Smooks Dev Tools Reference Guide

Version: 1.1.0.CR2

Introduction

This chapter gives you a short introduction to Smooks, Smooks tools and its installation.

First, have a look at the key features of Smooks tools:

1.1. Key Features of Smooks Tools

Here, we provide you with a key functionality which is integrated in Smooks tools.

Table 1.1. Key Functionality for Smooks Tools

Feature	Benefit	Chapter
Smooks Configuration File Wizard	Smooks tools allows to create/edit the Smooks configuration file for Java2Java data transformation.	*Smooks Configuration File Wizard*
Smooks Editor	Smooks Editor helps configure the created Smooks configuration file.	*Smooks Editor*

1.2. What is Smooks?

Smooks is a Java Framework/Engine for processing XML and non XML data (CSV, EDI, Java, JSON etc).

I. **Transformation**: Perform a wide range of Data Transforms. Supports many different Source and Result types -XML/CSV/EDI/Java/JSON to XML/CSV/EDI/Java/JSON.

II. **Java Binding**: Bind into a Java Object Model from any data source (CSV, EDI, XML, Java, JSON etc).

III. **Huge Message Processing**: Process huge messages (GBs) - Split, Transform and Route message fragments to JMS, File, Database etc destinations. Route multiple message formats to multiple destinations in a single pass over a message.

IV. **Message Enrichment**: Enrich a message with data from a Database, or other Datasources.

V. **Combine**: Combine the above features in different ways e.g. add Message Enrichment as part of a Splitting and Routing process.

Figure 1.1. Smooks

For more informations about Smooks, please visit *Smooks official site* [http://www.smooks.org].

1.3. What is Smooks Tools?

Smooks tools is a set of graphical tools for editing Smooks configuration file based on Eclipse.
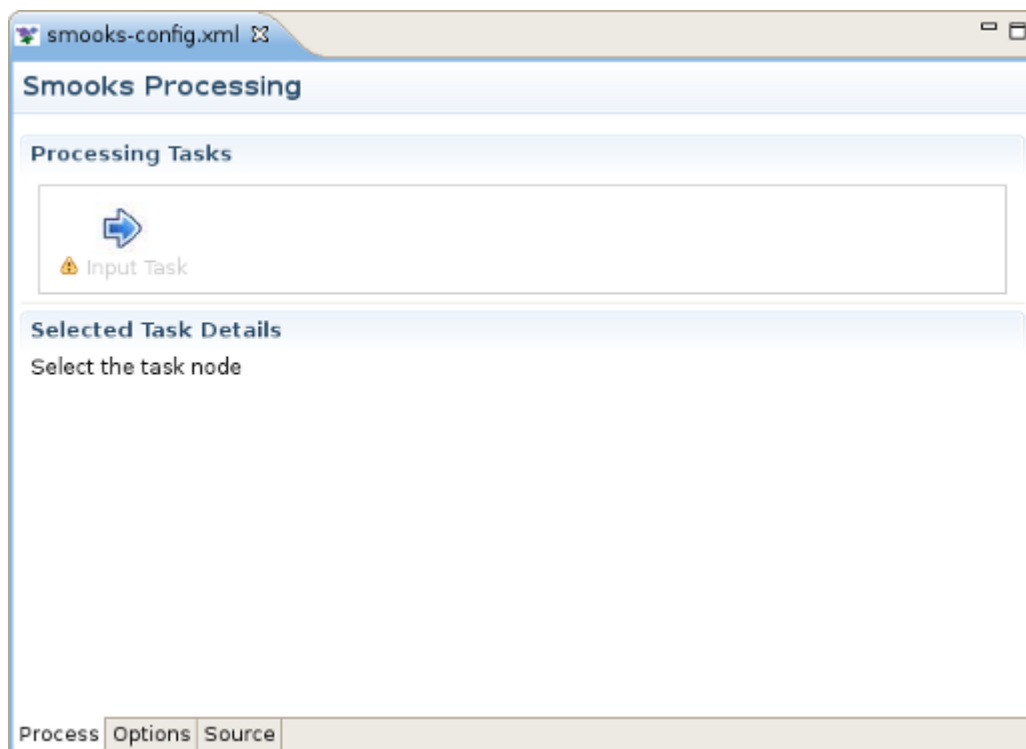
Figure 1.2. Smooks Form editor

The easiest way to use the Smooks Configuration Editor is to create a project (Java project, an ESB project, etc.), right-click on it and select New -> Other to open the New wizard. Drill into Smooks -> Smooks Configuration File and continue through the wizard. We recommend using a

minimum Smooks level of 1.1 or 1.2, but if you're using it in a deployed service, it depends on what version your runtime supports. Once the file is created, it will open in the Smooks Configuration Editor.

1.4. How to install Smooks dev tools?

The Smooks tools was included by the JBoss Tools since 3.0.0 Beta1 version. You can download the JBoss Tools from *JBoss download site* [http://jboss.org/tools/download.html].

Smooks tools (JBoss Tools) run with the latest Eclipse and other required plug-ins (GEF, EMF, etc.).

You should download the latest IDE for Java EE developers from *Eclipse site* [http://www.eclipse.org/downloads/]. It contains many plug-ins (GEF, EMF, etc.) required by Smooks dev tools.

The Smooks Configuration depends on having all of the appropriate Smooks runtime jars in the path of the Eclipse Plug-in or Java Project in the Eclipse workspace. The easiest solution is to do the following:

- 1) Download the Smooks distribution from here: http://www.smooks.org/mediawiki/index.php?title=Downloads . Grab the latest "ALL" distribution (as of today, it is the Smooks v.1.2.2 "ALL" distribution) and it will include binaries, examples, etc.

- 2) Extract files from the archive somewhere on your machine.

- 3) In your Eclipse workspace, copy the Smooks jars into a directory of your Eclipse plug-in or Java project named "lib".

- 4) For your Eclipse Plug-in or Java Project, right-click on the project and select Properties.

- 5) Select the "Java Build Path" item in the Properties list, select the Libraries tab, and click "Add JARs"

- 6) In the Jar Selection dialog, select all the jars in the "lib" directory mentioned in step 3 and click OK.

- 7) Click OK to close the Properties dialog. Now you should see a "Referenced Libraries" node that appeared in your project hierarchy in Eclipse.

Now let's progress to more advanced topics.

Tasks

This chapter describes the main tasks a user can be faced during Smooks tools usage.

2.1. New Smooks Configuration File Creation

Select the project where you want to create new Smooks Configuration File and right-click on it, select in the menu *New > Other,* then find *Smooks > Smooks Configuration File.* Click the *Next* button.
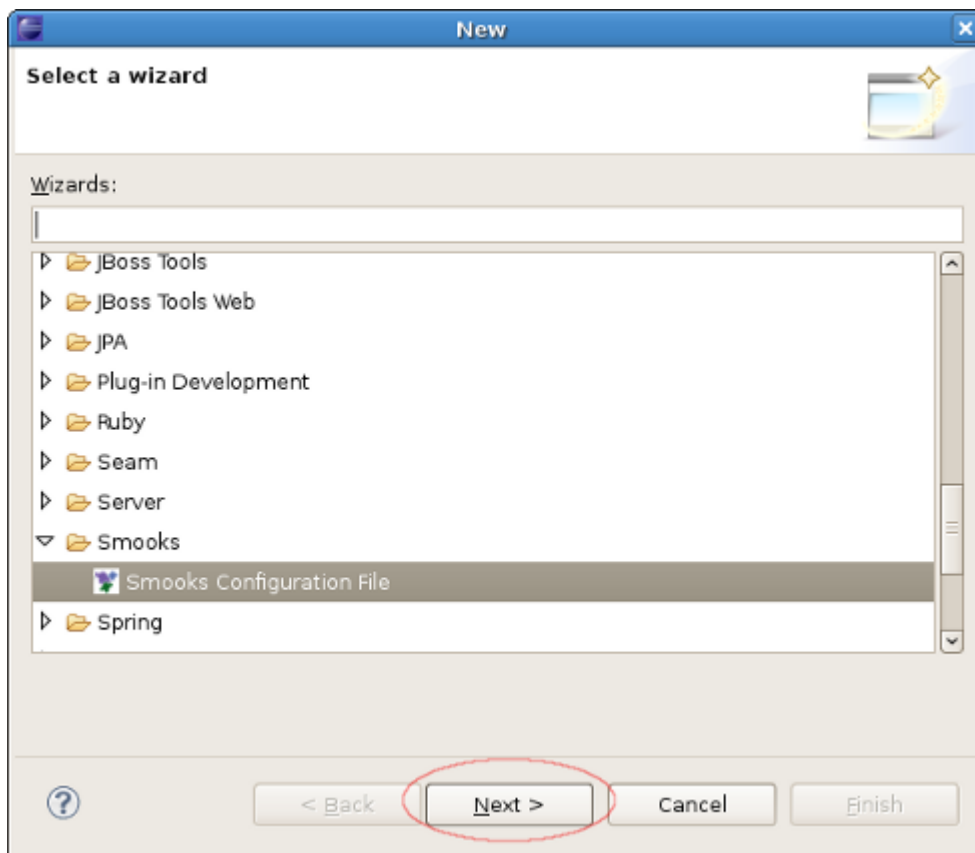

