréntesis: `%{pkg.fmri}`. En el punto en el que se aplica la directiva de transformación, el atributo se debe haber definido en una acción `set`, de lo contrario, se trata como `notfound`, descrito anteriormente. Cuando el procesamiento alcanza el final del archivo de manifiesto que describe el paquete, los atributos se borran para el paquete siguiente.

Es útil no sólo hacer referencia a los atributos del paquete como si fueran atributos de acción, sino también buscar coincidencias en ellos e, incluso, modificarlos temporalmente. Por lo tanto un nombre de acción sintético, `pkg`, está disponible (sólo en el contexto de pkgmogrify) para su uso en estas situaciones.

Cuando pkgmogrify termina de leer un manifiesto especificado en la línea de comandos, y dicho manifiesto define un atributo de paquete `pkg.fmri`, pkgmogrify crea esta acción sintética `pkg`, cuyos atributos son los atributos del paquete. Entonces, una directiva `<transform>` puede encontrar coincidencias en esta acción, al igual que cualquier otro tipo de acción.

Las operaciones en una acción `pkg` son especiales porque se realizan sólo en la memoria y no afectan directamente el manifiesto emitido. Por ejemplo, si se intenta definir un atributo en

una acción pkg por medio de las operaciones add, default o set, no se provocará que se agregue una acción set al manifiesto, aunque ésta estará disponible para que otras directivas <transform> busquen coincidencias. Si se intenta realizar la operación emit con pkg, se generará un error. Para agregar un atributo de paquete, realice la operación emit con la acción set.

El tercer tipo de sustitución es una retroreferencia. Esta sustitución no es como aquellas que se pueden utilizar en la operación edit, es una referencia a grupos que aparecen en la coincidencia de transformación a la izquierda de ->. Se indica mediante %<1>, %<2> y así sucesivamente, en el orden en el que aparecen en los criterios de coincidencia.

El orden de procesamiento es el siguiente:

1. Se leen las líneas desde los archivos de entrada.
2. Se aplican los macros.
3. Se procesan las directivas <include ...> y <transform>, lo que provoca que se encuentren y se lean archivos adicionales.
4. Una vez que se acumula toda la entrada, todas las líneas de entrada se convierten en acciones y se aplican todas las transformaciones.
5. Una vez que el procesamiento finaliza correctamente, se escribe la salida.

Ejemplos EJEMPLO 1 Adición de etiquetas a manifiestos SMF

Agregue etiquetas a los manifiestos de Service Management Facility (SMF) para que se importen cuando el paquete se instale en un sistema activo.

```
<transform file path=(var|lib)/svc/manifest/*.xml -> \  
    add restart_fmri svc:/system/manifest-import:default>
```

EJEMPLO 2 Movimiento de archivos

Mueva archivos de usr/sfw/bin a usr/bin.

```
<transform file -> edit path usr/sfw/bin usr/bin>
```

EJEMPLO 3 Especificación de reinicio necesario

Agregue etiquetas reboot-needed a archivos de /kernel que no sean archivos .conf. Tenga en cuenta que en este ejemplo se aprovecha cómo se aplican las transformaciones en cada acción, en el orden visto en los archivos de entrada.

```
<transform file path=kernel/*. * -> set reboot-needed true>  
<transform file path=kernel/*. *.conf -> delete reboot-needed .*>
```

Esto también se puede hacer en una sola regla de transformación con expresiones regulares.

EJEMPLO 4 Conversión de un atributo FMRI en una acción de dependencia

Convierta el atributo de paquete `pkg.fMRI` en una acción `depend` para que pase a formar parte de una incorporación.

```
<transform set name=pkg.fMRI -> \
    emit depend type=incorporate fMRI=%(value)>
<transform set name=pkg.fMRI -> drop>
```

EJEMPLO 5 Impresión de una lista de números de error

Imprima una lista de números de error entre comillas y con prefijo, separados por comas.

```
set name=bugs value=12345 value=54321 value=13579 value=97531
<transform set name=bugs -> \
    print %(value;sep=",";prefix="bug='";suffix="')>
```

EJEMPLO 6 Admisión de falta de atributos

Imprima un mensaje de manera segura, incluso si falta un atributo.

```
<transform driver -> print Found aliases: %(alias;notfound=<none>)>
```

EJEMPLO 7 Establecimiento de valores predeterminados

Establezca los valores predeterminados de propietario, grupo y permiso.

```
<transform file dir -> default owner root>
<transform file dir -> default group bin>
<transform file -> default mode 0444>
<transform dir -> default mode 0755>
```

EJEMPLO 8 Adición de dependencias a paquetes que no están marcados como obsoletos

Para cualquier paquete que no esté marcado como obsoleto, agregue una dependencia en la incorporación para la consolidación que ofrece el paquete. Este conjunto de transformaciones se deben realizar después de que el manifiesto se haya leído, o se emitirá la dependencia siempre. Como la modificación de una acción `pkg` no tiene un efecto permanente, no es necesario eliminar los atributos que coinciden con `pkg.obsolete=false`.

```
<transform pkg -> default pkg.obsolete false>
<transform pkg pkg.obsolete=false -> emit depend \
    fMRI=consolidation/$(CONS)/$(CONS)-incorporation type=require>
```

EJEMPLO 9 Cierre e impresión de un mensaje cuando se encuentra un error

Cuando se encuentra un atributo obsoleto en un manifiesto, se genera un error y un mensaje.

```
<transform file dir link hardlink opensolaris.zone=.* -> \
    exit 1 The opensolaris.zone attribute is obsolete.>
```

EJEMPLO 10 Establecimiento de la faceta de configuración regional adecuada

Establezca la faceta de configuración regional adecuada para el nombre de ruta en consideración.

```
<transform dir file link hardlink path=.* /locale/([^\s]+).* -> \
    default facet.locale.%<1> true>
```

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 Ejecución correcta.
- 1 Sucedió algo malo pero anticipado.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 99 Error de procesamiento inesperado.

Archivos /usr/share/pkg/ttransforms Este directorio contiene archivos con transformaciones útiles para establecer facetas, actuadores y otros atributos.

Atributos Consulte attributes(5) para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkg\(1\)](#), [pkg\(5\)](#)

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

Nombre pkgrecv – utilidad de recuperación de contenido de Image Packaging System

Sinopsis /usr/bin/pkgrecv [-s *uri_origen*] [-a] [-d (*ruta*|*uri_destino*)]
 [-c *dir_antem*] [-kr] [-m *coincidencia*] [-n] [--raw]
 [--key *archivo_clave* --cert *archivo_cert*] (*fmri*|*modelo*) ...
 /usr/bin/pkgrecv [-s *uri_origen*] --newest

Descripción pkgrecv permite al usuario recuperar paquetes de un archivo de paquetes o un depósito pkg(5). pkgrecv también le permite al usuario volver a publicar los paquetes recuperados en otro depósito de paquetes o archivarlos. De manera predeterminada, los paquetes se recuperan en un formato de depósito de paquetes adecuado para utilizar con pkg(1), pkg.depotd(1M), y herramientas de publicación de paquetes.

Después de una operación de pkgrecv, ejecute pkgrepo refresh o pkgrepo rebuild en el repositorio para crear índices de búsqueda.

Opciones Se admiten las siguientes opciones:

- a Almacena los datos de paquetes recuperados en un archivo pkg(5), en la ubicación especificada por -d. El archivo no puede existir previamente. Esta opción se puede utilizar sólo con destinos basados en sistemas de archivos. Aunque no sea obligatorio, se recomienda utilizar una extensión de archivo .p5p (por ejemplo, archive.p5p). Esta opción no se puede combinar con --raw.
- c *dir_antem* La ruta de un directorio que se utilizará para almacenar el contenido descargado en la antememoria. Si este directorio no se suministra, el cliente selecciona automáticamente un directorio de antememoria. En el caso de que se interrumpa la descarga y se haya elegido automáticamente un directorio de antememoria, utilice esta opción para reanudar la descarga. Consulte la sección “Variables de entorno” a continuación, para ver más detalles sobre cómo establecer la ubicación utilizada para el almacenamiento temporal de datos.
- d *ruta_o_uri* El URI o la ruta del sistema de archivos del destino en el que volverán a publicar los paquetes. Si se especifica -a, el destino es un archivo de paquetes nuevo que ya no puede existir. De lo contrario, el destino tiene que ser un repositorio de paquetes que ya existe. Se pueden crear nuevos depósitos mediante pkgrepo(1).
- h Muestra un mensaje de uso.
- k Mantiene comprimido el contenido del paquete recuperado. Esta opción se ignora al volver a publicar. El contenido del paquete comprimido no se debe utilizar con pkgsend(1).
- m *coincidencia* Controla el comportamiento de las coincidencias mediante los valores siguientes:

	<code>all-timestamps</code>	Incluye todos los indicadores de fecha y hora, no sólo el más reciente (incluye <code>all-versions</code>).
	<code>all-versions</code>	Incluye todas las versiones coincidentes, no sólo la última.
<code>-n</code>		Lleva a cabo una ejecución de prueba sin realizar cambios.
<code>-r</code>		Recupera recursivamente todas las dependencias para la lista de paquetes suministrada.
<code>-s uri_dep_origen</code>		Un URI que representa la ubicación de un archivo de paquetes o un depósito pkg(5) desde el que se recibirán los datos de los paquetes.
<code>--cert archivo</code>		Especifica el archivo de certificados SSL de cliente que se utilizará para recuperar paquetes desde un depósito HTTPS.
<code>--key archivo</code>		Especifica el archivo de claves SSL de cliente que se utilizará para recuperar paquetes desde un depósito HTTPS.
<code>--newest</code>		Muestra las versiones más recientes de los paquetes disponibles en el depósito especificado y se cierra. (Se ignoran todas las demás opciones excepto <code>-s</code>).
<code>--raw</code>		Recupera y guarda los datos de paquetes sin formato en un conjunto de estructuras de directorio, por recurso y versión, en la ubicación especificada por <code>-d</code> . Esta opción se puede utilizar sólo con destinos basados en sistemas de archivos. Estos datos de paquete se pueden utilizar para modificar y volver a publicar paquetes de manera conveniente, tal vez, mediante la corrección del contenido del archivo o el suministro de metadatos adicionales del paquete. Esta opción no se puede combinar con <code>-a</code> .

Ejemplos EJEMPLO 1 Visualización de los paquetes más recientes

Visualice los paquetes más recientes disponibles en el depósito, en el sistema denominado `test`.

```
$ pkgrecv -s http://test --newest
pkg://solaris/system/library/c++-runtime@0.5.11,5.11-0.174.0.0.0.0:20110921T190358Z
pkg://solaris/system/library/freetype-2@2.4.8,5.11-0.175.1.0.0.7.1234:20120109T215840Z
pkg://solaris/system/library/math@0.5.11,5.11-0.174.0.0.0.0.0:20110921T190432Z
```

EJEMPLO 2 Recuperación de datos de paquetes sin formato

Reciba el paquete `c++-runtime` del Ejemplo 1 en un formato adecuado para utilizar con `pkgsend publish`.

```
$ pkgrecv -s http://test \
-d /local/repo --raw \
```

EJEMPLO 2 Recuperación de datos de paquetes sin formato *(Continuación)*

```

c++-runtime@0.5.11,5.11-0.174.0.0.0.0:20110921T190358Z
Processing packages for publisher solaris ...
Retrieving and evaluating 1 package(s)...
PROCESS                                ITEMS      GET (MB)    SEND (MB)
Completed                             1/1        3.5/3.5      0.0/0.0
$ ls /local/repo
pkg5.repository publisher system%2Flibrary%2Fc%2B%2B-runtime

```

EJEMPLO 3 Recuperación de dependencias de un sistema

Reciba el paquete editor/vim y todas sus dependencias del sistema denominado test.

```
$ pkgrecv -s http://test -d /local/repo -r editor/vim
```

EJEMPLO 4 Recuperación de todas las versiones

Reciba todas las versiones del paquete editor/vim del sistema denominado test.

```

$ pkgrecv -s http://test -d /local/repo -m all-versions editor/vim
Processing packages for publisher solaris ...
Retrieving and evaluating 2 package(s)...
PROCESS                                ITEMS      GET (MB)    SEND (MB)
Completed                             2/2       16.7/16.7   44.9/44.9

```

EJEMPLO 5 Recuperación de todas las versiones y nueva publicación remota

Reciba todas las versiones del paquete library/zlib del sistema denominado test y vuelva a publicarlo en un repositorio remoto, en el sistema denominado remote.

```
$ pkgrecv -s http://test -d http://remote:10000 -m all-versions library/zlib
```

EJEMPLO 6 Recuperación de dependencias de un depósito

Reciba el paquete editor/gnu-emacs y todas sus dependencias del repositorio ubicado en /export/repo.

```
$ pkgrecv -s /export/repo -d /local/repo -r editor/gnu-emacs
```

EJEMPLO 7 Recuperación de paquetes adicionales

Reciba todos los paquetes que no existan previamente del depósito ubicado en http://example.com:10000.

```
$ pkgrecv -s http://example.com:10000 -d /my/pkg/repo '*'
```

EJEMPLO 8 Creación de un archivo de paquetes

Cree un archivo de paquetes que contenga el paquete editor/gnu-emacs y todas sus dependencias del repositorio ubicado en http://example.com:10000.

