Struts Tools Reference Guide



Version: 3.1.0.CR1

1. Introduction	. 1
1.1. Key Features of Struts Tools	. 1
1.2. Other relevant resources on the topic	. 2
2. Projects	. 3
2.1. Creating a New Struts Project	. 3
2.2. Importing an Existing Struts Project with Any Structure	8
2.3. Adding Struts Capability to an Existing Web Application	. 9
2.4. Relevant Resources Links	16
3. Editors	17
3.1. Web.xml Editor	17
3.2. Graphical Editor for Struts Configuration Files	17
3.2.1. Diagram View	17
3.2.2. Tree View	20
3.2.3. Source View	22
3.3. Graphical Editor for Tiles Files	24
3.3.1. Create New Tiles File	24
3.3.2. Tree View	25
3.3.3. Diagram View	27
3.3.4. Source	29
3.4. Graphical Editor for Struts Validation Files	31
4. Modules	38
4.1. When Importing a Struts Project	38
4.2. Editing Modules in an Existing Project	39
4.3. Adding New Modules	40
5. Code Generation	41
6. Struts Configuration File Debugger	44
7. Customizable Page Links Recognizer	45
8. Struts Project Verification	46
9. Relevant Resources Links	49

Introduction

If you prefer to develop web applications using Struts technology JBoss Tools also meet your needs. The professional developer toolset provides all necessary editors and wizards for creating Struts resources that enhances the process of building high-quality web applications.

Note:

Note that JBoss Tools support the Struts 1.1, 1.2.x versions.

In this guide you will learn how to take advantage of Struts support that JBoss Tools provide.

1.1. Key Features of Struts Tools

For a start, we propose you to look through the table of main features of Struts Tools:

Table 1.1. Key Functionality for Struts Tools

Feature	Benefit	Chapter
Struts Support	Step-by-step wizards for creating a new struts project with a number of predefined templates, importing existing ones and adding struts capabilities to non-struts web projects.	<u>struts support</u>
Support for Struts Configuration File	Working on file using three modes: diagram, tree and source. Synchronization between the modes and full control over the code. Easy moving around the diagram using the Diagram Navigator. Working with struts projects that have multiple modules. Possibility to use Struts configuration file debugger allowing to set break points on struts diagram and then launch the server in debug mode.	graphical editor for struts configuration file debugger
Support for Struts modules	A Struts module (struts-config.xml) is automatically created while creating a new project. There is also possibility to add new ones or edit already existing modules in your existing project or while importing Struts project.	modules
Verification and Validation	All occurring errors will be immediately reported by verification feature, no matter in what view you are working. Constant validation and errors checking allows to catch many of	<u>verification and</u> <u>validation</u>

Feature	Benefit	Chapter
	the errors during development process that	
	significantly reduces development time.	

1.2. Other relevant resources on the topic

All JBoss Developer Studio/JBoss Tools documentation you can find on <u>JBoss Tools project page</u> [http://docs.jboss.org/tools/].

The latest documentation builds are available on *nightly docs page* [http://download.jboss.org/ jbosstools/nightly-docs/].

Projects

JBoss Tools provide the following functionality when working with Struts:

- Create new Struts projects
- Import (open) existing Struts projects. You can import any project structure
- Add Struts capabilities to any existing Eclipse project
- Import and add Struts capabilities to any existing project created outside Eclipse.

Now, we'll focus on all these points more fully.

2.1. Creating a New Struts Project

JBoss Tools provides a New Struts Project Wizard that radically simplifies the process for getting started with a new Struts project. You just need to follow these steps:

Select File > New > Project... from the menu bar. Then, select JBoss Tools Web > Struts > Struts Project in this dialog box. Click Next:

•	New Project	×
Select a wizard		
<u>W</u> izards:		
type filter text		
JBoss jBPM		•
▽ 🗁 JBoss Tools Web		
👂 🗁 JSF		
🗢 🗁 Struts		
🕉 Struts Project		
👂 🗁 JPA		
Plug-in Development		
👂 🗁 Seam		
h Carina		
(2) < <u>B</u> ack	Next >	Enish Cancel

Figure 2.1. Selecting Struts Wizard

• On this form, provide the Project Name. You can also select where to create the project or use the default path.

Next to Struts Environment set which Struts version to use.

New Struts Project X		
Create Struts Proje	ct 🚳	
The Create New Project a brand new project. If project, just use the Im with it in JBoss Tools.	ct Wizard is used for creating ⁵ you already have a pre-existing aport Project Wizard to start working	
Project Name*	StrutsProject	
	Use default path*	
Location*	opt/work/workspace_jbds/StrutsProject	
Struts Environment*	Struts 1.1	
Template*	Struts 1.2 Didlik	
0	< Back Next > Finish Cancel	

Figure 2.2. Creating Struts Project

Тір:
Don't put spaces in project names since some OS could experience problems with their processing and searching these files.

You can select the KickStart template, then the project created will include a simple Hello World type application that is ready to run.

