



# Gant

Replacing XML with Groovy

Orientation in Objects GmbH

Weinheimer Str. 68  
68309 Mannheim

[www.oio.de](http://www.oio.de)  
[info@oio.de](mailto:info@oio.de)

Version: 1.0



## Gliederung

- Buildmanagement & Tools
- Sick of XML
- Getting started
- Resume
- What's next?

Falk Sippach

*Training*

*Beratung*

*Projekte*



# Java und XML

Besuchen Sie uns:  
OIO @ JAX Expo

## ) Software Factory )

- Schlüsselfertige Realisierung von Java Software
- Individualsoftware
- Pilot- und Migrationsprojekte
- Sanierung von Software
- Software Wartung

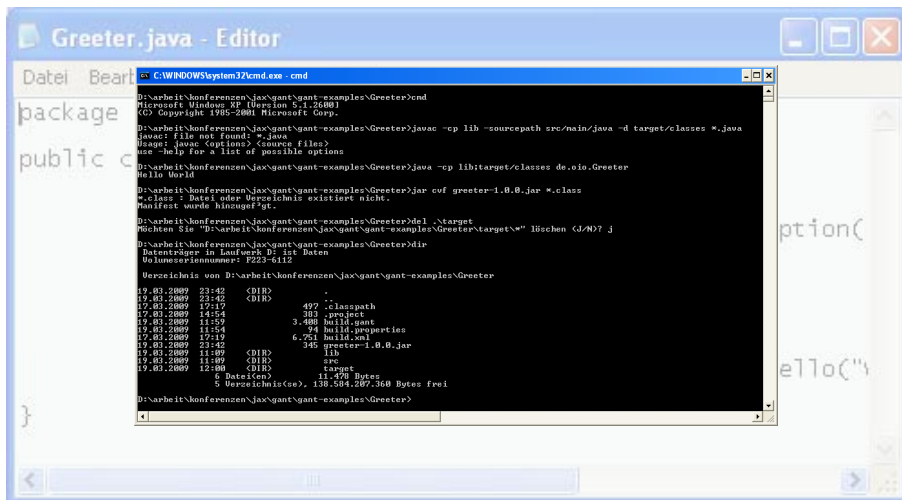
## ) Object Rangers )

- Unterstützung laufender Java Projekte
- Perfect Match
- Rent-a-team
- Coaching on the project
- Inhouse Outsourcing

## ) Competence Center )

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme
- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmensweiten Systemen

