Smooks Dev Tools Reference Guide



Version: 1.1.0.CR1

1. Introduction	1
1.1. Key Features of Smooks Tools	1
1.2. What is Smooks?	1
1.3. What is Smooks Tools?	2
1.4. How to install Smooks dev tools?	3
2. Quick Start	4
2.1. Download Smooks Java-to-Java Example	4
2.2. Mapping rules	4
2.3. Smooks Configuration File Wizard	6
2.4. Smooks Editor	7
2.4.1. Adding the input data	8
2.4.2. Adding the target Bean 1	11
2.5. Mapping 1	15
3. Smooks Design Page 2	20
3.1. Selector generate dialog 2	21
3.2. Smooks Parse Type 2	22
3.3. Import Smooks Configuration page 2	24
3.4. File selection 2	25
4. Smooks Source Editor Page	28
4.1. XML Source Editor 2	28
4.2. Error underlining in Graphical Editor 2	28
4.3. Smooks Configuration File Validator 2	29

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.	<u>Smooks</u> Configuration File Wizard
Smooks Editor	Smooks Editor helps configure the created Smooks configuration file.	<u>Smooks Editor</u>

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.

¥ smooks-config.xml 🛙				- 8
Overview				
Smooks configuration			Configuring Smooks Input	
Set the description for this S	mooks configuration file.			
			input : Define a new input	
Smooks Platform Version :	1,2	0	typeand edit the input type	
Name				
Norrie 1			Configuring Smooks	
Author :			Transform Processes	
 Global Paramters 			Processing Page : Use the	
Define the global parameters	s for the Smooks configuration file			
Param Type stream.filter	type	bleve		_
		New		-
		Remove		
		Up		
		Down		
	[Properties		
Smooks Default Setting	9			
Conditions				
▶ Profiles				~
Overview Input Process Sour	rce			

Figure 1.2. Smooks Form 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 <u>Eclipse site</u> [http:// www.eclipse.org/downloads/]. It contains many plug-ins (GEF, EMF, etc.) required by Smooks dev tools.

Now let's progress to more advanced topics.

Quick Start

This "Quick Start" will show how to use the Smooks tools to create/edit the Smooks configuration file for Java2Java data transformation.

This chapter gives you a brief understanding of the Smooks tools.

2.1. Download Smooks Java-to-Java Example

Visit <u>smooks examples page</u> [http://www.blogjava.net/Files/reloadcn/smooks_java_java.zip] to download the "java-to-java" example project.

Tip:

The example project is based on Maven, we suggest you to create a new eclipse Java project, copy the example code and .jar files from the example project and paste them into the eclipse Java project.

2.2. Mapping rules

At the begining of working with Smooks tools, we need to understand the Java-to-Java mapping rules of this example. The transform input Java class is *example.sourcemodel.Order*, its structure looks like this :



Figure 2.1. Order class

Let's describe the structure of *example.sourcemodel.Order* in short:

- Order class contains a Header class and a List class.
- The Header class has customerName, customerNumber, priority as properties.
- In Order class the objects of the OrderItem class are stored in the List.
- OrderItem has price, productId, quantity as its properties.

Let's look at the transform target Java class : LineOrder.

On the screen below you can see its structure:



Figure 2.2. LineOrder class

- LineOrder class contains 3 properties:customerId,customerName,priority and an array of lineItems.
- The Header class has customerName, customerNumber, priority as its properties.
- OrderItem objects are stored in a List.

Now it's the time to define the mapping rules to transform the input source java class to the target java class:

- Each instance of the Order class should be mapped with an instance of the LineOrder class.
- The value of customerName of the Header class should be assigned to the customerName property of a LineOrder instance.
- The value of customerNumber of the Header class should be assigned to the customerId property of a LineOrder instance.

- The value of priority of the Header class should be assigned to the priority property of LineOrder instance.
- For the orderitems List of the Order instance, it's necessary to map a new lineItems array instance of LineOrder, and each OrderItem instance in the orderItems List should be transformed to a LineItem instance, those instances will be put into the lineItems array.
- The value of productId of the OrderItem instance should be assigned to the 'productCode' of LineItem.
- The value of price of OrderItem should be assigned to the unitPrice of LineItem.

2.3. Smooks Configuration File Wizard

Select the Smooks Java-to-Java example project that you created and right-click, select the menu *New > Other,* then find *Smooks > Smooks Configuration File.* Click the *Next* button.