¢	New Struts Project X
Create Struts Proje	ect 🚳
The Create New Project a brand new project. I project, just use the In with it in JBoss Tools.	ct Wizard is used for creating f you already have a pre-existing nport Project Wizard to start working
Project Name*	StrutsProject
	✓ Use default path*
Location*	opt/work/workspace_jbds/StrutsProject
Struts Environment*	Struts 1.1
Template*	Blank
	KickStart
0	< <u>Back</u> Next > <u>Finish</u> Cancel

Figure 2.3. Choosing Struts Template

 Next, you register this application with the current servlet container defined for your workspace (JBoss AS, by default) in order to allow you to test your application more easily while still developing it. A new entry will be added in the servlet container configuration file to enable running the application in-place (called null deployment or link deployment). Uncheck the "Target Server" check box if for some reason you don't want to register your application at this point.

	New Struts Project	×
Web		
Servlet Version	2.4	
Context Path*	strutsApplication	
Runtime:*	JBoss 4.2 Runtime	New
Target Server:	JBoss Application Server 4.2	New Select All Deselect All
0	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

Figure 2.4. Registering The Project at Server

• On the next form, you can select the TLD files to include in this project:

🗧 New Struts Project 🗙
Tag Libraries Select tag library files you want to use in your project
TLDs
 struts-nested.tid fmt.tid sql.tid c.tid x.tid struts-html.tid struts-bean.tid struts-logic.tid struts-tiles.tid
⑦ < Back Next > Enish Cancel

Figure 2.5. Selecting Tag Libraries

After the project is created, you should have the following project structure (if you used the KickStart template):



Figure 2.6. Project Structure

Tip:

If you want to hide the jar files from Web App Libraries in view, select the downpointing arrow in the upper right corner, select *Filters...*, check the box next to Name filter patterns (matching names will be hidden), and type *.jar into the field. Then, click OK.

2.2. Importing an Existing Struts Project with Any Structure

For detailed information on migration projects to JBoss Developer Studio see <u>Migration Guide</u> [../../Exadel-migration/html_single/index.html].

2.3. Adding Struts Capability to an Existing Web Application

Here, we'll consider how to add Struts functionality (Struts libraries, tag libraries and a Struts configuration file) to any existing Web application project in your Eclipse workspace.

By adding a Struts Nature to your project, you can now edit files using JBoss Tools editors, such as the <u>Struts configuration editor</u> and the JBoss Tools JSP editor. To take advantage of this just right-click the project and select <u>JBoss Tools > Add Struts Capabilities</u> from the context menu. This will start the process of adding all necessary libraries and files to make a Web Struts project from your one.



Figure 2.7. Adding Struts Capabilities

In the wizard you should point to location of your deployment descriptor file web.xml and name of the project.

\$	Import Struts Project	×
Project Location Please select web.xr	nl location	
web.xml Location*	/opt/work/jbds_GA/workspace/JSFHello/WebContent/WEB-INF/web.xml	<u>C</u> hange
Project Name*	JSFHello	
0	Help < Back Next > Enish	Cancel

Figure 2.8. Choosing Project Location

After hitting *Next*, you will see the following screen. This screen simply means that you need to add at least one Struts module to your project to make this project a Struts project. Adding a Struts module means that a new struts-config.xml will be added to your project and registered in the web.xml file. In addition, all required Struts libraries will be added. To add a Struts module, select the *Add Struts Support* button.

Import Struts Project Project Modules The project should contain at least one Struts module to be imported		
ath on Disk		
ources Path		<u>⊆</u> hange
lodule Root		
lodule Root		<u></u>
	Add Struts Support	
0	Help < Back Next >	Enish Cancel

Figure 2.9. Project Modules

Here you can select what Struts Version, Servlet Class, URL Pattern and TLDs to add to this project.

\$	Add Struts Support		
Struts Project Image: Struts Support options			
Version*	1.2 💌		
Servlet Class:*	org.apache.struts.action.ActionServlet		
URL Pattern:*	*.do		
TLD Files:*	 struts-nested.tld fmt.tld sql.tld c.tld x.tld 		
	 struts-html.tid struts-bean.tid struts-logic.tid struts-tiles.tid 		
0	<u>Finish</u> Cancel		

Figure 2.10. Selecting Struts Support Options

When done, you will see the default Struts module configuration information. See how to Edit <u>Struts modules</u>.

	Import Struts Project				
Project Modules Configure Project Modules					
Name	URI				
<default></default>	/WEB-INF/struts-config.xml				
Name	<default></default>				
JRI	/WEB-INF/struts-config.xml				
Path on Disk	pt/work/jbds_GA/workspace/JSFHello/WebContent/WEB-INF/struts-config.xml	ange			
Sources Path	/opt/work/jbds_GA/workspace/JSFHello/JavaSource	ange			
Web Root	/opt/work/jbds_GA/workspace/JSFHello/WebContent	nange			

Figure 2.11. Project Configuration Information

On the last screen you can set the different folders for your project as well as register this application with a servlet container. If you want the libraries (.jar files) will be automatically added to your project, click on the checkbox *Add Libraries*.

