


# Struts 2 Plugin Features & User Guide

## Introduction

Struts 2 Plugin for JetBrains IntelliJ IDEA provides tight and feature-rich integration to develop [Struts 2](#) applications. Real productivity features (including refactoring support), smart assistance just everywhere as well as being a tool understanding the structure and framework will make you more productive instantly. No more countless wizards just generating some XML or code without much intelligence!


 **Screencasts**  
To get a quick overview watch these screencasts to see Struts 2 support "in action"

[Navigation and Autocompletion](#): demonstrates basic navigation and autocompletion features

[Spring integration](#): setting up Spring integration for your Struts 2 project

★ [Validator integration](#): new features in IDEA 11

more to follow soon...



 **IDEA versions**  
Some features may not be available in all versions.  
IDEA 8+ means "IDEA 8 or any higher version"


Please see [Struts2PluginChangeNotes](#) for detailed information.

 **Getting started**  
"Hello World" with [Struts 2 and IntelliJ IDEA 8](#) -  
Detailed introductory blog post by Tabrez Iqbal

## About this page

All (main) features are grouped and listed below. Please check the following table to understand the status indicators.

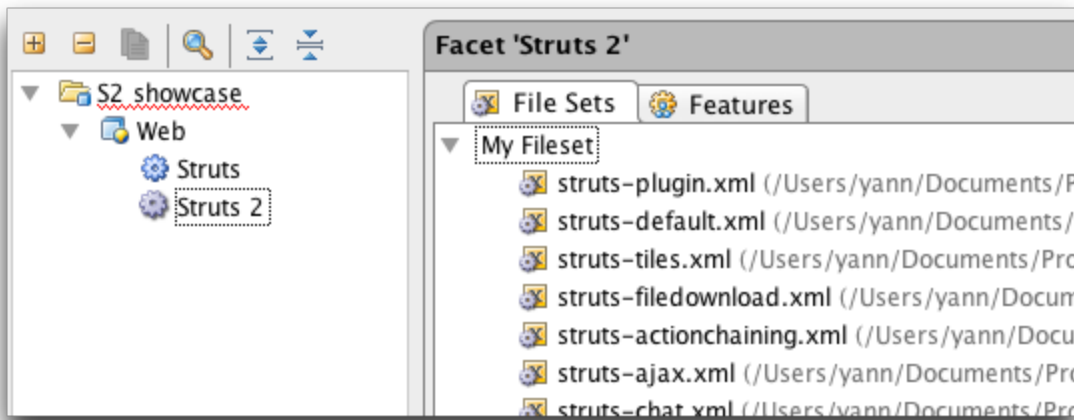
Symbol	Status
	planned
	in progress/mostly done
	done
	future planning

 **Screenshots**  
Starting with IntelliJ IDEA 12.1 icons have changed slightly.

## Struts 2 facet

All plugin configuration settings are bundled in a dedicated facet. Simply attach a new Struts 2-facet to an existing Web-facet, create a fileset and add your `struts.xml`-files from all auto-scanned configuration files in your project and the plugin is ready to work.

 **Fileset configuration**  
`struts-default.xml` from `struts2-core.jar` as well as any `struts-plugin.xml` from plugins (e.g. Spring integration) have to be added additionally to the fileset.  
Also make sure to add all files used via `<include>`.  
10.0.2+: projects created using project wizard will include all `struts.xml`-files from Struts-JARs automatically  
 Many features will NOT work if this setup has not been performed properly 



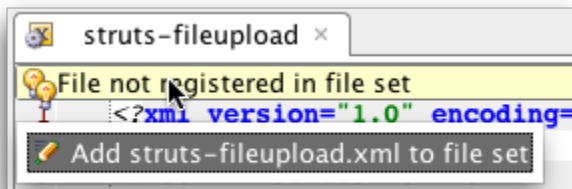
✔ Dedicated facet with configuration options

✔ File Set manager

Group and manage your `struts.xml`-files with an easy to use editor, providing maximum flexibility w/r to your project setup.

✔ Detect `struts.xml` files not mapped to file sets

Spot setup errors easily, additionally providing a quickfix to add the unmapped `struts.xml`-files to any existing fileset.



✔ Library validator/setup

Creates IDEA project/module-library and - if required - downloads all necessary JAR files required for Struts 2.

✖ Library validator/setup for standard plugins (Spring, Tiles, JSF, ..)

✔ Autodetection of Struts 2 facets

Existing `struts.xml`-files are recognized upon opening your project and a new S2 facet will be created automatically.