Figure 2.1. Selecting Smooks Configuration File Wizard

The first wizard page is a file path creation page. Select the *src* folder to be the files container, and input the name *smooks-config.xml.* Click *Next.*
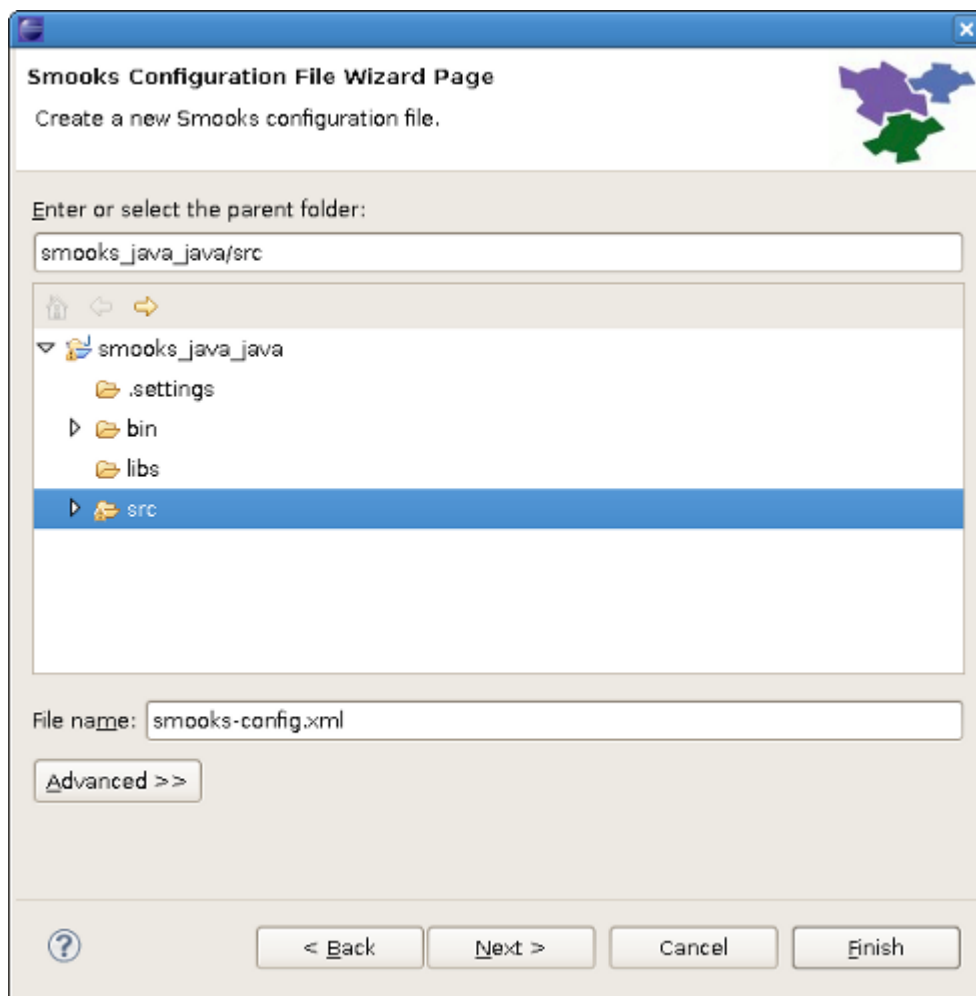
Figure 2.2. Choosing the configuration file container and the file name

The second wizard page allow you to select Smooks configuration file version. Select the appropriate one and click *Finish* to complete the wizard.

2.2. Input Task Configuring

Input task configuring is an obligatory step for your smooks project creation. You can configure it on the Process page of the editor: look for the "Input Task" in the Process Map at the top of the page.
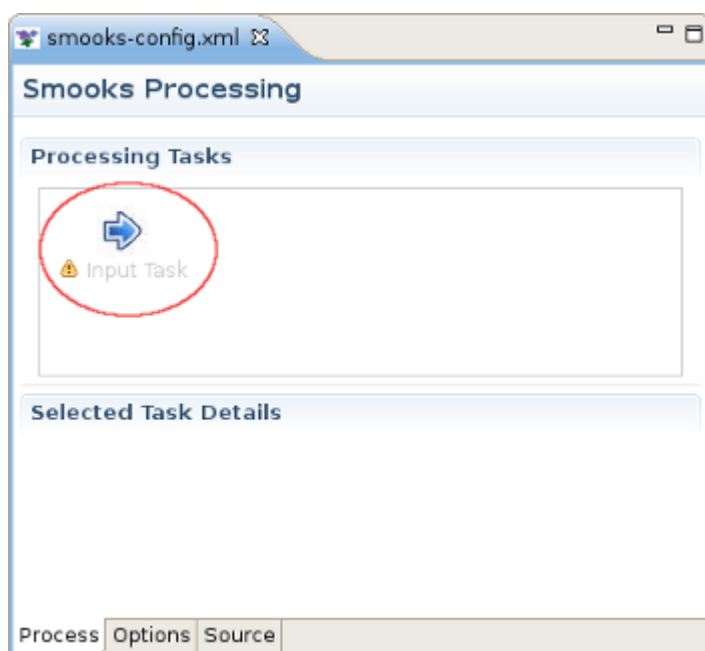
Figure 2.3. Input Task Configuring

Select it and you will see all the properties to set for the Input reader of your Smooks configuration. "Input type" corresponds to the type of data that you will be working with. For example, to work with incoming CSV (Comma-separated Values) data, you would specify "CSV" in the drop-down list. Each reader type has slightly different configuration details that must be set in the "Input configuration" area. For instance, the CSV reader requires you to specify details such as the encoding, quote character, separator character, and the list of incoming fields. The EDI reader requires the encoding and the path to the Mapping Model describing the incoming data. In the *Input data* section, you specify some sample data that conforms to your reader configuration.

