Usage:

<action name="transform" class="org.jboss.soa.esb.smooks.SmooksAction">

<property name="smooksConfig" value="smooks-config.xml" />

</action>

Optional properties:

<property name="get-payload-location" value="input" />

<property name="set-payload-location" value="ouput" />

<property name="mappedContextObjects" value="object1,object2" />

<property name="resultType" value="STRING" />

<property name="reportPath" value="/tmp/smooks-report.html" />

<property name="messageProfile" value="fromServiceA" />

Description of configuration properties:

* *smooksConfig* - the Smooks configuration file. Can be a path on the file system or on the classpath.
* *get-payload-location* - the body location which contains the object to be transformed. See [MessagePayloadProxy](http://docs.jboss.org/jbossesb/docs/4.7/javadoc/esb/org/jboss/soa/esb/message/MessagePayloadProxy.html).
* *set-payload-location* - the body location where the transformed object will be placed. See [MessagePayloadProxy](http://docs.jboss.org/jbossesb/docs/4.7/javadoc/esb/org/jboss/soa/esb/message/MessagePayloadProxy.html).
* *mappedContextObjects* - comma separated list of Smooks ExecutionContext objects to be mapped into the [EXECUTION\_CONTEXT\_ATTR\_MAP\_KEY](http://docs.jboss.org/jbossesb/docs/4.7/javadoc/esb/org/jboss/soa/esb/smooks/SmooksAction.html#EXECUTION_CONTEXT_ATTR_MAP_KEY) Map on the ESB Message. Default is an empty list. See [Exposing the Smooks ExecutionContext to other ESB Actions](http://docs.jboss.org/jbossesb/docs/4.7/javadoc/esb/org/jboss/soa/esb/smooks/SmooksAction.html#exposing-smooks-exec).
* *resultType* - type of result expected from Smooks ("STRING", "BYTES", "JAVA", "NORESULT"). Default is "STRING". For more on specifying and controlling the Smooks filtering result, see [Specifying the Source and Result Types](http://docs.jboss.org/jbossesb/docs/4.7/javadoc/esb/org/jboss/soa/esb/smooks/SmooksAction.html#specify-result).
* *javaResultBeanId* - specifies the Smooks bean context beanId to be mapped as the result when the resultType is "JAVA". If not specified, the whole bean context bean Map is mapped as the result.
* *reportPath* - specifies the path and file name for generating a Smooks Execution Report. This is a development tool.
* *messageProfile* - specifies the default message "profile" name to be used in creation of the Smooks ExecutionContext. See [Message Profiling](http://docs.jboss.org/jbossesb/docs/4.7/javadoc/esb/org/jboss/soa/esb/smooks/SmooksAction.html#profiling).

**Exposing the Smooks ExecutionContext to other ESB Actions**

After Smooks has performed the filtering operation on the message payload, it can map the contents of the ExecutionContext onto a Map on the the ESB message, making it available to other actions in the ESB. This Map can be accessed by using the [EXECUTION\_CONTEXT\_ATTR\_MAP\_KEY](http://docs.jboss.org/jbossesb/docs/4.7/javadoc/esb/org/jboss/soa/esb/smooks/SmooksAction.html#EXECUTION_CONTEXT_ATTR_MAP_KEY) key as follows:

message.getBody().get( SmooksAction.EXECUTION\_CONTEXT\_ATTR\_MAP\_KEY );

The ExecutionContext objects to be mapped must be specified in the "mappedContextObjects" action property. The objects must also be Serializable.

**Specifying the Source and Result Types**

From the ESB Message data type, this action is able to automatically determine the type of Source to use (via the Smooks PayloadProcessor). The Result type to be used can be specified via the "resultType" property, as outlined above.

It is expected that the above mechanism will be satisfactory for most usecase, but not all. For the other usecases, this action supports SourceResult payloads on the ESB Message. This allows you to manually specify other Source and Result types, which is of particular interest with respect to the Result type e.g. for streaming the Result to a file etc.

**Message Profiling**

Smooks Profiling allows you to use a single Smooks instance to transform multiple source messages. As an example, imagine a situation where messages of different formats are delivered to a Service. Before consuming the messages, the Service needs to transform these message payloads to a common format. To accomplish this, you can use profiling.

The action can have the default profile name configured through the "messageProfile" property. Each incoming ESB message can specify it's profile name through the message property of the same name ("messageProfile"). For more on profiling, see the [profiling example](http://milyn.codehaus.org/Smooks+Example+-+profiling).