- Buildmanagement & Tools
- Sick of XML
- Getting started
- Resume
- What's next?



```
Greeter.java - Editor
Datei Bearb... C:\WINDOWS\system32\cmd.exe - cmd
package ...
public class Greeter {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}

D:\arbeitskonferenzen\jax\gant\gant-examples\Greeter>cmd
Microsoft Windows [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\arbeitskonferenzen\jax\gant\gant-examples\Greeter>javac -cp lib \sourcepath src\main\java -d target\classes *.java
javac: file not found: *.java
Usage: javac [options] [source files]
use -help for a list of possible options

D:\arbeitskonferenzen\jax\gant\gant-examples\Greeter>java -cp lib;target\classes de.oio.Greeter
Hello World

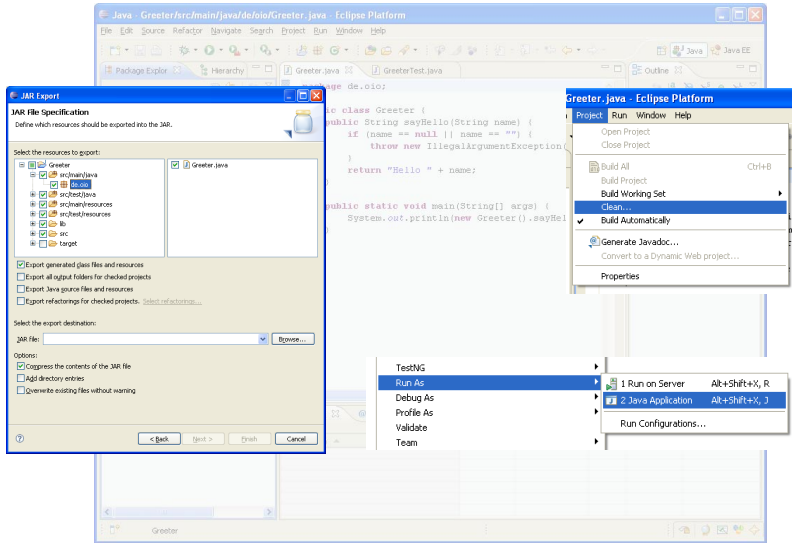
D:\arbeitskonferenzen\jax\gant\gant-examples\Greeter>jar cvf greeter-1.0.0.jar *.class
*.class: Datei oder Verzeichnis existiert nicht.
Manifest wurde hinzugefuegt.

D:\arbeitskonferenzen\jax\gant\gant-examples\Greeter>del *target
Wuechten Sie "D:\arbeitskonferenzen\jax\gant\gant-examples\Greeter>target\*" loeschen (Y/N)? J

D:\arbeitskonferenzen\jax\gant\gant-examples\Greeter>dir
Dateienverzeichnis in Laufwerk D:
Datei                Bytes
Verzeichnisnummer: F223-6412

Verzeichnis von D:\arbeitskonferenzen\jax\gant\gant-examples\Greeter
19.03.2009 23:42 <DIR>          .
19.03.2009 23:42 <DIR>          ..
19.03.2009 17:19 <DIR>          .497 -classpath
19.03.2009 14:54 <DIR>          .383 project
19.03.2009 11:59 <DIR>          .3.488 build.gant
19.03.2009 11:54 <DIR>          .24 build.properties
19.03.2009 17:19 <DIR>          .6.751 build.xml
19.03.2009 23:40 <DIR>          lib
19.03.2009 11:09 <DIR>          src
19.03.2009 11:09 <DIR>          target
19.03.2009 12:08 <DIR>          6 Dates(Ge), 11 <KB Bytes
19.03.2009 12:08 <DIR>          5 Verzeichnis(Ge), 138.584.207.368 Bytes Frei

D:\arbeitskonferenzen\jax\gant\gant-examples\Greeter>
```



## Java-Welt:

- Ant
- Maven
- **Builder**
- **Gant**

external DSL

## andere Sprachen:

- Make, CMake
- Autotools
- NAnt
- **SCons**
- **Rake, Raven, Rant**

internal DSL

## Gant in Kürze

- Version 1.6.1
  - Standalone (mitgeliefertes Groovy 1.6)
  - oder für vorinstallierte Groovy-Versionen 1.5+
- Apache License 2.0
- Kommandozeilen-Tool, Bedienung ähnlich wie Ant

## Gliederung

- Buildmanagement & Tools
- **Sick of XML**
- Getting started
- Resume
- What's next?

- XML, das "häßliche Entlein"
- falscher Abstraktionsgrad
  - "XML is not a programming language."
  - "XML is fantastic for data exchange but when you need conditional or iterative logic, things just begin to break down."
- zu deklarativ
  - Mißbrauch von Targets als Prozeduren
  - Kontrollflußstrukturen (if/else)
  - schlecht wiederverwendbar

- Mißbrauch von Targets als Prozeduren
- Kontrollflußstrukturen
- Ant Skripte bzw. Teile umständlich wiederverwendbar (DRY)

- "The problem with Ant is that it violates something we learned with Unix. **Tasks aren't composable.**"
- "**Ant** is nothing more than the **sum of its parts.**"
- "James Gosling once said that **every configuration file** becomes a **programming language**, so you might as well think that way."

[http://weblogs.java.net/blog/arnold/archive/2003/06/duncan\\_davidson.html](http://weblogs.java.net/blog/arnold/archive/2003/06/duncan_davidson.html)

- Buildmanagement & Tools
- Sick of XML
- **Getting started**
- Resume
- What's next?

