

Smooks Example - profiling

The primary goals of this example are to introduce you to the following:

1. Transformer that profiling.
2. The Smooks configuration file.
3. Executing the Smooks Transformation.

SVN - Download - Other Tutorials

To Build: "mvn clean install"

To Run: "mvn exec:java"

Transforming using profiling

This is a simple example that illustrates how Smooks can be used to apply "profile" based transformations on a message. Using profiles, you can share common transformation configurations across all messages that share the same profiles.

[work in progress]

So here's the source xml that is to be transformed:

```
<a>
  <b>
    <c>
      <b></b>
    </c>
  </b>
</a>
```

And this is the expected result of the transform for target profile 'message-exchange-1':

```
<from-a>
  <to-b>
    <c>
      <b></b>
    </c>
  </to-b>
</from-a>
```

And this is the expected result of the transform for target profile 'message-exchange-2':

```
<from-a>
  <b>
    <c>
      <to-c></to-c>
    </c>
  </b>
</from-a>
```

And this is the expected result of the transform for target profile 'message-exchange-3':

```
<from-a>
  <b>
    <to-d>
      <b></b>
    </to-d>
  </b>
</from-a>
```

And this is the expected result of the transform for target profile 'message-exchange-4':

```
<a>
  <to-b>
    <c>
      <b></b>
    </c>
  </to-b>
</a>
```

And this is the expected result of the transform for target profile 'message-exchange-5':

```
<a>
  <to-b>
    <message-exchange-5>
      <d-to-b></d-to-b>
    </message-exchange-5>
  </to-b>
</a>
```

The Smooks Configuration

In order to apply this transformer to a message fragment, a [Smooks Configuration](#) needs to be created. This configuration will target the transformer at a particular message fragment. For more information on configuring XSLT resources, see [XslContentDeliveryUnitCreator](#).

Here's the configuration ("smooks-config.xml"):

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

  <profiles>
    <profile base-profile="message-exchange-1"
sub-profiles="from-a,to-b" />
    <profile base-profile="message-exchange-2"
sub-profiles="from-a,to-c" />
    <profile base-profile="message-exchange-3"
sub-profiles="from-a,to-d" />

    <profile base-profile="message-exchange-4"
sub-profiles="from-c,to-b" />
    <profile base-profile="message-exchange-5"
sub-profiles="from-d,to-b" />
  </profiles>

  <resource-config selector="a" target-profile="from-a">
    <resource>example.BasicJavaTransformer</resource>
    <param name="newName">from-a</param>
  </resource-config>

  <resource-config selector="a b" target-profile="to-b">
    <resource>example.BasicJavaTransformer</resource>
    <param name="newName">to-b</param>
  </resource-config>

  <resource-config selector="c b" target-profile="to-c">
    <resource>example.BasicJavaTransformer</resource>
    <param name="newName">to-c</param>
  </resource-config>

  <resource-config selector="c" target-profile="to-d">
    <resource>example.BasicJavaTransformer</resource>
    <param name="newName">to-d</param>
  </resource-config>

  <resource-config selector="b c" target-profile="message-exchange-5">
    <resource>example.BasicJavaTransformer</resource>
    <param name="newName">message-exchange-5</param>
  </resource-config>

  <resource-config selector="c b" target-profile="to-b AND from-d">
    <resource>example.BasicJavaTransformer</resource>
    <param name="newName">d-to-b</param>
  </resource-config>

</smooks-resource-list>
```

Executing The Transformation

```
// Instantiate Smooks with the config...
Smooks smooks = new Smooks("smooks-config.xml");

// Filter the input message to the outputWriter, using the execution
context...
smooks.filter(new StreamSource(messageIn), new StreamResult(outputWriter));
```

Of course, you'd typically cache the [Smooks](#) instance.

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