EJEMPLO 8 Creación de un archivo de paquetes (Continuación)

```
$ pkgrecv -s http://example.com:10000 -d /my/emacs.p5p -a -r editor/gnu-emacs
```

EJEMPLO 9 Copia de paquetes de un archivo a un depósito

Copie todos los paquetes de un archivo de paquetes a un depósito existente ubicado en /export/repo.

```
$ pkgrecv -s /my/archive.p5p -d /export/repo '*'
```

Variables del entorno Se admiten las siguientes variables de entorno:

- PKG_DEST** La ruta de un directorio para guardar el paquete recuperado, o el URI o la ruta del sistema de archivos de un archivo de paquetes o un depósito donde se copiarán los paquetes.
- PKG_SRC** Un URI o una ruta del sistema de archivos que representa la ubicación de un archivo de paquetes o un depósito pkg(5) desde el cual se recuperarán paquetes.
- TMPDIR** La ruta absoluta del directorio donde se deben almacenar los datos temporales durante la ejecución del programa. Si no se establece, de manera predeterminada los datos temporales se almacenan en /var/tmp.

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 El comando se ejecutó correctamente.
- 1 Se produjo un error.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 3 Se solicitaron varias operaciones, pero sólo algunas se realizaron correctamente.
- 99 Se produjo una excepción no anticipada.

Atributos Consulte attributes(5) para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también pkgrepo(1), pkgsend(1), pkg(5)

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

Nombre pkgrepo – utilidad de gestión de depósitos de Image Packaging System

Sinopsis `/usr/bin/pkgrepo create [--version ver] uri_o_ruta`
`/usr/bin/pkgrepo add-publisher -s uri_o_ruta_dep editor ...`
`/usr/bin/pkgrepo get [-F formato] [-p editor ...]`
`-s uri_o_ruta_dep [sección/propiedad ...]`
`/usr/bin/pkgrepo info [-F formato] [-H] [-p editor ...]`
`-s uri_o_ruta_dep`
`/usr/bin/pkgrepo list [-F formato] [-H] [-p editor ...]`
`-s uri_o_ruta_dep [mod_ID rec ges err_paq ...]`
`/usr/bin/pkgrepo rebuild [-p editor ...]`
`-s uri_o_ruta_dep [--no-catalog] [--no-index]`
`/usr/bin/pkgrepo refresh [-p editor ...]`
`-s uri_o_ruta_dep [--no-catalog] [--no-index]`
`/usr/bin/pkgrepo remove [-n] [-p editor ...]`
`-s uri_o_ruta_dep mod_ID rec ges err_paq ...`
`/usr/bin/pkgrepo set [-p editor] -s uri_o_ruta_dep`
`sección/propiedad=[valor] ... or`
`sección/propiedad=([valor]) ...`
`/usr/bin/pkgrepo help`
`/usr/bin/pkgrepo version`

Descripción pkgrepo ofrece la posibilidad de crear y gestionar depósitos de paquetes pkg(5). Los depósitos de paquetes son un conjunto predefinido de directorios y archivos que permiten el almacenamiento y la recuperación de los datos de los paquetes mediante pkg(1) y clientes de publicación, como pkgsend (1) o pkgrecv(1). Además, cuando se necesita acceso basado en red a un depósito de paquetes, pkg.depotd(1m) puede ofrecer a los clientes acceso al depósito para almacenar y/o recuperar los datos de los paquetes.

Opciones Se admiten las siguientes opciones:

`--help o -?` Muestra un mensaje de uso.

Subcomandos Se admiten los subcomandos siguientes:

`create [--version ver] uri_o_ruta`
 Crea un depósito pkg(5) en la ubicación especificada.

Este subcomando se puede utilizar sólo con depósitos basados en sistemas de archivos.

Con `--version`, puede crear un depósito en un formato compatible con la versión especificada. De manera predeterminada, se crean depósitos de versión 4. Las versiones admitidas son:

- 3 Admite el almacenamiento de paquetes para un único editor, la versión de catálogo 1 y la versión de búsqueda 1.
- 4 Admite el almacenamiento de paquetes para múltiples editores, la versión de catálogo 1 y la versión de búsqueda 1.

`add-publisher -s uri_o_ruta_dep editor ...`

Agrega los editores especificados al depósito. Los editores nuevos no tienen paquetes ni contenido.

Este subcomando se puede utilizar sólo con depósitos basados en sistemas de archivos versión 4.

`get [-F formato] [-p editor ...] -s uri_o_ruta_dep [sección/propiedad ...]`

Muestra la información de propiedad para el depósito o sus editores.

De manera predeterminada, cada propiedad y su valor se imprimen en líneas diferentes. Los valores de cadena ASCII vacíos están representados por un par de comillas dobles (""). Los siguientes metacaracteres de shell Bourne, y nueva línea, espacio y tabulación, en los valores de cadena ASCII deben llevar el carácter de escape de la barra diagonal inversa (\):

`; & () | ^ < > \ " ' ``

Consulte la sección “Ejemplos”.

Para obtener una lista de las propiedades posibles, y el objetivo y el valor de cada propiedad, consulte el subcomando `set` a continuación.

Con `-F`, puede especificar un formato de salida alternativo. El valor de *formato* puede ser `tsv` (valores separados por tabuladores), `json` (JavaScript Object Notation, como una sola línea) o `json-formatted` (JavaScript Object Notation, en formato para mejorar la legibilidad).

Con `-H`, puede omitir los encabezados de la lista.

Con `-p`, puede mostrar la información de las propiedades del editor determinado. El valor especial `all` muestra las propiedades de todos los editores. Esta opción se puede especificar varias veces.

Con `-s`, puede realizar operaciones en el depósito ubicado en la ruta del sistema de archivos o el URI determinados.

`info [-F formato] [-H] [-p editor ...] -s uri_o_ruta_dep`

Muestra una lista de los editores de paquetes conocidos por el depósito. La lista incluye el número de paquetes para cada editor, cuándo se actualizaron por última vez los datos de los paquetes del editor y el estado de los datos de los paquetes del editor (por ejemplo, si se están procesando en ese momento).

Con `-F`, puede especificar una formato de salida alternativo. El valor de *formato* puede ser `tsv` (valores separados por tabuladores), `json` (JavaScript Object Notation, como una sola línea) o `json-formatted` (JavaScript Object Notation, en formato para mejorar la legibilidad).

Con `-H`, puede omitir los encabezados de la lista.

Con `-p`, sólo puede mostrar los datos del editor determinado. Si no se proporciona, se muestran los datos de todos los editores. Esta opción se puede especificar varias veces.

Con `-s`, puede realizar operaciones en el depósito ubicado en la ruta del sistema de archivos o el URI determinados.

```
list [-F formato] [-H] [-p editor ...] -s uri_o_ruta_dep [mod_ID rec ges err_paq ...]
```

Muestra los paquetes del depósito *uri_o_ruta_dep* que coinciden con los modelos especificados *mod_ID rec ges err_paq*. Si no se especifica ningún modelo, se muestran todos los paquetes del depósito.

En la salida predeterminada, la primera columna contiene el nombre del editor del paquete. La segunda columna contiene el nombre del paquete. La tercera columna es un indicador que muestra el estado del paquete. El valor `o` en la columna de estado indica que el paquete es obsoleto. El valor `r` en la columna de estado indica que se ha cambiado el nombre del paquete, lo cual es una manera de convertirse en obsoleto. La cuarta columna contiene las versiones de lanzamiento y ramificación del paquete. Consulte `pkg(5)` para obtener información sobre las versiones de lanzamiento y ramificación.

Con `-F`, puede especificar una formato de salida alternativo. El valor de *formato* puede ser `tsv` (valores separados por tabuladores), `json` (JavaScript Object Notation, como una sola línea) o `json-formatted` (JavaScript Object Notation, en formato para mejorar la legibilidad).

Con `-H`, puede omitir los encabezados de la lista.

Con `-p`, sólo puede mostrar los paquetes del editor determinado. Si no se proporciona, se muestran los paquetes de todos los editores. Esta opción se puede especificar varias veces.

Con `-s`, puede realizar operaciones en el depósito ubicado en la ruta del sistema de archivos o el URI determinados.

```
rebuild [-p publisher ...] -s repo_uri_or_path [--no-catalog] [--no-index]
```

Descarta toda la información de catálogo, búsqueda, y otra información almacenada en la antememoria encontrada en el depósito y, a continuación, la vuelve a crear según el contenido actual del depósito.

Con `-p`, puede realizar la operación sólo para un editor determinado. Si no se proporciona, o si se especifica el valor especial `all`, la operación se realiza para todos los editores. Esta opción se puede especificar varias veces.

Con `-s`, puede realizar operaciones en el depósito ubicado en la ruta del sistema de archivos o el URI determinados.

Con `--no-catalog`, no puede volver a generar los datos de los paquetes.

Con `--no-index`, no puede volver a generar índices de búsqueda.

`refresh [-p editor ...] -s uri_o_ruta_dep [--no-catalog] [--no-index]`

Cataloga los paquetes nuevos encontrados en el depósito y actualiza todos los índices de búsqueda. Esto está pensado para ser utilizado con la publicación diferida (las opciones `--no-catalog` o `--no-index` de `pkgsend`).

Con `-p`, puede realizar la operación sólo para un editor determinado. Si no se proporciona, o si se especifica el valor especial `all`, la operación se realiza para todos los editores. Esta opción se puede especificar varias veces.

Con `-s`, puede realizar operaciones en el depósito ubicado en la ruta del sistema de archivos o el URI determinados.

Con `--no-catalog`, no puede agregar ningún paquete nuevo.

Con `--no-index`, no puede actualizar índices de búsqueda.

`remove [-n] [-p editor ...] -s uri_o_ruta_dep mod_ID rec ges err_paq ...`

Elimina los paquetes que coinciden con los modelos especificados del depósito, incluidos los archivos a los que hacen referencia que no estén siendo utilizados por ningún otro paquete.

Nota – Se eliminan todos los datos del índice de búsqueda de los editores relacionados.

Este subcomando se puede utilizar sólo con depósitos basados en sistemas de archivos.

Precaución – Esta operación no se puede revertir y no se debe utilizar mientras otros clientes están accediendo al depósito, ya que podría causar fallos durante las operaciones de recuperación.

Con `-n`, puede llevar a cabo una ejecución de prueba de la operación sin realizar cambios en los paquetes. Antes del cierre, se muestra una lista de los paquetes que se van a eliminar.

Con `-p`, sólo puede eliminar los paquetes coincidentes para el editor determinado. Si no se proporciona, se eliminan todos los paquetes coincidentes para todos los editores. Esta opción se puede especificar varias veces.

Con `-s`, puede realizar operaciones en el depósito ubicado en la ruta del sistema de archivos o el URI determinados.

`set [-p editor] -s uri_o_ruta_dep sección/propiedad=[valor] ... or
sección/propiedad=([valor]) ...`

Establece el valor de las propiedades especificadas para el depósito o el editor.

Este subcomando se puede utilizar sólo con depósitos basados en sistemas de archivos.

Con `-p`, sólo puede establecer datos de propiedad para el editor determinado. Si el editor aún no existe, se agrega. El valor especial `all` se puede utilizar para establecer la propiedad de todos los editores.

Con `-s`, puede realizar operaciones en el depósito ubicado en la ruta del sistema de archivos o el URI determinados.

Las propiedades y los valores se pueden especificar de una de las siguientes maneras:

<code>sección/propiedad=</code>	Borra el valor de la propiedad.
<code>sección/propiedad= valor</code>	Sustituye el valor de la propiedad con el valor especificado.
<code>sección/propiedad=(valor1 valor2 valorN)</code>	Sustituye el valor de la propiedad con la lista de valores.

Para las versiones de depósito 3 y 4, se pueden definir las siguientes propiedades:

<code>publisher/prefix</code>	Una cadena que representa el nombre del editor predeterminado. El primer carácter debe estar comprendido entre <code>a</code> y <code>z</code> , <code>A</code> y <code>Z</code> o <code>0</code> y <code>9</code> . El resto de la cadena sólo puede contener los caracteres de <code>0</code> a <code>9</code> , <code>-</code> , <code>.</code> , <code>de</code> a <code>a</code> y <code>de</code> a <code>Z</code> . Este valor indica el editor que se debe utilizar cuando hay más de un paquete de editor, o cuando se publican paquetes en el repositorio, pero no se especifica ningún editor.
-------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Para las versiones de depósito 3 y 4, se pueden definir las siguientes propiedades para cada editor del depósito:

<code>publisher/alias</code>	Una cadena que representa el alias predeterminado que deben utilizar los clientes al agregar un editor mediante los datos de configuración del depósito. El primer carácter debe estar comprendido entre <code>a</code> y <code>z</code> , <code>A</code> y <code>Z</code> o <code>0</code> y <code>9</code> . El resto de la cadena sólo puede contener los caracteres de <code>0</code> a <code>9</code> , <code>-</code> , <code>.</code> , <code>de</code> a <code>a</code> y <code>de</code> a <code>Z</code> .
<code>repository/collection_type</code>	<p>Puede tener el valor <code>core</code> o <code>supplemental</code>, que indica el tipo de paquetes disponibles en este depósito.</p> <p>El tipo <code>core</code> indica que el depósito contiene todas las dependencias declaradas por los paquetes del depósito. El tipo <code>core</code> se utiliza principalmente para depósitos del sistema operativo.</p> <p>El tipo <code>supplemental</code> indica que el depósito contiene paquetes que dependen de paquetes ubicados en otro depósito, o que fueron diseñados para ser utilizados con ellos.</p>

<code>repository/description</code>	Un párrafo de texto sin formato que describe la finalidad y el contenido del depósito.
<code>repository/detailed_url</code>	Un URI que representa la ubicación de un documento (como una página web) que proporciona información adicional sobre el depósito.
<code>repository/legal_uris</code>	Una lista de las ubicaciones (URI) de los documentos que proporcionan más información legal sobre el depósito.
<code>repository/mirrors</code>	Una lista de ubicaciones (URI) de depósitos que contienen una copia del contenido de los paquetes del depósito pero no de los metadatos del paquete.
<code>repository/name</code>	Una cadena de texto sin formato que contiene el nombre del depósito.
<code>repository/origins</code>	Una lista de ubicaciones (URI) de depósitos que contienen una copia completa de los metadatos y el contenido de los paquetes del depósito.
<code>repository/refresh_seconds</code>	Un valor entero que representa el número de segundos que los clientes deben esperar antes de comprobar si existen datos de paquetes actualizados en el depósito después de cada comprobación de actualización.
<code>repository/registration_uri</code>	Un URI que representa la ubicación de un recurso que se debe utilizar para obtener credenciales para acceder al depósito. Un ejemplo es una página web de registro.
<code>repository/related_uris</code>	Una lista de las ubicaciones (URI) de los depósitos que contienen los paquetes en los que los usuarios pueden estar interesados.

