Image Packaging System

Michal Pryc
Michal.Pryc@Sun.Com
http://blogs.sun.com/migi
Agenda

- **14:30-14:45** - Setting up
- **14:45-14:55** - Introduction to the tutorial
- **14:55-15:10** - Common Build Environment - introduction and first sample package
- **15:10-15:20** - Break
- **15:20-15:50** - CBE - your first own svr4 package
- **15:50-16:00** - IPS - introduction
- **16:00-16:10** - Break
- **16:10-16:25** - IPS - setting up repository
- **16:25-16:45** - IPS - creating and sending package using pkgsend(1)
- **16:45-17:00** - IPS - creating and sending package from svr4 package
- **17:00-17:10** - Break
- **17:10-17:30** - Using pkg(1) and packagemanager to install packages in different images
What you will need:

✔ Patience and good humour
✔ Internet Access

✔ Operating system:
  ✔ OpenSolaris 2008.05
  ✔ Sun Studio
  ✔ Common Build Environment (CBE)

OR:

✔ VirtualBox
Installing CBE

Follow up the instructions in the materials:

3. Setting up environment
Common Build Environment #1

pkgtool
will do everything for us!

so just switch to CBE environment
and
rock&roll
Common Build Environment #2

• http_proxy needed?
• Flags:
  -- download (this will download sources)
  build (this will build and install package)

build-only (if you want only build packages, without installing, note, that it is easier to install packages, because other ones might depend on that package)
Common Build Environment #3

This is the default folder, where the packages are builded in svr4 format. It will be used in next parts of this presentation, so keep it in mind!

~/packages/PKGS
CBE - your first own svr4 package

Follow up the instructions in the materials:

4b. My first own svr4 package
How can I get some software?
Image Packaging System

pkg (5)
IPS overview #1

IPS - Project is under development

• Multi-platform
• Multi-install
• Non-root install
• Handling ELF dependencies
• Content filtering
• Virtualization-ready
• ZFS aware
• Filters
Three things that need to be clarified

• It is not a build environment
• Network based software lifecycle management
• No scripting
**FMRI** (Fault Management Resource Identifier)

```
pkg://opensolaris.org/library/libc@5.11,5.11-0.75:20071001T163427Z
```

- **Authority**: opensolaris.org
- **Scheme**: library
- **Package Name**: libc
- **Component Version**: 5.11, 5.11-0.75
- **Build Version**: 20071001T163427Z
- **Branch Version**: 'uname -r' or release number
- **Timestamp**: (when package was published)
pkg (5) in parts

IPS retrieval client
pkg (1)

IPS server
pkg.depot (1M)

Build machine
(svr4 packages)
solaris.py

Publication client
pkgsend (1)

HTTP/HTTPS

HTTP/HTTPS

HTTP/HTTPS

HTTPS/HTTPS
pkg.depot (1M)

IPS retrieval client
pkg (1)

Build machine
(svr4 packages)
solaris.py

IPS server
pkg.depot (1M)

Publication client
pkgsend (1)

HTTP/HTTPS
Setting up IPS repository

- using
  /usr/lib/pkg.depotd

- using smf service
  svc:/application/pkg/server
/usr/lib/pkg.depotd

- Possible options:
  -d – server directory
  -p – server port
  --readonly
  --rebuild
pkg.depot (1M)

- HTTP/HTTPS session
- SHA-1 hash for each file
- Default /var/pkg/repo
- May be run by normal user
svc:/application/pkg/server

• Possible options:
  pkg/port
  pkg/inst_root
Your task!

• Set up your local repository :-) 

5a. Setting up IPS repository
pkgsend (1)

IPS retrieval client
pkg (1)

IPS server
pkg.depot (1M)

Build machine
(svrr4 packages)
solaris.py

Publication client
pkgsend (1)
pkgsend (1)

- Allows publication of new packages
- Each publication is structured as a transaction
pkgsend (1)

Task
Your task!

- Create sample package in your repository!

5b. Creating and sending package using pkgsend(1)
svr4 → IPS server

IPS retrieval client
pkg (1)

HTTP/HTTPS

Build machine
(svr4 packages)
solaris.py

IPS server
pkg.depot (1M)

Publication client
pkgsend (1)

HTTP/HTTPS

HTTP/HTTPS
svr4 → IPS server

solaris.py

- handling ELF components
- finding package dependencies
- publishing
svr4 → IPS server

Demo
pkg (1)

IPS retrieval client
pkg (1)

IPS server
pkg.depot (1M)

Build machine
(svr4 packages)
solaris.py

Publication client
pkgsend (1)
pkg (1)

ips@opensolaris:~$ pkg
Usage:
    pkg [options] command [cmd_options] [operands]

Basic subcommands:
    pkg install [-nvq] package...
    pkg uninstall [-nrqv] package...
    pkg list [-aHsuv] [package...]
    pkg image-update [-nvq]
    pkg refresh [--full]
    pkg version
    pkg help

Advanced subcommands:
    pkg info [-lr] [--license] [pkg_fMRI_pattern ...]
    pkg search [-lr] [-s server] token
    pkg verify [-THqv] [pkg_fMRI_pattern ...]
    pkg contents [-Hmr] [-o attribute ...] [-s sort_key] [-t action_type ...]
    pkg_fMRI_pattern [...]
    pkg image-create [-FPUz] [--full|--partial|--user] [--zone]
        [-k ssl_key] [-c ssl_cert] -a <prefix>=<url> dir
    pkg set-authority [-P] [-k ssl_key] [-c ssl_cert]
        [-0 origin_url] authority
    pkg unset-authority authority ...
    pkg authority [-H] [authname]

Options:
    -R dir

Environment:
    PKG IMAGE
ips@opensolaris:~$
pkg (1)

Let's do something!

fmri matching

image plan

package plan

Install
Remove
Update

done
pkg (1)

TASK
GUI tool – Package Manager

“One picture tells a thousand words”
Thank you!

Michal Pryc
Michal.Pryc@Sun.Com
http://blogs.sun.com/migi

“open” artwork and icons by chandan:
http://blogs.sun.com/chandan