

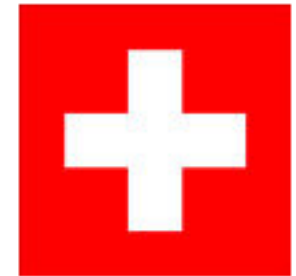
Walkthrough Groovy's AST Transformations

Andres Almiray – Canoo Engineering AG

`andres.almiray@canoo.com`

About the speaker

- Ⓜ Java developer since '95
- Ⓜ Open Source believer
- Ⓜ Groovy dev team member since '07
- Ⓜ Griffon co-founder and project lead



canoo

[@aalmiray](http://jroller.com/aalmiray)



US 2011

Minneapolis, MN

What are AST Transformations?

- Ⓜ Essentially byte code generation
- Ⓜ They enable compile-time metaprogramming
- Ⓜ There are two types: global and local
- Ⓜ Local transformations require an specific context

Available in Groovy 1.7

@Delegate

```
1 class Event {  
2     @Delegate Date when  
3     String title, url  
4 }  
5  
6 df = new java.text.SimpleDateFormat("dd.MM.yyyy")  
7 so2gx = new Event(title: "SpringOne2GX",  
8     url: "http://springone2gx.com",  
9     when: df.parse("25.10.2011"))  
10 gr8conf = new Event(title: "Gr8conf",  
11     url: "http://gr8conf.org",  
12     when: df.parse("27.06.2011"))  
13 assert gr8conf.before(so2gx.when)
```

@Singleton

```
1 class Account {  
2     BigDecimal balance = 0  
3 }  
4  
5 @Singleton  
6 class AccountManager {  
7     void open(Account account) {  
8         account.balance = 100  
9     }  
10 }  
11  
12 account = new Account()  
13 assert account.balance == 0  
14 AccountManager.instance.open(account)  
15 assert account.balance == 100
```

@Immutable

```
1 class Person { String name }
2
3 @Immutable
4 class ImmutablePerson { String name }
5
6 person1 = new Person(name: 'Duke')
7 person2 = new Person(name: 'Duke')
8 assert person1 != person2
9
10 person1 = new ImmutablePerson('Duke')
11 person2 = new ImmutablePerson(name: 'Duke')
12 assert person1 == person2
13
14 shouldFail(ReadOnlyPropertyException) {
15     person1.name = 'boom!'
16 }
```

@Category

```
1 @Category(Integer)
2 class Pouncer {
3     String pounce() {
4         (1..this).collect { 'boing!' }.join(' ')
5     }
6 }
7
8 use(Pouncer) {
9     assert 3.pounce() == 'boing! boing! boing!'
10 }
```


@Mixin

```
1 class Bouncer {
2     int times
3
4     String bounce() {
5         (1..this.times).collect { 'boing!' }.join(' ')
6     }
7 }
8
9 @Mixin(Bouncer) class Cat { }
10 @Mixin(Bouncer) class Dog { }
11
12 cat = new Cat(times: 2)
13 assert cat.bounce() == 'boing! boing!'
14 dog = new Dog(times: 3)
15 assert dog.bounce() == 'boing! boing! boing!'
16
```

@Grab

```
1 @Grab('net.sf.json-lib:json-lib:2.4:jdk15')
2 import net.sf.json.groovy.JsonGroovyBuilder
3
4 builder = new JsonGroovyBuilder()
5
6 json = builder.books {
7     book(title: 'Groovy in Action',
8         name: 'Dierk Koenig')
9 }
10
11 assert json.books.book.name == 'Dierk Koenig'
12
```

@Synchronized

```
1 import groovy.transform.Synchronized
2
3 class Worker {
4     private myLock = new Object()
5
6     @Synchronized
7     void doStuff() { /* uses auto-generated lock */ }
8
9     @Synchronized('myLock')
10    void workSomeMore() { /* uses myLock */ }
11
12    @Synchronized
13    static void compute() {
14        // works with static methods too!
15    }
16 }
```

@Lazy

```
1 class Sneaky {  
2     @Lazy List values = {[1,2,3]}()  
3     // thread - safe  
4     @Lazy volatile String id  
5 }  
6  
7 sneaky = new Sneaky()  
8 assert !sneaky.@$values  
9 assert sneaky.values == [1,2,3]  
10 assert sneaky.@$values == [1,2,3]
```

@Newify

```
1 class Apple {}
2
3 class Basket {
4     @Newify(Apple)
5     List bushel() {
6         [new Apple(), Apple(), Apple.new()]
7     }
8 }
9
10 basket = new Basket()
11 bushel = basket.bushel()
12 assert bushel*.getClass() == [Apple, Apple, Apple]
```

@Bindable (@Vetoable)

```
1 @groovy.beans.Bindable
2 class Model {
3     String name
4 }
5
6 counter = 0
7 model = new Model()
8 model.addPropertyChangeListener({
9     counter++
10 } as java.beans.PropertyChangeListener)
11
12 model.name = 'Duke'
13 model.name = 'Tux'
14 assert counter == 2
```