Las propiedades que no se documentan aquí, pero que figuran en la salida del subcomando `get`, están reservadas para uso interno y no se deben definir.

versión

Muestra una cadena única que identifica la versión del sistema `pkg(5)`. Los valores producidos por la operación `version` no se pueden clasificar y no son seguros para la comparación más allá igualdad.

Ejemplos

EJEMPLO 1 Creación de un depósito de paquetes

```
$ pkgrepo create /my/repository
```

EJEMPLO 2 Visualización de información

Visualice un resumen de los editores y el número de paquetes de un depósito.

```
$ pkgrepo info -s /my/repository
PUBLISHER PACKAGES STATUS UPDATED
example.com 5 online 2011-07-22T18:09:09.769106Z
$ pkgrepo info -s http://pkg.oracle.com/solaris/release/
PUBLISHER PACKAGES STATUS UPDATED
solaris 3941 online 2010-11-12T19:24:25.967246Z
```

EJEMPLO 3 Reconstrucción de catálogos y datos de búsqueda

Reconstruya los catálogos y los datos de búsqueda del depósito.

```
$ pkgrepo rebuild -s /my/repository
```

EJEMPLO 4 Refrescamiento de catálogos y datos de búsqueda

Refresque los catálogos y los datos de búsqueda del depósito.

```
$ pkgrepo refresh -s /my/repository
$ pkgrepo refresh -s http://example.com/repository
```

EJEMPLO 5 Visualización de todas las propiedades del depósito

```
$ pkgrepo get -s /my/repository
SECTION PROPERTY VALUE
publisher prefix ""
repository version 4
$ pkgrepo get -s http://pkg.oracle.com/solaris/release/
SECTION PROPERTY VALUE
publisher prefix solaris
repository version 4
```

EJEMPLO 6 Visualización de todas las propiedades del editor

```
$ pkgrepo get -s http://pkg.oracle.com/solaris/release/ -p all
PUBLISHER SECTION PROPERTY VALUE
solaris publisher alias
solaris publisher prefix solaris
solaris repository collection-type core
solaris repository description This\ repository\ serves\ the\ Oracle\
Solaris\ 11\ Package\ repository.
solaris repository legal-uris ()
solaris repository mirrors (http://pkg-cdn1.oracle.com/solaris.release/)
solaris repository name Oracle\ Solaris\ 11\ Package\ Repository
solaris repository origins ()
solaris repository refresh-seconds
solaris repository registration-uri ""
solaris repository related-uris ()
```

EJEMPLO 7 Establecimiento del editor predeterminado

```
$ pkgrepo set -s /my/repository publisher/prefix=example.com
```

EJEMPLO 8 Establecimiento de una propiedad de editor

```
$ pkgrepo set -s /my/repository -p example.com \
repository/origins=http://example.com/repository
```

EJEMPLO 9 Adición de un editor nuevo al depósito

```
$ pkgrepo add-publisher -s /my/repository example.com
```

- Estado de salida** Se devolvieron los siguientes valores de salida:
- 0 El comando se ejecutó correctamente.
 - 1 Se produjo un error.
 - 2 Se especificaron opciones de línea de comandos no válidas.
 - 3 Se solicitaron varias operaciones, pero sólo algunas se realizaron correctamente.
 - 4 No se realizaron cambios, no se requiere ninguna acción.
 - 99 Se produjo una excepción no anticipada.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkg\(1\)](#), [pkgrecv\(1\)](#), [pkgsend\(1\)](#), [pkg.depotd\(1m\)](#), [pkg\(5\)](#)

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

Nombre pkgsend – cliente de publicaciones de Image Packaging System

Sinopsis /usr/bin/pkgsend [*opciones*] *comando* [*opciones_comando*] [*operandos*]

```
/usr/bin/pkgsend generate [-T modelo] [--target archivo]
                        origen ...
```

```
/usr/bin/pkgsend publish [-b paquete ...] [-d origen ...]
                        [-s uri_o_ruta_dep] [-T modelo] [--no-catalog]
                        [manifiesto ...]
```

Descripción pkgsend permite la publicación de paquetes nuevos y versiones de paquetes nuevas en un depósito de empaquetado de imágenes mediante manifiestos de paquetes. Para crear o gestionar depósitos, consulte pkgrepo(1). Para crear archivos de paquetes a partir de paquetes de un depósito existente, consulte pkgrecv(1). Para obtener más información sobre manifiestos de paquetes, consulte pkg(5).

Después de una operación pkgsend, ejecute pkgrepo refresh o pkgrepo rebuild en el repositorio para crear índices de búsqueda.

Opciones Se admiten las siguientes opciones:

--help o -? Muestra un mensaje de uso.

Subcomandos Se admiten los subcomandos siguientes:

```
generate [-T modelo] [--target archivo] origen ...
```

Lee cada *origen* (como un paquete SVR4, un directorio, o un archivo tar) y emite el manifiesto que describe el *origen* en stdout. En el manifiesto de salida, las acciones de file y dir tienen el propietario establecido en root y el grupo establecido en bin.

Al manifiesto de salida se le puede realizar anotaciones, se le puede agregar o analizar dependencias usando pkgdepend(1), y se le puede verificar la exactitud mediante pkglint(1) antes de pasar al subcomando publish.

Se admiten los siguientes orígenes:

- Paquetes SVR4 de formato de sistema de archivos
- Paquetes SVR4 de formato de secuencia de datos
- Archivos tar
- Directorios

Si el nombre de base de los archivos del origen coinciden con los modelos especificados con -T, el indicador de fecha y hora del archivo se agrega a la acción para ese archivo. El *modelo* utiliza reglas de coincidencia de shell:

- * Coincide con todo.
- ? Sustituye a un carácter cualquiera.
- [*seq*] Coincide con cualquier carácter incluido en *seq*.

! [*seq*] Coincide con cualquier carácter no incluido en *seq*.

Cuando el origen especificado es un directorio, no existe una manera clara de distinguir una acción `file` de una acción `hardlink` cuando hay varios nombres de ruta para un único inode. Generalmente, el primer nombre de ruta que se encuentra en el recorrido del sistema de archivos se trata como un archivo y el resto como enlaces físicos. Esto puede ser arbitrario, según la implementación del sistema de archivos. Para especificar qué nombres de ruta se deben tratar como archivos, pase cada nombre de ruta como argumento a la opción `--target`. Esta opción no tiene ningún efecto sobre otros tipos de orígenes, porque éstos son capaces de expresar qué nombres de ruta son archivos y qué nombres de ruta son enlaces físicos.

Cuando se proporcionan paquetes SVR4 como origen, `pkgsend` comprueba que no haya archivos con secuencias de comandos de acción de clase ni secuencias de comandos `preinstall`, `postinstall`, `preremove` o `postremove`. Se crea una excepción para cualquier manifiesto SMF instalado con la clase `manifest`. `BASEDIR` se elimina de todas las rutas reubicables.

El parámetro SVR4 `DESC` se convierte en un valor `pkg.description`. El parámetro SVR4 `NAME` se convierte en un valor `pkg.summary`.

```
publish [-b paquete ...] [-d origen ...] [-s uri_o_ruta_dep] [-T modelo]  
[--no-catalog] [manifiesto ...]
```

Publica un paquete utilizando los manifiestos de paquete especificados en el depósito de paquetes de destino, y recupera archivos para el paquete a partir de los orígenes proporcionados. Si se especifican varios manifiestos, se unen en el orden proporcionado. Si no se especifica un manifiesto, éste se lee desde `stdin`.

Con `-b`, puede agregar el paquete especificado a la lista de orígenes en los que se deben buscar los archivos del manifiesto. Los paquetes son orígenes como archivos `tar` y paquetes SVR4. Si esta opción se especifica varias veces, los orígenes se buscan en el orden en el que aparecen en la línea de comandos. Si se especifican `-b` y `-d`, los orígenes `-d` se buscan en primer lugar. Para obtener una descripción de los paquetes admitidos, y de cómo utilizarlos, consulte el subcomando `generate` descrito anteriormente.

Con `-d`, puede agregar el directorio especificado a la lista de orígenes en los que se deben buscar los archivos del manifiesto. Si esta opción se especifica varias veces, los orígenes se buscan en el orden en el que aparecen en la línea de comandos. Para obtener una descripción de los orígenes admitidos, y de cómo utilizarlos, consulte el subcomando `generate` descrito anteriormente.

Con `-s`, puede publicar el paquete en el depósito ubicado en el URI o la ruta del sistema de archivos determinados. Consulte la sección “Notas” a continuación, para obtener más información sobre las restricciones y las sugerencias para la publicación. Consulte también la sección “Variables de entorno”.

Con `--no-catalog`, no puede agregar el paquete al catálogo del editor. Se recomienda usar esta opción cuando se publiquen varios paquetes a la vez, ya que las actualizaciones de los catálogos del editor se deben realizar en serie. Una vez finalizada la publicación, se puede utilizar el subcomando `refresh` de `pkgrepo` (1) para agregar los paquetes nuevos a los respectivos catálogos de editor.

Para todas las demás opciones, consulte el subcomando `generate` descrito anteriormente, para ver el uso y sus efectos.

Variables del entorno `PKG_REPO` La ruta o el URI del depósito de destino.

Ejemplos **EJEMPLO 1** Generación y publicación de un paquete

Cree un paquete mediante `pkgsend generate` y publíquelo.

```
$ pkgsend generate /path/to/proto > /path/to/manifests/foo.p5m
```

Agregue el FMRI de paquete para el editor `example.com` al principio de `foo.p5m`.

```
set name=pkg.fmri value=pkg://example.com/foo@1.0
```

El manifiesto resultante debe tener este aspecto:

```
set name=pkg.fmri value=pkg://example.com/foo@1.0
dir group=sys mode=0755 owner=root path=usr
dir group=bin mode=0755 owner=root path=usr/bin
file usr/bin/foo group=bin mode=0555 owner=root path=usr/bin/foo

$ pkgsend publish -s http://example.com:10000 -d /path/to/proto \
/path/to/manifests/foo.p5m
```

EJEMPLO 2 Creación y publicación de un paquete trivial

Cree un manifiesto para el editor `example.com` que contenga las líneas siguientes:

```
set name=pkg.fmri value=pkg://example.com/foo@1.0-1
file /exdir/foo mode=0555 owner=root group=bin path=/usr/bin/foo
```

Publique el paquete:

```
$ pkgsend publish -s http://example.com:10000 -d /exdir
```

EJEMPLO 3 Uso de un manifiesto preexistente

Publique un paquete con una publicación basada en sistema de archivos y un manifiesto preexistente.

```
$ pkgsend publish -s /tmp/example_repo -d /tmp/pkg_files \
/tmp/pkg_manifest
```

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 El comando se ejecutó correctamente.
- 1 Se produjo un error.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 99 Se produjo una excepción no anticipada.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkgdepend\(1\)](#), [pkgrepo\(1\)](#), [pkg.depotd\(1m\)](#), [pkg\(5\)](#)

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

Notas Debido a limitaciones de protocolo de publicación, la publicación basada en sistema de archivos se debe utilizar al publicar archivos de paquetes individuales cuyo tamaño sea superior a los 128 MB. La publicación basada en sistema de archivos también se recomienda cuando se necesita control de acceso para un depósito.

Cuando se utiliza la publicación basada en sistema de archivos, una vez completada la publicación, se debe reiniciar cualquier proceso `pkg.depotd` que esté prestando servicio al depósito de destino, para que los cambios se reflejen en la interfaz web o en las respuestas de las búsquedas. Consulte `pkg.depotd(1M)` para obtener más información.

Nombre pkgsign – utilidad de firma de Image Packaging System

Sinopsis /usr/bin/pkgsign [-a *algoritmo_hash*]
 [-c *ruta_para_certificado_firma*]
 [-i *ruta_para_certificado_intermedio*] ...
 [-k *ruta_para_clave_privada*] [-n] -s *ruta_o_uri*
 [--help] [--no-index] [--no-catalog]
 (*fmri|modelo*) ...

Descripción pkgsign actualiza el manifiesto para los FMRI determinados en el lugar en el depósito mediante la adición de una acción de firma con la clave y los certificados especificados. El paquete modificado conserva el indicador de fecha y hora original.

