Groovy Runtime

GMaven uses an abstraction layer to access a specific version of the Groovy language runtime.

This allows builds to work with different Groovy versions without much fuss.

- Features
- Providers
- Runtime Selection
- Provider Information
- Custom Versions
- Resources
  - Goal Documentation

The abstraction layer consists of a runtime *provider* implementation which exposes a set of *features*.

Features

- Class Loading
- Class Compilation
- Stub Compilation
- Script Execution
- Command-line Shell (ie. Groovy Shell)
- GUI Console (ie. Groovy Console)
- Stack Trace Sanitization

> ⚠ **TODO**
>
> Document which goals require which features.

Providers

	1.0	1.5	1.6	1.7	1.8	2.0
Class Loading	✅	✅	✅	✅	✅	✅
Class Compilation	✅	✅	✅	✅	✅	✅
Stub Compilation	✅	✅	✅	✅	✅	✅
Script Execution	✅	✅	✅	✅	✅	✅
Command-line Shell	✅	✅	✅	✅	✅	✅

GUI Console	✅	✅	✅	✅	✅	✅
Stack Trace Sanitization	✅	✅	✅	✅	✅	✅

✅ Supported	❌ Unsupported	⚠️ Partial/Crippled Support	❓ Unknown

Runtime Selection

The selection of which Groovy runtime will be used is primarily controlled by the `gmaven.runtime` property.

For example, to use the Groovy `1.5` runtime:

```
mvn -Dgmaven.runtime=1.5 install
```

To always use the latest supported runtime the `default` provider key can be used:

```
mvn -Dgmaven.runtime=default install
```

This property is a comma-separated list of runtime provider keys, which will be used in the order in which they are listed. So in the example below, first the `1.5` runtime will be queried for availability (and support), then the `1.0` provider and then finally the `default` provider will be used.

```
mvn -Dgmaven.runtime=1.5,1.0,default install
```

Provider Information

Sometimes it helps to see what providers are configured. Use the `groovy:providers` goal to show some details about the discovered providers as well as which is selected based on the current execution environment.

```
mvn groovy:providers
```

This should split up some details about the providers like:

```
[INFO] [groovy:providers]
```

```
[INFO]
[INFO]  Provider selection: default
[INFO]
[INFO]  Found 2 registered providers:
[INFO]      [1.5] 'Groovy v1.5.1' (version:
1.5.1, type:
org.codehaus.groovy.maven.runtime.v15.Groovy
Runtime_v15)
[INFO]          Features:
[INFO]
org.codehaus.groovy.maven.runtime.Console
[INFO]
org.codehaus.groovy.maven.runtime.ClassFacto
ry
[INFO]
org.codehaus.groovy.maven.runtime.TraceSanit
izer
[INFO]
org.codehaus.groovy.maven.runtime.StubCompil
er
[INFO]
org.codehaus.groovy.maven.runtime.ClassCompi
ler
[INFO]
org.codehaus.groovy.maven.runtime.Shell
[INFO]
org.codehaus.groovy.maven.runtime.ScriptExec
utor
[INFO]
[INFO]      [default] 'Groovy v1.5.1'
(version: 1.5.1, type:
```

```
org.codehaus.groovy.maven.runtime.vdefault.D
efaultGroovyRuntime)
[INFO]          Features:
[INFO]
org.codehaus.groovy.maven.runtime.Console
[INFO]
org.codehaus.groovy.maven.runtime.ClassFacto
ry
[INFO]
org.codehaus.groovy.maven.runtime.TraceSanit
izer
[INFO]
org.codehaus.groovy.maven.runtime.StubCompil
er
[INFO]
org.codehaus.groovy.maven.runtime.ClassCompi
ler
[INFO]
org.codehaus.groovy.maven.runtime.Shell
[INFO]
org.codehaus.groovy.maven.runtime.ScriptExec
utor
[INFO]
[INFO]  Selected provider:
[INFO]      [default] 'Groovy v1.5.1'
(version: 1.5.1, type:
org.codehaus.groovy.maven.runtime.vdefault.D
efaultGroovyRuntime)
[INFO]
```

To test a configuration set the `gmaven.runtime` property and see what it produces:

```
mvn groovy:providers
-Dgmaven.runtime=1.5,1.0,default
```

