



## Groove Is In The Ant

Gilles QUERRET  
Riverside Software



---

---

---

---

---

---

---

---

---

---

### About The Speaker

- Started Riverside Software 10 years ago
- Based in Lyon, France
- Continuous integration and technical expertise around Java / OpenEdge
- Code analysis for OpenEdge

---

---

---

---

---

---

---

---

---

---

### Ant in build automation

- Lots of tasks used in build automation
- Not specifically meant for deployment
- PCT widely used in the OpenEdge world
  
- Deployment needs more...
- So your script grew...

---

---

---

---

---

---

---

---

---

---

Pointy-Bracket Hell?



---

---

---

---

---

---

---

---

Ant-Contrib Hell...



---

---

---

---

---

---

---

---

Here's Groovy !



---

---

---

---

---

---

---

---

## What's Groovy?

- Dynamic language for the Java Virtual Machine
- Built on Java, gives access to the entire Java ecosystem
- Easy to learn (especially if you know Java)
  - A bit more difficult to master ☹
- Used in Jenkins

---

---

---

---

---

---

---

---

## Hello, PUG Challenge!

```
Hello.groovy
1 println "Hello, PUG Challenge"
```

```
gquerret ➤ groovy Hello.groovy
Hello, PUG Challenge
gquerret ➤
```

---

---

---

---

---

---

---

---

## GStrings

```
Var1.groovy
1 def x = "PUG Challenge"
2 println "Hello, $x"
```

```
gquerret ➤ groovy Var1.groovy
Hello, PUG Challenge
gquerret ➤
```

---

---

---

---

---

---

---

---

## Multi-line Strings

```

Var2.groovy
1 def str = """This is
2   a multiline
3   string"""
4 println str

```

```

gquerret → groovy Var2.groovy
This is
a multiline
string
gquerret →

```

---

---

---

---

---

---

---

---

## Regular Expressions

```

regexp.groovy
1 def m = "Groovy is groovy" =~ /(G|g)roovy/
2 println m[0][0] // The first whole match (i.e. the first word Groovy)
3 println m[0][1] // The first group in the first match (i.e. G)
4 println m[1][0] // The second whole match (i.e. the word groovy)
5 println m[1][1] // The first group in the second match (i.e. g)

```

```

gquerret → groovy regexp.groovy
Groovy
G
groovy
g
gquerret →

```

---

---

---

---

---

---

---

---

## Closures

```

closure.groovy
1 def fn = { name ->
2   println "Hello, $name"
3 }
4 fn('PUG Challenge')

```

```

gquerret → groovy closure.groovy
Hello, PUG Challenge
gquerret →

```

---

---

---

---

---

---

---

---



## Ant support in Groovy

- Supported by default in Groovy

```
ant.groovy
1 def ant = new AntBuilder()
2 ant.echo('hello from Ant!')

groovy ant.groovy
(echo) hello from Ant!
```

---

---

---

---

---

---

---

---

---

---

## Ant support in Groovy

```
ant2.groovy
1 ant.zip(destfile: 'sources.zip', basedir: 'src')
2 ant.mkdir(dir: myDir)
3 ant.copy(todir: myDir) {
4   fileset(dir: "src/test") {
5     include(name: "**/*.groovy")
6   }
7 }
```

---

---

---

---

---

---

---

---

---

---

## Groovy support in Ant ☺

```
groovy.xml
1 <?xml encoding="utf-8"?>
2 <project>
3   <taskdef name="groovy" classname="org.codehaus.groovy.ant.Groovy" />
4   <groovy>
5     println "Hello, world!"
6   </groovy>
7 </project>
```

---

---

---

---

---

---

---

---

---

---



## PCT XMLNS

```
xmlns.groovy
1 def ant = new AntBuilder()
2 ant.taskdef(uri: 'antlib:eu.rssw.pct', resource: 'eu.rssw/pct/antlib.xml')
3 def pct = groovy.xml.NamespaceBuilder.newInstance(ant, 'antlib:eu.rssw.pct')
4 pct.run(procedure: 'rssw.p', dlcHome: System.env['DLC'])
```

---

---

---

---

---

---

---

---

## What for ?

- Deployment script
- Dev environment setup
- Database update script
- ...

---

---

---

---

---

---

---

---

## Download

- <https://github.com/Riverside-Software/groovy-pct>

---

---

---

---

---

---

---

---



## Groovy fun...

```
fun.groovy
1 10.times { println it }
```

```
gquerret ~> groovy fun.groovy
0
1
2
gquerret ~> █
```

---

---

---

---

---

---

---

---

## Groovy fun...

```
fun.groovy
1 def str = 'example of method reference'
2 def fun = str.&toUpperCase
3 def upper = fun()
4 assert upper == str.toUpperCase()
```

---

---

---

---

---

---

---

---

```
fun.groovy
1 class Car {
2     String make
3     String model
4 }
5 def cars = [
6     new Car(make: 'Peugeot', model: '508'),
7     new Car(make: 'Renault', model: 'Clio')]
8 def makes = cars*.make
9 assert makes == ['Peugeot', 'Renault']
```

---

---

---

---

---

---

---

---

Questions?



---

---

---

---

---

---

---