¢	Import Struts Project	5
Project Folders Select Project Fol	s ders	0
Classes Folder	/opt/work/jbds_GA/workspace/JSFHello/WebContent/WEB-INF/classes	e
Lib Folder	/opt/work/jbds_GA/workspace/JSFHello/WebContent/WEB-INF/lib	e
Ant Build File	Chang	e
	Add Libraries	
Environment	Struts 1.2	\$
Servlet Version:	2.4	
Context Path*	JSFHello	
Runtime:*	JBoss 4.2 Runtime New.	
Target Server:	Choice list is empty. New. Select Deselect	All
0	Help < Back Next > Finish Canc	el

Figure 2.12. Registering the Project at Server

When done, you can open end edit the struts-config.xml file using useful Struts configuration file editor provided by JBDS. (The Struts configuration is shown below in the Tree viewer).



Figure 2.13. Struts-config.xml File

2.4. Relevant Resources Links

You can find more in-depth explanation on how to work with special wizards, editors and views that can be used in various scenarios while developing Struts applications in our <u>Visual Web Tools</u> <u>guide</u> [../../jsf/html_single/index.html].

Editors

In this chapter we'll introduce you to featured graphical editors for specific Struts files such as Struts Configuration files, Tiles files, Struts Validation files and web.xml.

3.1. Web.xml Editor

The *web.xml* file inside the *WEB-INF* folder is a deployment descriptor file for a Web Application. It describes the servlets and other components and deployment properties that make up your application.

JBoss Tools add the *web.xml* file to created Struts project automatically and provides a special editor for its editing. See the Visual Web Tools guide that gives a descriptive information on the *web.xml editor* [../../jsf/html_single/index.html#GraphicalWebApplicationFileEditor].

3.2. Graphical Editor for Struts Configuration Files

First, let's dwell on the Struts Configuration file editor.

This editor has three views with different representation of *struts-config.xml:* Diagram, Tree and Source. The views can be selected via the tabs at the bottom of the editor. Any changes made in one view are immediately visible when you switch to any other view.

Now, we'll consider every view in more detail.

3.2.1. Diagram View

The Diagram view graphically displays the Web flow of the application defined in the Struts configuration file.

suus-coning.xim ta		
etName		
/page s/ii	nputname.isp	
->	<u> </u>	
	/greeting	
	> GetNameForm	
	sayHello	
		Inspectareeting ico
		/pages/greeting.jsp

Figure 3.1. Diagram View

The Diagram view allows to edit navigation in your Struts application. Just by right-clicking anywhere on the diagram, you can use a context menu to create the building blocks of a Struts application:

- Actions
- Global forwards
- · Global exceptions
- JSP Pages

🥐 getName		
	Add	Action
	Create Comment	💍 Global Forward
	Paste Ctri	+ V Global Exception
	Select Element	
	Auto-Layout	
	👺 Verify	/pages/greeting.jsp
	Properties	> <u>A</u>
	ᄎ Generate Java Code	
	Input Methods	•

Figure 3.2. Diagram Context Menu

Along the upper-left side of the editor is a stack of seven icons for changing the behavior of the cursor in the diagram.



Figure 3.3. Editor Icons

The first icon switches to the default regular selection cursor, the second to the marquee selection cursor and the third to the new connection cursor. The last four icons switch the cursor to an insert cursor for each type of Struts build block listed above (and in the order listed).

For instance, clicking on the first of these four icons (the one with the gears) will switch the cursor to insert actions. Clicking anywhere in the diagram with this cursor has the same effect as right-click and selecting Add > Action... from the context menu with the regular cursor active. It's just more efficient to use this cursor if you're adding more than one action at once.

3.2.2. Tree View

The Tree view represents the different elements of the Struts application that are organized into functional categories on the left-hand side and a form for editing the properties of currently selected items on the right-hand side.

🕏 struts-config.xml 🛙				- 0
Struts Config Editor				
	• Properties Ed	itor		-
マ 龄 struts-config.xml	forward			
🙈 data-sources	ld:			
S form-beans	ClassName:		<u>B</u> rowse	i III
v a global-exceptions v a global-forwards	ContextRelative:		-	1
🔗 getName	Module:			i =
	Name:	sayHello		j
♂ sayHello	Path:	/pages/greeting.js	<u>C</u> hange	
3 controller	Redirect:		-	
a resources	Small-Icon:			i III
co plug-ins	Large-Icon:	[i 🗌
< III >>	Display-Name:			j
Diagram Tree Source				

Figure 3.4. Tree View

You can right-click on any node in the category tree and perform appropriate operations through a context menu. For instance, by right-clicking on the action-mappings category node, you can add new actions to the application.

🕏 struts-config.xml 🛙				- 0
Struts Config Editor				
▼ struts-config	• Propertie	s Editor		
 ✓ ➡ struts-config.xml ➡ data-sources ➡ form-beans ➡ global-exceptions ➡ ➡ global-forwards ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡ ➡	action-maj ld: Type: Comment:	ppings	<u>C</u> hange	
 ✓ A action-mappings ✓ A /greeting A /greeting <	Create Action Sort Paste Properties	Ctrl + V		
Diagram Tree Source	/erify			

Figure 3.5. Tree Context Menu

Let's consider the tree on the left more closely.

