



# pkg (5)

Image Packaging System

Lubos  
Kocman

---

# Agenda

---

- Introduction to the IPS
- How to set up an IPS server
- How to create / publish own packages
- How to get involved?
- Q&A

# Introduction to the IPS

---



- Overview
- What is an image?
- pkg (5) components
- Package FMRI and versions
- Metapackages & Incorporations
- Example usage of pkg (1)
- Package manager
- pkg authorities

# Overview

---

- IPS is a packaging system used by OpenSolaris 2008.\* releases
- The IPS software is a network-centric packaging system written in Python
- IPS enables you to install and manage packages on your system  
Packages are identified by FMRI's

# Overview

---

- Scripting free (no checkinstall, preinstall, postinstall ...)
- Network repository support :-)
- Default authority is set to pkg.opensolaris.org
- Not all packages are installed by default (caused by live cd capacity)
- Resolving dependencies
- Branded zone support (create -t SUNWipkg)
- User images

# What is an image?

---

- A collection of software in a package that comprises an entire operating system.
- An installed OpenSolaris 2008.\* release on your system, constitutes an image. Packages can be installed directly within this image.
- New images are created to mark a boundary between the packages installed in different images.

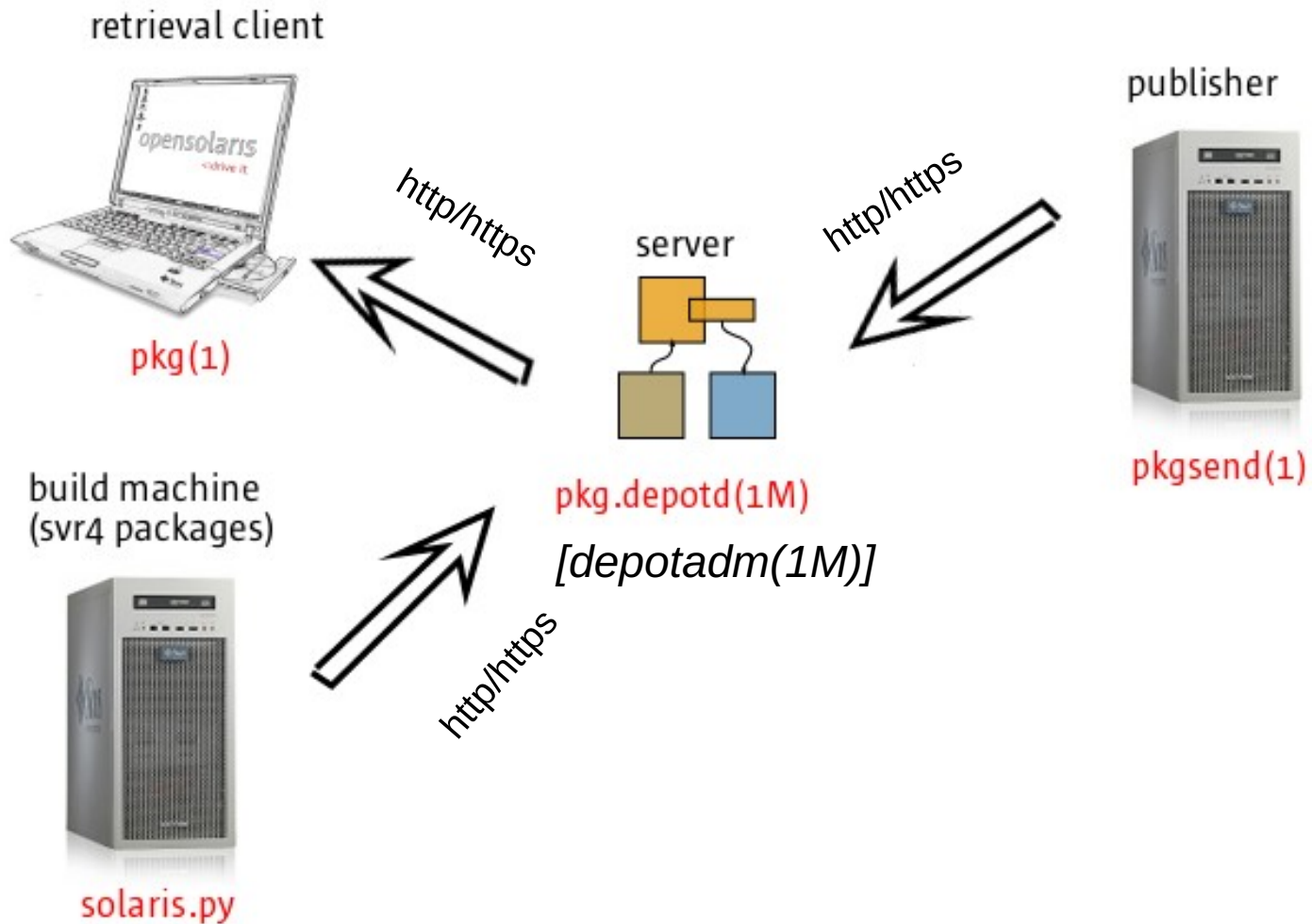
# What is an image?