	New	×
Select a wizard		
<u>W</u> izards:		
type filter text		
🕨 🗁 jBoss Tools Web)	^
👂 🗁 JPA		
👂 🗁 Plug-in Develop	iment	
👂 🗁 Remote System	n Explorer	
👂 🗁 Seam		
👂 🗁 Server		
🗢 🗁 Smooks		
🔤 EDI Message	Mapping Configuration File	
🚏 Smooks Cor	figuration File	
🕨 🗁 SQL Developm	ent	
?	< <u>B</u> ack <u>N</u> ext > Cancel	Einish

Figure 2.3. 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*.

Sec. (* 1997)
Smooks Configuration File Wizard Page
Create a new Smooks configuration file.
Enter or select the parent folder:
smooks_java_java/src
マ ஜ⊌ smooks_java_java
🗁 .settings
🕨 🗁 bin
🗁 libs
🕨 🚌 src
File na <u>m</u> e: smooks-config.×ml
Advanced >>
Advanced 22
< Back Next > Cancel Finish

Figure 2.4. Choosing the configuration file container and the file name

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

2.4. Smooks Editor

When all the wizard steps are passed, a new Smooks Configuration file is created and the Smooks Configuration Editor will be opened automatically:

¥ smooks-config.xml 🛙			- 8
Overview			
Smooks configuration		Configuring Smooks Input	
Set the description for this S	mooks configuration file.	Input : Define a new input	
Smooks Platform Version :	1.2	typeand edit the input type	
Name :			
Author :		Configuring Smooks Transform Processes	
 Global Paramters Define the global parameters 	s for the Smooks configuration file	Processing Page : Use the	
Param Type stream.filter	type N	lew	
	Ren	move	=
		qu	
	Do	own	
	Prope	erties	
Smooks Default Setting	g		
Conditions			
Profiles			
Overview Input Process Sou	rce		~

Figure 2.5. Smooks Editor

On the bottom of this editor there are 5 tabs, they are : Overview, Reader, Message Filter, Graph, Source.

2.4.1. Adding the input data

At first, we need to add the Order class as the input data. To do this, click the Reader tab, swich to the Reader/Input editor page

To choose the transform data you should open *Reader* tab and click *Add* button in the Input Data section.

📽 smooks-config.xml 🛛	- B
Reader/Input	
Reader Edit the smooks readers	Add Remove Up Down
▼ Input Data	
Type Path	Extension Paramers
Overview Reader Message Filter Graph Sour	>

Figure 2.6. Input Data section

After clicking the button the Data Type Selection wizard is opened, and there are many data types in the list: Java, XML etc.

Note
There will become more and more data types on the list as the development goes on.
on.

Smooks Input Data Selection Wizard
Input Data Selection
Choose "Input Data Type" to open wizard dialog to select the input data.
Input Data Type
? EDI
? Json
S XSD/WSDL
X XML
D java
? CSV
Select a Java type(class,interface) as the input data.
(?) < Back Next > Cancel Enish

Figure 2.7. Data Type Wizard

Select the Java data type and click Next, you will see the JavaBean class selection page:

E	Ja	vaBean Mod	el Wizard		×
Select Java Select Java	abean Ibean model				
Class : Class				ls Array	Add Remove
?	< <u>B</u> ack	Next >	Cance	el 🗌	<u>F</u> inish

Figure 2.8. JavaBean Data Selection

Let's select the Order class from the example project as the source JavaBean class.After that click *Finish* and you should see the just added java class, being displayed in the Input Data Section.

📽 smooks-config.xml 🕴	- 8
Reader/Input	
Reader Edit the smooks readers	Add Remove Up Down
▼ Input Data	
Type Path	Extension Paramers
java example.srcmodel.Order	
K	
Overview Reader Message Filter Graph Source	O

Figure 2.9. Displaying the Structure of the Transformed Data

2.4.2. Adding the target Bean

After adding the input data, let's switch to the Message Filter page:

🐨 smooks-config.xml 🛛	- 8
Message Filter	
Message Filtering Resources	
Add.	ε ε
Up Down	
Overview Reader Message Filter Graph Source	

Figure 2.10. Displaying the Structure of the Transformed Data

Click the Add button, the Add Child wizard page will be shown, let's select the Java Binding v1.2 -> Bean v1.2 and click *OK* to add the Bean :