Opciones Se admiten las siguientes opciones:

Con -a, puede utilizar el algoritmo de firma *algoritmo_hash* en lugar del valor predeterminado. El algoritmo de firma predeterminado es *rsa-sha256*. Los algoritmos de firma admitidos son *rsa-sha256*, *rsa-sha384*, *rsa-sha512*, *sha256*, *sha384* y *sha512*. Un algoritmo de firma que sólo especifica un algoritmo hash hace que el valor de firma sea el hash del manifiesto del paquete. Un algoritmo de firma que especifica *rsa* y un algoritmo hash hace que el valor de firma sea el hash del manifiesto firmado con la clave privada proporcionada (consulte las opciones -c y -k).

Con -c, puede agregar el certificado *ruta_para_certificado_firma* como el certificado que se utilizará al verificar el valor de la firma de la acción. La opción -c sólo se puede utilizar con la opción -k.

Con -i, puede agregar el certificado *ruta_para_certificado_intermedio* como un certificado que se utilizará al validar el certificado *ruta_para_certificado_firma* proporcionado como un argumento para -c. Se pueden proporcionar varios certificados especificando -i varias veces.

Con -k, puede utilizar la clave privada almacenada en *ruta_para_clave_privada* para firmar el manifiesto. La opción -k sólo se puede utilizar con la opción -c. Si no se especifica -k, el valor de firma es el hash del manifiesto.

Con -n, puede realizar una ejecución de prueba, que no cambia el depósito de ninguna manera.

Con -s, puede firmar paquetes del depósito en *ruta_o_uri*.

Con --help, puede visualizar un mensaje de uso.

Con --no-index, no puede actualizar los índices de búsqueda del repositorio una vez que el manifiesto firmado se ha publicado nuevamente.

Con --no-catalog, no puede actualizar el catálogo del depósito una vez que el manifiesto firmado se ha publicado nuevamente.

Ejemplos EJEMPLO 1 Firma con el valor hash del manifiesto

Firme un paquete publicado en `http://localhost:10000` con el valor hash del manifiesto. A menudo esto es útil para realizar pruebas.

```
$ pkgsign -s http://localhost:10000 -a sha256 \
example_pkg@1.0,5.11-0:20100626T030108Z
```

EJEMPLO 2 Firma con una clave y un certificado

Firme un paquete publicado en el depósito de archivos en `/foo/bar` mediante `rsa-sha384` para aplicar hash al manifiesto y firmarlo. La clave de firma está en `/key/usr2.key`, su certificado asociado está en `/key/usr2.cert` y un certificado necesario para validar el certificado está en `/icerts/usr1.cert`.

```
$ pkgsign -s file:///foo/bar/ -a rsa-sha384 \
-k /key/usr2.key -c /key/usr2.cert -i /icerts/usr1.cert \
example_pkg@1.0,5.11-0:20100626T031341Z
```

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 El comando se ejecutó correctamente.
- 1 Se produjo un error.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 3 Se solicitaron varias operaciones, pero sólo algunas se realizaron correctamente.
- 99 Se produjo una excepción no anticipada.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkg\(1\)](#), [pkgrecv\(1\)](#), [pkgsend\(1\)](#), [pkgrepo\(1\)](#), [pkg\(5\)](#)

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

Nombre pm-updatemanager – aplicación para actualizar paquetes

Sinopsis /usr/bin/pm-updatemanager [*opciones*]
 /usr/bin/pm-updatemanager [-hdR] [--help] [--debug]
 [--image-dir *directorio*]

Descripción pm-updatemanager comprueba si existen actualizaciones disponibles de los paquetes instalados en el sistema y las instala.

Nota – Si los paquetes package/pkg, package/pkg/package-manager o package/pkg/update-manager deben ser actualizados, pm-updatemanager, primero, actualiza estos paquetes y, luego, reinicia para llevar a cabo las actualizaciones restantes.

Opciones Se admiten las siguientes opciones:

-h o --help	Muestra un mensaje de uso.
-d o --debug	Ejecuta pm-updatemanager en el modo de depuración.
-R o --image-dir <i>directorio</i>	Realiza operaciones en la imagen que tiene raíz en el <i>directorio</i> , en lugar de la imagen detectada automáticamente.

Ejemplos EJEMPLO 1 Actualización de la imagen actual

Invoque pm-updatemanager en la imagen actual. Esto permite comprobar si existen actualizaciones disponibles para los paquetes instalados en la imagen actual e instalarlas todas.

```
$ /usr/lib/pm-launch pm-updatemanager
```

Este es el mismo comando que invoca la opción de menú de escritorio Sistema>Administración>Administrador de actualizaciones.

EJEMPLO 2 Actualización de una imagen especificada

Invoque pm-updatemanager en la imagen almacenada en /aux0/example_root.

```
$ /usr/lib/pm-launch pm-updatemanager -R /aux0/example_root
```

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 Ejecución correcta.
- 1 Se produjo un error.
- 2 Se especificaron opciones de línea de comandos no válidas.

Atributos Consulte attributes(5) para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg/update-manager

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Estabilidad de interfaz	Sin asignar.

Véase también [packagemanager\(1\)](#), [pkg\(1\)](#), [pkg\(5\)](#)

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

Notas Al realizar operaciones en una imagen que no le pertenece, debe invocar a `pm-updatemanager` con suficiente privilegio. Normalmente invoca a `pm-updatemanager` mediante `/usr/lib/pm-launch` en estas circunstancias.

REFERENCIA

Comandos de administración del sistema

Nombre pkg.depotd – Servidor de depósitos de Image Packaging System

Sinopsis /usr/lib/pkg.depotd [-a *address*] [-d *inst_root*] [-p *port*]
 [-s *threads*] [-t *socket_timeout*] [--add-content]
 [--cfg] [--content-root] [--debug *feature_list*]
 [--disable-ops=*op*[/1][,...]]
 [--log-access] [--log-errors] [--mirror]
 [--proxy-base *url*] [--readonly] [--rebuild]
 [--ssl-cert-file] [--ssl-dialog] [--ssl-key-file]
 [--writable-root]

Descripción pkg.depotd es el servidor de depósitos de Image Packaging System. Proporciona acceso de red a los datos incluidos en un repositorio de paquetes. Los clientes que no admiten acceso directo a un repositorio por medio del sistema de archivos, o para los cuales el acceso de red es el único método de transporte disponible o preferido, suelen utilizar el depósito de paquetes.

Los clientes, como pkg(1), el cliente de recuperación, pueden recuperar una lista de paquetes y metadatos de paquetes desde un repositorio directamente o mediante el servidor de depósitos. pkgsend(1), el cliente de publicación, puede enviar versiones nuevas de paquetes a un repositorio directamente o mediante el servidor de depósitos. pkgrepo(1) se puede utilizar para crear repositorios para usar con el servidor de depósitos o para gestionarlos directamente o mediante el servidor de depósitos.

pkg.depotd se ejecuta habitualmente como un servicio en el sistema. Los desarrolladores de software y paquetes pueden ejecutar copias privadas para la realización de pruebas.

El depósito no proporciona ningún método de control de acceso propio. De manera predeterminada, todos los clientes que se pueden conectar pueden leer los datos de todos los paquetes y publicar versiones nuevas de paquetes. La excepción es que cuando se ejecuta en la utilidad de gestión de servicios (SMF), el valor predeterminado es que se ejecute en modo de sólo lectura. En la sección "Notas" que aparece a continuación, se describen algunas de las prácticas recomendadas para el mantenimiento de un servidor de depósitos público con contenido en evolución.

Propiedades Smf El servidor pkg.depot se suele configurar mediante las propiedades de smf(5) asociadas a su servicio. Se reconocen las siguientes propiedades:

pkg/address	(net_address). La dirección IP en la que se escuchan conexiones. El valor predeterminado es 0.0.0.0 (INADDR_ANY), que escucha en todas las interfaces activas. Para escuchar en todas las interfaces IPv6 activas, utilice ::. Sólo se utiliza el primer valor.
pkg/content_root	(astring). La ruta del sistema de archivos en la que la instancia debe buscar su contenido web estático y de otro tipo. El valor predeterminado es /usr/share/lib/pkg.

pkg/debug	(astring). Una lista separada por comas de funciones de depuración para activar. Los valores posibles son: headers Registra los encabezados de cada solicitud en el registro de errores.
pkg/disable_ops	(astring). Una lista separada por comas de operaciones que se deben desactivar para el servidor de depósitos. Las operaciones se indican de la siguiente manera: <i>operation[/version]</i> (por ejemplo, <i>catalog</i> o <i>search_1</i>).
pkg/image_root	(astring). La ruta a la imagen cuya información de archivo se utilizará como antememoria para los datos de archivo.
pkg/inst_root	(astring). La ruta del sistema de archivos en la que la instancia debe buscar los datos de su repositorio. Requerido a menos que se haya proporcionado <i>file_root</i> o <i>PKG_REPO</i> . El valor predeterminado es <i>/var/pkgrepo</i> .
pkg/log_access	(astring). El destino de cualquier información relacionada con el acceso registrada por el proceso del depósito. Los valores posibles son: <i>stderr</i> , <i>stdout</i> , <i>none</i> o un nombre de ruta absoluta. El valor predeterminado es <i>stdout</i> si <i>stdout</i> es <i>tty</i> . Si <i>stdout</i> no es <i>tty</i> , el valor predeterminado es <i>none</i> .
pkg/log_errors	(astring). El destino de cualquier error u otra información registrados por el proceso del depósito. Los valores posibles son: <i>stderr</i> , <i>stdout</i> , <i>none</i> o un nombre de ruta absoluta. El valor predeterminado es <i>stderr</i> .
pkg/mirror	(boolean). Establece si se utiliza el modo de reflejo de paquetes. Si es <i>true</i> , se desactivan las operaciones de publicación y metadatos, y sólo se proporciona una interfaz de usuario del navegador limitada. Esta propiedad no puede ser <i>true</i> cuando la propiedad <i>pkg/readonly</i> es <i>true</i> . El valor predeterminado es <i>false</i> .
pkg/port	(count). El número de puerto en el que la instancia debe escuchar solicitudes de paquetes entrantes. Si no se proporcionó información sobre la clave y el

	certificado SSL, el valor predeterminado es 80; de lo contrario, el valor predeterminado es 443.				
pkg/proxy_base	(uri). Esto cambia la URL base para el servidor de depósitos y es muy útil cuando se ejecuta en segundo plano con Apache u otro servidor web en una configuración de proxy inverso.				
pkg/readonly	(boolean). Establece si las operaciones de modificación, como las iniciadas por pkgsend(1), se encuentran desactivadas. Las operaciones de recuperación todavía están disponibles. Esta propiedad no puede ser true cuando la propiedad pkg/mirror es true. El valor predeterminado es true.				
pkg/socket_timeout	(count). El número máximo de segundos durante los que el servidor debe esperar una respuesta de un cliente antes de cerrar la conexión. El valor predeterminado es 60.				
pkg/sort_file_max_size	(count). El tamaño máximo del archivo de clasificación del indexador. Se utiliza para limitar la cantidad de RAM que utiliza el depósito para la indexación, o para aumentarla para acelerar el proceso.				
pkg/ssl_cert_file	(astring). El nombre de ruta absoluta para un archivo de certificado con codificación PEM. El valor predeterminado es none. Esta propiedad se debe usar con ssl_key_file. El depósito sólo responde a las solicitudes SSL si se proporcionan ssl_cert_file y /ssl_key_file.				
pkg/ssl_dialog	(astring). Especifica el método que se debe utilizar para obtener la frase de contraseña utilizada para descifrar ssl_key_file. Los valores posibles son: <table> <tr> <td>builtin</td><td>Solicita la frase de contraseña. Éste es el valor predeterminado.</td></tr> <tr> <td>exec:/path/to/program</td><td>Ejecuta el programa externo especificado para obtener la frase de contraseña. El primer argumento para el programa es '' y está</td></tr> </table>	builtin	Solicita la frase de contraseña. Éste es el valor predeterminado.	exec:/path/to/program	Ejecuta el programa externo especificado para obtener la frase de contraseña. El primer argumento para el programa es '' y está
builtin	Solicita la frase de contraseña. Éste es el valor predeterminado.				
exec:/path/to/program	Ejecuta el programa externo especificado para obtener la frase de contraseña. El primer argumento para el programa es '' y está				

	reservado. El segundo argumento para el programa es el número de puerto del servidor. La frase de contraseña se imprime en stdout.
smf:fmri	Intenta recuperar el valor de la propiedad pkg_secure/ssl_key_passphrase de la instancia de servicio relacionada con el FMRI.
pkg/ssl_key_file	(astring). El nombre de ruta absoluta para un archivo de claves privadas con codificación PEM. Esta propiedad se debe utilizar junto con la propiedad ssl_cert_file. El depósito sólo responde a las solicitudes SSL si se proporcionan /ssl_key_file y ssl_cert_file.
pkg/threads	(count). El número de subprocesos iniciados para atender solicitudes. El valor predeterminado es 60. Sólo es adecuado para implementaciones pequeñas. Este valor debe equivaler aproximadamente al número de clientes simultáneos multiplicado por 20. El valor máximo de threads es 5.000.
pkg/writable_root	(astring). La ruta del sistema de archivos de un directorio en el que el programa tiene acceso de escritura. Se utiliza con la opción -readonly para activar el servidor de depósitos con el fin de crear archivos, como índices de búsqueda, sin necesidad de disponer de acceso de escritura a la información del paquete.
pkg_secure/ssl_key_passphrase	(astring). La contraseña que se utilizará para descifrar pkg/ssl_key_file. Este valor está protegido con autorización de lectura mediante el atributo solaris.smf.read.pkg-server.
La presentación y el comportamiento de la interfaz de usuario del navegador (BUI) del servidor de depósitos se controlan mediante las siguientes propiedades:	
pkg_bui/feed_description	(astring). Un párrafo descriptivo para la fuente RSS/Atom.
pkg_bui/feed_icon	(astring). El nombre de ruta de una imagen pequeña utilizada para representar visualmente la fuente RSS/Atom.