Once you've specified your reader configuration and sample data, you can see the input model rendered in a tree form in the *Input model* section. On the picture below you can see the correct configuration of some XML input task.
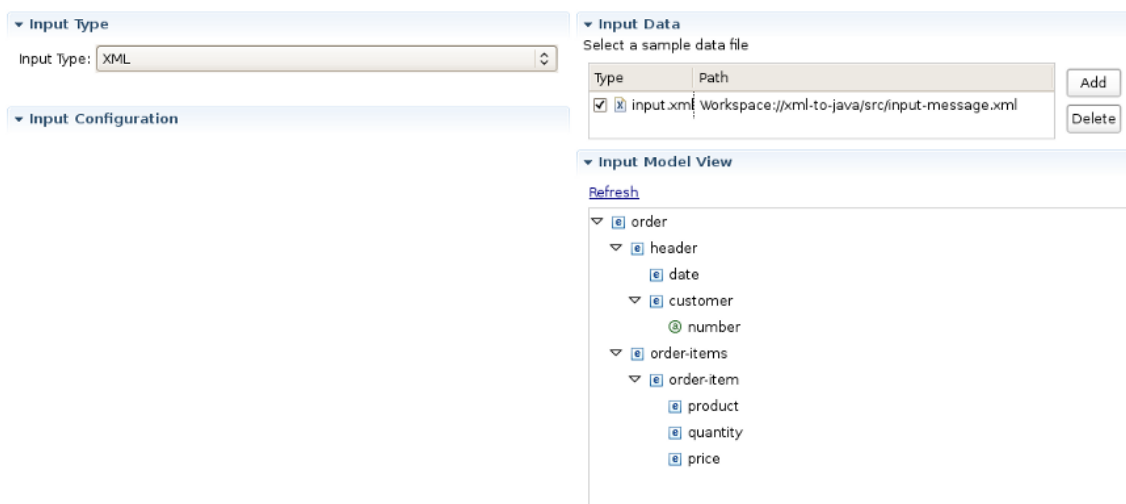
Figure 2.4. Input Task Configuring

2.3. "Java Mapping" or "Apply Template"?

Though there are many options in Smooks as far as what you can do with input data such as transformation, routing, and persistence, this version of the Smooks Configuration Editor focuses only on these areas: mapping to java and applying templates to create different output formats. If you have a set of Java classes you want to use the incoming data for, you can use the "Java Mapping" task to specify those classes and use drag and drop to map between the input model generated by the reader and elements in the output model. Or if you simply want to transform your output to one or more formats, you can use the "Apply Template" task to map it to a CSV file, XML or XSD file (and other formats in the future).

> **Note**
>
> Now you can't transform your output directly, using only Input and Template tasks. You should use Mapping as an interagent between these tasks.

2.4. Java Mapping Task

If you decide to do Java Mapping, you need to make sure that your Input reader has been set up and you have some sample data specified. Then you should select *Input Task* in the Process tab and click the plus (+) sign to the right of the icon. Select *Java Mapping* from the popup menu and it will appear to the right, connected to *Input Task* . Then select *Java Mapping* task.
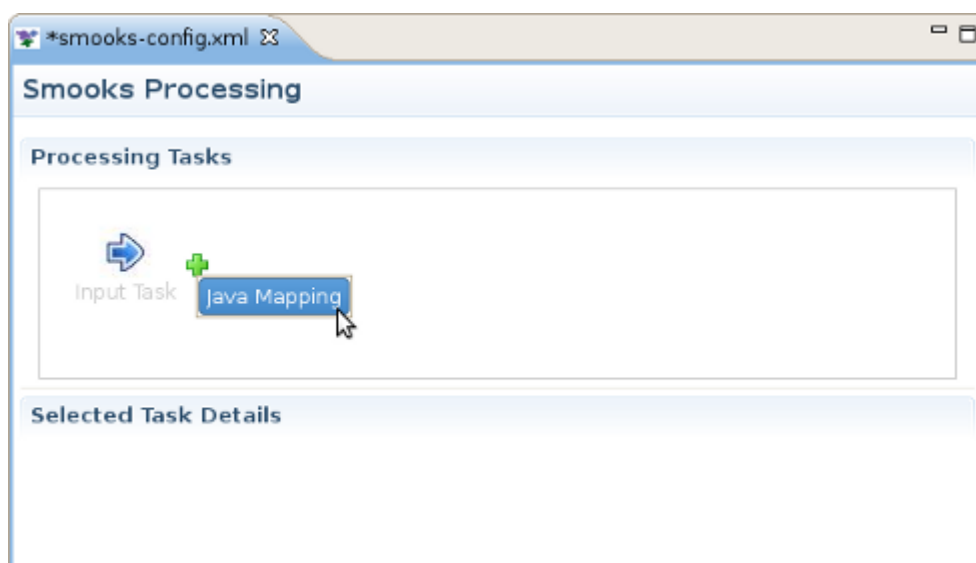
Figure 2.5. Java Mapping configuration

Another method of adding *Java Mapping* element to the canvas in the Processing Tasks section is to right click Input Task element and select *Java Mapping* in the popup menu
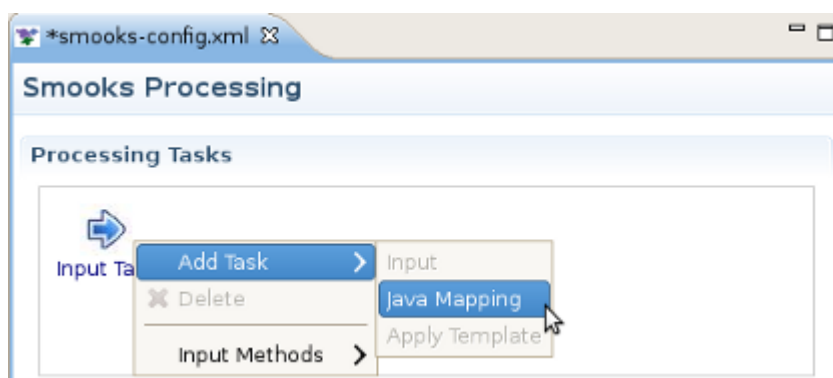

Figure 2.6. Java Mapping configuration

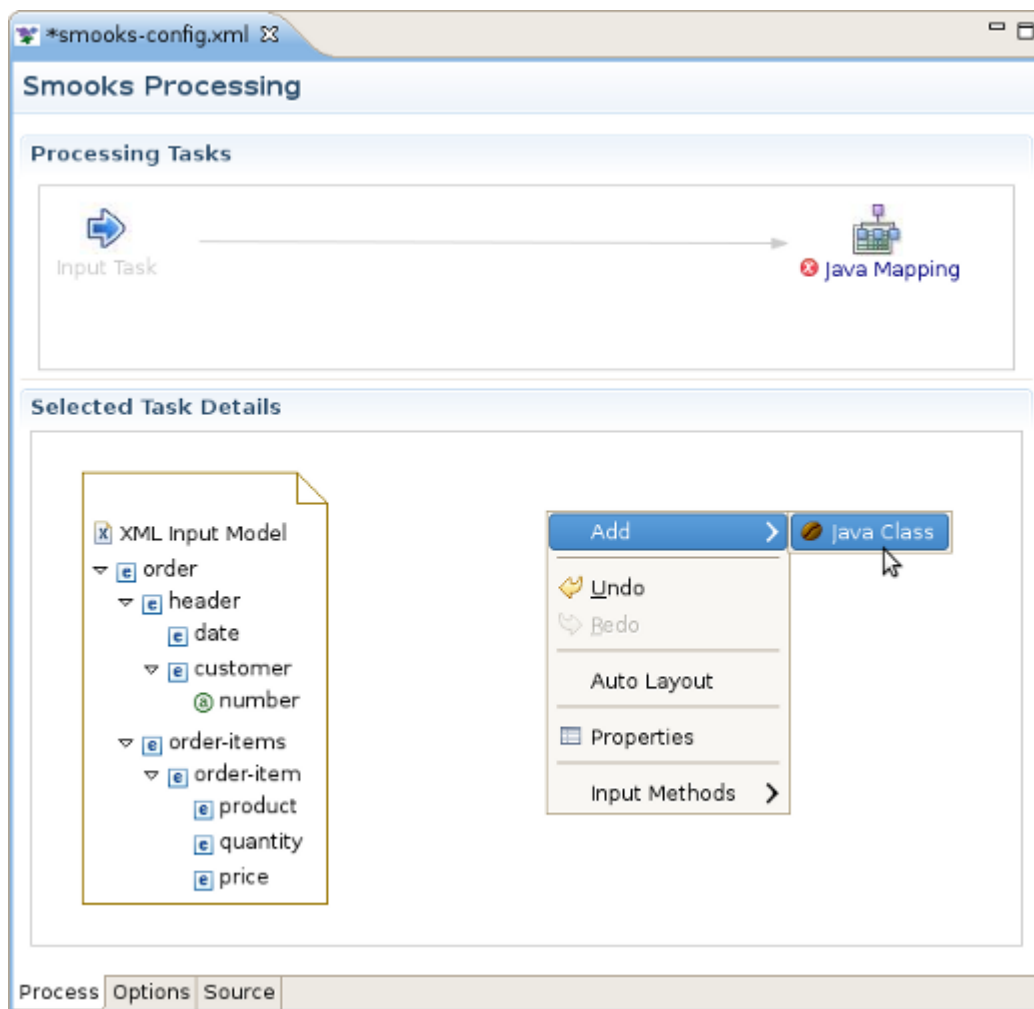Right-click on the canvas in an empty space and select "Add ->Java Class".

