



INSPIRE and Beyond Workshop: Groovy Scripting

Day 2 – 24.05.2018



scripting

Source: https://en.wikipedia.org/wiki/Apache_Groovy

Agenda

1. Motivation
2. Groovy basics
3. Groovy in hale studio

1. Motivation

Using Groovy scripts in hale we can

- Add functionality not covered by default transformation functions
- Adapt behavior of type transformations
- Implement logic across transformation functions (e.g. using collectors)
- Control exactly how objects are created

Thus using Groovy scripts is a very powerful tool, but it requires some programming knowledge.

1. Motivation

Your motivation

- Is there anything specific you want to learn?
- Did you have any challenges in your work with hale that you think could be solved with scripts?
- Are there aspects where you found the documentation lacking?

2. Groovy basics

- Dynamic programming language based on Java
- Playing around with Groovy
 - e.g. with the Groovy Console (part of Groovy installation) or the Online Groovy Playground (<https://groovy-playground.appspot.com/>)
 - Allows to enter groovy scripts for testing
 - Displays script output and return value

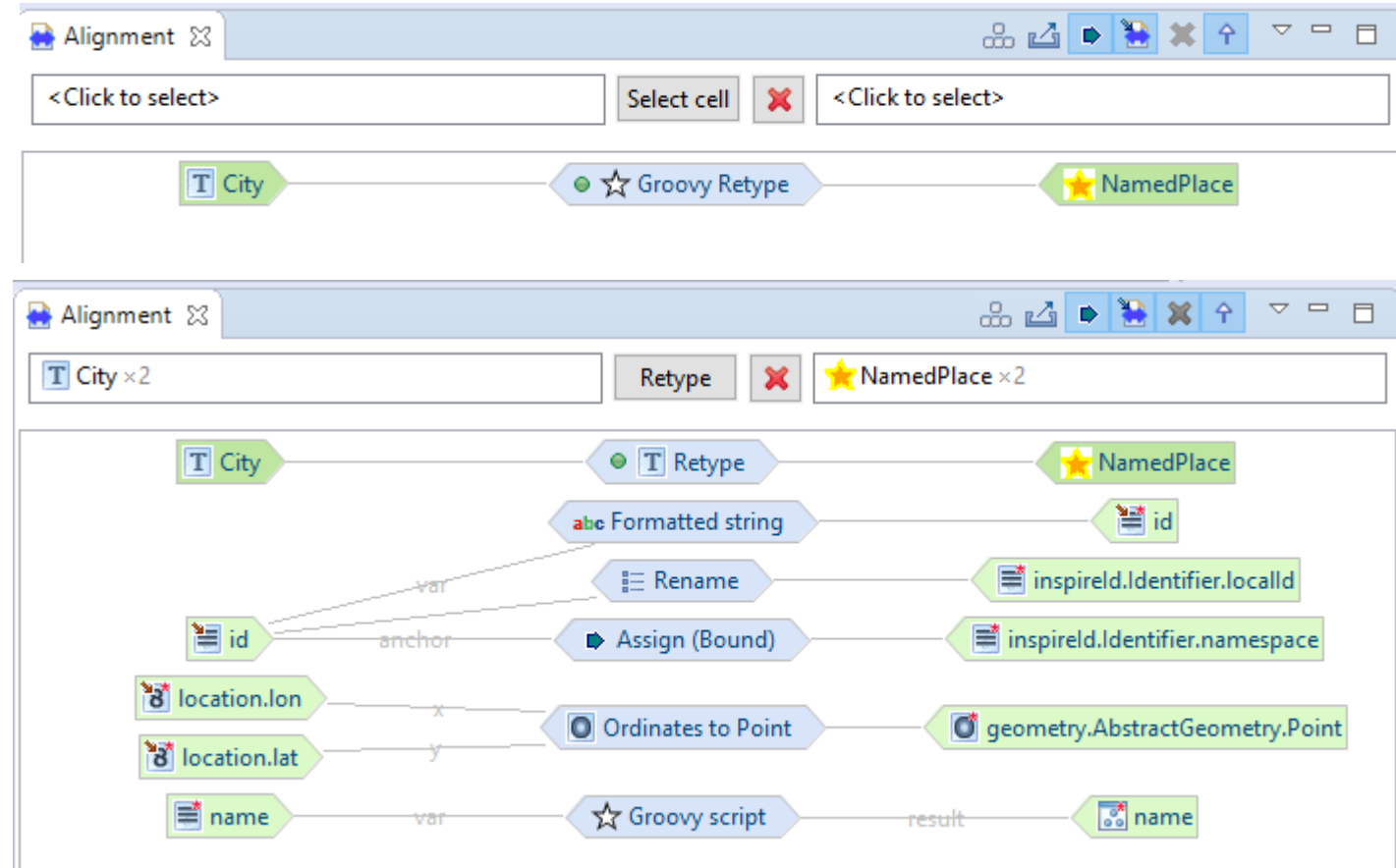
Basics on the Groovy Playground

<https://goo.gl/GGnkHD>

3. Groovy in hale studio

3. Groovy in hale studio

- Groovy scripts can be used in
 - type relations
 - property relations
 - custom functions
 - snippets



Using a custom function

Advantages

- Easily reuse custom functionality
- Function behaves like a normal transformation function, i.e. usage is transparent for the user

Disadvantages / restrictions

- Currently only usable for property mappings
- Currently no UI support for defining parameters

Using an external Groovy script (Snippet)

Advantages

- Script can be edited and tested externally (e.g. with GroovyConsole)
- Functionality can be reused in different contexts (e.g. different type relations)

Disadvantages / restrictions

- Not usable in base alignments
- Dependencies added via Groovy Grapes are not supported in hale

Using an external Groovy script (Snippet)

For calling a snippet there are two recommended ways:

- Run the snippet script **or**
- run a closure on the snippet script

```
// run the snippet "util"  
def res1 = _snippets.util()  
// run the snippet passing binding variables  
def res2 = _snippets.util(limit: 10, verbose: true)  
// run a closure  
// assuming the snippet defines the method "format"  
def res3 = _snippets.util {  
    format(source_field)  
}
```

3. Groovy in hale studio

Examples