	El nombre de ruta se debe relacionar con <code>content_root</code> . El valor predeterminado es <code>web/_themes/pkg-block-icon.png</code> .
<code>pkg_bui/feed_logo</code>	(<code>string</code>). El nombre de ruta de una imagen de gran tamaño que se utilizará para marcar o identificar visualmente la fuente RSS/Atom. Este valor se debe relacionar con <code>content_root</code> . El valor predeterminado es <code>web/_themes/pkg-block-icon.png</code> .
<code>pkg_bui/feed_name</code>	(<code>string</code>). Un nombre corto y descriptivo para las fuentes RSS/Atom generadas por el depósito que sirve al repositorio. El valor predeterminado es "package repository feed".
<code>pkg_bui/feed_window</code>	(<code>count</code>). La cantidad de horas transcurridas antes de que se generara por última vez la fuente para el repositorio, para incluirla al generar la fuente.

El depósito de paquetes también puede actuar como servidor de reflejo para imágenes de clientes locales desde `pkg(5)`. Esto permite a los clientes que comparten una subred en una LAN reflejar sus antememorias de archivo. Los clientes pueden descargar archivos entre ellos, y, de esta manera, reducir la carga del servidor de depósitos de paquetes. Esta funcionalidad está disponible como un servicio de depósito alternativo configurado mediante `smf(5)`. Utiliza `mDNS` y `dns-sd` para la detección de servicio.

El reflejo `mDNS` generalmente se configura mediante las propiedades `smf(5)` asociadas a su servicio. Se reconocen las siguientes propiedades:

<code>pkg/image_root</code>	(<code>string</code>). La ruta a la imagen cuya información de archivo se utilizará como antememoria para los datos de archivo. El valor predeterminado es <code>/</code> .
<code>pkg/port</code>	(<code>count</code>). El número de puerto en el que la instancia debe escuchar solicitudes de paquetes entrantes. El valor predeterminado es 80.

Opciones `pkg.depotd` puede leer la información de configuración base desde un archivo o desde los datos de la propiedad de una instancia de servicio `smf(5)` existente.

<code>--cfg source</code>	El nombre de ruta del archivo que se utilizará al leer y escribir los datos de configuración, o una cadena con el formato <code>smf:fmri</code> , donde <i>fmri</i> es el identificador de recurso de gestión de errores (FMRI) del servicio de la instancia de la cual se leerán los datos de configuración. Consulte "Configuración de depósito" a continuación para obtener información sobre el formato del archivo especificado.
---------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Si no hay ningún origen de configuración preexistente disponible, o si se deben sustituir los valores leídos desde el archivo de configuración proporcionado mediante `--cfg`, se pueden utilizar las siguientes opciones para modificar el comportamiento predeterminado del servidor de depósitos:

<code>-a address</code>	Consulte pkg/address arriba.
<code>--content-root root_dir</code>	Consulte pkg/content_root arriba.
<code>-d inst_root</code>	Consulte pkg/inst_root arriba.
<code>--debug-features</code>	Consulte pkg/debug arriba.
<code>--disable-ops op_list</code>	Consulte pkg/disable_ops arriba.
<code>--image-root path</code>	Consulte pkg/image_root arriba.
<code>--log-access dest</code>	Consulte pkg/log_access arriba.
<code>--log-errors dest</code>	Consulte pkg/log_errors arriba.
<code>--mirror</code>	Consulte pkg/mirror arriba.
<code>-p port</code>	Consulte pkg/port arriba.
<code>--proxy-base url</code>	Consulte pkg/proxy_base arriba. Esta opción se ignora si se proporciona un valor vacío.
<code>--readonly</code>	Consulte pkg/readonly arriba.
<code>-s threads</code>	Consulte pkg/threads arriba.
<code>-s-ort-file-max-size bytes</code>	Consulte pkg/sort_file_max_size arriba.
<code>--ssl-cert-file source</code>	Consulte pkg/ssl_cert_file arriba.
<code>--ssl-dialog type</code>	Consulte pkg/ssl_dialog arriba.
<code>--ssl-key-file source</code>	Consulte pkg/ssl_key_file arriba.
<code>-t socket_timeout</code>	Consulte pkg/socket_timeout arriba.
<code>--writable-root path</code>	Consulte pkg/writable_root arriba.

Se proporciona funcionalidad de gestión y administrativa adicional para repositorios de paquetes mediante `pkgrepo(1)`.

Configuración de depósito Cuando se proporciona un archivo de configuración (en lugar de un FMRI `smf(5)`) mediante la opción `--cfg`, el servidor de depósitos lee y escribe todos los datos de configuración en un formato de texto simple. Los datos de configuración se describen en la sección “Propiedades SMF” arriba. Los datos de configuración están compuestos por secciones, que empiezan con el

encabezado *[section]* y siguen con las entradas *name = value*. Las continuaciones están en el estilo de RFC 822. Los valores se pueden dividir en varias líneas comenzando líneas de continuación con un espacio en blanco.

Los valores requeridos que no se hayan proporcionado en el archivo de configuración se deben proporcionar mediante la opción incluida en “Opciones” anteriormente. Un archivo de configuración de ejemplo podría tener este aspecto:

```
[pkg]
port = 80
inst_root = /export/repo

[pub_example_com]
feed_description = example.com's software
update log
```

Ejemplos EJEMPLO 1 Activación del servidor de depósitos

```
# svcadm enable application/pkg/server
```

EJEMPLO 2 Cambio del puerto de escucha del servidor

```
# svccfg -s application/pkg/server setprop pkg/port = 10000
# svcadm refresh application/pkg/server
# svcadm restart application/pkg/server
```

EJEMPLO 3 Activación del reflejo

```
# svcadm enable application/pkg/dynamic-mirror
```

Variables del entorno

PKG_REPO	Especifica el directorio que contiene el repositorio que se debe atender. Este valor se ignora si se especifica -d.
PKG_DEPOT_CONTENT	Especifica el directorio que contiene contenido estático atendido por el depósito. Los archivos que aparecen a continuación, en “Archivos”, deben estar presentes en este directorio, aunque su contenido puede diferir del contenido predeterminado suministrado.

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 Operación correcta.
- 1 Se produjo un error.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 99 Se produjo una excepción no anticipada.

Archivos /usr/share/lib/pkg Ubicación predeterminada del contenido de la presentación. Modifique pkg/content_root para seleccionar una ubicación alternativa.

Atributos Consulte attributes(5) para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también dns-sd(1M), mDNSd(1M), [pkg\(1\)](#), [pkgrepo\(1\)](#), [pkgsend\(1\)](#), syslogd(1M), [smf\(5\)](#)

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

Notas El servicio pkg.depotd se gestiona mediante SMF con el identificador de servicio svc:/application/pkg/server.

El servicio de reflejo mDNS se gestiona mediante SMF con el identificador de servicio svc:/application/pkg/dynamic-mirror.

Para controlar el acceso de lectura al depósito, puede utilizar un proxy inverso HTTP en combinación con métodos de autenticación, como acceso de certificado SSL basado en cliente, que pkg(1) admite de manera nativa.

Los cambios en la configuración o los cambios en los datos de los paquetes mediante operaciones del proceso del servidor de depósitos para que los cambios se puedan reflejar en las operaciones y la salida. Utilice uno de los métodos siguientes para reiniciar el proceso del servidor de depósitos:

- Use svcadm(1M) para reiniciar la instancia application/pkg/server.
- Envíe una señal SIGUSR1 al proceso del servidor de depósitos mediante kill(1). Con esto se ejecutará un “reinicio normal” que dejará el proceso intacto, pero volverá a cargar todos los datos de configuración, paquete y búsqueda:

```
# kill -USR1 pid
```

Nombre pkg.sysrepo – Configuración del repositorio del sistema Image Packaging System

Síntaxis pkg.sysrepo -p *port* [-c *cache_dir*] [-s *cache_size*]
[-w *http_proxy*] [-W *https_proxy*]

Descripción pkg.sysrepo se utiliza para generar los archivos de configuración para el repositorio del sistema Image Packaging System (IPS). pkg.sysrepo es llamado por el servicio de la utilidad de gestión de servicios (SMF) svc:/application/pkg/system-repository. Se deben realizar cambios en la configuración de las propiedades del servicio SMF.

El repositorio del sistema es responsable de proporcionar el acceso a los repositorios de paquetes configurados en una imagen de referencia mediante un proxy centralizado. Cualquier cliente que esté configurado para utilizar el repositorio del sistema puede ver de inmediato los cambios de configuración del editor realizados en esa imagen de referencia.

El repositorio del sistema se utiliza principalmente en la zona global para permitir que las zonas no globales accedan a los repositorios configurados en la zona global. Los servicios SMF svc:/application/pkg/zones-proxy y svc:/application/pkg/zones-proxy-client son los responsables de facilitar el transporte entre las zonas no globales y la zona global. Este transporte sólo es utilizado por pkg(5).

Tenga en cuenta que sólo se admiten los repositorios de archivos http, https y v4. Los repositorios de archivos basados en p5p y los formatos de repositorios de archivos más antiguos no se admiten. Consulte pkgrepo(1) para obtener más información sobre versiones de repositorios.

Opciones Se admiten las siguientes opciones:

-c *dir_antem* La ruta absoluta a un directorio que debe ser utilizado por el repositorio del sistema para almacenar en la antememoria las respuestas de los editores configurados.

De manera predeterminada, se utiliza una antememoria de archivos. Sin embargo, se puede utilizar el valor especial *memory* para indicar que se debe utilizar una antememoria de memoria. El valor especial *None* se puede utilizar para indicar que el repositorio del sistema no debe realizar el almacenamiento en antememoria. Este valor se debe configurar mediante la propiedad SMF *config/cache_dir*.

-p *port* El puerto que el repositorio del sistema debe utilizar para escuchar solicitudes. Este valor se debe configurar mediante la propiedad SMF *config/port*.

-s *cache_size* Un valor entero en megabytes, que define el tamaño máximo de la antememoria del repositorio del sistema. Este valor se debe configurar mediante la propiedad SMF *config/cache_max*.

- `-w http_proxy` Una cadena con el formato *scheme://hostname[:port]*, que define un proxy web que el repositorio del sistema puede usar para acceder a los repositorios de paquetes basados en http. Este valor se puede configurar mediante la propiedad SMF `config/http_proxy`.
- `-w https_proxy` Una cadena con el formato *scheme://hostname[:port]*, que define un proxy web que el repositorio del sistema puede usar para acceder a los repositorios de paquetes basados en https. Este valor se puede configurar mediante la propiedad SMF `config/https_proxy`.

Ejemplos EJEMPLO 1 Activación del repositorio del sistema

```
$ svcadm enable svc:/application/pkg/system-repository
```

Estado de salida Se devolvieron los siguientes valores de salida:

- 0 El comando se ejecutó correctamente.
- 1 El comando no pudo escribir una configuración válida.
- 2 Se especificaron opciones de línea de comandos no válidas.
- 99 Se produjo una excepción no anticipada.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkg\(1\)](#), [pkg.depotd\(1m\)](#), [pkg\(5\)](#)

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

REFERENCIA

Estándares, entornos y macros

Nombre	pkg – Image Packaging System
Descripción	El sistema de empaquetado de imágenes, pkg(5), es una estructura que facilita la administración del ciclo de vida del software (instalación, actualización y eliminación). El sistema de empaquetado de imágenes administra el software en unidades de paquetes, que son recopilaciones de acciones, definidas por un conjunto de pares de clave y valor y, posiblemente, una carga útil de datos. En muchos casos, las acciones son archivos que se encuentran en un sistema de archivos, pero también representan otros objetos instalables, como controladores, servicios y usuarios.
Versiones e identificadores de recursos de gestión de errores de paquetes	<p>Cada paquete es representado por un identificador de recursos de gestión de errores (FMRI) con el esquema pkg: . El FMRI completo de un paquete está compuesto por el esquema, un editor, el nombre del paquete y una cadena de versión con el formato siguiente:</p> <pre>pkg://solaris/system/library/c++-runtime@0.5.11,5.11-0.174.0.0.0.0:20110921T190358Z</pre> <p>solaris es el editor. system/library/c++-runtime es el nombre del paquete. Si bien el espacio de nombres es jerárquico y tiene una profundidad arbitraria, no se aplica una contención; el nombre es básicamente arbitrario. La información del editor es opcional, pero debe estar precedida por pkg: // si está presente. Si un FMRI incluye al editor a menudo se lo denomina "completo". Si la información del editor no está presente, el nombre del paquete debe estar precedido por pkg: /.</p> <p>A menudo los clientes de empaquetado permiten que el esquema de un FMRI se omita si no contiene información del editor. Por ejemplo, pkg: /system/library/c++-runtime se puede escribir como system/library/c++-runtime. Si el esquema se omite, los clientes también permiten la omisión de todos los componentes de un nombre de paquete excepto el último para favorecer la coincidencia. Por ejemplo, system/library/c++-runtime se podría escribir como library/c++-runtime o c++-runtime, que coincidiría con los paquetes denominados c++-runtime o con los nombres de paquetes que finalicen en /c++-runtime.</p> <p>Un nombre de editor identifica a una persona, un grupo de personas o una organización como el origen de uno o varios paquetes. Para evitar conflictos entre nombres de editores y ayudar a identificar al editor, se aconseja utilizar como nombre de editor un nombre de dominio que represente a la entidad que publica los paquetes.</p> <p>La versión se indica a continuación del nombre del paquete, separada por un signo arroba (@). La versión consta de cuatro secuencias de números, separados por puntuación. Los elementos de las tres primeras secuencias están separados por puntos, y las secuencias tienen una longitud arbitraria. No se permiten ceros al principio de los componentes de la versión (por ejemplo, 01.1 o 1.01). Se permiten ceros a la derecha (por ejemplo, 1,10).</p> <p>La primera parte de la versión es la versión del componente. Para los componentes estrictamente ligados al sistema operativo, éste suele ser el valor de uname -r para esa versión del sistema operativo. Para un componente con su propio ciclo de desarrollo, esta secuencia es un número de versión separado por puntos, como 2.4.10.</p>

La segunda parte de la versión, que si está presente debe estar precedida por una coma (,), es la versión de generación. La versión de generación especifica en qué versión del sistema operativo se generó el contenido del paquete, lo cual proporciona una limitación mínima respecto de la versión del sistema operativo en la que se puede esperar que el contenido se ejecute correctamente.