- Under the *data-sources* node you can create a Data Source object that will be configured and made available as a servlet context attribute.
- The *form-beans* node is meant for creating a set of form bean descriptors for this module. Every created element under this node is a JavaBean that implements the org.apache.struts.action.ActionForm class. Use the Properties editor on the right to adjust properties specific to every created form-bean.
- The *global-exceptions* node is intended for registering the handlers for the exceptions that might be thrown by an Action object.
- Use the *global-forwards* node to add, edit or delete a global forwards that represent ActionForward objects available to all Action objects as a return value.
- The *controller* node allows you to configure the controller properties.
- Under the *resources* node you can add, delete, or edit message resources.
- Under the *plug-ins* node you can define a Struts plug-in. Right-click the node, select *Create Plug-in* and specify the plug-ib *Id* and *ClassName* by pointing to the Java class which implements the org.apache.struts.action.PlugIn interface.

3.2.3. Source View

In the Source view, you have complete editing control of the underlying XML coding.

B) :	truts-config.xml 🕱 🗧	• O
	<pre>c?xml version="1.0" encoding="ISO-8859-1"?> c!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD St</pre>	=
(
Dia	gram Tree Source	

Figure 3.6. Source View

When working in Source view, you always have all the following features available:

- Content Assist
- Open On Selection
- File Folding

You	can	take	advantage	of	code	<u>assist</u>	[//jsf/html_single/
index.h	ntml#Code	AssistAnd	DynamicCodeAs	ssist42E	BasedOnPro	ojectData].	



Figure 3.7. Code Assist

The editor will also immediately flag any errors.



Figure 3.8. Errors in Source View

Finally, you can use the Outline view with the editor to easily navigate through the file.



Figure 3.9. Outline View

Find more information about editor features *in the editor features chapter* [../../jsf/html_single/ index.html#editors_features].

3.3. Graphical Editor for Tiles Files

Here, you'll know how to make use of the special graphical editor for Tiles configuration files.

The editor has three main views: Tree, Diagram and Source. The views can be selected via the tabs at the bottom of the editor. Any changes made in one view are immediately visible when you switch to any other view.

Before we consider each view of the editor, let's look at the way of creating new Tiles files.

3.3.1. Create New Tiles File

To create new Tiles files, right click any folder and select New > Tiles File.



Figure 3.10. Creating a New Tiles File

3.3.2. Tree View

The Tree view represents the different elements of the Tiles file that are organized into functional categories on the left-hand side and a form for editing the properties of currently selected items on the right-hand side.

5 tiles-defs.xml 🛛				- 0
Tiles Editor				
✓ tiles-defs	▼ Tiles Cor	fig Description	l.	
🗢 📓 tiles-defs.xml	Name:	tiles-defs		
	Encoding:			•
action	- Definitio	ns		
🔚 tiles.head	name	extends	path	<u>A</u> dd
tiles.action	tiles.layout	:	tiles/layout.jsp	Remove
tiles.root	tiles.head		tiles/head.jsp	Edit
tiles layout errors	tiles.action	1	tiles/action.jsp	<u></u>
	tiles.foot		tiles/foot.jsp	Цр
	tiles.error	errer tiles laveut	tiles/error.jsp	Down
	ules.layou	Lerror tiles.layout		
				_
iree Diagram Source				

Figure 3.11. Tree View

To edit the file, simply right click any node and select among the available actions.

🕒 tiles-defs.xml 🕴				- 0
Tiles Editor				
▼ tiles-defs		▼ Basic		4
▽ 🗟 tiles-defs.xml		Name:	tiles.layout	
Tiles.layout	🖹 Add Pu	ıt	· · · ·	_
action	👫 Add Pu	ıtList	tiles/layout.jsp	
🔚 tiles.head	Renam	ne	Browse	
🔚 tiles.action	[Сору	Ctrl + C	Change	H
tiles.foot	🚯 Paste	Ctrl + V		
🔚 tiles.layout.	X Delete	Delete		
	Proper	ties		
		Description:	<u>C</u> hange	
		Role:	•	
4	I	Small-Icon:		•
Tree Diagram Source				

Figure 3.12. Editing in Tiles Editor

3.3.3. Diagram View

The Diagram view allows you to create complex Tiles files in the form of a diagram.

🖪 til	les-defs.xml 🛱	- 8
	tiles.layout	
	tiles.error	
	tiles.foot	
	tiles.action	
	tiles.head	
Tree	Diagram Source	

Figure 3.13. Diagram Mode

To create new definitions, simply right click anywhere in the diagram.

🖪 ti	les-defs.xml ස		- 8
	tiles.layout	tiles.layout.errors	
	tiles.error		
	tiles.foot	New Definition	
		o∱ Cut Ctrl + X	
	tiles action	Copy Ctrl + C	
		Paste Ctrl + V	
	tiles.head	🗱 Delete 🛛 Delete	
		Preferences	
		Input Methods	
Tree	e Diagram Source		

Figure 3.14. Creating New Definition

You can also use the Diagram toolbar to make editing easier.