Available in Groovy 1.8

@Log (@Commons, @Log4j, @Slf4j)

```
1 @groovy.util.logging.Log
2 class User {
3     String username
4     void login() {
5         log.info("$username has logged into the system")
6     }
7 }
8
9 user = new User(username: 'Duke')
10 user.login()
```


@InheritConstructors

```
1 @groovy.transform.InheritConstructors
2 class MyException extends RuntimeException {}
3
4 msg = 'Error message'
5 x1 = new MyException(msg)
6 x2 = new MyException(x1)
7 x3 = new MyException(msg, x1)
8 assert x1.message == msg
9 assert x2.cause == x1
10 assert x3.message == msg
11 assert x3.cause == x1
```

@ToString

```
1 @groovy.transform.ToString(includeNames=true)
2 class Customer {
3     String first, last
4     Address address
5 }
6
7 @groovy.transform.ToString
8 class Address {
9     String street, country
10 }
11
12 customer = new Customer(first: 'Andres', last: 'Almiray',
13     address: new Address(street: 'Kirschgartenstrasse 5',
14         country: 'Schweiz'))
15 println customer
```

@EqualsAndHashCode

```
1 @groovy.transform.EqualsAndHashCode
2 class Thing {
3     String name
4     int id
5 }
6
7 thing1 = new Thing(name: 'One', id: 1)
8 thing2 = new Thing(name: 'One', id: 1)
9 thing3 = new Thing(name: 'Two', id: 2)
10
11 assert thing1 == thing2
12 assert thing1 != thing3
13 assert thing1.hashCode() == thing2.hashCode()
14 assert thing1.hashCode() != thing3.hashCode()
```

@TupleConstructor

```
1 @groovy.transform.TupleConstructor
2 class Animal {
3     String species = 'goldfish'
4     String name = 'Bubbles'
5     int age = 2
6 }
7
8 animals = [
9     new Animal(species: 'dog', name: 'Wolfie', age: 4),
10    new Animal('cat', 'Mr. Whiskers', 3),
11    new Animal()]
12
13 assert animals.species == ['dog', 'cat', 'goldfish']
14 assert animals.name == ['Wolfie', 'Mr. Whiskers', 'Bubbles']
15 assert animals.age == [4, 3, 2]
```

@IndexedProperty


```
1 class Band {  
2     @groovy.transform.IndexedProperty  
3     List members = []  
4 }  
5  
6 beatles = new Band()  
7 assert beatles.members == []  
8 beatles.members << 'John'  
9 beatles.setMembers(1, 'Paul')  
10 beatles.members[2] = 'George'  
11 beatles.members.add('Ringo')  
12  
13 assert beatles.members == ['John', 'Paul', 'George', 'Ringo']  
14 assert beatles.getMembers(3) == 'Ringo'  
15
```

And may more!

- Ⓜ @AutoClone – @AutoExternalize
- Ⓜ @ConditionalInterrupt – @TimedInterrupt
- Ⓜ @ThreadInterrupt
- Ⓜ @PackageScope
- Ⓜ @WithReadLock – @WithWriteLock
- Ⓜ @Field
- Ⓜ @ListenerList

Found in other projects

 @Entity

 Grails 1.4 has more coming!




 @EventPublisher

 @PropertyListener

 @Threading

 @MVCAware

 @ThreadingAware

 @Validatable

 alpha stage: @Entity – @Scaffold

GContracts

 @Invariant

 @Requires

 @Ensures

GContracts

```
1 import org.gcontracts.annotations.*
2
3 @Invariant({this.@balance >= 0})
4 class Account {
5     private BigDecimal balance = 0
6
7     BigDecimal getBalance() { balance }
8
9     Account() {}
10
11     @Requires({amount > 0})
12     @Ensures({balance == amount})
13     Account(BigDecimal amount) {
14         balance = amount
15     }
16
17     @Requires({amount > 0})
18     @Ensures({balance == old.balance + amount})
19     BigDecimal deposit(BigDecimal amount) {
20         balance += amount
21     }
22 }
```

Somewhere else in the 'net

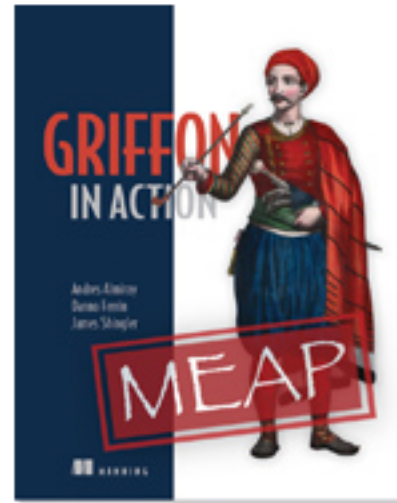
 @Scalify

<http://groovy.codehaus.org/Scalify>

 @ByteCode

<http://bit.ly/fLanEH>

Thank you!



@aalmiray