Figure 2.11. Add Child wizard

Then we can find on the Message Filter page the Bean v1.2 item added, click on it ,a Bean v1.2 section should appear to the right of the page. The section represents a GUI for editing Bean v1.2 properties:

💱 *smooks-config.xml 🛛 🖾		- 6
Message Filter		
Message Filtering Resources		Bean v1.2
○ Bean v1.2	Add Remove	Create On Element Name : Browse Namespace : Bean Id* : Bean Id* : Class* : Class* : Target Profile : Extend Lifecycle : Add Binding
Overview Reader Message Filter Graph Source		

Figure 2.12. Bean v1.2 section

You should input lineOrder as the Bean's id in the "Bean Id" field.Then you should click Browse button near the Class textfield and select LineOrder as the Bean's class.

Note

When you enter a value for the Class text field, you can type 'Alt + /' and the content assist pop dialog will be shown, so it's will increase the efficiency. If 'Alt + /' doesn't work, try to use the 'Ctrl + space'('Key Assist' of Eclipse).

Let's click the Add Binding link at the bottom of this page, it will show a dialog to ask the binding of what Bean properties you will prefer to add. Select all and click OK, you will see a list of Binding elements added under the Bean item.

Then you should add an array of beans called lineItems and a Bean LineItem. These two classes and Id values are:

Array Bean:

• BeanId : lineItems;

• Class : example.trgmodel.LineItem[]

LineItem Bean :

- BeanId : lineItem;
- Class : example.trgmodel.LineItem

Note

In Smooks , to be an array of Beans , the class value should end with '[]'. This means that this bean is an array type.

To finish this work, use the Add Binding link to add all the bindings for the Bean.

2.5. Mapping

How to map the java class just added?

First, you should know what to transform:

- a. Bind the "customerName" property of the Order#header to the "customerName" property of LineOrder.
- b. Bind the "customerNumber" property of the Order#header to the "customerId" property of LineOrder.
- c. Bind the "price" property of the OrderItem to the "unitPrice" property of LineItem.
- d. Bind the "productId" property of the OrderItem to the "productCode" property of LineItem.
- e. Bind the "quantity" property of the OrderItem to the "unitQuantity" property of LineItem.

Make sense? You should open Design tab and click Add button. In opened Add Child Wizard, please, select Add Smooks Resource>Java Binding>Bindings.

First of all, you should bind *Order* to *LineOrder*. To do this one *Bindings* Item should be added to the Resource List.

¥ smooks-config.xml 🛙		- 8
Smooks 1.1 - Design Page		
Message Filtering Resources	Resource List	
Resource List		
Add Smooks Resource 💦 🔪	Reader >	~
⊘ Undo Ctrl+Z	Templating >	
Sedo Ctrl+Y	Java Binding 💦 🔪	📽 Bindings
	Datasources >	WS
X Delete	Scripting >	Browne
of Cut	Fragment Routing	BIOWSE
Eopy	Database >	
📴 Paste	♦2 Global Parameters	
Validate	Conditions	
Validate	* Profiles	
	* Import Smooks Configuration	
	* Custom Resource Configuration	
	🕸 IO Router	
		1
- Innut Data		
• Input Data		
Type Path		Add
		Delete
		Delete
Design Source		

Figure 2.13. Mapping

Click on it and you should see *Bindings* Section to the right of *Message Filtering Resources*.Write *lineOrder* in the *Bean id* field.

Select *example.trgmodel.LineOrder* in the *Class* field and *example.srcmodel.Order* in the *Name* field.Your *Smooks Configuration Editor* should look like this:

¥ *smooks-config.xml छ		- 8
Smooks 1.1 - Design Page		
Message Filtering Resources	Bindings	
 ✓ ◆ Resource List ▷ ◆ Global Parameters 	Bean Id* :	lineOrder
🕨 🚸 Bindings	<u>Class* :</u>	.trgmodel.LineOrder Search Class
	Target Profile :	
	Extend Lifecycle :	0
	Create On Eler	nent
	Name :	example.srcmodel.Order Browse
	Namespace :	
▼ Input Data		
Type Path		Add
		Delete
Design Source		

Figure 2.14. Mapping

Now you should bind two values *"customerNumber"* property of the Order#header to the *"customerId"* property of LineOrder.To do this right click your just created *Bindings Item >Add Smooks Resource>Value*.

essage Filtering Resources	Value header/c	usto
'	Decoder :	
▽ ♦ Bindings	Default :	
Value header/custo	Property :	customerName Select property
	Setter Method :	Select method
	Data	
	Name* :	header/customerName Browse
	Namespace :	
Input Data		
Type Path		Add

Figure 2.15. Mapping Editor

Note:
Please, remember that the root nodes of the source data and target data *MUST* be connected.
This rule is not only for Java-to-Java, but also is relevant for XML-to-Java

Do the same operation with the other properties in this and other classes.

