



Simplify your external  
dependency management

Stephan Hochdörfer, bitExpert AG

## About me

- Stephan Hochdörfer, bitExpert AG
- Department Manager Research Labs
- enjoying PHP (and a little bit of Java) since 1999
- S.Hochdoerfer@bitExpert.de
- @shochdoerfer

Simplify your ext. dependency management

---

## Dependencies?



## Dependencies?

internal  
Dependencies



external  
Dependencies

Simplify your ext. dependency management

---

## What are external dependencies?



## What are external dependencies?

Anything your application requires  
to run...

## What are external dependencies?

Extensions, Systemtools, PHP Libs



Simplify your ext. dependency management

Our first project....





Simplify your ext. dependency management

---

Our first project....



+



## Our first project – the structure

```
yourproject
| -application
| -library
| - - -Facebook
| - - -Zend
| -public
| - - -js
| - - -css
| -tests
```

## And some more projects...

yourproject

- | -application
- | -library
- | - - -Facebook
- | - - -Zend
- | -public
- | - - -js
- | - - -css
- | -tests

yourproject2

- | -application
- | -library
- | - - -Zend
- | -public
- | - - -js
- | - - -css
- | -tests

Simplify your ext. dependency management

A close-up, high-contrast photograph of a vampire's mouth. The lips are parted, revealing a row of sharp, yellowish fangs. The lighting is dramatic, highlighting the texture of the skin and the sharp edges of the teeth. The background is dark and out of focus.

**Maintenance  
nightmare!**

Simplify your ext. dependency management

---

First try...

Let`s use what we already have...



Simplify your ext. dependency management

---

First try...

...our version control system!



Simplify your ext. dependency management

---

First try...

For Subversion users: SVN externals





## First try – SVN externals

```
yourproject
| -application
| -library
| - - -Facebook
| - - -Zend
| -public
| - - -js
| - - -css
| -tests
```

## First try – SVN externals

```
yourproject
|-application
|-library
|---Facebook
|-public
|---js
|---css
|-tests
```

### 1. Step

Delete Zend directory

## First try – SVN externals

```
yourproject
| -application
| -library
| - - - Facebook
| -public
| - - - js
| - - - css
| -tests
```

### 2. Step

```
svn propedit \  
svn:externals library
```

Add line:

```
Zend/ http://framework...
```

## First try – SVN externals

```
yourproject
| -application
| -library
| - - - Facebook
| - - - Zend
| -public
| - - - js
| - - - CSS
| -tests
```

### 3. Step

svn commit &&  
svn update

## SVN externals - Recap

- Links local dirs to remote SVN dirs
- Synced on every (local) update
- Pull-based system

## SVN externals - Drawbacks

- You can only reference other SVN repos!
- You need access rights for the external repo!
- You can commit to externals!

## SVN externals – Pro tip I

Always link to a tag  
or branch, not to trunk!



## SVN externals – Pro tip II

Add `svn:externals` only to direct parent directory!

Simplify your ext. dependency management

---

First try – Alternatives?



Simplify your ext. dependency management

---

## First try – Alternatives?

Git + Git submodules



## Git submodules

- Clone external history and re-merge local changes
- is a checkout of an external (Git) project
- partial checkout possible
- pull-based system
- you need to manually merge fixes in local repo!

Simplify your ext. dependency management

---

Some recap



## Some recap

A VCS is not made for  
dependency management!

Simplify your ext. dependency management

---

What else?





What else?

Requirements first!

## What else?

1. Do not hide dependency metadata

## What else?

### 2. Support transitive dependencies

## What else?

### 3. Seperation from VCS

## What else?

4. Do not reinvent the wheel

Simplify your ext. dependency management

---

What else?



## What else? - Packaging standards

\*nix packages

PEAR

## What else? - Build Tools

Phing

Maven



Simplify your ext. dependency management

---

\*nix packages – 4 steps to enlightenment

1. Create directory yourproject-0.0.1



Simplify your ext. dependency management

---

\*nix packages – 4 steps to enlightenment

2. dh\_make -createorig -s



Simplify your ext. dependency management

---

\*nix packages – 4 steps to enlightenment

### 3. Add Makefile



## \*nix packages – 4 steps to enlightenment

```
BINDIR = $(DESTDIR)/usr/bin
clean:
install:
    mkdir -p $(BINDIR)
    cp info.php $(BINDIR)
uninstall:
    rm -f $(BINDIR)/info.php
```

Simplify your ext. dependency management

---

\*nix packages – 4 steps to enlightenment

4. fakeroot debian/rules binary



Simplify your ext. dependency management

---

\*nix packages – Dependencies?