Figure 2.7. Java Mapping configuration

Java Bean Creation wizard appears. Specify a unique identifier for the new class, the class path, whether it's an array or not, and if it is a collection, also specify the collection class. If the Java class is specified, you'll see a list of the properties in the box below. Click *Finish* when you're done. Now with the input and output models on the canvas, you can click and drag from the various input elements to corresponding output elements. Make sure to connect collection elements to corresponding collection elements. Finally your mapping should look nearly like the one on the picture below.
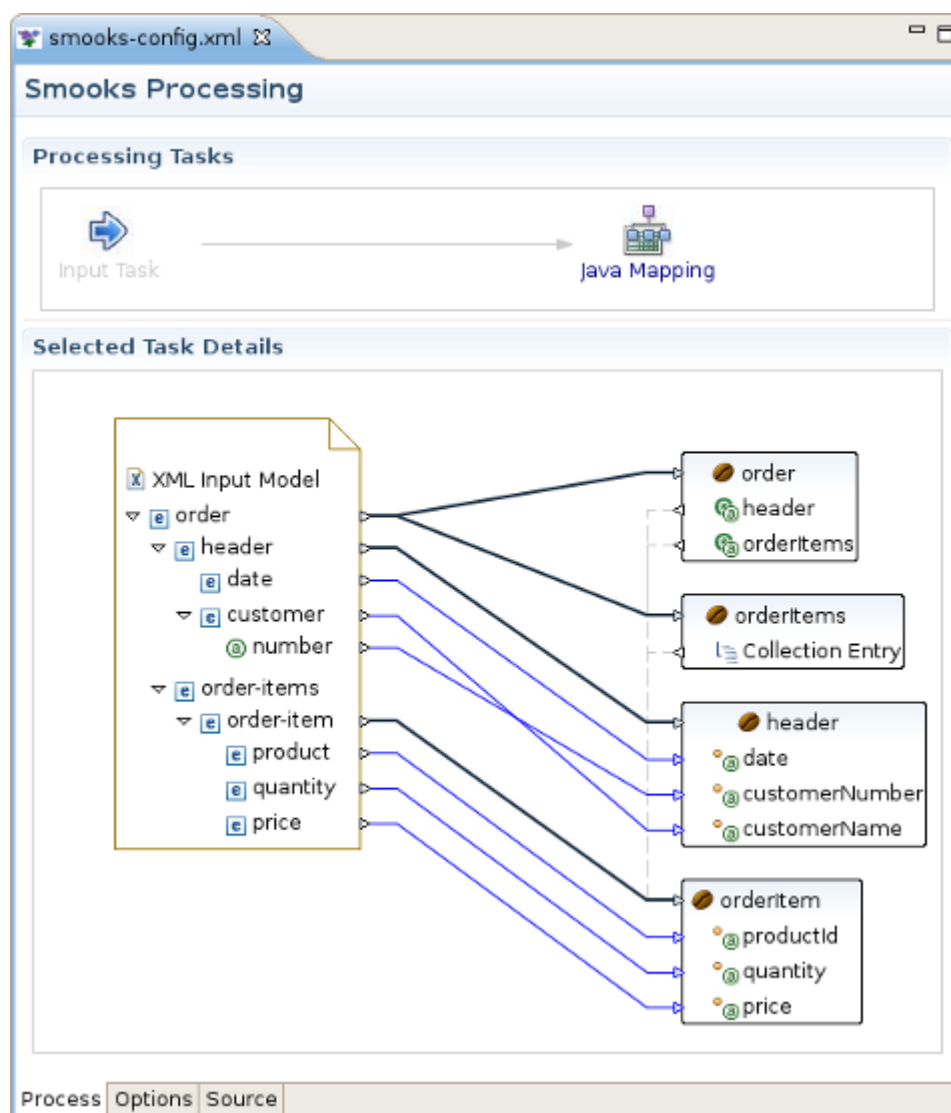
Figure 2.8. Final Mapping schema

2.5. Apply Template Task

The "Apply Template" task works very similarly to the *"Java Mapping" task* , where you map between an input model and an output model. Select the *Java Mapping* task you want to use as the input model in the Process Map pane and click the plus (+) sign to the right of the icon.
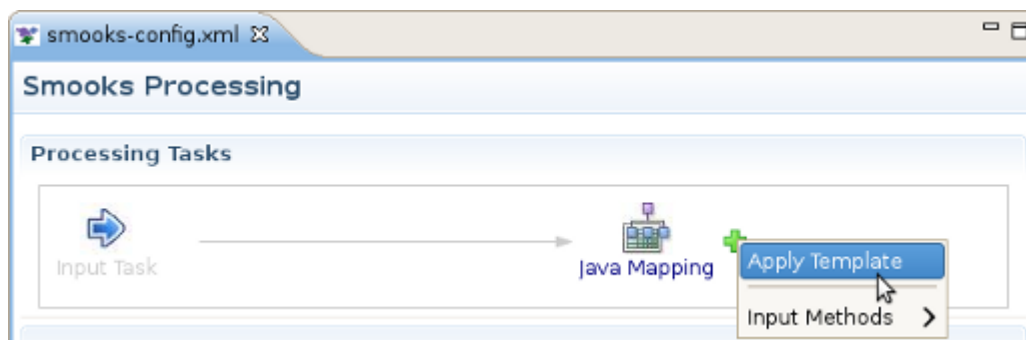
Figure 2.9. Apply Template configuration

Select "Apply Template" from the popup menu and it will appear to the right, connected to the task you created it from. Then select the "Apply Template" task. Once you've defined your template model, you can click and drag from the various input elements to corresponding output elements in the template. Make sure to connect collection elements to corresponding collection elements

2.6. Smooks Configuration testing using Smooks Run Configuration

This option is intended to view the results of Smooks transforming procedure. For more detailed information about this option please go *here* [http://community.jboss.org/wiki/UsingtheSmooksRunConfigurationtotestSmooksConfigurations].

Reference

This chapter includes detailed reference information about Smooks Tools.

3.1. Smooks Configuration Editor

This chapter describes the following tabs of the Smooks Configuration Editor:

- *Process tab*

- *Options tab*

- *Source tab*

3.1.1. Process tab

The Process tab of the Smooks Configuration Editor helps to configure different types of transformations. By default smooks configuration file is opened in this editor.If you have another default settings for editor opening you should left click smooks configuration file and select:*Open With->Smooks Configuration Editor*.