Figure 3.15. Diagram Toolbar

It contains four icons for changing the cursor state. The first one is the default cursor state for selecting existing nodes. The second icon is marquee selector. The third is used for creating new connections and the last one is for adding definition template to the content.

3.3.4. Source

The other view of the Tiles editor is the Source view that gives you full control over the source. Any changes here will immediately appear in other modes when you switch to them.

When working in Source view, you always have all following features available:

Content Assist

• Open On Selection

```
- 8
🖥 tiles-defs.xml 😫
  <?xml version="1.0"?>
  <!DOCTYPE tiles-definitions PUBLIC *-//Apache Software Foundation//DTD Tiles Conf
                                        "http://jakarta.apache.org/struts/dtds/tiles-c
  <tiles-definitions>
  <definition name="tiles.layout" path="tiles/layout.jsp">
   <put name="title"/>
   <put name="action"/>
   </definition>
   <definition name="tiles.head" path="tiles/head.jsp"/>
   <definition name="tiles.action" path="tiles/action.jsp"/>
   <definition name="tiles.foot" path="tiles/foot.jsp"/>
<definition name="tiles.error" path="tiles/error.jsp"/>
   <definition extends="tiles.layout" name="tiles.layout.errors"/>
  </tiles-definitions>
                                                                                         Þ
Tree Diagram Source
```

Figure 3.16. Source View

<u>Code</u> <u>assist</u> [../../jsf/html_single/ index.html#CodeAssistAndDynamicCodeAssist42BasedOnProjectData] is available in the Source mode.

tiles-defs.xml 🕴	- 8
<pre><?xml version="1.0"?> <!DOCTYPE tiles-definitions PUBLIC *-//Apache Softwa</th> <th><pre>pache.org/struts/dtds/tiles-c</pre></th></pre>	<pre>pache.org/struts/dtds/tiles-c</pre>
<> put	Element : description
<> putList	======== Info Elements
«» description	"description" element contains descriptive (paragraph length)
 display-name icon comment - xml comment XSL processing instruction - XSL processing instruction 	text about the surrounding element, suitable for use in GUI tools. operties: value, link, icon, tooltip. This properties are to be interpreted by the jsp page using them. By default the bean is of type "org.apache.struts.tiles.beans.SimpleMenuItem". This bean is useful to create list of beans used as menu items. value The bean 'value' property. link The bean 'link' property. icon The bean 'icon' property. tooltip The bean 'tooltip' property. classtype The fully qualified classname for this bean. If specified, the classname must be a subclass of the interface "org.apache.struts.tiles.beans.MenuItem". For compatibility with
<pre><definition name="tiles.foot" path="tiles/foot.jsp" <definition name="tiles.error" path="tiles/error.js <definition extends="tiles.layout" name="tiles.layo </tiles-definitions> </pre>	/> p*/> ut.errors*/>

Figure 3.17. Code Assist

Any errors are immediately reported as shown below:

😫 Packag 😫 🔝 Web Pro 📟		🖫 tiles-defs.xml 🕴	-	٥
C C C C C C C C C C C C C C C C C C C	~	<pre><?xml version="1.0"?> <!DOCTYPE tiles-definitions PUBLIC "-//Apache Software Foundation//DTD Tiles Conf</pre> </pre>	-	
🗢 🖶 MEB-INF	Π	<pre>"http://jakarta.apache.org/struts/dtds/tiles-c</pre>		
🕨 🗁 lib		<pre><tiles-definitions> <definition isn's)<="" lowent;="" normalities="" notheltiles(lowent="" pre=""></definition></tiles-definitions></pre>		-
📾 carDetail.jsp		<pre><pre><pre><pre>cres.cayouc pace cres/cayouc.jsp > <pre><pre><pre><pre>cput name="title"/></pre></pre></pre></pre></pre></pre></pre></pre>		
X faces-config.xml		<put name="action"></put>		
🔷 struts-bean.tid		<sometag></sometag>		
struts-config.xml	=	<definition name="tiles.head" path="tiles/head.jsp"></definition>		
struts-html.tid		<pre><definition name="tiles.action" path="tiles/action.jsp"></definition> <definition name="tiles.foot" path="tiles/foot.jsp"></definition></pre>		
struts-logic.tld		<pre><definition name="tiles.error" path="tiles/error.jsp"></definition></pre>		
tiles-defs.xml		<definition extends="tiles.layout" name="tiles.layout.errors"></definition>		
🗟 web.xml		<pre></pre>	=	-

Figure 3.18. Errors Reporting

You can also use the Outline view together with the editor's Source mode. It provides an easier navigation through the file.



Figure 3.19. Outline View

3.4. Graphical Editor for Struts Validation Files

Providing full support for development Struts applications JBoss Tools comes with a visual validation editor. To open the editor double-click on the validation file or if you don't have it create a new one.