It's also necessary to bind orderItems property of the Order to lineItems property of LineOrder. The difference from the previous steps is that the orderItems is presented by the list of Order and lineItems is the massive of LineItem. To do the binding, you should use Wiring Item. First you should create a new Binding item. This is a configuration that allows you to construct an object graph. Select example.trgmodel.LineItem[] in the Class field and orderItems in the Name field, in bean Id write lineItems. Your Smooks Configuration Editor should look like this:

¥ smooks-config.xml 🛛		- 8
Smooks 1.1 - Design Page		
Message Filtering Resources	Bindings	
 ✓ ◆ Resource List ✓ ◆ Global Parameters 	Bean Id* :	lineItems
▶ ◆ Param stream.filter.type ♥ ◆ Bindings	<u>Class* :</u>	el.LineItem[] Search Class
Wiring lineItems	Target Profile :	
 Value header/custo Value header/custo 	Extend Lifecycle :	
🕨 🚸 Bindings	Create On Elem	nent
	Name :	orderitems Browse
 Value example.srcm Value example.srcm 	Namespace :	
▶ Input Data		
Design Source		

Figure 2.16. Creating Wiring Item

Then you should right click *lineOrder* binding item *>Add Smooks Resource>Wiring*.Select *lineItem* in *Bean id Ref*.

The final step is to add new Wiring item to lineOrder Binding.In the Bean id Ref select *lineItems* and in *Property* select *lineItems*.

Now, you should save the file and the Smooks tools generate the correct configuration file content.

OK, let run the test class: example.Main. (Please, make sure that the Smooks engine loads the correct configuration file).

Success!

Smooks Design Page

In this section you will explore the details of Smooks Design Page usage.

Smooks Design Page is one tab of the Smooks Editor.

¥ smooks-config.xml 🕴	- 8
Smooks 1.1 - Design Page	
Message Filtering Resources	Resource List
 ✓ * Resource List ✓ * Global Parameters ▷ * Param stream.filter.type ✓ * Bindings * Value header/custo * Value header/custo ✓ * Bindings * Wiring line!tem ✓ * Bindings * Wiring line!tem ✓ * Bindings * Value example.srcm * Value example.srcm 	Default Condition Ref :
▼ Input Data Type Path	Add
	Delete
Design Source	

Figure 3.1. Design Page

This page consists of 3 sections: Message Filtering Resources, Input Data and the section devoted to the configurations of a selected in a Message Filtering Resources section Item.

You can bind classes using Bindings item on the Design page of the Smooks Editor.

Resource list item in *Message Filtering Resources* corresponds to the root element of the *smooks-config.xml* file.You can configurate the next four properties using corresponding fields:

- default-selector:Defines the selector that will be applied to all resource-config elements in the smooks-config.xml, where a selector is not defined.
- *default-selector-namespace*:Defines the default selector namespace.

- default-target-profile: Defines the default target profile that will be applied to all resources in the smooks-config.xml.
- default-condition-ref: The condition is applied to resources where an empty "condition" element (i.e. <condition/>)is defined. The condition element does not reference a globally defined condition.

The configuration editor also edits the <u>Smooks parse type</u> and manages the <u>Import Files</u>.

3.1. Selector generate dialog

Using Selector generate dialog you can allow the user to select a node for generation "selector" for smooks-config file. To open the dialog you should left click Browse button in *Default Selector* section.