The Process tab has two sections:

- *Processing Task section*

- *Selected Task Details section*

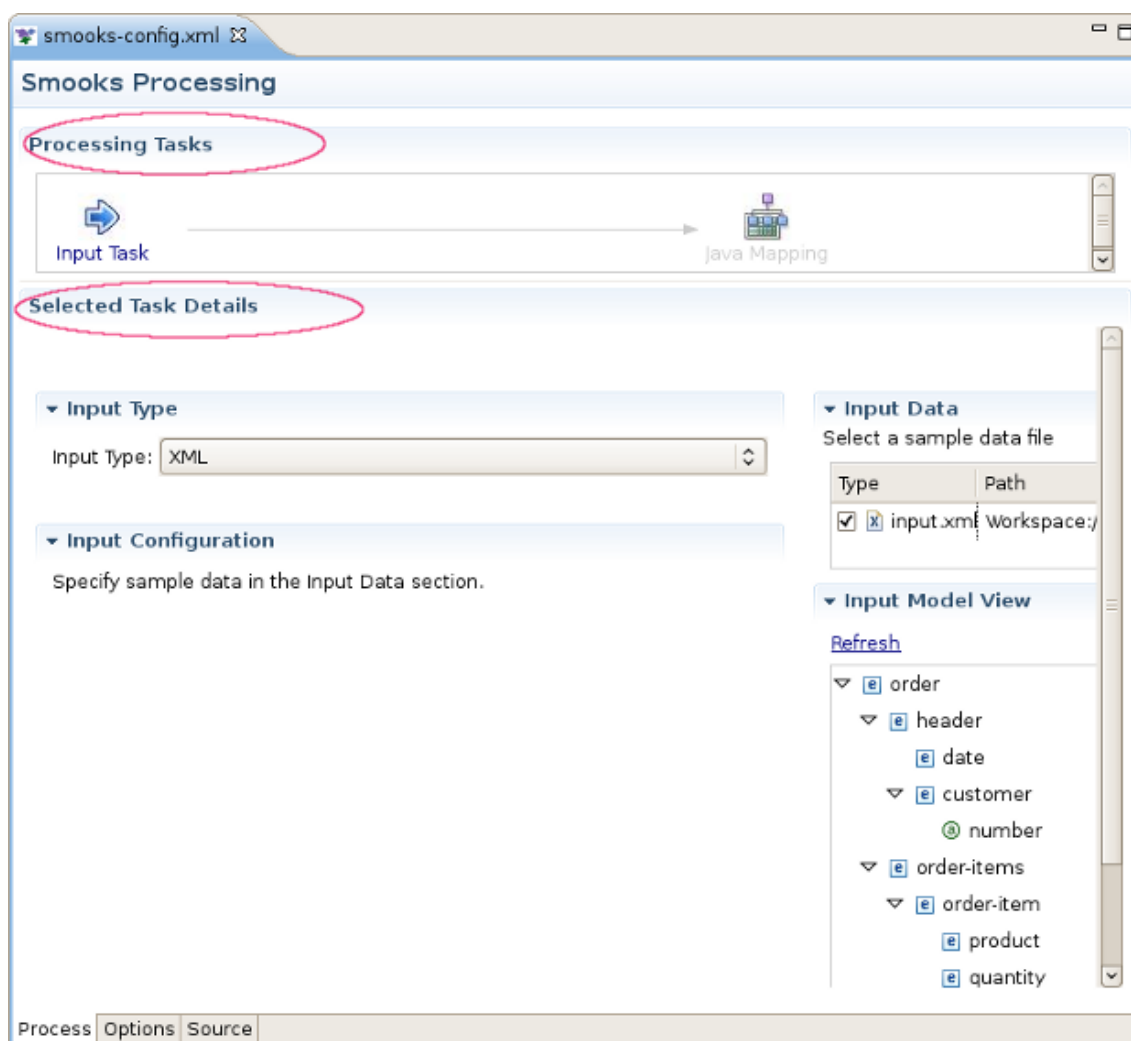You can see them on the picture below.

Figure 3.1. Two Sections of the Process tab.

3.1.1.1. Processing Task section

Using the popup menu in the Processing Task section you can select which types of technologies(templating or mapping ones) you will use for transofmation:

The descriptions of the popup menu options are in the following table.

Table 3.1. Process Tab. Processing Task section.

Option	Description	Default
Add Task	Select one of the following tasks according to the necessary type of Source and Result types of the files: • *Input* -this task is required and appears automatically when Smooks config file is created. You should just configure it properly.	

Option	Description	Default
	• *Java Mapping* • *Apply Template*	
Delete	Click this option if you want to delete some task from the section. Note:you can't delete input task because it's required.	
Input Methods	Choose one of the following methods: • System • Simple • Amharic(EZ+) • Cedilla • Cyrillic • Inuktitut • IPA • Multipress • SCIM Bridge Input Method • SCIM Input Method • Thai-Lio • Tigrigna-Eritrean(EZ+) • Tigrigna-Ethiopian(EZ+) • Vietnamese • X input Method	System

3.1.1.2. Selected Task Details Section

The options of this section depends on the selected task in the Processing Task section. Because there are 3 types of tasks there are 3 different sets of its options in the Selected Task Details Section. They will be described one by one.

3.1.1.2.1. Selected Task Details Section for Input Task.

On the picture below you can find an example of Selected Task Details Section view if XML is selected as input type.

Figure 3.2. Selected Task Details Section for Input XML Task.

As you can see on the picture above Input Configuration section is empty for XML input file. But this section has special configuration options for CSV,EDI,JSON,Custom input files.

Here are the screens of these configuration options:

- CSV:

Figure 3.3. Selected Task Details Section for Input CSV Task.

- EDI:

Figure 3.4. Selected Task Details Section for Input EDI Task.

- JSON:

Figure 3.5. Selected Task Details Section for Input JSON Task.

- Custom:

Figure 3.6. Selected Task Details Section for Input Custom Task.

All the input task configuration positions can be found in the table below:

Table 3.2. Selected Task Details Section. Options for Input Task.

Option	Description	Default
Input type	Select your type of input file. If don't find your type in the list,you should use Custom type: • No Input • XML • Java • XSD/WSDL	XML

Option	Description	Default
	• CSV • EDI • JSON • Custom	

Input configuration	• *No Input* -no info required
	
	• *XML* -no info required
	
	• *Java* -no info required
	
	• *XSD/WSDL* -no info required
	
	• *CSV*
	
	   • *Fields* -Comma separated list of CSV record field names
	
	   • *Separator Character* -Field separator character. Default of ','.
	
	   • *Quote Character* -Quote character. Default of '"'.
	
	   • *Record Name* -Name of csv record element.Default:csv-record
	
	   • *Root Name* -Name of csv root element.Default:csv-set
	
	   • *indent* -Add indentation character data to the generated event stream. This simply makes the generated event stream easier to read in its serialized form. Useful for testing etc.Default:true
	
	   • *Skip Lines* -Number of lines to skip before processing starts. Default of 0.
	
	• *EDI*
	
	   • *Target Profile* -Defines the target profile
	
	   • *Encoding* -The character encoding. Default "UTF-8"
	
	   • *Mapping Model* -Defines the EDI Mapping Model configuration for processing the EDI message
		• *CSV*
		
		   • not defined
		
		   • ','
		
		   • '"'
		
		   • csv-record
		
		   • csv-set
		
		   • true
		
		   • 0
		
		• *EDI*
		
		   • not defined
		
		   • UTF-8
		
		   • not defined
		
		   • true
		
		• *JSON*
		
		   • not defined
		
		   • element
		
		   • UTF-8
		
		   • not defined
		
		   • false
		
		   • not defined
		
		   • not defined
		
		   • ""(an empty string)

Option	Description	Default
	stream to a stream of SAX events that can be processed by Smooks. • *Validate* -This attribute turns on/off datatype validation in the EDI Parser. Validation is on by default. It makes sense to turn datatype validation off on the EDI Reader if the EDI data is being bound into a Java Object model. • *JSON* • *Target Profile* -Defines the target profile • *Array Element Name* -The element name of a array element. Default of 'element'. • *Encoding* -encoding: The default encoding of any JSON message InputStream processed by this Reader. Default of 'UTF-8'. • *Illegal Element Name Char Replacement* -If illegal characters are encountered in a JSON element name then they are replaced with this value. By default this is not defined, so that the reader doesn't doesn't search for illegal characters. • *Indent* -Add indentation character data to the generated event stream. This simply makes the generated event stream easier to read in its serialized form. Useful for testing etc.Default:false. • *Key Prefix on Numeric* -The prefix character to add if the JSON node name starts with a number. By default this is not defined, so that the reader doesn't search for element names that start with a number. • *Key Whitespace Replacement* -The replacement character for whitespaces in a JSON map key. By default this not defined, so that the reader doesn't search for whitespaces. • *Null Value Replacement* -The replacement string for JSON NULL values. Default is ""(an empty string).	• 'json' • not defined • *Custom* • no defaults

Option	Description	Default

	• *Root Name* -The element name of the document root. Default of 'json'.
	
