DEPLOYMENT

WITH

PHING

2013-12-03

Modern PHP User Group

PHING

- **PH**ing **I**s **N**ot **G**NU make
- http://www.phing.info
- 아파치 Ant 프로젝트 기반
- PHP 언어로 개발
- XML로 정의

INSTALLATION

```
$ pear channel-discover pear.phing.info
```

```
$ pear install [--alldeps] phing/phing
```

COMPONENTS

- Project
- Target
- Task
- Type
- Condition

PROJECT

- 최상위 엘리먼트로 사용
- 프로젝트 이름, 디폴트 태스크, 설명 등을 기술
- 디폴트 파일은 build.xml
- -f 옵션으로 별도 지정 가능

```
<project name="..." default="...">
    <target>
        ...
    </target>
</project>
```

TARGET

- 작업(Task)들을 묶은 덩어리
- depends
  - 의존 관계가 있는 다른 Target의 name
  - 콤마(,)로 여러개를 적을 수 있다
  - 실행 순서는 보장되지 않는다

```
<target name="..." depends="..." description="...">...</target>
```

TASK

- 하나의 작업
- 주요 작업들
  - property(변수), echo(출력), if(조건문), foreach(반복문)
  - exec(외부 실행), fail(단정문), phingcall(호출)
  - reflexive(재귀)
- Core - 38개, Optional - 85개

TYPE - FILESET

- 파일 집합을 규칙으로 지정
- include, exclude, patternset을 포함

```
<fileset dir="/etc">
    <include name="httpd/**" />
</fileset>
```

```
<fileset dir="/etc">
    <patternset>
        <include name="**/*.php" />
        <exclude name="**/*.bak" />
    </patternset>
</fileset>
```

FILELIST

- 파일 목록을 직접 지정

```
<filelist dir="/etc" files="httpd/conf/httpd.conf,php.ini"/>
```

FILTERCHAIN

- 19개 필터
- ReplaceTokens, Head, Iconv, Tail, Tidy 등
- 유닉스 파이프 개념으로 이해하면 쉽다

FILEMAPPER

- 5개 맵퍼
- Flatten, Glob, Identity, Merge, Regexp
- 필터는 파일 내용을 바꾸고, 맵퍼는 파일 이름을 바꾼다

CONDITION

- 조건 및 논리 연산
- equals, isset, contains, istrue, isfalse, available
- not, and, or

```
<not>
    <equals arg1="abc" arg2="abc" />
</not>
```

```
<if>
    <and>
        <equals arg1="${name}" arg2="bookworm" />
        <contains string="${name}" substring="worm" />
    </and>
    <then>
        <echo msg="I'm a ${name}" />
    </then>
</if>
```

JUST DO IT!

TIP

- 각 컴포넌트 사이에는 조합 또는 내포 가능 여부가 정해져 있다.
- 조합 또는 내포 가능 여부는 매뉴얼을 통해서 확인 가능하다.
- 불명확한 조합은 테스트 코드를 만들어서 실험을 해보라.
- 오류 발생 시 메시지를 살펴 보는 것이 원인 파악이 도움이 된다.
- 오류 파악이 힘든 경우 Phing 소스 코드를 디버깅 하면 쉽게 파악되기도 한다
- 필요한 Task가 없는 경우 구글링 하라.
- Exec와 PhpEval은 만능에 가깝지만 되도록 이미 정의된 Task를 사용하라.
- 자주 쓰이는 독자적인 기능은 사용자 정의 Task로 만드는 것이 낫다.

ONE MORE THING...

INSTALLING PEAR WITH COMPOSER

이제는 컴포저로 PEAR를 설치하세요.

PHPBrew & Homebrew와 궁합이 좋습니다.

```
{
    "repositories": [
        {
            "type": "pear",
            "url": "http://pear.php.net"
        }
    ],
    "require": {
        "phing/phing" : "2.6.*",
        "pear-pear.php.net/VersionControl_Git": "0.4.*"
    }
}
```

THANK YOU