## Was ist Gant nicht?

- **kein** Out-Of-The-Box-Buildframework wie Gradle oder Maven
- **keine** Neuimplementierung von Ant



## Was ist Gant dann?

- "Skriptsprache" für Ant-Tasks (Groovy statt XML)
- leichtgewichtige Hülle um Groovys AntBuilder
- Kommandozeilen-Tool: `>gant [option]* [target]*`





```
<target name="hello" description="Saying Hello">
  <echo message="Hello World"/>
</target>
```

> ant hello

```
target(hello:"Saying hello") {
  echo(message:"Hello World")
}
```

> gant hello

## Umwandlung: Ant → Gant

- Tool Ant2Gant
- manuell

<property />	Groovy final Variablen
<target />	target-Methode mit Groovy Closure
Ant Tasks: <javac />, <jar />	AntBuilder-Aufrufe

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<project default="run" name="HelloGant">
  <property file="build.properties" />
  <property name="src.dir" value="src" />
  <property name="target.dir" value="target" />

  <target name="clean" description="cleans up the build directory">
    <delete dir="${target.dir}" />
  </target>

  <target name="compile" description="compiles the sources">
    <mkdir dir="${target.dir}" />
    <javac srcdir="${src.dir}" destdir="${target.dir}" />
  </target>

  <target name="run" depends="compile" description="runs project">
    <java classname="${mainClassname}" classpath="${target.dir}" />
  </target>
</project>
```

```
ant.property(file: 'build.properties')
final srcDir = 'src'
final targetDir = 'target'

includeTargets << gant.targets.Clean
cleanDirectory << targetDir

target(compile:'compiles the source directories') {
  ant.mkdir(dir:targetDir)
  ant.javac(srcdir:srcDir, destDir:targetDir)
}

target(run:'runs the project') {
  depends(compile)
  java(classname:mainClassname, classpath:targetDir)
}

setDefaultTarget('run')
```

- jedes Gant-Skript hat vordefinierten AntBuilder: **ant**

```
ant.property(aProperty:'foobar')

ant.taskdef(name:'groovyc',
            classname:'org.codehaus.groovy.ant.Groovyc')

ant.javac(srcdir:sourceDir, destdir:buildDir)
ant.groovyc(srcdir:sourceDir, destdir:buildDir)
```

- Binden einer Closure an einen Namen
- Angabe des AntBuilder (ant) in Target Closure optional

```
target ( name : target-name ) target-closure
target ( name : target-name , description : target-desc ) target-closure
⇒ target ( target-name : target-desc ) target-closure
```

```
target(compile:'compiles the source directories') {
    mkdir(dir:targetDir)
    ant.javac(srcdir:srcDir, destDir:targetDir)
}
```

## Default Target

- entweder Name 'default'

```
target(package: 'package project') { ... }  
target('default', 'The default target.') {'package'() }
```

- oder Methode setDefault()

```
target(run: 'runs the project') { ... }  
setDefaultTarget(run)
```

## Target Abhängigkeiten

- Aufruf von Targets aus anderen Targets

```
target(compile: 'compiles the source directories') {  
    clean()  
    mkdir(dir: targetDir)  
    ant.javac(srcdir: srcDir, destDir: targetDir)  
}
```

- Aufruf nur, wenn noch nicht erfolgt

```
target(compile: 'compiles the source directories') {  
    depends(clean)  
    mkdir(dir: targetDir)  
    ant.javac(srcdir: srcDir, destDir: targetDir)  
}
```

- `includeTargets`:
  - Hinzufügen von Targets aus anderen Skripten bzw. Java-Klassen

```
includeTargets << gant.targets.Clean  
cleanDirectory << 'build'  
cleanPattern << ['**/*~', '**/*.class']
```

- `includeTools`:
  - Hinzufügen von anderweitig definierten Services (Shell, LaTeX)