	• *Key Maps* -Defines a JSON element name mapping The "from" key will be replaced with the "to" key or the contents of the element.
	
	• *Custom*
	
	  • *Target Profile* -
	
	  • *Class* -Custom reader class.
	
	  • *Handlers* -Set a handler on the reader instance e.g. an EntityResolver, ErrorHandler etc.
	
	  • *Features* -Reader Features List
	
	  • *Parametres* -Resource Parameters	

Input Data	You should select a data file using *Add* and *Delete* buttons	
Input Model View	Using this view you can see the structure of your input file.If the file has been changed, to see the changes click *Refresh*link.	

3.1.1.2.2. Selected Task Details section for Java Mapping Task.

Selected Task Details section for this task is presented by the graf, that lighten the process of java mapping.
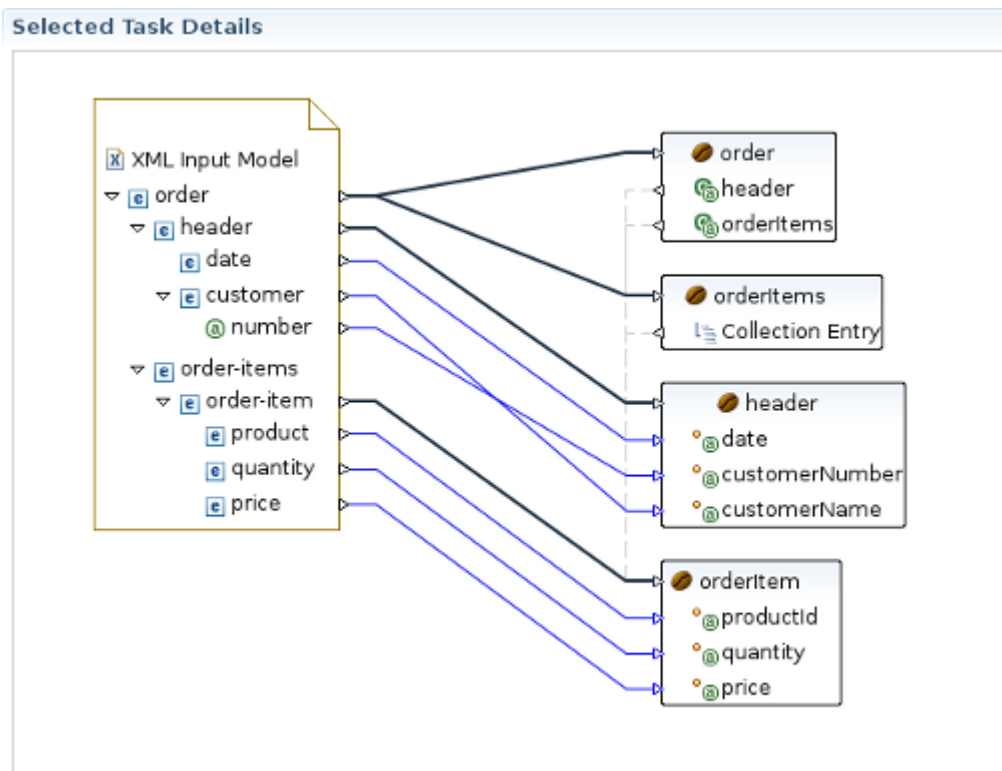
Figure 3.7. Selected Task Details Section for Mapping Task.

This graphical editor allow you to perform drug/drop operations with the nodes of transform data to map the source data to target data. When you save the changes in the graphical editor the correct Smooks configuration file content will be generated.

3.1.1.2.3. Selected Task Details section for Template Task.

Selected Task Details section for this task is presented by the graf, that is similar to the one in the *previous section*.
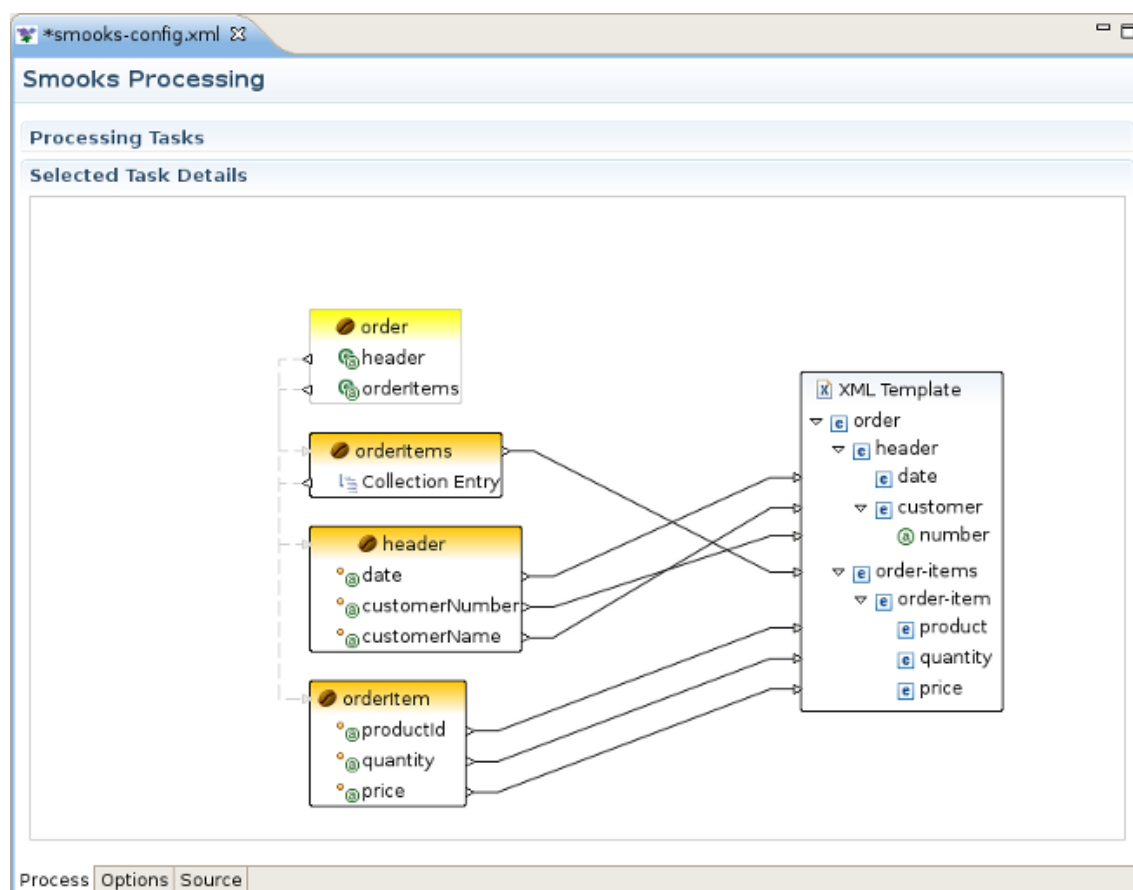
Figure 3.8. Selected Task Details Section for Template Task.

