

# Smooks Example - groovy

This example demonstrates how [Groovy](#) can be used to perform Fragment based Transforms through Smooks.

- [Overview](#)
- [The Input Message](#)
- [The Smooks Configuration](#)
- [Executing Smooks](#)

## SVN - Download - Other Tutorials

[To Build](#): "mvn clean install"

[To Run](#): "mvn exec:java"

See [Mixing DOM and SAX Models with Smooks](#)

## Overview

In this example, we want to modify the "supplies" category in a Shopping list message, adding 2 to the quantity, where the item is "Pens".

## The Input Message

The input message is a Shopping list:

```
<shopping>
  <category type="groceries">
    <item>Chocolate</item>
    <item>Coffee</item>
  </category>
  <category type="supplies">
    <item>Paper</item>
    <item quantity="4">Pens</item>
  </category>
  <category type="present">
    <item when="Aug 10">Kathryn's Birthday</item>
  </category>
</shopping>
```

## The Smooks Configuration

To do this, we write a simple little Groovy script and target it at the <category> elements in the message. The script simply iterates over the <item> elements in the category and increments the quantity by 2, where the category type is "supplies" and the item is "Pens":

```

<?xml version="1.0"?>
<smooks-resource-list xmlns="http://www.milyn.org/xsd/smooks-1.1.xsd"
xmlns:g="http://www.milyn.org/xsd/smooks/groovy-1.1.xsd">

  <params>
    <param name="stream.filter.type">SAX</param>
  </params>

  <g:groovy executeOnElement="category">
    <g:script>
      <!--
        use(DOMCategory) {

          // Modify "supplies": we need an extra 2 pens...
          if (category.'@type' == 'supplies') {
            category.item.each { item ->
              if (item.text() == 'Pens') {
                item['@quantity'] =
item.'@quantity'.toInteger() + 2;
              }
            }
          }
        }

        // When using the SAX filter, we need to explicitly write the
fragment
        // to the result stream...
        writeFragment(category);
        -->
      </g:script>
    </g:groovy>

  </smooks-resource-list>

```

The result of running this transform on the specified input message will simply modify the quantity value on the "Pens" item from 4 to 6.

## Executing Smooks

```

protected static String runSmooksTransform(Reader messageIn) throws
IOException, SAXException, SmooksException {
    Smooks smooks = new Smooks("smooks-config.xml");
    StringResult result = new StringResult();

    smooks.filter(new StreamSource(messageIn), result);

    return result.toString();
}

```

Of course, you'd typically cache the [Smooks](#) instance, and if the Result was potentially a huge message, then you'd more likely stream the result

to a file (for example).

See the [example/Main.java](#) in the example source.