Which may spit out something like this:

```
[INFO] [groovy:providers]
[INFO]
[INFO]  Provider selection: 1.5,1.0,default
[INFO]
[INFO]  Found 3 registered providers:
[INFO]      [1.5] 'Groovy v1.5.1' (version:
1.5.1, type:
org.codehaus.groovy.maven.runtime.v15.Groovy
Runtime_v15)
[INFO]          Features:
[INFO]
org.codehaus.groovy.maven.runtime.Console
[INFO]
org.codehaus.groovy.maven.runtime.ClassFacto
ry
[INFO]
org.codehaus.groovy.maven.runtime.TraceSanit
izer
[INFO]
org.codehaus.groovy.maven.runtime.StubCompil
er
[INFO]
org.codehaus.groovy.maven.runtime.ClassCompi
ler
[INFO]
```

```
org.codehaus.groovy.maven.runtime.Shell
[INFO]
org.codehaus.groovy.maven.runtime.ScriptExec
utor
[INFO]
[INFO]      [1.0] 'Groovy v1.0' (version:
1.0.0, type:
org.codehaus.groovy.maven.runtime.v10.Groovy
Runtime_v10)
[INFO]          Features:
[INFO]
org.codehaus.groovy.maven.runtime.Console
[INFO]
org.codehaus.groovy.maven.runtime.ClassFacto
ry
[INFO]
org.codehaus.groovy.maven.runtime.TraceSanit
izer
[INFO]
org.codehaus.groovy.maven.runtime.StubCompil
er
[INFO]
org.codehaus.groovy.maven.runtime.ClassCompi
ler
[INFO]
org.codehaus.groovy.maven.runtime.Shell
[INFO]
org.codehaus.groovy.maven.runtime.ScriptExec
utor
[INFO]
[INFO]      [default] 'Groovy v1.5.1'
```

```
(version: 1.5.1, type:
org.codehaus.groovy.maven.runtime.vdefault.D
efaultGroovyRuntime)
[INFO]          Features:
[INFO]
org.codehaus.groovy.maven.runtime.Console
[INFO]
org.codehaus.groovy.maven.runtime.ClassFacto
ry
[INFO]
org.codehaus.groovy.maven.runtime.TraceSanit
izer
[INFO]
org.codehaus.groovy.maven.runtime.StubCompil
er
[INFO]
org.codehaus.groovy.maven.runtime.ClassCompi
ler
[INFO]
org.codehaus.groovy.maven.runtime.Shell
[INFO]
org.codehaus.groovy.maven.runtime.ScriptExec
utor
[INFO]
[INFO]  Selected provider:
[INFO]      [1.5] 'Groovy v1.5.1' (version:
1.5.1, type:
org.codehaus.groovy.maven.runtime.v15.Groovy
```

```
Runtime_v15)
[INFO]
```

Custom Versions

Custom versions of Groovy can be used if needed. Simply include a similar provider as a dependency, exclude the dependencies which need to be changed, then add them as dependencies.

Below is an example of using the `1.7` provider and configuring the provider to actually use Groovy `1.0`.

```
<dependencies>
  <dependency>
    <groupId>groovy</groupId>
    <artifactId>groovy-all</artifactId>
    <version>1.0</version>
  </dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.codehaus.groovy.maven</gr
oupId>
      <artifactId>gmaven-plugin</artifactId>
      <version>1.3</version>
      <dependencies>
        <dependency>

<groupId>org.codehaus.groovy.maven.runtime</
groupId>

<artifactId>gmaven-runtime-1.7</artifactId>
```

```
            <version>1.3</version>
            <exclusions>
              <exclusion>

<groupId>org.codehaus.groovy</groupId>

<artifactId>groovy-all</artifactId>
              </exclusion>
            </exclusions>
          </dependency>
          <dependency>
            <groupId>groovy</groupId>
            <artifactId>groovy-all</artifactId
>
            <version>1.0</version>
          </dependency>
        </dependencies>
        <configuration>
          <providerSelection>1.7</providerSele
ction>
        </configuration>
      </plugin>
```

```
  </plugins>
</build>
```

> **NOTE**
>
> Be aware that the providers *key* will **not** change; the custom provider is still configured with a selection of `1.5`.

If you end up needing a specific version of Groovy then you might be better of rolling your own provider model, and put all the dependency overrides in one place. Don't worry, in most cases you can use the provider/feature implementations from a supported provider and just customize the Groovy (and related) dependencies.

Resources

Goal Documentation

- groovy:providers