3.1.2. Options Tab

This section describes Options tab of the Smooks Configuration File editor, gives short recommendations how this tab can be used during the project configuring.
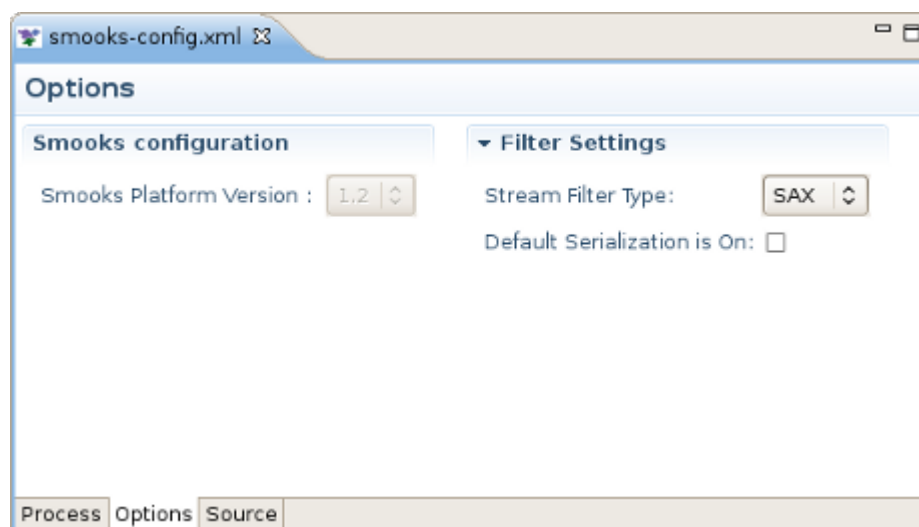

Figure 3.9. Options tab of the Smooks Configuration File editor

3.1.2.1. Smooks Configuration section

In the Smooks Configuration section of Options Tab only one element is availible:Smooks Platform Version

Figure 3.10. Smooks Configuration section of Options tab

This parameter is not rechangable, and is set according to the vesion of the Smooks libraries that are added to the project.

3.1.2.2. Filter Settings Filter section

In Filter Settings section you can set the following global options responsible for Smooks filtering configuring:
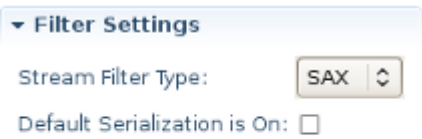

Figure 3.11. Filter Settings section of Options tab

This behavior can be turned off using this global configuration parameter and can be overriden on a per fragment basis by targetting a Visitor implementation at that fragment that takes ownership of the Result writer (in the case of SAX filtering), or simply modifies the DOM (in the case of DOM filtering). As an example of this, see the FreeMarkerTemplateProcessor.

Table 3.3. Options Tab. Filter Settings section.

Option	Description	Default
Stream Filter Type	Determines the type of processing model that will be used. Please refer to *Filtering Process Selection section* [http://www.smooks.org/mediawiki/index.php?title=V1.2:Smooks_v1.2_User_Guide#Filtering_Process_Selection_.28DOM_or_SAX.3F.29] of the official Smooks User Guide for more information about these models: • *SAX*	DOM

Option	Description	Default
	• *DOM*	
Default Serialization is On	Defines whether default serialization should be switched on. Default serialization being turned on leads to locating StreamResult/DOMResult to the Result objects provided to the Smooks.filterSource method and to serialization all the events to that Result.	true

3.1.3. Source Tab

This section provides information about Smooks Source Editor Page.

3.1.3.1. XML Source Editor

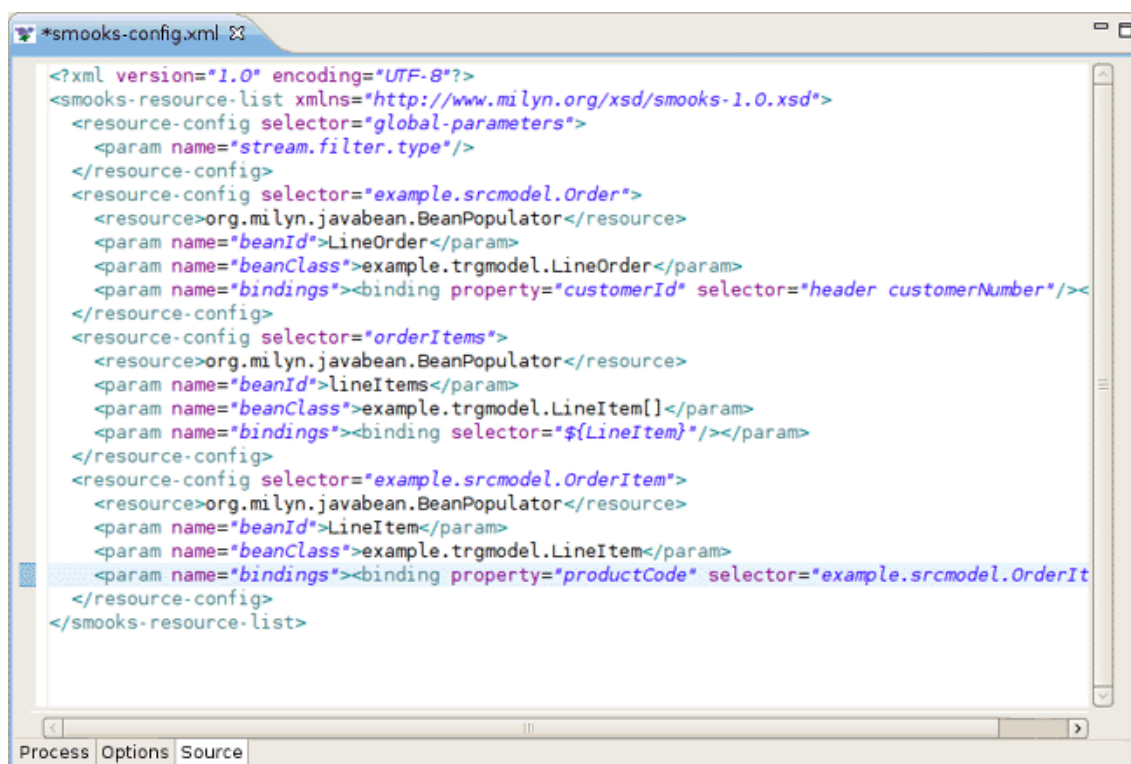You can use this editor to edit the Smooks Configuration file directly.

Figure 3.12. Graphical Editor

3.1.3.2. Error underlining in Graphical Editor

If the Smooks tools can't understand the configuration file or the configuration file is illegal (XML structure isn't right for Smooks Configuration file, etc.), the error is underlined.
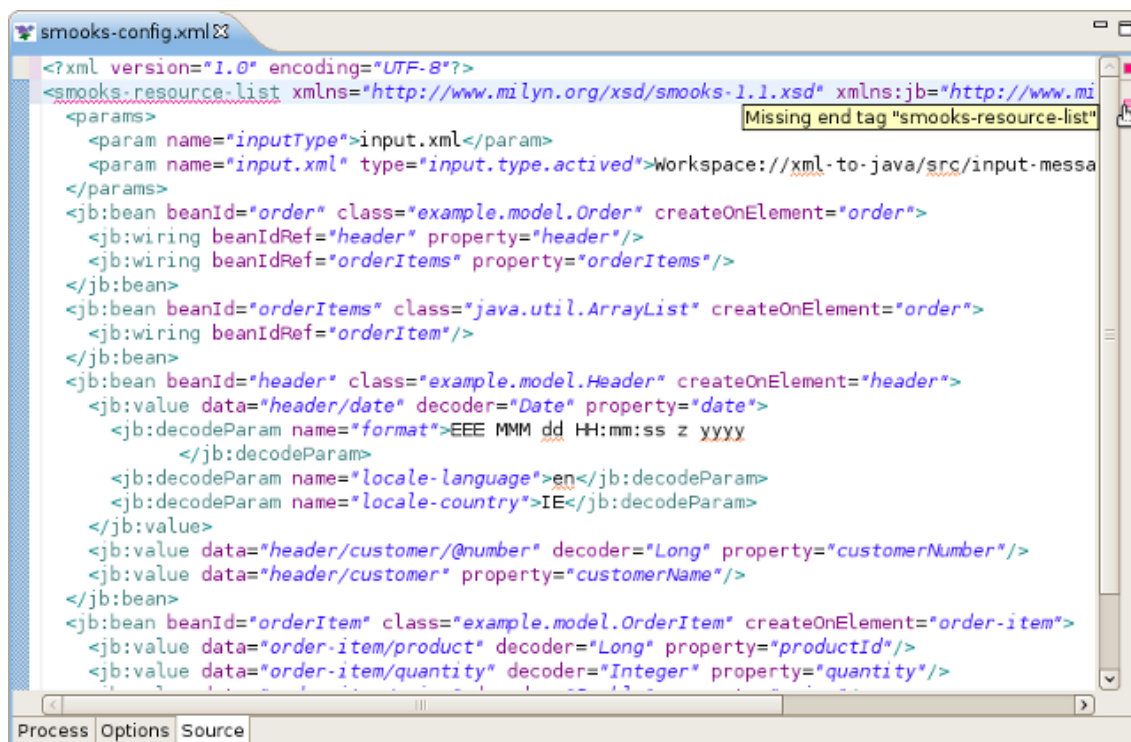
Figure 3.13. Graphical Editor