To create a new validation file, right click any folder in Project Explorer and select *File > New > Other...* from the context menu and then *JBoss Tools Web > Struts > Validation File.*

E New	×
Select a wizard	-
Wizards:	
type filter text	
🗢 Boss Tools Web	A
🚳 CSS File	
🚵 HTML File	
🚿 JS File	
🔯 JSP File	
📸 Properties File	
🔆 TLD File	
💾 Web Descriptor	
🚵 XHTML File	
D 🗁 JSF	
🔊 Struts Config	
Struts Project	
🖄 Tiles File	
🛃 Validation File	
D 🧁 JPA	•
(2) < <u>Back</u> <u>Next</u> Enish	Cancel

Figure 3.20. Creating New Validation File

The validation editor works with five modes: Formsets, Validators, Constants and standard Tree and Source that you can easily switch over using tabs at the bottom of the editor.

The Formsets view shows forms and their elements on the left side and the dialogue for defining their validation rules on the right side.

MyValidation.xml 🛙				- 0
Formsets { 🖹 🇊 🛛 🕅 👔	Page Indexed List Prope	1		Edit
▽ 1 formset (default) Constants	Msg - Correspond	ed Message Template	resource	Add
 Ø AlertWizard Ø dashboardManagerForm 				Edit
▽ 🥔 dashboardVeiwForm ▽ 🍳 name	Arg - Replacemer	III It Value for Message T	ſemplate	Delete
1 ⁹ required	name	arg key	resource	Add
🧐 mask	required	arg0 Nmae	false	Edit
	4	Ш		Delete
	-Var - Validator Par	ameter var-value		Add
	mask	^[()A-Za-z0-9()_]+	[A-Za-z0-9']	Edit
				Delete
III	5			
Formsets Validators Constants T	ree Source			

Figure 3.21. Formsets View

The Constants view let you set constant values for your validation rules.

🗹 *validation.xml 🕴		- 8
Current Global Section glob	a 🗢 🛛 🖺 🎁	
constant-name	constant-value	∆dd
standartName	^[()A-Za-z0-9()_]+[A-Za-z0-9']	Edit
		Delete
Formsets Validators Constar	ts Tree Source	

Figure 3.22. Constansts View

The validation file can also be viewed in a Tree view.



Figure 3.23. Tree View

At any point you have full control over the source by switching to the Source view. Any editing in this view will immediately be available in other views of the editor.

•	*validation.xml 🖾	-	8
	<pre><?xml version="1.0" encoding="ISO-8859-1"?> <!DOCTYPE form-validation PUBLIC "-//Apache Software Foundation//DTD Commons Vali</th> <th>i</th><th>-</th></pre>	i	-
	<form-validation></form-validation>		
	<global></global>		
	<constant></constant>	=	
	<constant-name>standartName</constant-name>		
	<constant-value>^[()A-Za-z0-9()_]+[A-Za-z0-9']</constant-value>		_
	<formset></formset>		
	<form name="AlertWizard"></form>		
	<form name="dashboardManagerForm"></form>		
	<form name="dashboardVeiwForm"></form>		
	<field depends="required,mask" property="Name"></field>		
	<arg key="name" name="required" position="arg0"></arg>		
	<var></var>		
	<var-name>mask</var-name>		
	<var-value>^[()A-Za-z0-9()_]+[A-Za-z0-9']</var-value>		
	<th></th> <th></th>		
	<th></th> <th></th>		
	<pre></pre>	-	
	Ster All example formset	Ē	
	Armente Validatore Constante Teo Source)	-
FI	ormsets validators Constants life Source		

Figure 3.24. Source View

You can also open your own custom or Struts-standard *validation-rules.xml* file.

The Validators view shows the validation rules for a selected validator. You can of course add your own rules.

MyValidation.xml 🛙		- 8
Current Global Section globa 💠	🗅 👔	
Validators	Depends	Change
name	Message Key errors.timeornumber	Change
long	Java Class Name com.echopass.provisioning.rra.validator.TimeOrNumbe	Change
double	Method validateTimeOrNumber	Change
date	Method Param java.lang.Object, org.apache.commons.\	Change
intRange	JavaScript	
creditCard	Function Name	Change
email	Function Body	
url time_number time_numberforthresholds	var timeMask = "^([0-9]){1,3}(:[0-5][0-9]){0,2}\$"; var numberMask = "^[0-9]+.?[0-9]?\$"; var booleanMask = "^active inactive\$"; var anyMask = "^.*\$"; var percentMask = "^[0-9]+.?[0-9]?%?\$";;	
Formsate Validatore Constants Th	var statistic = new Array();	v
valuators constants in	Jource	

Figure 3.25. Validators View

Here are the validation rules shown in the Source mode.

MyValidation.xml 🕱	- 0
msg=" "	
<pre>jsFunction="org.apache.commons.validator.javascript.validateUtilities"/></pre>	
<validator <="" name="time_number" td=""><td></td></validator>	
<pre>classname="com.echopass.provisioning.rra.validator.TimeOrNumber"</pre>	
method="validateTimeOrNumber"	
methodParams="java.lang.Object,	
org.apache.commons.validator.ValidatorAction,	H
org.apache.commons.validator.Field,	
org.apache.struts.action.ActionMessages,	
org.apache.commons.validator.Validator,	
javax.servlet.http.HttpServletRequest"	
depends="	
msg=errors.tlmeornumber">	
var timemask = "([0,9])[1,3]([0,5]](0,9])[0,2]	
var hundermask = [0·3];*; j	
var bovtedningsk = "^ *C".	
var percentMask = *^[0-9]+ ?[0-9]?%?\$"	
var statistic = new Arrav():	
statistic("*1=anvMask:	•
	•
Formsets Validators Constants Tree Source	

Figure 3.26. Validation Rules

Modules

JBoss Tools support working with Struts projects that have multiple modules. You can easily do the following:

- Add new modules
- Edit modules for an existing project or during Struts project import

Now, let's discuss this functionality in more detail.