Simplify your ext. dependency management

---

\*nix packages – Dependencies?

Add them to debian/control file



## \*nix packages - Pros

- Independant of your repo state, build it manually
- Dependency metadata not hidden!
- \*nix packages can depend on other packages
- Allows dependencies on other system packages



## \*nix packages - Cons

- Packaging diversity: .deb, .rpm, .tgz, ...
- Only one version of a package for one server

Simplify your ext. dependency management

---

## PEAR package – 3 steps to enlightenment

1. Create directory yourproject-0.0.1



Simplify your ext. dependency management

---

## PEAR package – 3 steps to enlightenment

### 2. Create package.xml



## PEAR package – 3 steps to enlightenment

```
<?xml version="1.0" encoding="UTF-8"?>
<package packagerversion="1.8.0" version="2.0"
xmlns="http://pear.php.net/dtd/package-2.0"
xmlns:tasks="http://pear.php.net/dtd/tasks-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://pear.php.net/dtd/tasks-1.0
    http://pear.php.net/dtd/tasks-1.0.xsd
    http://pear.php.net/dtd/package-2.0
    http://pear.php.net/dtd/package-2.0.xsd">
  <name>Yourapp</name>
  <channel>pear.yourapp.com</channel>
  <summary>Yourapp summary</summary>
  <description>Yourapp description.</description>
  <lead>
    <name>Stephan Hochdoerfer</name>
    <user>sh</user>
```

Simplify your ext. dependency management

---

## PEAR package – 3 steps to enlightenment

### 3. pear package



## PEAR packages - Pros

- Similar to \*nix packages, but platform independant
- Transitive dependency management
- Multiple versions of a package on one server (with work-a-round)
- PEAR is a PHP standard way of distribution
- Easy to set-up a server (Pyrus, Pirim, ...)

## Phing

- Is a build tool, is not aware of dependency management!
- Can communicate with CVS, SVN, Git, ...
- You need to provide your own dependency logic
- No task to automatically deploy to PEAR / Pyrus / PIRUM

## What else?

Unfortunately the PHP community misses ONE tool to rule them all!



Simplify your ext. dependency management

---

What else?

Java, anyone?



## What is Maven?

Maven is a declarative project management tool.

## Maven - Project Object Model

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <groupId>com.zend</groupId>
  <artifactId>framework</artifactId>
  <packaging>jar</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>Zend Framework</name>
</project>
```

## What can Maven do?

After first downloading it,  
almost nothing, but...

## What can Maven do?

- Run goals
- Run phases (which are a collection of goals)
- Download (transitive) dependencies automatically
- Download plugins as well!
- Once downloaded provides offline access to the packages

## Maven Beginners guide

### 1. Add Maven PHP repository

## Maven Beginners guide

```
<pluginRepository>
  <id>release-repo1.php-maven.org</id>
  <name>PHP-Maven 2 Release Repository</name>
  <url>http://repo1.php-maven.org/release</url>
  <releases>
    <enabled>true</enabled>
  </releases>
</pluginRepository>
```

## Maven Beginners guide

### 2. Create new Maven project



## Maven Beginners guide

```
mvn archetype:generate \  
-DarchetypeGroupId=org.phpmaven \  
-DarchetypeArtifactId=php5-lib-archetype \  
-DarchetypeVersion=1.1 \  
-DgroupId=org.yourproject \  
-DartifactId=appname
```

## Maven Beginners guide

### 2. Deploy your project

Simplify your ext. dependency management

---

## Maven Beginners guide

```
mvn deploy
```



## Maven Bonus points

- Lot`s of plugins, eg. Liquibase
- Automatic release management
- Automatic deployment
- Build profiles
- IDE support, e.g. Eclipse



Thank you!

## Image Credits

- <http://www.sxc.hu/photo/1092493>