3.1.3.3. Smooks Configuration File Validator

Smooks configuration file validator will validate your Smooks configuration file. Just right-click on the file and then click on the *Validate* button. The validator can be enabled/disabled in *Window -> Preferences -> Validation*:
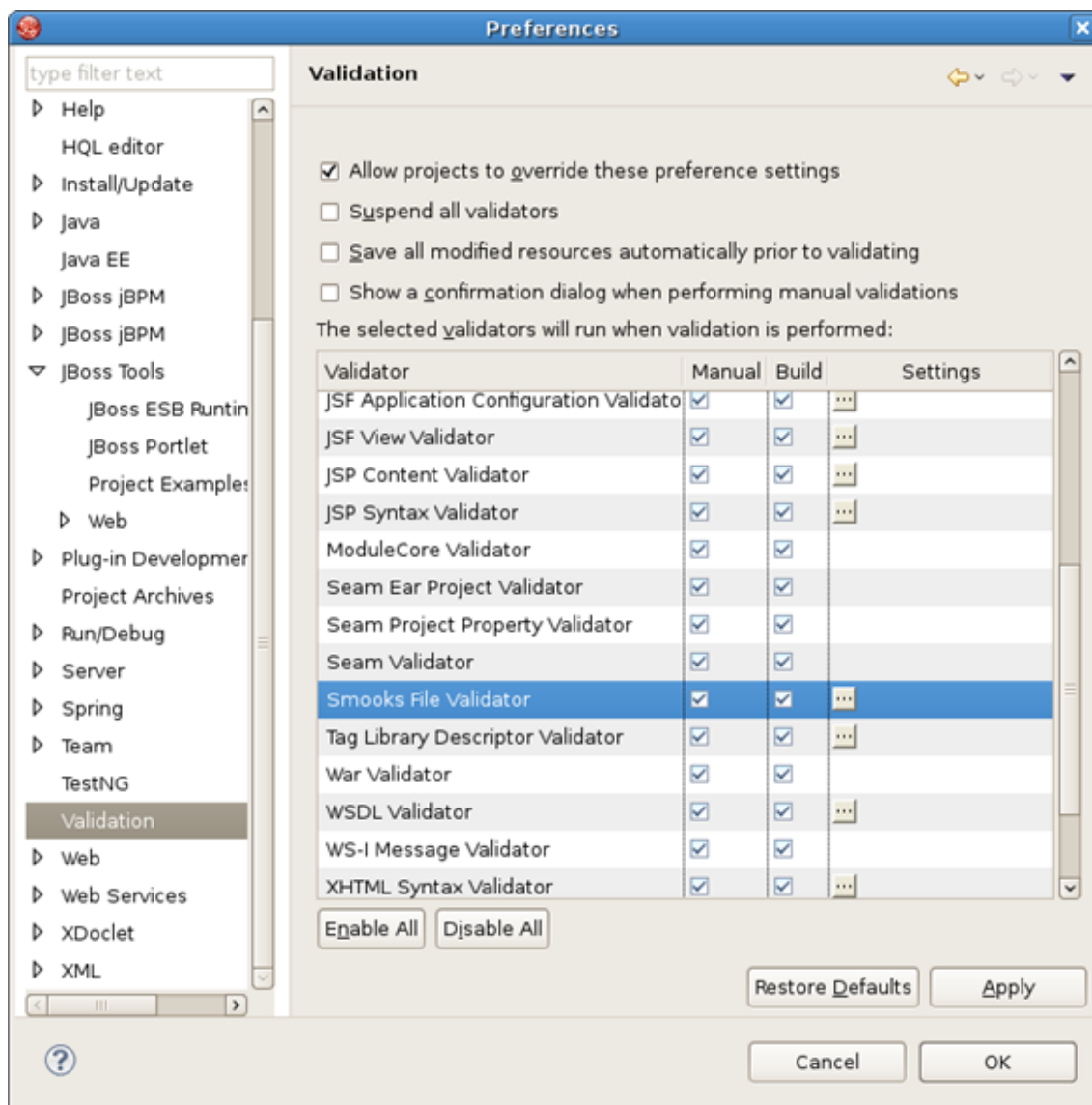
Figure 3.14. Validation: Smooks Configuration File Validator

You can set up your Smooks validator to include, exclude groups to validate and specify rules for validation. Just click on the Settings button and use the options provided:
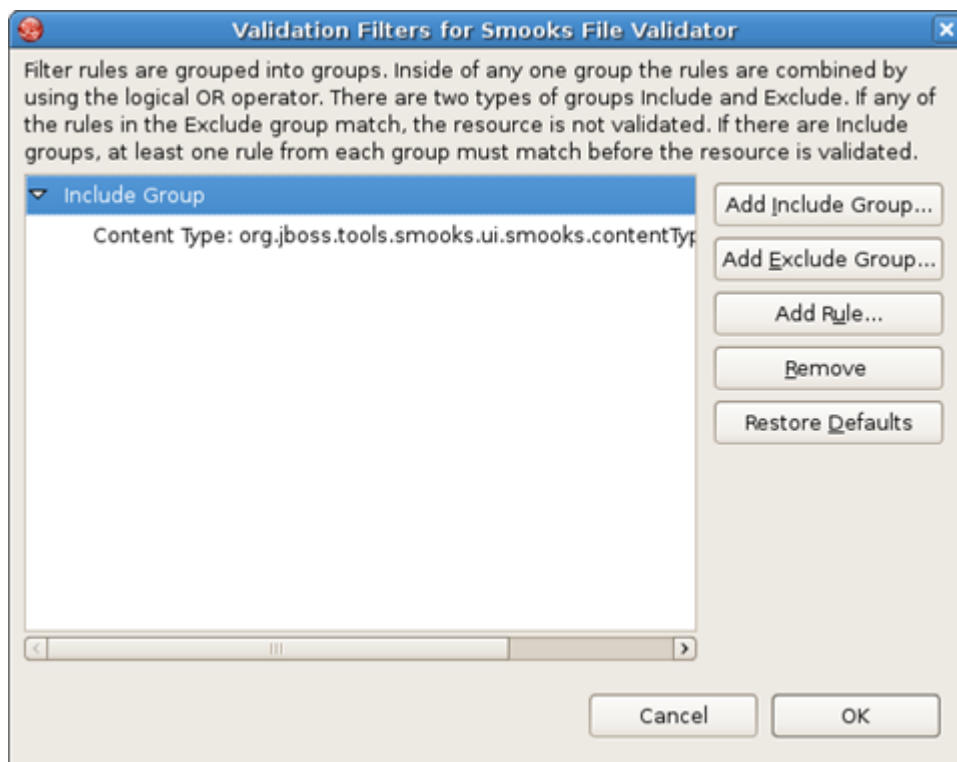
Figure 3.15. Smooks Configuration File Validator Settings

Summary

In conclusion, with this document you know all the capabilities of Smooks Tools and could easily start with them. The chapters above walked you through the steps on how to create and configure some XML to JAVA mapping project. If you have questions or suggestions concerned both the documentation and tools behavior, you are welcome to JBoss Tools Users forum. Please, use Jira to report bugs and requests on documentation.

4.1. Other relevant resources on the topic

All JBoss Developer Studio/JBoss Tools release documentation you can find at *http:// docs.jboss.org/tools* in the corresponding release directory.

The latest documentation builds are available at *http://download.jboss.org/jbosstools/nightly-docs* .

For more information about Smooks technology please visit *Smooks Technology Home Page* [http://www.smooks.org/mediawiki/index.php?title=Main_Page]