

Groovy 1.7.4 and the first beta drop of Groovy 1.8

This is with great pleasure that the Groovy development team is happy to announce the joint releases of **Groovy 1.7.4** and **Groovy 1.8-beta-1!**

Groovy 1.7.4

Groovy 1.7 is our official stable branch, and 1.7.4 is a new minor release of that breed.

Among other things, we fixed a few issues with automatic reloading when using the GroovyScriptEngine, with the joint compiler, with some minor syntax issues with Enums. We added a few new GDK methods like File.renameTo(String path), minus and intersect for sets, etc. And we have new option in the Groovy swing console to compile scripts -- versus running them right away.

You can have further details on Groovy 1.7.4 in the [JIRA release notes](#).

Groovy 1.8-beta-1

Groovy 1.8-beta-1 is the first drop of what the next major version of Groovy will be -- we plan on releasing 1.8-final at the end of the year.

From a practical Groovy developer perspective, we've decided to move to **Gradle** for our build, thanks to the help of [Hans Dokter](#). This release was still done with Ant, but our final build should be ready soon for prime-time, and we're also seizing this opportunity to work in parallel on making **Groovy more modular**. Some early work towards more granular Groovy JAR(s) have already started on modules for JMX and BSF, as first experiments.

In this first beta, we obviously included all the bug fixes and minor enhancements from the 1.7 branch. And we've been working on new features as well:

- closures are now "callable" (inheriting from java.util.concurrent.Callable)
- new AST transformations like
 - @Log to inject a logger in your classes
 - @ScriptField for creating a field in a script (should be renamed to @Field in the next beta)
 - @PackageScope now working also on methods and classes (not just fields as before)
 - @Synchronized for providing safer synchronization semantics
 - @InheritConstructors to inherit constructors, like often the case when extending exception classes
 - @IndexedProperties to add JavaBeans indexed property support
 - @AutoClone providing automatic cloning support to your beans
 - @AutoExternalize providing automatic externalization of your POGOs
 - @Canonical adding proper equals(), hashCode(), toString() methods
 - @EqualsAndHashCode adding an equals() and hashCode() method
 - @ToString for creating a default readable toString() method
 - @TupleConstructor for adding a tuple constructor
- an additional syntax for strings, with \$/.../\$, to circumvent various escaping corner cases, like for example \$/a/b\c\$\$ \$//
- new GDK methods like Map.countBy(), Map.collectEntries(), Date.putAt() (subscript operator), Date.updated()
- + obviously all other incremental improvements and new features from the Groovy 1.7 branch like
 - the new String methods like tr(), stripMargin(), stripIndent(), (un)expand(),
 - Map's withDefault{} method,
 - Closure's ncurry() and rcurry()
 - Sql's withBatch{} and withTransaction{}

A feature I'd also like to highlight in particular is "closure annotation parameters". In Java, there's a limited set of types you can use as annotation parameters (String, primitives, annotations, classes, and arrays of these). But in Groovy 1.8, we're going further and let you use closures as annotation parameters – which are actually transformed into a class parameter for compatibility reasons.

```
import java.lang.annotation.*

@Retention(RetentionPolicy.RUNTIME)
@interface Invariant {
    Class value() // will hold a closure class
}

@Invariant({ number >= 0 })
class Distance {
    float number
    String unit
}

def d = new Distance(number: 10, unit: "meters")

def anno = Distance.getAnnotation(Invariant)
def check = anno.value().newInstance(d, d)

assert check(d)
```

Closure annotation parameters open up some interesting possibilities for framework authors!
You can have a look at the [JIRA release notes](#) for Groovy 1.8-beta-1.

You can download Groovy 1.7.4 and Groovy 1.8-beta-1 in the [download section](#) of the Groovy website.

Thanks a lot to everybody for their contributions and involvement in those two new milestones!