---

There are three types of image:

- Full images, capable of providing a complete system.
- Partial images, which are linked to a full image (parent image), but do not provide a complete system on their own.
- User images, which should contain only relocatable packages.

# pkg (5) components



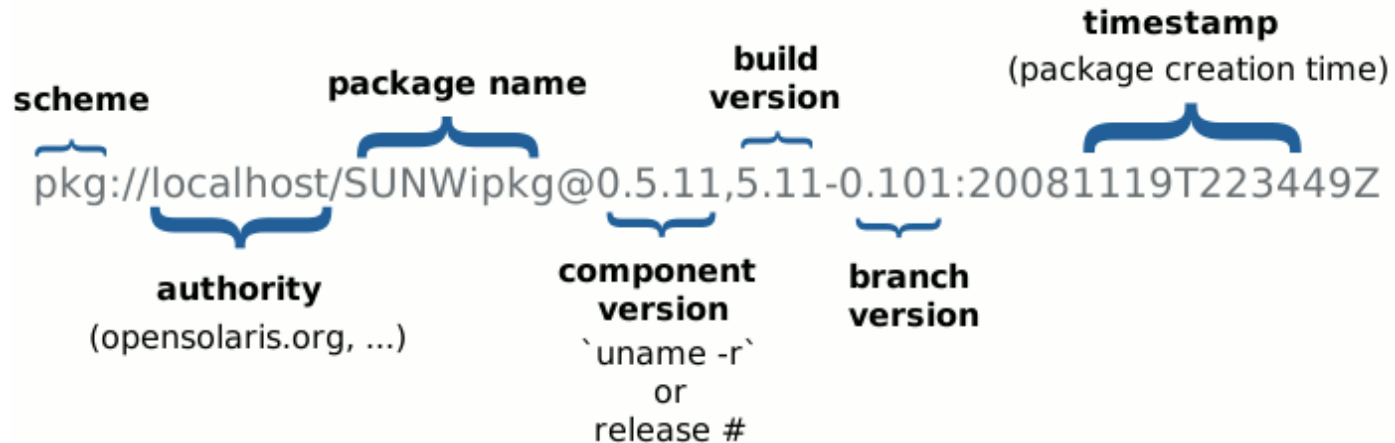
# FMRI - Fault Management Resource Identifier

---

- Each IPS package is represented by a Fault Management Resource Identifier (FMRI). The `pkg(1)` command uses valid FMRI package information to perform its command actions.
- The FMRI includes descriptive information about the package, such as the package name, version information, and date.

# FMRI scheme

---



# Metapackages & Incorporations

---

Typical pkg(5) package contains files and other actions (e.g. link, hardlink, user driver, ...)

But there are some packages just contains other packages. These are metapackages and incorporations.

# Metapackages or (package clusters)

---

- A metapackage is a package that consists entirely of dependencies of the type require.
- Installing a metapackage installs all of the packages it depends on.
- An example should be the `ss-dev` metapackage.