```
includeTool << gant.tools.Execute  
execute.shell 'echo "Hello"'
```

- Gant auf der Kommandozeile
- Gant in IntelliJ



- Buildmanagement & Tools
- Sick of XML
- Getting started
- **Resume**
- What's next?

- ~~just works~~... rocks
- alle Ant-Tasks verwendbar (auch von Drittanbietern)
- Groovy statt XML (Schleifen, Bedingungen, Funktionen)
- Verwendung in CI Servern
  - Hudson hat Gant-Plugin
  - Teamcity, Bamboo, Cruisecontrol benötigen Umweg über Ant-Task:

```
<taskdef name="gant" classname="org.codehaus.gant.ant.Gant"/>
<gant file="build.gant" target="test"/>
```

## ... die schlechten ins Kröpfchen

- noch keine Unterstützung in freien IDEs
  - <http://jira.codehaus.org/browse/GANT-1> (Eclipse)
  - <http://jira.codehaus.org/browse/GANT-36> (NetBeans)
- "Gant should behave more Ant like"
  - <http://jira.codehaus.org/browse/GANT-77>
- Exception-Stacktrace groovytypisch lang und kryptisch
- seltsame Fehler durch Groovys dynamische Magie
- Vorsicht, bei Target-Name = Groovy Schlüsselwort ('package')

## Wann verwenden?

- bei zu komplizierten Ant Build Files
  - speziell if/else
- bei zu vielen selbst implementierten Ant-Tasks
  - Ant-Task = eigene Java-Klasse: schwer zu pflegen
- wenn bereits Skriptsprachen in Ant benutzt werden
  - `<groovy>println ... </groovy>`

## Eating it's own dog food

- Gant wird durch Gant gebaut
  - <http://gant.codehaus.org/Gant's+Build+Script>
- Grails Befehle sind Gant Skripte
  - <http://docs.codehaus.org/display/GRAILS/Command+Line+Scripting>

[http://en.wikipedia.org/wiki/Eat\\_one%27s\\_own\\_dog\\_food](http://en.wikipedia.org/wiki/Eat_one%27s_own_dog_food)

## Gliederung

- Buildmanagement & Tools
- Sick of XML
- Getting started
- Resume
- **What's next?**

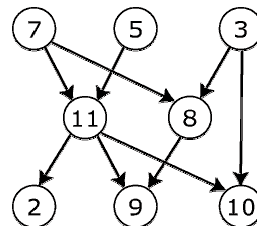


## What else?

- Gradle
  - Maven-like Build Framework
- Grape
  - Auflösen der Abhängigkeiten zur Laufzeit

## Gradle

- schwergewichtiger und mächtiger als Gant
  - komplettes Ivy-basiertes Build Framework
  - Dependency Management
  - Verwaltung des kompletten Projekt-Zyklus
  - nutzt einen DAG (Directed Acyclic Graph)
- verwendet auch Groovy für Build-Skripte
- **zur Erinnerung:**
  - Gant baut keine Datenstruktur auf
  - führt einfach nur Code aus



- **Groovy Adaptable/Advanced Packaging Engine**
- Groovy Skript enthält Beschreibung der abhängigen Bibliotheken
  - AST Transformation mit `@Grab`
- Nachladen aus Repository (bei Bedarf), auch transitive Abh. (Ivy)

```
import com.google.common.collect.HashMap

@Grab (group='com.google.code.google-collections',
        module='google-collect', version='snapshot-20080530')
def getMediumSizedVehicles() {
    [vw:'golf', opel:'astra', ford:'focus'] as HashMap
}
```



Fragen ?



# Vielen Dank für Ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68  
68309 Mannheim

[www.oio.de](http://www.oio.de)  
[info@oio.de](mailto:info@oio.de)

Version: 1.0

## Mehr von OIO zum Thema...

- Artikel: Test-Coverage für Grails
  - <http://www.oio.de/public/opensource/tutorial-grails-test-coverage.htm>
- Beratung durch OIO zum Einsatz von Groovy und Grails
  - <http://www.oio.de/beratung-consulting/open-source-software/java-frameworks/groovy-grails-consulting.htm>