



Groovy in 15 minutes...

Johannes Carlén

Callista Enterprise AB

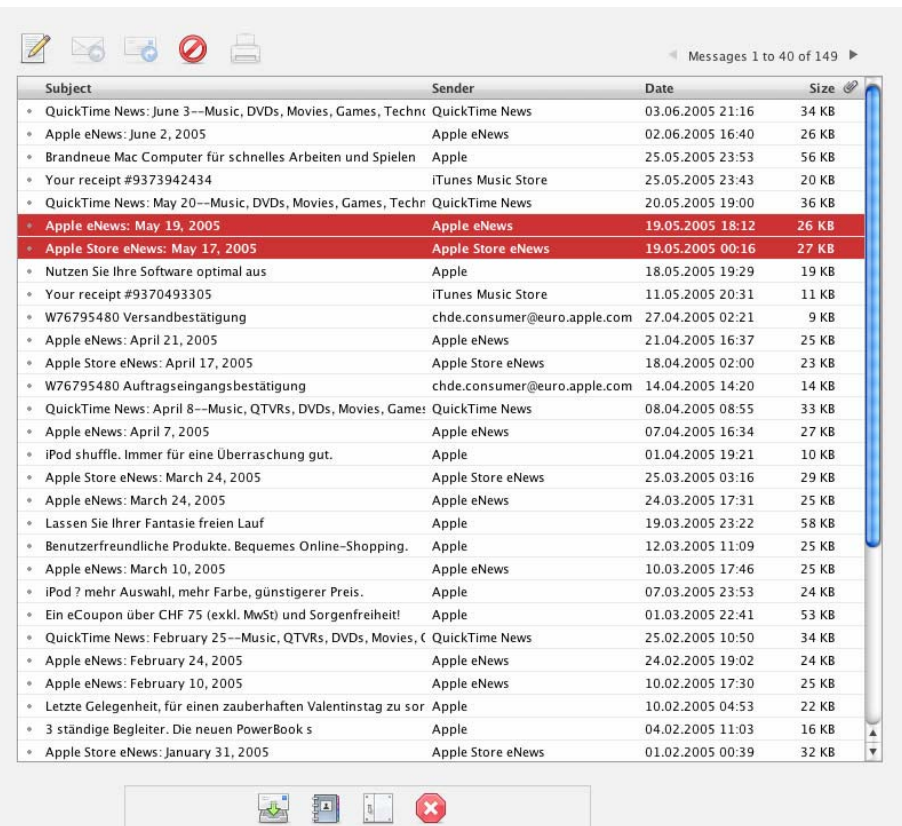
johannes.carlen@callista.se



Today...

- Building systems requires broad *and* deep skills
- Complex applications requires a complex platform
- But... same mechanisms used for building simpler applications as for more complex.

PHP with MySQL



Java keeping up...

- Integrating existing script languages
- Beanshell
- Building a new script language -
Groovy

What 's Groovy?

- Standard - JSR #241
- A scripting language that is tightly integrated into the Java platform.
- A dynamic, agile OO language for the JVM
- Full access to the Java API
- Groovy Scripts can be compiled into Java bytecode
- Static *or* dynamic typing

Java

```
public class Utils {  
  
    public static void main(String[] args) {  
        List<String> list = new ArrayList<String>();  
        list.add("Dupond");  
        list.add("Dupont");  
        list.add("Tintin");  
        List<String> filteredList = Utils.findAll("Dup", list);  
        for (String name : filteredList) {  
            System.out.println(name);  
        }  
    }  
  
    public static List<String> findAll(String filter, List<String> items) {  
        List<String> result = new ArrayList<String>();  
        for (String item : items) {  
            if (item.contains(filter)) {  
                result.add(item);  
            }  
        }  
        return result;  
    }  
}
```

Groovy

```
list = ["Dupond", "Dupont", "Tintin"]  
duponts = list.findAll { it.contains("Dup") }  
duponts.each { println it }
```

Strings

GStrings

```
name = "Tintin"
```

```
message = "Hello ${name}"
```

Multiline

```
someXML = """
```

```
    <character>
```

```
        <name>Tintin</name>
```

```
    </character>"""
```

Collections - list

```
list = [1,2,3]
longerlist = list + ["four","five"]
longerlist now equals [1,2,3,"four","five"]
```

indexing:

```
list[2..3] equals [3,"four"]
list[-1] equals "five"
```

Collections - Map

```
niceMap = ["tintin":true,"Rastapopulous":false]

println niceMap["tintin"]
> True

niceMap.calculus = true
Println niceMap
> ["tintin":true,"Rastapopulous":false,"calculus":true]
```

Closures

```
def list = [1,2,3]
```

```
def square = { numberToSquare ->
  numberToSquare*numberToSquare }
```

```
def squaredlist = list.collect (square)
```

squaredlist equals [1,4,9]

```
newlist.each { println it }
```

1

4

9

Example

- **Extract data from a database table into XML**

From database into XML

```
def sql = groovy.sql.Sql.newInstance(
    "jdbc:hsqldb:hsqldb://localhost/groovy",
    "sa", "", "org.hsqldb.jdbcDriver")

def xml = new groovy.xml.MarkupBuilder()

xml.product_catalogue() {
    sql.eachRow("select * from product") { row ->
        xml.product() {
            name(row.name)
            description(row.description)
        }
    }
}
```

The XML

```
<product-catalogue>
  <product>
    <name>iPod</name>
    <description>mp3 player with video</description>
  </product>
  <product>
    <name>MacBook Pro</name>
    <description>Intel based Apple</description>
  </product>
</product-catalogue>
```

What about the other way?

```
def file = new java.io.File("/groovy/products.xml")

def products = new
    groovy.util.XmlParser().parseText(file.getText())

def sql = groovy.sql.Sql.newInstance(
    "jdbc:hsqldb:hsqldb://localhost/groovy", "sa", "",
    "org.hsqldb.jdbcDriver")

products.each { product ->
    sql.execute("insert into product values (?,?)",
        [product.name.text(), product.description.text()])
}
```

Unit testing

- JUnit built into runtime => script your tests for Groovy and Java classes with Groovy syntax
- Groovy provides several new assertions
- Easily scripted with Ant or Maven
- Integrate Groovy unit tests with your existing suite

Unit testing

```
class StringSplitTest extends GroovyTestCase {
    void testFullSplit() {

        splitArray = StringSplitter.split(
            "groovy.util.GroovyTestCase", ".")

        expect = ["groovy", "util", "GroovyTestCase"].toArray()

        assertArrayEquals(expect, splitAr)
    }
}
```

Other features

- Ant Scripting, Templates
- Groovlets, GSP, Swing, SWT, XMLRPC
- GroovyBeans

```
class Product {  
    @Property String name  
    @Property String description  
}
```

Grails

- "Coding by convention"-paradigm
- A toolkit of Spring, Hibernate, SiteMesh...
- Smaller applications
 - forums, blogs etc.

What else can you do?

- Configuration
- Simple tasks
- Prototypes
- Building and testing
- Agile development
- Rules for rules engines
- ESB transformations
- ...

Q&A

johannes.carlen@callista.se
<http://www.callista.se/enterprise>