# Incorporations

---

- An incorporation is a package that consists entirely of dependencies of the type incorporate.
- Installing an incorporation ensures that packages already installed on the system are brought up to the release levels stated in the dependencies in the incorporation.
  
- Useful when application has more release families
- A good example is incorporation “entire”

# Example usage of pkg (1)

---

## Install / Update package

```
$ pfexec pkg install package_FMRI
```

*#This will take all the dependencies and get latest software*

*# good to use -n for a dry run action*

## Uninstall package

```
$ pfexec pkg uninstall package_FMRI
```

## Update entire image

```
$ pfexec pkg image-update # update icorporation entire
```

# Example usage of pkg (1)

---

**Search for file in local (installed) / remote package**

```
$ pkg search [-lr] ggrep
```

**Get package information**

```
$ pkg info [-lr] package_FMRI
```

# Example usage of pkg (1)

---

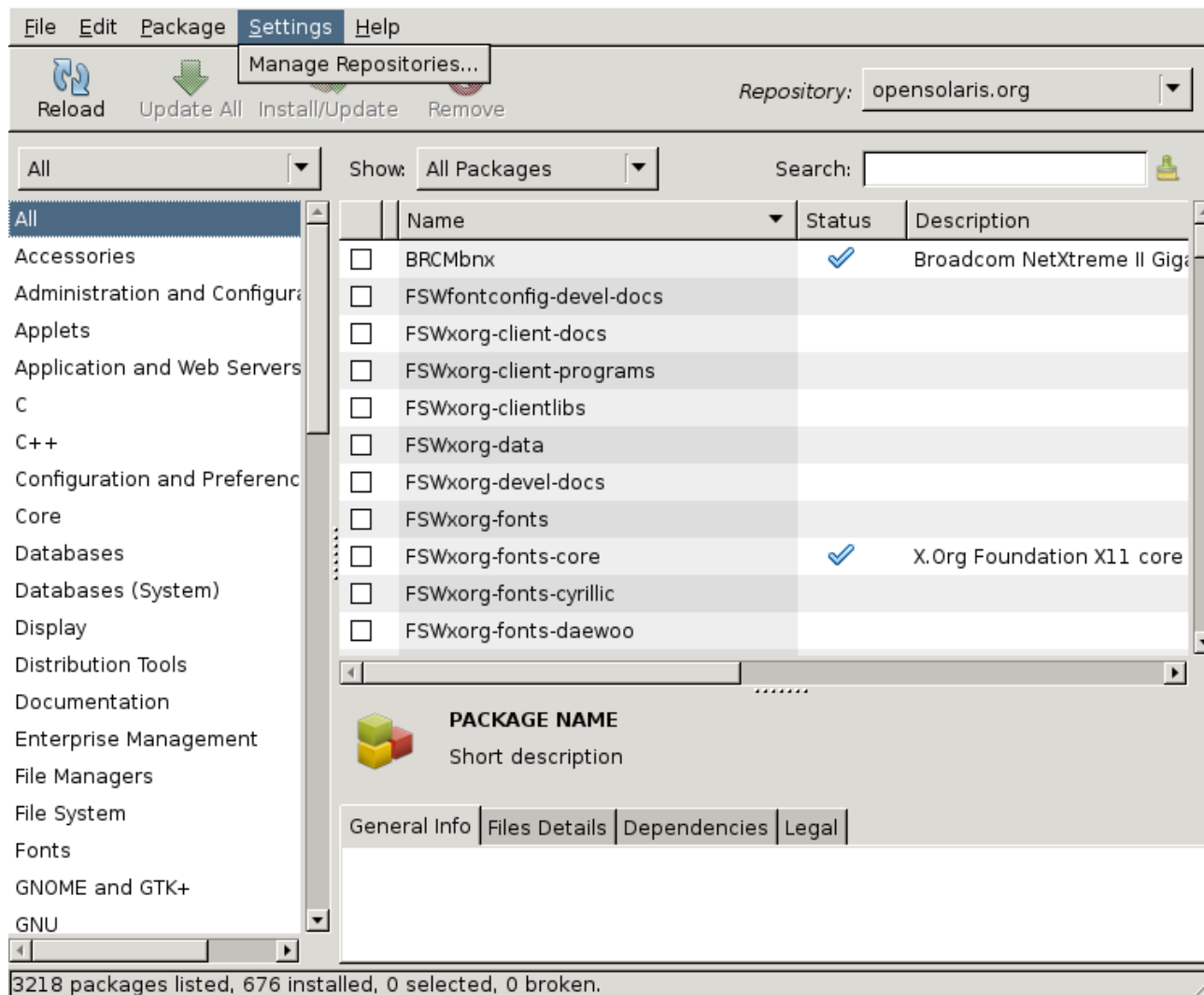
## Show dependencies of the package

```
$ pkg contents -t depend -o action.raw SUNWzfs
```

