

# FreeBSD ports & packages



# FreeBSD ports & packages - overview

- Different UNIX distributions use different package systems for distributing software
  - Debian GNU/Linux and Ubuntu use .DEB
  - SuSE, RedHat, Fedora use .RPM
  - etc..
- FreeBSD uses a simple format (.TGZ)
- Where do they come from ?



# Installing software on FreeBSD

- Normally 3 ways to install software on FreeBSD:
  - from the « source »  
`# ./configure; make; make install`
  - from the port
  - from the package



# The FreeBSD ports collection

- Described in detail at [http://www.freebsd.org/doc/en\\_US.ISO8859-1/books/handbook/ports.html](http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ports.html)
- The ports is a collection of « skeleton » instructions (Makefile, patches) to retrieve, configure, build and install software
- The ports collection contains instructions for over 16.000 software programs as of 2007, and it is still growing.



# Installing software ...

- Open source software is shipped in source form; to be usable on a system:
  - 1. It must be unpacked
  - 2. It must be adapted to the system one is running on (« porting »)
  - 3. It may be compiled (source -> binary)
  - 4. It must installed
  - 5. It may need to be configured



# Installing software ...

- The ports system takes care of steps 1 – 4, and sometimes 5 (preconfiguration)
- The build and installation process might trigger the build and installation of other required software – this is known as « dependencies ».
  - For example, a graphics program might need a JPEG software library to be able to read and write JPEG images



# From port to package

- Once the port is built, one can make a **package**
- A package is a pre-built port
- Packages can be installed...
  - from the network via FTP
  - from the FreeBSD CDRROM
- Packages can be those built by the FreeBSD project or your own



# Ports vs packages

## • **PACKAGES**

- precompiled
- easy to install
- no need to have ports collection installed

## **PORTS**

- more and better configuration control / options (for example Apache and PHP)
- local patches possible
- tuning options





# Installing packages

- `pkg_add [-r] <package name>`
- For instance:
  - `# pkg_add -r bash`
- This will attempt to install the bash package from the network
- Problems:
  - what if bash depends on other software ?
  - which version of bash ? 2 or 3 ?
- `pkg_add` will try to install dependencies, if they are available



# Installing packages

- If you have the package on CD, and the CD is mounted, you can install it directly, for instance:

```
# pkg_add /cdrom/packages/All/bash-4.0.10_2.tbz
```

- If you know the URL (Link) to the package on the Internet, you could:

```
# pkg_add ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/7.2-RELEASE/packages/All/bash-4.0.10_2.tbz
```



# Ports

- To install a port, you can search by name or keyword:
  - # cd /usr/ports; make search name=<name>
  - # cd /usr/ports; make search key=<keywrd>
- Once you know where the port resides (its **category**), you can go to that directory, and install it:
  - # cd /usr/ports/**shells**/bash
  - # make
  - # make install
  - # make clean
- That's it!



# Ports

- If you want to build a **package**, you just need to type:

# make package

- ... from the directory where you built the port from.



# Best of both worlds

- But what if...
  - you don't know in which category the port is located or which version you want;
  - you can't find up-to-date packages for the version of FreeBSD you are running (maybe it's a bit older) ;
  - you want to upgrade a package, but other packages depend on it ;
- For all the above reasons, it is strongly recommended to use a tool called **portupgrade**