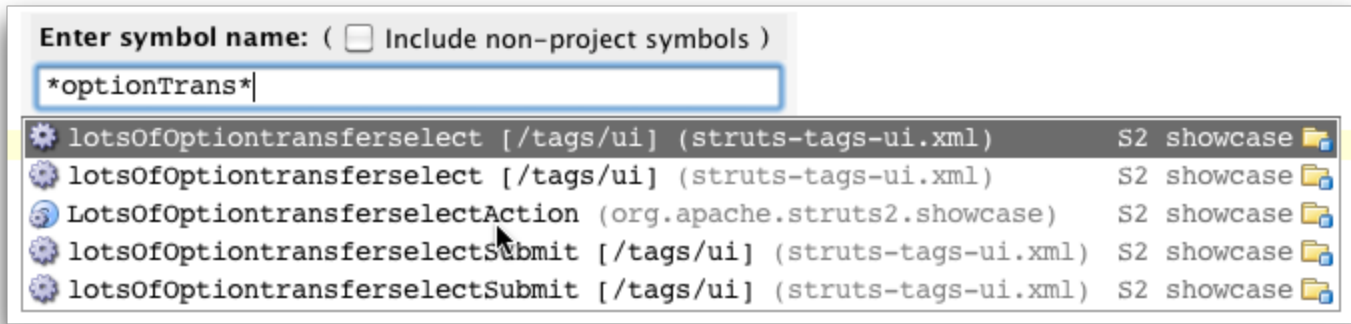
✔ Integration with "New Project"

Add a blank ready-to-start Struts 2 application template by selecting Struts 2 from the list of available technologies. A S2-facet with default fileset and an empty `struts.xml`-file already mapped in `web.xml` will be created automatically, all necessary libraries will be setup as well.

## IDE integration

✔ "Go To Action/Package" by name

Press `Ctrl+Shift+Alt+N` (Go To -> Symbol) to quickly navigate to any Action/Package by simply entering its name (including wildcards):



✔ "Navigate -> Related File"

Tip: just start typing to narrow down the result list of related items and press ENTER to navigate.

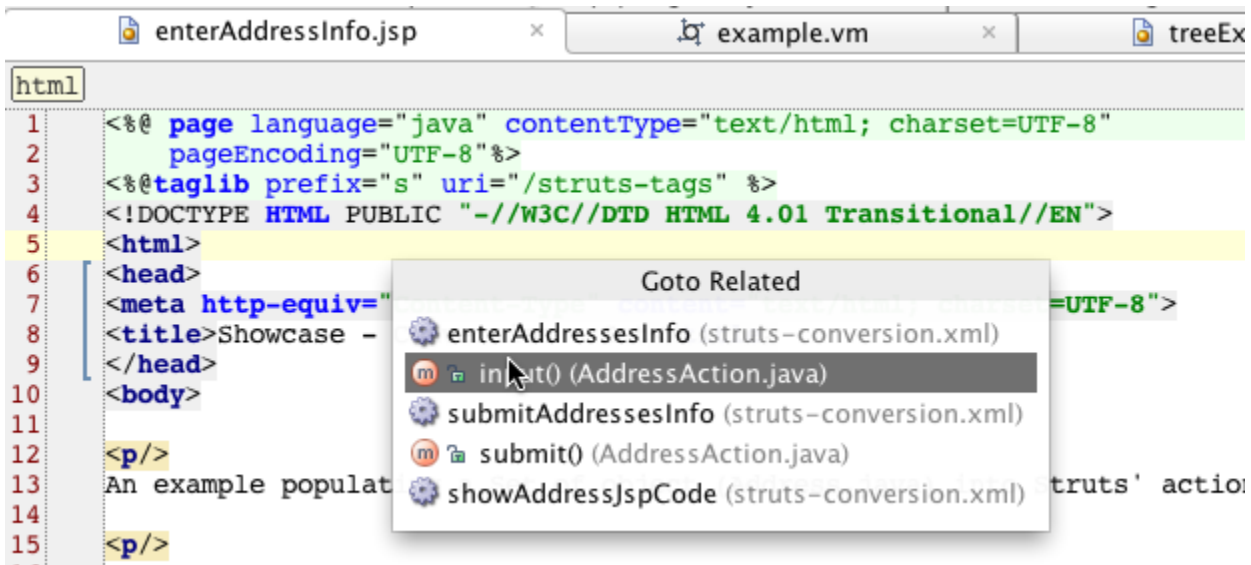
IDEA 10.5+

Use this shortcut to quickly navigate from any web-resource used in <result> to its <action>-declaration and/or Action-method.