E Selector g	jenerate dialog	×
Input Message:	Click to add Inp	ut Data
Sperator Char I		
Selector generate policy :		~
Full Path Only Name		
Contractine Contractine		
	Cancel	

Figure 3.2. Selector generate dialog

When the dialog is opened you should click "Click to add input data"to set the selector. In the opened *Data Type Selection* Wizard you should choose the type of the file and the file itself you want to load to the *Selector generate dialog*. After that your dialog should look nearly like that:

E Selector g	enerate dialog 🛛 🗙
Input Message:	Click to add Input Data
🗢 🖻 order	
🗢 🖻 header	
e date	
▽ 🖻 customer	
Inumber	
⊽ 🖻 order-items	
⊽ 🖻 order-item	
e product	
🖲 quantity	
e price	
Sperator Char : /	0
Selector generate policy :	
● Full Path ○ Only Name	
© rain ann O only Name	
	Cancel OK

Figure 3.3. Selector generate dialog after adding Input Data

The final step is to choose the node you want to make a selector for *smooks-config.xml*.

3.2. Smooks Parse Type

If your config file doesn't contain *Global Parametres* item yet, you should add it by right click *Resource List>Add Smooks Resource>Global Parametres*.After this step you should add *Param* to just created *Global Parametres*.

¥ *smooks-config.xml 🛛 🖾			- 8
Smooks 1.1 - Design Page			
Message Filtering Resources		Global Parameters	
Add Creative Description	N I A	Para an	
Add Smooks Resource		* Param	
💛 <u>U</u> ndo Add	Ctrl+Z		
Sedo	Ctrl+Y		
💢 <u>D</u> elete			
of Cut			
Еору			
💼 <u>P</u> aste			
Validate			
Input Data			
Design Source			

Figure 3.4. Adding Param to created Global Parametres

Then left click *Param* and in the Param section, you will see a set of text fields: *Name*, *Type* and *Text*. *Name* value is required. To set a parse type you should write SAX/DOM to the *Text* field.

¥ *smooks-config1.xml ⊠			- 8
Smooks 1.1 - Design Page			
Message Filtering Resources ▼	Param strea Name* : Type : Text Value :	stream.filter.type	
	► CDATA V	'alue	
▶ Input Data			
Design Source			

Figure 3.5. Filling Param section field Name and Text Value for set a Parse Type

To apply changes to the *smooks-config.xml* click Save button (Ctrl+S).

3.3. Import Smooks Configuration page

Right click *Resource List>Add Smooks Resource>Import Smooks Configuration*, then you should see a new *Import Smooks Configuration* item in the list (see the figure below).

¥*smooks-config.xml ⊠	- B
Smooks 1.1 - Design Page	
Message Filtering Resources	Import Smooks Configuration
 Resource List order Global Parameters Custom Resource Configuration Bindings Bindings Bindings Bindings 	File* : Browse

Figure 3.6. Import Data

Left click the added item and to the left of *Message Filtering Resources Import Smooks Configuration* section should appear. Click the *Browse* button and select the file you want to import(see *File selection* section).

Click Save and the <import> tag with the corresponding file path as a *file* property will be added to the *smooks-config.xml*.

3.4. File selection

Some attributes in Smooks elements should store path to a file as a value (e.g. attribute "mappingModel" of the element "edi:reader"). You can see a Browse button near the fields, that set such a value to the attribute (e.g. "Mapping Model" field set a value to the "mappingModel" attribute).

¥ smooks-config.xml 🛛 🖾		- 8
Smooks 1.1 - Design Page		
Message Filtering Resources	EDI Reader	
✓	Target Profile : Encoding : <u>Mapping Model :</u>	Browse
✓ Input Data Type Path		Add)elete

Figure 3.7. EDI:Reader element

When you click *Browse*, the dialog prompts you click one of two buttons: *Browse File System* and *Browse WorkSpace*.

E	×
File :	
	Browse File System Browse WorkSpace
(?)	Cancel
_	

Figure 3.8. Browse dialog

Using the buttons you can locate the file you need. If you click *Browse File System* button you specify the path of, or browse to, a file system directory.

v w user pro	eclipse	
Places	Name	✓ Modified
🔯 Desktop	🎾 about_files	05/01/2009
🗇 File System	📁 configuration	17:09
	📁 dropins	17:07
📁 Documents	🎾 features	07/22/2009
🎾 Music	🎾 jdbc:hsqldb:hsql:	Tuesday
芦 Pictures	🎾 p2	05/19/2009
📁 Videos 📃	📁 plugins	07/22/2009
	🎾 readme	05/01/2009
🖶 🗚 dd 📄 😑 Bernove	🍘 about.html	05/01/2009
		🞇 <u>C</u> ancel

Figure 3.9. Browse file system

If you click *Browse WorkSpace* button, you specify the path of, or browse to, a workspace relative working directory.