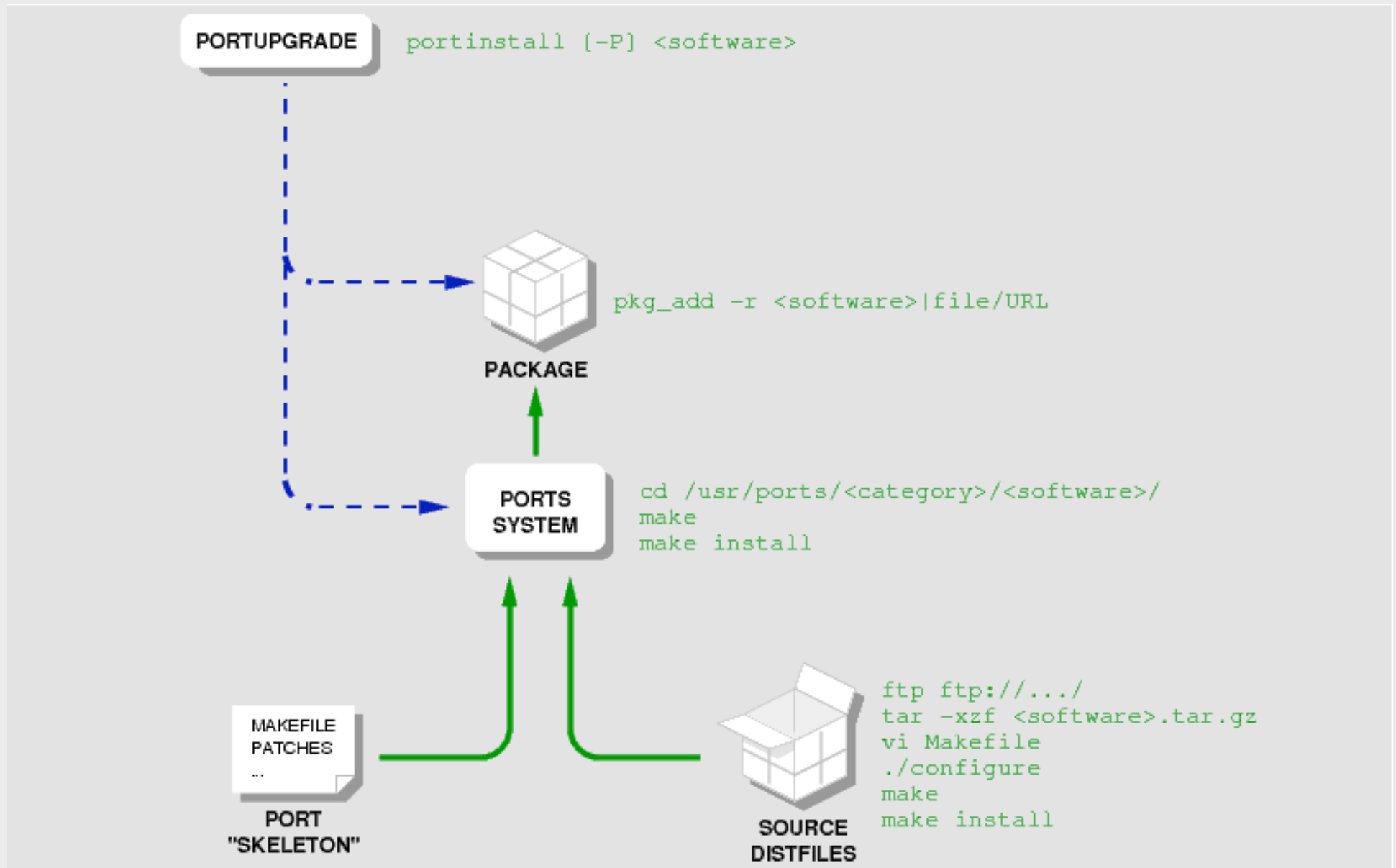


# Portupgrade

- **Portupgrade** is a « meta » package manager. It sits on a layer above the ports and package system, and makes your life easier
- Portupgrade greatly simplifies package installation, upgrades.



# The big picture



# Installing portupgrade

- Portupgrade is itself a port, and can be installed as such, or as a package:

```
# cd /usr/ports/sysutils/portupgrade
```

```
# make
```

```
# make install
```

```
# make clean
```

- or

```
# pkg_add -r portupgrade
```

- It can also be installed at system install time (via sysinstall)





# Using portupgrade

- To install a program with portupgrade

```
# portinstall <port name>
```

- For instance:

```
# portinstall bash
```



# Using portupgrade

- Portupgrade can be told to try and install a pre-built package, using the -P option, one or more times:
  - Try and install from a package, fallback to the port if the package isn't found:

```
# portinstall -P <port name>
```

- Try and install from a package, stop if it's not found:

```
# portinstall -P -P <port name>
```



# Using portupgrade

- To upgrade an already installed software package:

```
# portupgrade <package name>
```

- For instance:

```
# portupgrade apache
```



# Summary

- FreeBSD has three main ways to install software
  - from the source
  - from the port
  - from the package
- Ports are “skeletons” that help you build from source and make sure it works with FreeBSD
- portupgrade is an interesting port that helps you manage your other ports.



# Questions ?



# Extra: Building FreeBSD from source

FreeBSD ships a copy of the source code in `/usr/src` as well as the tools needed to compile this as well.

Once built it allows you to install and make sure the configuration for the new system is correct in case changes have been made between FreeBSD versions.

Documentation of what to do is both on the system and online on the FreeBSD website.

*\* we may or may not get the chance to do this in class depending on how fast or slow we go.*

