

GroovyDevCon 3

GroovyDevCon #3

So far, we've held two Groovy Developer Conferences: one in London in November 2004, and one in Paris in November 2005. With the late releases of RC-1 and 1.0, the third instance will take place in 2007. The goal of this page is to work on the dates, the place, and the goals and topics of the next session.

Agenda

With Groovy 1.0 released, it'll be time to focus on the JSR-241 and decide of the roadmap of Groovy 2.0

Please suggest the various topics you'd wish to speak about or features you'd like to have in Groovy 2.0:

Project

Roadmap

- Branching scheme : 1.x and 2.x
- Numbering scheme : 1.0.1, 1.1, 1.5/2.0 alpha/beta/RC
- Define a roadmap towards the next major milestone
- Set a policy for keeping 1.0 compatibility
- Provide a process for languages changes with a real proposal and prototype (à la PEP: Python Extension Proposal)

Unit testing

- More unit tests
- Have the wiki samples being automatically tested
- Reuse GinA tests
- Split the tests to create tests for:
 - TCK
 - Good & bad error messages checking
 - Tests for non-compilable code
 - GDK vs. language tests

JSR-241

- Give the status and setup a roadmap for the JSR
- Who can lead the effort?

Misc.

- Coding style issues, e.g. TODOs, javadoc or groovydoc, metrics

Documentation

- What can be taken from GinA?
- What are the areas we need to work on?
- Who can work on this effort?

Features

Syntax

- JDK 1.5+ features
 - Make 1.5 the default target JVM for Groovy NG or can we support 1.4 still?
 - Full annotation support
 - Enum support
- Syntax for operator overloading
- New mechanism for DGM and possible syntax for extension methods: a mixin category, inspiration from C# 3
- Allow omitting parentheses when named arguments are used in top-level method calls: `monster.move x:100.meters, y:200.meters`
- A pass by ref keyword for implementing `swap(ref x, ref y)` (exists in multiple languages like C#)
- Multiple assignment: `a, b = foo()` (if a 2-element list is returned for instance)
- Class-scoped use directive
- Discuss possibility of adding a construct that can differentiate between a closure that can return from its outer scope and one that cannot.
- Move date arithmetic code from Google Data into DGM (1.day.ago)
- Tuples: `(a,b)`

MOP

- Adding new features to make Meta-programming easier ([ExpandoMetaClass](#)), for instance: `myString.metaClass.threeTimes = {}` and `myString.metaClass.someProp = 3`
- Remove hard references to `java.lang.reflect.*` types from `MetaClassImpl` making everything a `Meta*` (`MetaConstructor`, `MetaField`, `MetaMethod` etc.). This is only partially done.

GDK

- Commons-cli or JOpt Simple
- Store source date in the class itself? (comments, closure code which can be used for special serialization or AST operations during runtime)
- Date / Time calculations
- Math & scientific support: complex numbers, units handling, module wrapper around JScience, Choco, and the likes

Advanced features

- Continuations
- Tail recursion

Problems to address

- Serialization issue
- Registry clean-up (cf XWiki usage)
- Refactor DGM into difference classes that deal with each type (Object, Map, List) for maintainability

Tooling

- GroovyDoc
- IDE plugins
- Improved console

Build

- Rework the build system (Gant, Ivy, m2)
- Cleaning up the codebase (what can be removed, e.g. old eclipse plugin)
- Coverage report

- Move non core classes (AntBuilder, Mock...) out of main source tree and cut down jar dependencies
- Make sure we still provide poms for M1 / M2 repository distribution
- Continuous integration build

Distribution

- Groovy-all should be the rule (no double jars all vs. not all)
- Include a native launcher
- Update dependencies (antlr 3, asm 3, ant 1.7, xerces 2.9, junit 4.2...)
- Remove unneeded dependencies from the distributions (EJB, radeox, axion...)
- Modularize the groovy jars (groovy-core, groovy-gdk, groovy-jsr223 engine and embed asm & antlr by default in groovy-core)
- Add JSR-223 Groovy engine in the distribution
- Distinguish distributions for JDK 1.4 / 1.5 / 1.6 to embed jars like JMX, JSR-223, Xerces/Xalan

Performance

- Performance improvement (run time and compiler performance)

Location

The Groovy Developer Conference #3 will be held in **Paris**.

Sun will provide us with a big room named "**Jupiter**" with video-projector and so on.
You can find the details for going to the Sun's offices [there](#).

Dates

The conference will take place on **29 and 30 January 2007**.

(Other possible matching dates were 1-2 and 8-9 February 2007. The sooner, the better.)

The conference is a **two days conference**. All Groovy & Grails developers are welcome to this conference, and if space permits, we can potentially accept additional invitees