Select Files	×
▽ ≧ milvn-smooks-example-xslt-basic	
Isettings	
▷ isrc	
▷ 🗁 target	
☑ .classpath	
🗷 .project	=
🖹 input-message.xml	
S MySchema.xsd	
🗷 pom.xml	
README.TXT	-
🚏 smooks-config.×ml	
🖹 smooks-config.xml.ext	~
Cancel OK	

Figure 3.10. Browse Workspace

Smooks Source Editor Page

This section provides information about Smooks Source Editor Page.

4.1. XML Source Editor

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



Figure 4.1. Graphical Editor

4.2. 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.

🕱 *smooks-config.xml 🕴 🗖 🗖
<pre><?txml version="1.0" encoding="UTF-8"?> <mmooks-resource-list xmlns="http://www.milyn.org/xsd/smooks-1.0.xsd"> <mmooks-resource-list xmlns="http://www.milyn.org/xsd/smooks-1.0.xsd"> <mmooks-resource-list xmlns="http://www.milyn.org/xsd/smooks-1.0.xsd"> <mmooks-resource-list xmlns="http://www.milyn.org/xsd/smooks-1.0.xsd"> <mmooks-resource-list xmlns="http://www.milyn.org/xsd/smooks-1.0.xsd"> <mmooks-resource-list xmlns="http://www.milyn.org/xsd/smooks-1.0.xsd"> <mmooks-resource-config selector="global-parameters"> <mmooks-resource-config selector="global-parameters"> <mmooks-resource-config selector="example.srcmodel.Order"> </mmooks-resource-config> <mmooks-resource-config selector="example.srcmodel.Order"> </mmooks-resource-config> </mmooks-resource-config> </mmooks-resource-config> </mmooks-resource-list></mmooks-resource-list></mmooks-resource-list></mmooks-resource-list></mmooks-resource-list></mmooks-resource-list></pre> </td

Figure 4.2. Graphical Editor

4.3. 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*:

۲		Preferences				
ty	pe filter text	Validation			⇔ - ⇔ -	-
Þ	Help					
	HQL editor	🗆 allemensierte te energide these es				
Þ	Install/Update	Allow projects to override these pr	ererence	setting	js	
₽	Java	Suspend all validators				
	Java EE	Save all modified resources automatically prior to validating				
Þ	JBoss jBPM	Show a <u>c</u> onfirmation dialog when p	performin	g manı	ual validations	
Þ	JBoss jBPM	The selected validators will run when v	alidation	is perf	ormed:	
~	JBoss Tools	Validator	Manual	Build	Settings	
	JBoss ESB Runtin	JSF Application Configuration Validate				
	JBoss Portlet	JSF View Validator				
	Project Example:	JSP Content Validator				
	Þ web	JSP Syntax Validator				
Þ	Plug-in Developmer	ModuleCore Validator				
	Project Archives	Seam Ear Project Validator				
Þ	Run/Debug	Seam Project Property Validator				
Þ	Server	Seam Validator				
Þ	Spring	Smooks File Validator		⊻		
Þ	Team	Tag Library Descriptor Validator			<u></u>	
	TestNG	War Validator				
	Validation	WSDL Validator			<u></u>	H
Þ	Web	WS-I Message Validator				
Þ	Web Services	XHTML Syntax Validator				~
Þ	XDoclet	Enable All Disable All				
Þ	XML		-			
<	······································		F	lestore	Defaults Apply	
(?			Car	ncel ОК	

Figure 4.3. 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:

ilter rules are grouped into groups. Inside of any one group the ru Ising the logical OR operator. There are two types of groups Includ	les are combined by
he rules in the Exclude group match, the resource is not validated proups, at least one rule from each group must match before the	e and Exclude. If any of I. If there are Include resource is validated.
🛩 Include Group	Add Include Group
Content Type: org.jboss.tools.smooks.ui.smooks.contentTyr	Add <u>E</u> xclude Group
	Add Rule
	Bemove
	Restore <u>D</u> efaults
Cance	ОК

Figure 4.4. Smooks Configuration File Validator Settings

We hope, our guide will help you to get started with JBoss Smooks Tools. Besides, for additional information you are welcome on <u>JBoss forum</u> [http://www.jboss.com/index.html? module=bb&op=viewforum&f=201].