3. Installing software and package management

Option 1: Installing software from OpenIndiana repositories via IPS

OpenIndiana uses the Image Packaging System - IPS, pkg(5) , the same package management system as Oracle Solaris 11.
Oracle is continuing practice of providing well written manuals: http://download.oracle.com/docs/cd/E19963-01/html/820-6572/index.html

To find more information on IPS/ pkg(5), using man command, use:

```
$ man -s 5 pkg
```

The command line package manager command is '**pkg**'.
To find more information about pkg(1) , using man command, use:

```
$ man pkg
```

In IPS parlance, we are talking about adding **publishers** in place of term **repositories**.
Package management is performed using accounts with elevated **Role Based Access Control** (**RBAC**) privileges (using **pfexec** or **sudo** before commands shown below).

Adding publisher

```
$ pfexec bash
# pkg set-publisher -g repository_url repository_name
```

Replacing publisher

```
# pkg set-publisher -G old_repository_url -g new_repository_url repository_name
```

Replacing publisher - example

```
# pkg set-publisher \
-G http://pkg.openindiana.org/hipster-2015 \
-g https://pkg.openindiana.org/hipster openindiana.org
```

Listing publishers configured on the local system:

```
# pkg publisher

PUBLISHER                TYPE     STATUS P LOCATION
openindiana.org          origin   online F https://pkg.openindiana.org/hipster/
hipster-encumbered       origin   online F https://pkg.openindiana.org/hipster-encumbered/
```

'OpenIndiana hipster' repositories (as of February, 2017)

openindiana.org	https://pkg.openindiana.org/hipster	Hipster primary package repository
hipster-encumbered	https://pkg.openindiana.org/hipster-encumbered	Hipster encumbered license packages
localhostoih	http://sfe.opencsw.org/localhostoih	3rd party Spec Files Extra (SFE) packages

Syntax for searching remotely for IPS packages

```
# pkg search -rp package_name
```

Syntax for installing an IPS package

```
# pkg install package_name
```

Syntax for upgrading a package

```
# pkg update package_name
```

Upgrading all the packages on the system, with linked image Solaris Zones. (-r) and verbose output (-v)

```
# pkg update -v -r
```

Option pkg -r updates all Solaris Zones , if not issued, thay are kept unupdated untill -r command is issued on update or they are updated separately.

Option 2: Install software from SmartOS repositories via pkgin

All illumos-based operating systems, like OmniOS, Openindiana and SmartOS can use the repository from Joyent/SmartOS. Its main advantage is that you find there a lot of very up to date packages.

A list of available software: http://pkgsrc.joyent.com/packages/SmartOS/ in folder http://pkgsrc.joyent.com/packages/SmartOS/ (or http://pkgsrc.smartos.org/packages/illumos/).

If you want to install software via `pkgin` (installs every package to `/opt`), you need to (console as `root`):

1. add `/opt/local/{s,}bin` where all software is installed to your PATH (in your shell, maybe save to your `.profile`):

   ```
   PATH=/opt/local/sbin:/opt/local/bin:$PATH
   export PATH
   ```

2. install the `bootstrap-loader`: (use the loader according to your repository, see http://pkgsrc.joyent.com/packages/SmartOS/bootstrap/)

   ```
   :; curl https://pkgsrc.joyent.com/packages/SmartOS/bootstrap/bootstrap-2016Q4-x86_64.tar.gz | gtar -zxpf
   - -C /
   ```

3. update the repository database:

   ```
   :; pkgin -y update
   ```

4. install the needed package, for example – Apache 2.4.6:

   ```
   :; pkgin -y install apache-2.4.6
   ```

   or, just for newest 2.4:

   ```
   :; pkgin -y install apache-2.4
   ```

For more information see:

- http://wiki.smartos.org/display/DOC/Working+with+Packages
- http://www.perkin.org.uk/posts/whats-new-in-pkgsrc-2013Q2.html

- newest repository:http://pkgsrc.joyent.com/packages/SmartOS/

Option 3: Compile yourself

You need a compiler like `gcc`; download the sources, switch to the folder with your sources (make the content of your 'sourcefolder' executable recursively) and compile via:

```
:; cd /sourcefolder
:; ./configure
:; make
:; make install
```

You might want to first look into `./configure --help` to see what options are available for building the software – quite often, some features useful for you might notbe a general choice enabled by default, or might require other dependency software to be available first in binary or source form, in order to compile.

For building software from NetBSD pkgsrc from sources, see page: pkgsrc in OI .

Option 4: Build software using oi-userland

For contributing to oi-userland, see page: Building with oi-userland