La tercera parte de la versión, que si está presente debe estar precedida por un guión (-), es la versión de ramificación. La versión de ramificación es un componente de versión que proporciona información específica del proveedor. La versión de ramificación puede aumentar cuando se cambian los metadatos de empaquetado, independientemente de la versión de componente. La versión de ramificación puede incluir un número de generación u otra información.

La cuarta parte de la versión, que si está presente debe estar precedida por dos puntos (:), es un indicador de fecha y hora. El indicador de fecha y hora indica cuándo se publicó el paquete.

Al realizar comparaciones entre versiones, no se considera ningún componente de la versión completa a menos que los componentes de la izquierda sean iguales. Por lo tanto, “4.3-1” es mayor que “4.2-7” porque “4.3” es mayor que “4.2”, y “4.3-3” es mayor que “4.3-1” porque “3” es mayor que “1”.

Muchas partes del sistema, cuando corresponde, abrevian los FMRI cuando los muestran y aceptan entradas en formatos más cortos para reducir el volumen de información que se muestra o se requiere. Normalmente, el esquema, el editor, la versión de generación, y el indicador de fecha y hora se pueden omitir. A veces se puede omitir toda la información de versión.

Acciones Las acciones representan los objetos instalables en un sistema. Las acciones se describen en el manifiesto de un paquete. Cada acción consta básicamente de su nombre y un atributo clave. Juntos, estos hacen referencia a un objeto exclusivo como se desprende un historial de versiones. Las acciones pueden tener otros atributos. Algunos atributos son interpretados directamente por el sistema de empaquetado. Otros atributos pueden ser útiles sólo para el administrador del sistema o el usuario final.

Las acciones tienen una representación textual simple:

```
action_name attribute1=value1 attribute2=value2 ...
```

Los nombres de los atributos no pueden incluir espacios en blanco, comillas ni signos de igual (=). Todos los caracteres que aparecen después del primer signo de igual pertenecen al valor. Los valores pueden incluir todos estos, aunque los espacios deben aparecer entre comillas simples o dobles. A las comillas simples no es necesario aplicarles caracteres de escape dentro de una cadena que está entre comillas dobles, y a las comillas dobles no es necesario aplicarles caracteres de escape dentro de una cadena que está entre comillas simples. Una comilla puede estar precedida por una barra diagonal inversa (\) para evitar la terminación de la cadena citada. Una barra diagonal inversa se puede escapar con una barra diagonal inversa.

Los atributos pueden ser nombrados más de una vez con múltiples valores. Se tratan como listas desordenadas.

Las acciones con muchos atributos pueden crear líneas largas en un archivo de manifiesto. Estas líneas se pueden ajustar terminando cada línea incompleta con una barra diagonal inversa. Tenga en cuenta que este carácter de continuación se debe colocar entre los pares de atributo y valor. Ni los atributos ni sus valores ni la combinación se puede dividir.

Los atributos mencionados a continuación no abarcan la totalidad del conjunto. De hecho, los atributos que se pueden conectar a una acción son arbitrarios, y los conjuntos estándar de atributos son fáciles de aumentar para incorporar futuros desarrollos.

Determinados atributos de acción generan que se ejecuten operaciones adicionales fuera del contexto de empaquetado. Estos atributos se documentan en la sección “Activadores” a continuación.

Acciones de archivo La acción `file` representa un archivo normal. La acción `file` hace referencia a una carga útil y tiene cuatro atributos estándar:

<code>ruta</code>	La ruta del sistema de archivos donde se instala el archivo. Se trata del atributo clave de la acción <code>file</code> .
<code>modo</code>	Los permisos de acceso (en formato numérico) del archivo. Estos son sólo algunos permisos simples, no son listas de control de acceso (ACL).
<code>propietario</code>	El nombre del usuario que posee el archivo.
<code>grupo</code>	El nombre del grupo que posee el archivo.

La carga útil es un atributo posicional desde el punto de vista de que no tiene nombre. Es la primera palabra después del nombre de acción. En un manifiesto publicado, es el hash SHA-1 del contenido del archivo. Si está presente en un manifiesto que aún tiene que ser publicado, representa la ruta donde se puede encontrar la carga útil. Consulte `pkgsend(1)`. El atributo hash se puede utilizar en lugar del atributo posicional, si el valor incluye un signo igual. Ambos se pueden utilizar en la misma acción. Sin embargo, los hashes deben ser idénticos.

Entre los otros atributos se incluyen:

`preserve` Esto especifica que el contenido del archivo no se debe sobrescribir en la actualización si se determina que el contenido ha cambiado desde la instalación o la última actualización del archivo. En las instalaciones iniciales, si se detecta un archivo existente, el archivo se recupera (se almacena en `/var/pkg/lost+found`).

Si el valor de `preserve` es `renameold`, el archivo existente es renombrado con la extensión `.old` y el archivo nuevo se coloca en su lugar.

Si el valor de `preserve` es `renamenew`, el archivo existente no se modifica y el archivo nuevo se instala con la extensión `.new`.

Si el valor de `preserve` es `legacy`, este archivo no se instala para las instalaciones iniciales de paquete. En las actualizaciones, cualquier archivo existente es renombrado con la extensión `.legacy` y el archivo nuevo se coloca en su lugar.

Si el valor de `preserve` es `true` (o un valor que no figura en la lista anterior, como `strawberry`), el archivo existente no se modifica y el archivo nuevo no se instala.

`overlay` Especifica si la acción permite que otros paquetes entreguen un archivo en la misma ubicación o si entregan un archivo destinado a superponerse a otro. Esta funcionalidad está destinada únicamente para su uso con archivos de configuración que no participan en ningún autoensamblaje (por ejemplo, `/etc/motd`) y que se pueden sobrescribir de manera segura.

Si `overlay` no se especifica, varios paquetes no pueden entregar archivos en la misma ubicación.

Si el valor de `overlay` es `allow`, otro paquete puede entregar un archivo en la misma ubicación. Este valor no tiene efecto, a menos que también se establezca el atributo `preserve`.

Si el valor de `overlay` es `true`, el archivo entregado por la acción sobrescribe cualquier otra acción que haya especificado `allow`. Los cambios realizados en el archivo instalado se conservan en función del valor del atributo `preserve` del archivo que se superpone. Durante la eliminación, el contenido del archivo se conserva si la acción que se está superponiendo aún está instalada, independientemente de si se especificó el atributo `preserve`. Sólo una acción puede superponer a otra, y los atributos `mode`, `owner` y `group` deben coincidir.

Los archivos también se pueden “probar” y, en función del tipo, pueden tener atributos adicionales. Para los archivos ELF, se reconocen los siguientes atributos:

<code>elfarch</code>	La arquitectura del archivo ELF. Ésta es la salida de <code>uname -p</code> en la arquitectura para la que se crea el archivo.
<code>elfbits</code>	Este es 32 o 64.
<code>elfhash</code>	Este es el hash de las secciones ELF “interesantes” del archivo. Estas son las secciones que se asignan en la memoria cuando se carga el archivo binario. Estas son las únicas secciones que es necesario considerar al determinar si el comportamiento ejecutable de dos archivos binarios diferirá.
<code>original_name</code>	Este atributo se utiliza para manejar archivos editables que se mueven de un paquete a otro, desde un lugar a otro, o ambos. La forma que adopta es el nombre del paquete de origen, seguido de dos puntos y la ruta original

del archivo. Cualquier archivo que se suprime se registra con su paquete y ruta, o con el valor del atributo `original_name`, si se especifica. Cualquier archivo editable que se instale que tenga el conjunto de atributos `original_name` utiliza el archivo con ese nombre si se suprime como parte de la misma operación de empaquetado.

revert - tag Este atributo se utiliza para los archivos con etiquetas editables que se deben revertir como un conjunto. Se pueden especificar diversos valores `revert - tag`. El archivo vuelve al estado definido en el manifiesto cuando `pkg revert` se invoca con cualquiera de esas etiquetas especificadas. Consulte `pkg(1)`.

Acciones de directorio La acción `dir` es similar a la acción `file` en que representa un objeto del sistema de archivos. La acción `dir` representa un directorio en lugar de un archivo normal. La acción `dir` tiene los mismos cuatro atributos estándar que la acción `file`, y `path` es el atributo clave.

Los directorios se cuentan como referencia en IPS. Cuando el último paquete que explícita o implícitamente hace referencia a un directorio ya no lo hace, ese directorio se elimina. Si ese directorio contiene objetos del sistema de archivos desempaquetados, los elementos se mueven a `$IMAGE_META/lost+found`. Consulte la sección “Archivos” para obtener más información sobre `$IMAGE_META`.

Para mover el contenido desempaquetado a un directorio nuevo, el atributo siguiente podría ser útil:

salvage - from Esto nombra un directorio de elementos recuperados. Un directorio con dicho atributo hereda en el momento de su creación el contenido del directorio recuperado, si existe.

Acciones de enlace La acción `link` representa un vínculo simbólico. La acción `link` tiene los siguientes atributos estándar:

ruta

La ruta del sistema de archivos donde se instala el enlace simbólico. Se trata de un atributo clave de la acción `link`.

target

El destino del enlace simbólico. El objeto del sistema de archivos en el que se resuelve el enlace.

mediator

Especifica la entrada en el espacio de nombre de mediación compartido por todas las rutas que participan en un grupo de mediación determinado (por ejemplo, `python`). La mediación de enlace se puede realizar en función de `mediator-version` y/o `mediator-implementation`. Todos los enlaces mediados para un nombre de ruta determinado deben especificar el mismo `mediator`. Sin embargo, no todas las versiones e implementaciones de `mediator` necesitan proporcionar un enlace a una ruta determinada.

Si una mediación no proporciona un enlace, el enlace se elimina cuando se selecciona esa mediación. Un `mediator`, en combinación con una versión y/o implementación específica representa una mediación que se puede seleccionar para ser utilizada por el sistema de empaquetado.

`mediator-version`

Especifica la versión (expresada como una secuencia de números enteros no negativos separados por puntos) de la interfaz descrita por el atributo `mediator`. Este atributo es necesario si se especifica `mediator` y no se especifica `mediator-implementation`. Un administrador del sistema local puede establecer la versión que se va a utilizar explícitamente. El valor especificado, por lo general, debe coincidir con la versión del paquete que entrega el enlace (por ejemplo, `runtime/python-26` debe usar `mediator-version=2.6`), aunque esto no es necesario.

`mediator-implementation`

Especifica la implementación del mediador para utilizar además de `mediator-version`, o en su lugar. No se considera que las cadenas de implementación estén ordenadas y una cadena es seleccionada de manera arbitraria por `pkg(5)` si un administrador del sistema no la especifica explícitamente.

El valor puede ser una cadena de tamaño arbitrario, compuesta por caracteres alfanuméricos y espacios. Si la implementación en sí puede estar versionada o está versionada, la versión se debe especificar al final de la cadena, después de una `@` (y se debe expresar como una secuencia de números enteros no negativos separados por puntos). Si existen varias versiones de una implementación, de manera predeterminada, se selecciona la implementación con el versión superior.

Si en un sistema se instala sólo una instancia de un enlace de mediación de implementación en una ruta determinada, esa instancia se elegirá automáticamente. Si se instalan futuros enlaces a la ruta, el enlace no se cambia a menos que se aplique un proveedor, un sitio o una sustitución local, o si uno de los enlaces tiene una mediación de versión.

`mediator-priority`

Al resolver conflictos en enlaces mediados, `pkg(5)` normalmente elige el enlace con el mayor valor de `mediator-version` o, si no es posible, según `mediator-implementation`. Este atributo se utiliza para especificar una sustitución para el proceso normal de resolución de conflictos.

Si no se especifica este atributo, se aplica la lógica de selección de mediador predeterminado.

Si el valor es `vendor`, se prefiere el enlace a aquellos que no tienen `mediator-priority` especificado.

Si el valor es `site` se prefiere el enlace a los que tienen un valor de `vendor` o que no tienen `mediator-priority` especificado.

Un administrador del sistema local puede anular la lógica de selección descrita anteriormente.

Acciones de enlace La acción `hardlink` representa un enlace físico. Tiene los mismos atributos que la acción `link`, y `path` también es su atributo clave.