4.1. When Importing a Struts Project

During Struts project import, if the project has multiple modules, you will see a screen with all existing modules. You can select each module and edit its details.

•	Import Struts Project	×
Project Mode	lles	
Configure Proje	ect Modules	
Name	URI	
<default></default>	/WEB-INF/struts-config.xml	
Name	<default></default>	
URI	/WEB-INF/struts-config.xml	
Path on Disk	/opt/work/jbds_GA/workspace/Struts/WebContent/WEB-INF/struts-config.xml	<u>C</u> hange
Sources Path	/opt/work/jbds_GA/workspace/Struts/JavaSource	<u>C</u> hange
Web Root	/opt/work/jbds_GA/workspace/Struts/WebContent	<u>C</u> hange
0	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

Figure 4.1. Configuring Project Modules

4.2. Editing Modules in an Existing Project

To edit modules in an existing project, right click the project and select *JBoss Tools > Modules Configuration.*

Validate		
Add Spring Project Nature		
T <u>e</u> am	•	
Comp <u>a</u> re With	,	
Restore from Local History		
JBoss Tools		S Add Struts Capabilities
PDE Tools	•	1 Add JSF Capabilities
Java EE	,	Remove Struts Capabilities
Properties	Alt+Enter	Add Custom Capabilities
		Remove JSF Capabilities
		Verify
		Modules Configuration

Figure 4.2. Choosing Modules Configuration

You will see the same screen as above where you will be able to select a module and edit its details.

\$ (Modules Configuration	Þ
Struts Projec	t	
Name	URI De	eleted Add
<default></default>	/WEB-INF/struts-config.xml	Delete
Name	<default></default>	
URI	/WEB-INF/struts-config.xml	
Path on Disk	A/workspace/Struts/WebContent/WEB-INF/struts-cont	fig.xml <u>C</u> hange
Sources Path	/opt/work/jbds_GA/workspace/Struts/JavaSource	<u>C</u> hange
Web Root	/opt/work/jbds_GA/workspace/Struts/WebContent	<u>C</u> hange
	Fini	ish Cancel

Figure 4.3. Modules Configuration

4.3. Adding New Modules

Adding a new module is very simple. First switch to Web Project view. Expand your project to the Configuration folder. Under that folder you should see the current modules. Right click on Configuration and select *New > Struts Config.*



Figure 4.4. Adding New Modules

You will see the screen below. You can specify a new module name and also add the new Struts configuration file to web.xml file.

	Struts Config 🛛 🗙
Struts Co	onfig 🛞
Folder:*	/Struts/WebContent/WEB-INF Browse
Name:*	struts-config-mod1
Version:	1.1 \$
Module:	/mod1
	Register in web.xml
	Finish Cancel

Figure 4.5. Adding New Modules

Code Generation

JBoss Tools comes with a code generation feature. You can generate stub code for Struts Actions, FormBeans, Forwards and Exceptions.

The code generation that JBoss tooling provides is based on Velocity templates which can be modified for your use. The templates are located at {*JBossStudioHome*} > *templates* > *codegeneration.*

There are a number of ways to invoke code generation. One is simply right-clicking the Struts diagram and selecting *Generate Java Code....*



Figure 5.1. Selecting Generate Java Code

On this screen you can select for which elements to generate code. If you select Next you will be able to specify more options for each of the categories you selected.

¢	Generate - Step 1
Java Code Ste	p 1
	Generate classes for Actions
	Generate classes for FormBeans
	Generate classes for Forwards
	Generate classes for Exceptions
	Overwrite existing files
Base Package:	
	Next >> Generate Cancel

Figure 5.2. Generate - Step 1

Tip:
Please be careful not to override your existing files.

When generation is complete, a result window will appear letting you know how many classes were generated:

ê	Generate - Finish	×
Java Code Step 7 Generation finished.		0
Message:		
Generated classes: 1 Form beans: 1		
4	III	Þ
		Finish

Figure 5.3. Generation Finished

You don't always have to generate code for all elements at once. You can invoke generation for just an individual Struts artifact as well. Right-click an element on the diagram of the Struts configuration file and select *Generate Java Code...* from the context menu.