*<show output in terminal>*

**\$ see man -s 1 pkg for more information and examples**

# Package manager (picture for a thousand words)



# Some pkg authorities

---

- **<http://pkg.opensolaris.org>**
  - The default authority
  - # of packages: 22791
- **<http://blastwave.network.com:10000>**
  - Blastwave project
  - # of packages: 1714
- **<http://pkg.sunfreeware.com:9000>**
  - The Companion project (companion dvd)
  - # of packages: 297

# How to manage pkg authorities?

---

- Authorities can be managed via `pkg(1)` command

- Listing authorities

```
$pkg authority
```

- Adding authority

```
# pkg set-authority [-P] -O http://... Name
```

- Remove authority

```
#pkg unset-authority Name
```

# How to setup an IPS server

---



- via SMF services
- via pkg.depotd(1)

# How to set up an IPS server (smf way)

---

- How to set up pkg server via smf services?
- # svcadm enable application/pkg/server
  
- By default port is set to 80 and the server directory is /var/pkg
  
- Often used options
  - pkg/port
  - pkg/inst\_root

# How to set up an IPS server (the next one)

---

- You can also run
- `$ /usr/lib/pkg.depotd`
- By default port is set to 80 and the server directory is `/var/pkg/repo`
- Server could be started by normal user using this way
- Often used options
  - `-p [ port number ]`
  - `-d [ server directory ]`
  - `-s [ number of threads for requests ]` Def. 10

# How to publish package to IPS server

---



- pkgsend syntax & attributes
- Things to know before demo
- Demo

# pkgsend syntax

---

- pkgsend open [-en] package\_FMRI
- pkgesend add action arguments
- pkgsend import [-T] pattern
  - (pattern such as a SV4 package)

# pkgsend add actions

---

- depend
- dir
- driver
- file
- group
- hardlink
- legacy
- license
- link
- set
- user

To be done: service

# Things to know before demo

---

- There are two ways how to publish packages into the IPS repository
- pkgsend (1)
  - Creation of pkg(5) packages
  - Allows to publish svr4 package (do not resolve dependencies)
- solaris.py
  - Better when publishing multiple packages at once (resolving dependencies)
  - part of pkg sources (src/tools/distro-import/solaris.py)
  -
- **In the demo I'll work only with the pkgsend(1) utility**

# Things to know before demo

---

- Authority attribute is needed when we are working with authority different from <http://localhost:10000>

- Opening a transaction in different shells

## *In Bourne like shells*

```
eval` pkgsend [-s authority] open package_fmri`
```

## *In C like shells*

```
$ pkgsend [-s authority ] open package_fmri
```

```
export PKG_TRANS_ID=1228122616_pkg%3A%2Ftest  
%401.0.0%2C5.11%3A20081201T091016Z
```

```
$ setenv PKG_TRANS_ID 1228.....
```

# How to publish package to IPS server

---

PKG publication demo

# How to get involved?

---

## Get more information

<http://www.opensolaris.org/os/project/pkg/>

## Get the source

```
hg clone --ssh "ssh -C" ssh://anon@hg.opensolaris.org/  
hg/pkg/gate
```

*Join the discuss list, report bugs and contribute :-)*

# Q&A

---

Any questions?



**Thank you**

Lubos Kocman  
setuid@setuid.cc

<http://setuid.cc>

---