Acciones del controlador La acción `driver` representa un controlador de dispositivos. La acción `driver` no hace referencia a una carga útil. Los archivos de controlador se deben instalar como acciones `file`. Se reconocen los siguientes atributos (consulte `add_drv(1M)` para obtener más información):

nombre	El nombre del controlador. Suele ser, aunque no siempre, el nombre de archivo del binario de controlador. Este es el atributo clave de la acción <code>driver</code> .
alias	Esto representa un alias para el controlador. Un controlador determinado puede tener más de un atributo <code>alias</code> . No se requieren reglas de comillas especiales.
clase	Esto representa una clase de controlador. Un controlador determinado puede tener más de un atributo <code>class</code> .
permisos	Esto representa los permisos del sistema de archivos para los nodos de dispositivo del controlador.
clone_perms	Esto representa los permisos del sistema de archivos para los nodos menores del controlador clon de este controlador.
política	Esta opción especifica la política de seguridad adicional para el dispositivo. Un controlador determinado puede tener más de un atributo <code>policy</code> , pero no puede haber una especificación de dispositivo menor en más de un atributo.
privs	Especifica los privilegios utilizados por el controlador. Un controlador determinado puede tener más de un atributo <code>privs</code> .
devlink	Especifica una entrada en <code>/etc/devlink.tab</code> . El valor es la línea exacta para entrar en el archivo, con tabuladores indicados mediante <code>\t</code> . Consulte <code>devlinks(1M)</code> para obtener más información. Un controlador determinado puede tener más de un atributo <code>devlink</code> .

Acciones de dependencia La acción `depend` representa una dependencia entre paquetes. Un paquete puede depender de otro porque el primero requiere la funcionalidad del segundo para que la funcionalidad del primero funcione o, incluso, se instale. Las dependencias pueden ser opcionales. Si una dependencia no se cumple en el momento de la instalación, el sistema de empaquetado intenta instalar o actualizar el paquete dependiente a una versión lo suficientemente nueva, sujeta a otras restricciones.

Se reconocen los atributos siguientes:

fmri	El FMRI que representa el paquete dependiente. Este es el atributo clave de la acción <code>dependency</code> . El valor <code>fmri</code> no debe incluir el editor. Se asume que el nombre del paquete está completo. Las dependencias de tipo <code>require-any</code> pueden tener varios atributos <code>fmri</code> . Una versión es opcional en el valor <code>fmri</code> , aunque para algunos tipos de dependencias, un <code>fmri</code> sin ninguna versión no tiene ningún significado.
type	<p>El tipo de dependencia.</p> <p>Si el valor es <code>require</code>, la dependencia es necesaria y debe tener una versión igual o mayor a la versión especificada en el atributo <code>fmri</code>. Si no se especifica la versión, cualquier versión satisface la dependencia. Un paquete no se puede instalar si alguna de las dependencias necesarias no se puede satisfacer.</p> <p>Si el valor es <code>optional</code>, la dependencia, si está presente, debe estar en el nivel de versión especificado o superior.</p> <p>Si el valor es <code>exclude</code>, el paquete contenedor no se puede instalar si la dependencia está presente en el nivel de versión especificado o superior. Si no se especifica ninguna versión, el paquete dependiente no se puede instalar simultáneamente con el paquete que especifica la dependencia.</p> <p>Si el valor es <code>incorporate</code>, la dependencia es opcional, pero la versión del paquete dependiente es restringida. Consulte “Restricciones y congelación” a continuación.</p> <p>Si el valor es <code>require-any</code> cualquiera de los múltiples paquetes dependientes especificados mediante múltiples atributos <code>fmri</code> puede satisfacer la dependencia siguiendo las mismas reglas que el tipo de dependencia <code>require</code>.</p> <p>Si el valor es <code>conditional</code>, la dependencia sólo se necesita si el paquete definido por el atributo <code>predicate</code> está presente en el sistema.</p> <p>Si el valor es <code>origin</code>, la dependencia, si está presente, debe ser de un valor especificado o mejor en la imagen que se va a modificar antes de la instalación. Si el valor del atributo <code>root-image</code> es <code>true</code>, la dependencia debe estar presente en la imagen con raíz en / para poder instalar este paquete.</p> <p>Si el valor es <code>group</code>, la dependencia es necesaria, a menos que el paquete esté en la lista de imágenes para evitar. Tenga en cuenta que los paquetes obsoletos de manera silenciosa satisfacen la dependencia de grupo. Consulte el subcomando <code>avoid</code> en <code>pkg(1)</code>.</p> <p>Si el valor es <code>parent</code> y la imagen no es una imagen secundaria, la dependencia se omite. Si la imagen es una imagen secundaria, se requiere que la</p>

dependencia esté presente en la imagen principal. La versión del paquete coincidente para una dependencia de `parent` es la misma que se usó para las dependencias de `incorporate`.

`predicate` El FMRI que representa el predicado para las dependencias de `conditional`.

`root-image` Tiene efecto únicamente para las dependencias de `origin` como se mencionó anteriormente.

Acciones de licencia La acción `license` representa una licencia u otro archivo de información asociado con el contenido del paquete. Un paquete puede entregar licencias, renunciaciones de responsabilidad u otras pautas para el instalador del paquete para el uso de la acción `license`.

La carga útil de la acción `license` se entrega en el directorio de metadatos de imágenes relacionado con el paquete y sólo debe contener datos de texto legibles para el usuario. No debe contener HTML ni ningún otro tipo de marca. Por medio de los atributos, las acciones `license` pueden indicar a los clientes que la carga útil relacionada se debe mostrar y/o requiere aceptación. El método visualización y/o aceptación queda a discreción de los clientes.

Se reconocen los atributos siguientes:

`license` Se trata de un atributo clave de la acción `license`. Este atributo aporta una descripción significativa de la licencia para ayudar a los usuarios a determinar el contenido sin leer el texto de la licencia. Estos son algunos ejemplos de valores:

- Aviso de Copyright de ABC Co.
- Licencia personalizada de ABC Co.
- Licencia común de desarrollo y distribución 1.0 (CDDL)
- Licencia pública general de GNU 2.0 (GPL)
- Sólo Licencia pública general de GNU 2.0 (GPL)
- Licencia de MIT
- Licencia pública de Mozilla 1.1 (MPL)
- Licencia BSD simplificada

El valor de `license` debe ser único en un paquete. Se recomienda incluir la versión de la licencia en la descripción, como se mostró anteriormente en varios ejemplos. Si un paquete tiene un código bajo varias licencias, utilice varias acciones de `license`. La longitud del valor del atributo de la licencia no debe tener más de 64 caracteres.

`must-accept` Cuando es `true`, esta licencia debe ser aceptada por un usuario antes de que el paquete relacionado se pueda instalar o actualizar. La omisión de este atributo es equivalente a `false`. El método de aceptación (interactivo o basado en configuración, por ejemplo) queda a discreción de los clientes.

`must-display` Cuando es `true`, los clientes deben mostrar la carga útil de la acción durante las operaciones de empaquetado. La omisión de este valor es

equivalente a `false`. Este atributo no se debe utilizar para avisos de copyright, sólo para licencias reales u otro material que se deba mostrar durante las operaciones. El método de visualización queda a discreción de los clientes.

Acciones de herencia La acción `legacy` representa los datos del paquete utilizados por un sistema de empaquetado heredado. Los atributos asociados con esta acción se agregan a las bases de datos del sistema heredado, con el fin de que las herramientas que consultan dichas bases de datos puedan funcionar como si los paquetes heredados estuviesen instalados realmente. En particular, esto debería ser suficiente para convencer al sistema heredado de que el paquete nombrada por el atributo `pkg` está instalado en el sistema, para que el paquete se pueda utilizar para satisfacer las dependencias.

Se reconocen los siguientes atributos, nombrados de acuerdo con los parámetros de `pkginfo(4)`:

<code>categoría</code>	El valor para el parámetro <code>CATEGORY</code> . El valor predeterminado es <code>system</code> .
<code>desc</code>	El valor para el parámetro <code>DESC</code> .
<code>hotline</code>	El valor para el parámetro <code>HOTLINE</code> .
<code>nombre</code>	El valor para el parámetro <code>NAME</code> . El valor predeterminado es <code>none provided</code> .
<code>pkg</code>	La abreviación para el paquete que se está instalando. El valor predeterminado es el nombre del FMRI del paquete. Se trata de un atributo clave de la acción <code>legacy</code> .
<code>vendor</code>	El valor del parámetro <code>VENDOR</code> .
<code>versión</code>	El valor del parámetro <code>VERSION</code> . El valor predeterminado es la versión del FMRI del paquete.

Acciones de definición La acción `set` representa un atributo de nivel de paquete, o metadatos, como la descripción del paquete.

Se reconocen los atributos siguientes:

<code>nombre</code>	El nombre del atributo.
<code>valor</code>	El valor especificado para el atributo.

La acción `set` puede entregar cualquier metadato que elija el autor del paquete. Sin embargo, existen varios nombres de atributos bien definidos que tienen un significado específico para el sistema de empaquetado.

<code>pkg.fMRI</code>	Consulte “Versiones e identificadores de recursos de gestión de errores de paquetes” en la sección “Descripción”.
-----------------------	-------------------------------------------------------------------------------------------------------------------

	<code>info.classification</code>	Uno o varios tokens que un cliente pkg(5) puede utilizar para clasificar el paquete. El valor debe contar con un esquema (como “org.opensolaris.category.2008” o “org.acm.class.1998”) y la clasificación real, como “Applications/Games”, separados por dos puntos (:).
	<code>pkg.description</code>	Una descripción detallada del contenido y la funcionalidad del paquete, en general, aproximadamente un párrafo de longitud.
	<code>pkg.obsolete</code>	Cuando es <code>true</code> , el paquete se marca como obsoleto. Es posible que un paquete obsoleto no tenga más acciones que las de definición, y no se debe marcar como cambiado de nombre.
	<code>pkg.renamed</code>	Cuando es <code>true</code> , se ha cambiado el nombre del paquete. En el paquete también debe haber una o varias acciones depend que se refieran a la versiones de paquete en las que este paquete se ha cambiado de nombre. Un paquete no se puede marcar como cambiado de nombre y como obsoleto, pero puede tener cualquier cantidad de acciones de definición.
	<code>pkg.summary</code>	Una descripción breve, de una línea, del paquete.
Acciones de grupo	La acción <code>group</code> define un grupo de UNIX como se define en <code>group(4)</code> . Las contraseñas de grupo no se admiten. Los grupos definidos con esta acción inicialmente no tienen lista de usuarios. Con la acción <code>user</code> se pueden agregar usuarios. Se reconocen los atributos siguientes:	
	<code>groupname</code>	El valor para el nombre del grupo.
	<code>gid</code>	El identificador numérico único del grupo. El valor predeterminado es el primer grupo libre inferior a 100.
Acciones de usuario	La acción <code>user</code> define un usuario UNIX según lo definido en los archivos <code>/etc/passwd</code> , <code>/etc/shadow</code> , <code>/etc/group</code> y <code>/etc/ftpd/ftpusers</code> . Los usuarios definidos con este atributo tienen entradas agregadas en los correspondientes archivos. Se reconocen los atributos siguientes:	
	<code>nombre_usuario</code>	El nombre único del usuario.
	<code>contraseña</code>	La contraseña cifrada del usuario. El valor predeterminado es <code>*LK*</code> . Consulte <code>shadow(4)</code> .
	<code>uid</code>	El UID único del usuario. El valor predeterminado es el primer valor libre inferior a 100.
	<code>grupo</code>	El nombre del grupo principal del usuario. Se debe encontrar en <code>/etc/group</code> .

<code>gcos-field</code>	El valor del campo <code>gcos</code> en <code>/etc/passwd</code> . El valor predeterminado es <code>username</code> .
<code>home-dir</code>	El directorio principal del usuario. El valor predeterminado es <code>/</code> .
<code>login-shell</code>	El shell predeterminado del usuario. El valor predeterminado está vacío.
<code>lista_grupos</code>	Los grupos secundarios a los que pertenece el usuario. Consulte <code>group(4)</code> .
<code>ftpuser</code>	Se puede establecer en <code>true</code> o <code>false</code> . El valor predeterminado de <code>true</code> indica que el usuario se puede conectar por medio de FTP. Consulte <code>ftputils(4)</code> .
<code>lastchg</code>	El número de días entre el 1 de Enero de 1970 y la fecha en que la contraseña se modificó por última vez. El valor predeterminado está vacío. Consulte <code>shadow(4)</code> .
<code>min</code>	El número mínimo de días necesarios entre cambios de contraseña. Este campo se debe definir en 0 o más para habilitar la antigüedad de la contraseña. El valor predeterminado está vacío. Consulte <code>shadow(4)</code> .
<code>max</code>	El número máximo de días durante los que la contraseña es válida. El valor predeterminado está vacío. Consulte <code>shadow(4)</code> .
<code>warn</code>	Indica cuántos días antes de que caduque la contraseña se debe advertir al usuario. Consulte <code>shadow(4)</code> .
<code>inactive</code>	El número de días de inactividad permitidos para ese usuario. Se calcula por equipo. La información del último inicio de sesión se toma del archivo <code>lastlog</code> del equipo. Consulte <code>shadow(4)</code> .
<code>expire</code>	Una fecha absoluta expresada como el número de días desde UNIX Epoch (1 de enero de 1970). Cuando se alcanza este número, ya no se puede iniciar sesión. Por ejemplo, el valor de caducidad 13514 especifica la caducidad del inicio de sesión para el 1 de enero de 2007. Consulte <code>shadow(4)</code> .
<code>flag</code>	Se establece como vacío. Consulte <code>shadow(4)</code> .