GetNan	eFor
sayhello	Open Declaration
titi	Open Form-bean Source
	Create Exception
	💑 Create Forward
	📸 Create Property
	Create Comment
	Set as unknown
	Select Referring Item
	🍧 Generate Java Code

Figure 5.4. Generation For Individual Struts Artifact

🕏 struts-config.xml 🛙		- C
Struts Config Editor		
	+ Propertie	es Editor
🗢 🛸 struts-config.xml	struts-con	fig 1.1
🛵 data 🛛 Rename		struts-config
👂 🥪 form 🛛 Add	• on:	xml
🔯 glot	va Code	
Globarti Properties	ent:	Change
<pre>controller</pre>	Encoding	
a resources	cricoung.	
P 🧠 plug-ins		
Diagram Tree Source		

The same can be done from within the Tree viewer for the editor of the Struts configuration file.

Figure 5.5. Generation in Struts Config Editor

Struts Configuration File Debugger

JBoss Tools come with Struts configuration file debugger. It allows you to set break points on Struts diagram and then simply launch the server in debug mode.

🗟 struts-config.xml 🛿			- 0
	pages/inputname.jsp		
i getName	Open Page Rename Page		
	Create Link Create Comme Show/Hide Link Select Referring	ent s g Item	
	Run on Server		
	Copy Cut	Ctrl + C Ctrl + X Ctrl + V	/pages/greeting.jsp
	X Delete	Delete	
	Add Breakpoint		
	Input Methods	•	
Diagram Tree Source			<u>b-l-i-d-l-i-d-l-i-d-l-i-d-</u>

Simply right click an Action or a page and select Add Breakpoint.

Figure 6.1. Adding Breakpoint

Customizable Page Links Recognizer

Custom page links allow you to define custom Struts page links that will be recognizable in the Struts application diagram. You can define these links by selecting *Window > Preferences* from the menu bar and then selecting *JBoss Tools > Web > Struts > Customization* from the Preferences dialog box.

Preferences 🗙						
type filter text		Customizatio	n			⇔ ∙ ⇔≁
General	Â	Link Recognize	er			
Ant		Tag	Attribute	Refer to	Link Type	Add
Connectivity		html:link	action	action	Struts	
FreeMarker Editor		html:link	page	page	Struts	Edit
Help		html:link	forward	forward	Struts	Delete
HQL editor		html:frame	action	action	Struts	
Install/Update		html:frame	page	page	Struts	
Internet	_	html:frame	forward	forward	Struts	
⊅ Java	-	html:form	action	action	Struts	
JBoss jBPM		logic:forward	name	forward	Struts	
		logic:redirect	forward	forward	Struts	
JBoss Servers		logicileaneer	. or france	ronnara	5000	
Packaging Archives						
⊽ Web						
Editors						
▷ JSF	Ц					
Þ Seam						
Automation						
Customization						
Project						
Struts Pages						
Verification						
XDoclet		<u> </u>		Deate	n Defeutte	
				Resto	re <u>D</u> efaults	Арріу
٢					ОК	Cancel

Figure 7.1. Customization Panel

Struts Project Verification

In this section we'll consider one more functionality that JBoss Tools provide for Struts projects, namely adjusting projects verification.

To configure Struts project verification select *Window > Preferences* from the menu bar, select *JBoss Tools > Web > Verification* from the Preferences dialog box and then expand the Struts Rules node.



Figure 8.1. Struts Rules

Suppose you are working in the Source viewer for a Struts configuration file as shown below:



Figure 8.2. Struts Configuration File

While typing a class name or entering it from the graphical editor, you might make a minor typo (like *"sample.GreetingAction1"* instead of *"sample.GreetingAction"*). After saving the file, verification checks to make sure everything is correct and finds the error below:



Figure 8.3. Error Reporting

Notice that the Package Explorer View shows a marked folder and a marked file where the error is.

You can place the cursor over the line with the error to view a detailed error message:



Figure 8.4. Error Message

The verification also checks to make sure you have specified the correct JSP page for the forward:



Figure 8.5. JSP Page Verification

Once you place the cursor over the line, you can see the error message:

	<pre><action name="GetNameForm" path="/greeting" scope="request" type="sample.GreetingAction</pre></pre>				
ß	Forward sayHello of Action /greeting path attribute reference to non-existent page				

Figure 8.6. Error Message

You can always invoke the verification by switching to the Diagram viewer, right-clicking and selecting *Verify* from the context menu:

🕏 *struts-config.xml 🛿		- 0
Image: space of the space	/page s/inputname.jsp	
	Add Create Comment Paste Ctrl + V Select Element Auto-Layout	vages/greeting.jsp
	Properties	
Diagram Tree Source	Generate Java Code Input Methods	

Figure 8.7. Verify Command

Relevant Resources Links

Find out necessary information on <u>Struts technology</u> [http://struts.apache.org/] if you don't know enough.

In summary, this reference should help you to get familiar with those parts of JBoss Tools which are meant for development with Struts technology. If you've carefully gone through the document, you should know now how to create/import Struts project or enable Struts capabilities for an existing web project as well as organize and edit all necessary staff for your Struts application using a set of views and editors provided by JBoss Tools.

If you'd like to contribute your feedback is always appreciated. You can leave your questions and suggestions on our *Forum* [http://www.jboss.com/index.html?module=bb&op=viewforum&f=201].

We hope, this guide helped you to get started with the JBoss Struts 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].