Activadores En algunos contextos, puede ser apropiado ejecutar operaciones adicionales antes o después de la introducción de una acción determinada. Por lo general, estas operaciones adicionales sólo se necesitan en una imagen de sistema activo, y son específicas del sistema operativo. Cuando varias acciones involucradas en la instalación o eliminación de un paquete tienen los mismos activadores, la operación correspondiente a la presencia del activador se ejecuta una vez para esa instalación o eliminación.

Si los activadores se especifican incorrectamente se pueden producir errores en la instalación del paquete, si el activador no se puede determinar una manera de realizar una instalación segura.

Se definen los siguientes activadores:

`reboot - needed`

Se puede establecer en `true` o `false`. Si se instala o actualiza una acción con este activador establecido en `true` durante la instalación de un paquete, se puede anunciar que la transacción de empaquetado requiere un reinicio. Es posible que determinadas implementaciones de cliente requieran pasos adicionales, como realizar toda la operación de paquete con un clon de la imagen, en caso de que la imagen sea la imagen del sistema activo.

`disable_fmri, refresh_fmri, restart_fmri, suspend_fmri`

Cada uno de estos activadores toma el valor de un FMRI de una instancia de servicio en la cual funcionará durante la instalación o la eliminación de paquetes. El activador `disable_fmri` hace que el FMRI determinado se desactive antes de la eliminación de la acción, mediante el subcomando `disable` de `svcadm(1M)`. Los activadores `refresh_fmri` y `restart_fmri` hacen que el FMRI determinado se refresque o se reinicie después de la instalación, la actualización o la eliminación de la acción, según los respectivos subcomandos de `svcadm(1M)`. Por último, el activador `suspend_fmri` hace que el FMRI determinado se desactive temporalmente antes de la fase de instalación de la acción y, luego, se active una vez finalizada dicha fase.

El valor puede incluir un modelo que coincida con varias instancias de servicio. Sin embargo, debe hacerlo explícitamente con un modelo glob aceptado por `svcs(1)`, en lugar de hacerlo implícitamente sin indicar ninguna instancia.

Restricciones y congelación

Cuando un paquete se pasa a una versión nueva, o cuando se agrega o elimina del sistema, la selección de la versión o la posibilidad de la eliminación se determina mediante una variedad de restricciones impuestas al paquete. Dichas restricciones pueden ser definidas por otros paquetes mediante dependencias o por el administrador mediante congelaciones.

La forma más común de restricción es la dependencia `require`, como se describió anteriormente en “Acciones de dependencia”. Dicha restricción impide que el paquete pase a una versión anterior o se elimine.

La mayoría de las partes del sistema operativo están encapsuladas en paquetes denominados *incorporaciones*. Estos paquetes principalmente ofrecen restricciones representadas mediante la dependencia `incorporate`.

Como se describió anteriormente, un paquete incorporado no necesita estar presente en el sistema, pero si lo está, especifica la versión mínima de inclusión y la versión máxima de exclusión. Por ejemplo, si el FMRI dependiente tiene una versión 1.4.3, ninguna versión

anterior a 1.4.3 satisfaría la dependencia, y tampoco lo haría ninguna versión posterior o igual a 1.4.4. Sin embargo, las versiones que simplemente ampliaron la secuencia separada por puntos, como 1.4.3.7, estarían permitidas.

Las incorporaciones se utilizan para forzar a partes del sistema a actualizarse de manera sincronizada. Para algunos componentes, como la biblioteca de C y el núcleo, éste es un requisito básico. Para otros, como un componente simple del espacio de usuario en el que nada más tiene una dependencia, se utiliza la actualización sincronizada simplemente para proporcionar un conjunto de versiones de paquete conocido y probado al que una versión concreta de la incorporación pueda hacer referencia.

Dado que una incorporación es simplemente un paquete, se puede eliminar y, por lo tanto, todas las restricciones impuestas por ella se flexibilizan. Sin embargo, muchas de las incorporaciones impartidas por Oracle Solaris son requeridas por los paquetes que incorpora porque la flexibilidad mencionada no ofrecería seguridad.

Si se intenta realizar una actualización de un paquete a una versión que no está permitida por una incorporación instalada, no se intentará encontrar una versión más reciente de la incorporación para satisfacer la solicitud, por el contrario, se producirá un error. Si se debe mover la restricción y la incorporación que la especifica no se puede eliminar, la incorporación se debe actualizar a una versión que especifique una versión deseada de la restricción. Si se actualiza una incorporación, también se deben actualizar todos los paquetes que no satisfagan las restricciones impartidas por la nueva versión.

Un administrador del sistema puede restringir un paquete mediante el comando `pkg freeze`. Si no se proporciona ninguna versión, el paquete nombrado se restringe a la versión instalada en el sistema. Si se proporciona un paquete versionado, esta congelación o restricción administrativa actúa como si se instalara una dependencia de incorporación en la que el atributo `fmri` tuviera el valor de la versión de paquete proporcionada.

Una inmovilización nunca es levantada automáticamente por el sistema de empaquetado. Para flexibilizar una restricción, utilice el comando `pkg unfreeze`.

Editores y depósitos

Como se explicó anteriormente, un editor es simplemente un nombre que utilizan los clientes de paquetes para identificar al proveedor de paquetes. Los editores pueden distribuir los paquetes mediante depósitos de paquetes y/o archivos de paquetes. Actualmente el sistema de paquetes admite dos tipos de depósitos: depósitos de origen y depósitos de reflejo.

Un *origen* es un depósito de paquetes que contiene todos los metadatos (como catálogos, manifiestos e índices de búsqueda) y el contenido (archivos) de uno o varios paquetes. Si se configuran varios orígenes para un editor determinado en una imagen, la API del cliente del paquete intenta seleccionar el mejor origen del cual recuperar los datos del paquete. Éste es el tipo de depósito más común y se crea implícitamente cuando se utiliza `pkgsend` o `pkgrecv` en un depósito de paquetes.

Un *reflejo* es un depósito de paquetes que sólo incluye el contenido del paquete (archivos). Si se configura uno o varios reflejos para un editor determinado en una imagen, la API del cliente

prefiere los reflejos para la recuperación del contenido de los paquete e intenta seleccionar el mejor reflejo del cual recuperar el contenido de los paquetes. Si el reflejo es inaccesible, no tiene el contenido requerido o se ralentiza, la API del cliente recupera el contenido de cualquier depósito de origen configurado. La finalidad de los reflejos es el uso compartido de contenido entre un conjunto confiable de clientes mediante la funcionalidad de reflejo dinámica de `pkg.depotd(1M)`. Los reflejos también tienen la finalidad de autenticar el acceso a los metadatos del paquete y distribuir el contenido del paquete sin autenticación. Por ejemplo, un cliente puede estar configurado con un origen `https` que requiere un par de clave y certificado SSL para acceder, y con un reflejo `http` que proporciona el contenido del paquete. De esta manera, únicamente los clientes autorizados pueden instalar o actualizar los paquetes, mientras se evitan los costos generales de autenticación para la recuperación del contenido del paquete. Un reflejo se puede crear eliminando todos los subdirectorios de un depósito excepto los nombrados `file` y sus directorios principales. Un depósito de origen también se puede proporcionar como un reflejo mediante el uso del modo de reflejo de `pkg.depotd(1M)`.

Facetas y variantes El software puede tener componentes que son opcionales y componentes que son mutuamente excluyentes. Algunos ejemplos de componentes opcionales son las configuraciones regionales y la documentación. Algunos ejemplos de componentes mutuamente excluyentes son SPARC o x86, y archivos binarios de depuración y no depuración.

En IPS, los componentes opcionales se denominan *facetas* y los componentes mutuamente excluyentes se denominan *variantes*. Las facetas y variantes se especifican como etiquetas en acciones de paquetes. Cada etiqueta de faceta y variante tiene un nombre y un valor. Una sola acción puede tener varias etiquetas de facetas y variantes. Los ejemplos de componentes con varias etiquetas de facetas y variantes incluyen un archivo de encabezado específico de arquitectura utilizado por desarrolladores, o un componente que sólo es para una zona global SPARC.

Un ejemplo de una etiqueta de variante es `variant.arch=sparc`. Un ejemplo de una etiqueta de faceta es `facet.devel=true`. Por lo general, se hace referencia a las facetas y las variantes sin el inicio `facet.` ni `variant.`.

Las facetas y las variantes son propiedades especiales de la imagen y no se pueden establecer en paquetes individuales. Para ver los valores actuales de las facetas y las variantes definidas en la imagen, use los comandos `pkg facet` y `pkg variant`, como se muestra en la página del comando `man pkg(1)`. Para modificar los valores de las facetas y las variantes definidas en la imagen, use los comandos `pkg change-facet` y `pkg change-variant`.

Las facetas son booleanas: sólo se pueden establecer en `true` (activadas) o `false` (desactivadas). De manera predeterminada, se considera que todas las facetas están establecidas en `true`, en la imagen. Una etiqueta de faceta en una acción sólo debe tener el valor `true`; los demás valores tienen comportamiento indefinido. Una faceta establecida en la imagen puede ser una faceta completa, como `doc.man`, o un patrón, como `locale.*`. Esto resulta de gran utilidad cuando se desea desactivar una parte del espacio de nombre de la faceta, y cuando se desean activar solamente facetas individuales en él. Por ejemplo, puede

desactivar todas las configuraciones regionales y luego puede activar sólo una o dos configuraciones regionales específicas, como se muestra en el siguiente ejemplo:

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

La mayoría de las variantes pueden tener cualquier cantidad de valores. Por ejemplo, la variante `arch` se puede establecer en `i386`, `sparc`, `ppc` o `arm`, o en cualquier arquitectura que la distribución admita. (Sólo `i386` y `sparc` se utilizan en Oracle Solaris). La excepción son las variantes `debug`. Las variantes `debug` sólo se pueden definir en `true` o `false`; los demás valores tienen un comportamiento indefinido. Si la acción de un archivo tiene versiones de depuración y de no depuración, ambas versiones deben tener la variante `debug` aplicable explícitamente establecida, como se muestra en el siguiente ejemplo:

```
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
```

El valor de la variante se debe definir en la imagen para que un paquete que usa la variante se instale. Las variantes `arch` y `zone` son definidas por el programa que crea la imagen e instala su contenido inicial. De manera predeterminada, las variantes `debug.*` son `false` en la imagen.

Las facetas y las variantes establecidas en la imagen afectan la instalación de una acción concreta.

- Las acciones sin etiquetas de facetas o variantes siempre se instalan.
- Se instalan las acciones con etiquetas de facetas, a menos que todas las facetas o los patrones de facetas que coinciden con las etiquetas estén establecidos en `false`, en la imagen. Si alguna faceta está establecida en `true` o no está establecida de forma explícita (`true` es el valor predeterminado), se instala la acción.
- Las acciones con etiquetas de variantes se instalan sólo si los valores de todas las etiquetas de variantes son los mismos que los definidos en la imagen.
- Las acciones con etiquetas de facetas y variantes se instalan si tanto las facetas como las variantes permiten que la acción se instale.

Puede crear sus propias etiquetas de facetas y variantes. Las siguientes etiquetas se utilizan con frecuencia en Oracle solaris.

Nombre de variante	Valores posibles
<code>variant.arch</code>	<code>sparc</code> , <code>i386</code>

Nombre de variante	Valores posibles
variant.opensolaris.zone	global, nonglobal
variant.debug.*	true, false

A continuación, se muestran algunos ejemplos de las etiquetas de facetas que se utilizan en Oracle Solaris:

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

Políticas de imagen Las políticas de imagen se definen mediante las propiedades de la imagen con valores booleanos. Consulte "Propiedades de la imagen" en la página del comando `man pkg(1)` para obtener descripciones de las propiedades `flush-content-cache-on-success` y `send-uuid`, e información sobre cómo ver y modificar sus valores.

Archivos Dado que las imágenes `pkg(5)` se pueden ubicar arbitrariamente en un sistema de archivos más grande, se utiliza el token `$IMAGE_ROOT` para distinguir las rutas relacionadas. Para una instalación de sistema típica, `$IMAGE_ROOT` es equivalente a `/`.

<code>\$IMAGE_ROOT/var/pkg</code>	Directorio de metadatos para una imagen completa o parcial.
<code>\$IMAGE_ROOT/.org.opensolaris, pkg</code>	Directorio de metadatos para una imagen de usuario.

En los metadatos de una imagen en particular, ciertos archivos y directorios pueden contener información útil durante la reparación y la recuperación. El token `$IMAGE_META` se utiliza para hacer referencia al directorio de nivel superior que contiene los metadatos. `$IMAGE_META` generalmente es una de las dos rutas especificadas anteriormente.

<code>\$IMAGE_META/lost+found</code>	Ubicación de los archivos y directorios conflictivos movidos durante una operación de paquetes.
<code>\$IMAGE_META/publisher</code>	Contiene un directorio para cada editor. Cada directorio almacena los metadatos específicos de un editor.

Otras rutas dentro de la jerarquía de directorios de `$IMAGE_META` son privadas y están sujetas a cambios.

Atributos Consulte `attributes(5)` para ver descripciones de los atributos siguientes:

TIPO DE ATRIBUTO	VALOR DEL ATRIBUTO
Disponibilidad	package/pkg
Estabilidad de interfaz	Sin asignar.

Véase también [pkg\(1\)](#), [pkgsend\(1\)](#), [pkg.depotd\(1m\)](#), [pkg.sysrepo\(1m\)](#), [svcs\(1\)](#), [svcadm\(1M\)](#)
<http://hub.opensolaris.org/bin/view/Project+pkg/>