IDEA 11+

Also vice versa from Action-class to <action>-declaration and any referenced result(s) files.

Navigate from Action-class to all corresponding validation.xml files.



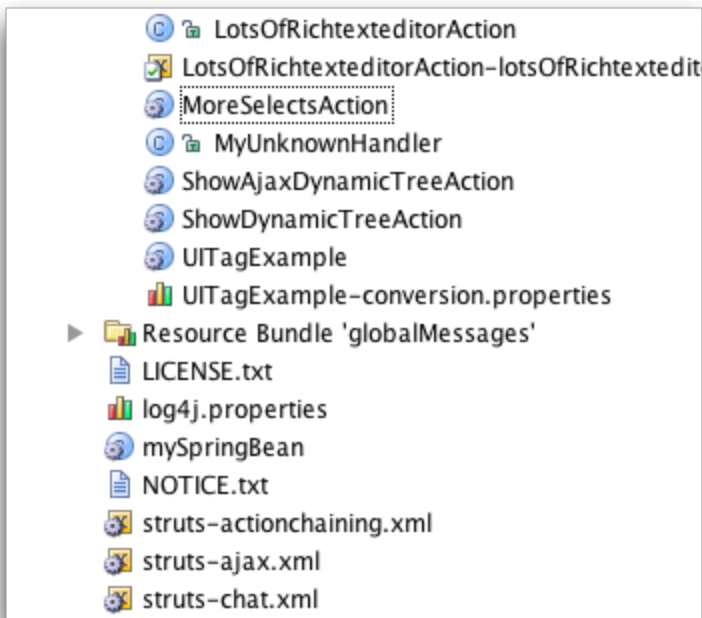
✔ Dedicated dictionary

IDEA 11+

A bundled dictionary providing Struts-specific words and acronyms integrates with IDEA's builtin spellchecker.

✔ Dedicated icons Struts elements

Recognize Struts related configuration files or code at a glance throughout the IDE (Project/Packages View, Editor tabs, Autocompletion, ...):



## struts.xml/validation.xml

### ✓ Autocompletion

Please see this [screencast](#) to see it "in action" (a small part only!).

Autocompletion (CTRL+Space) is available for virtually all tags and attributes, just give it a try:

```
<interceptors>
  <interceptor-stack name="jsfFullStack">
    <interceptor-ref name="params" />
    <interceptor-ref name="basicStack" />
  </interceptor-stack>
</interceptors>

<default-interceptor-ref name="basicStack" />

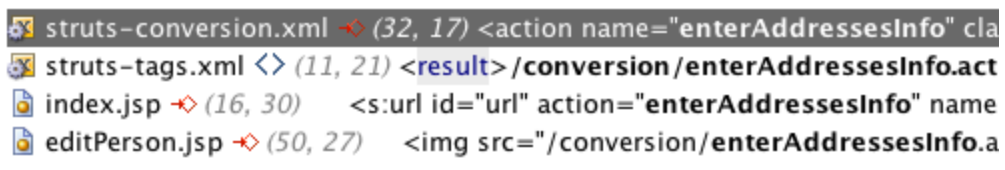
</package>

<package name="jsf.employ">
  <action name="list" class="MyAction">
    <result name="success" type="redirect" />
  </action>
</package>
```

### ✓ Rename refactoring/Find Usages

Need to refactor your S2 application? No more manual search and replace - just use Rename refactoring (Shift+F6) or perform Find Usages (Alt+F7/Ctrl+Alt+F7):

```
<!-- Example populating a Set inside an Action -->
<action name="enterAddressesInfo" class="org.apache.struts2.showcase.conversion.Ad
  <result>enterAddre
</action>
<action name="submit
  <result>showAddres
  <result name="inpu
</action>
<action name="showAddressjspcode >
  <result type="plainText">/conversion/enterAddressInfo.jsp</result>
```



✔ Navigation to referenced elements

Simply `Ctrl-click` (or press `Ctrl+B`) on any element to navigate to its declaration - or take a peek without leaving the current editor using Quick Definition Lookup (`Ctrl+Shift+I`).

✔ Highlighting of configuration/reference errors

Errors are highlighted immediately in the editor, additional validation is triggered for Make Project.

IDEA 7: see "Validation" tab in S2 facet for more options

IDEA 8+: see Settings->Compiler->Validation

+ Quickfixes

Various quickfixes to create missing elements etc.

✔ struts.xml: restrict resolving to `<result>` value depending on result-type (e.g. Tiles, Velocity)

IDEA 10+

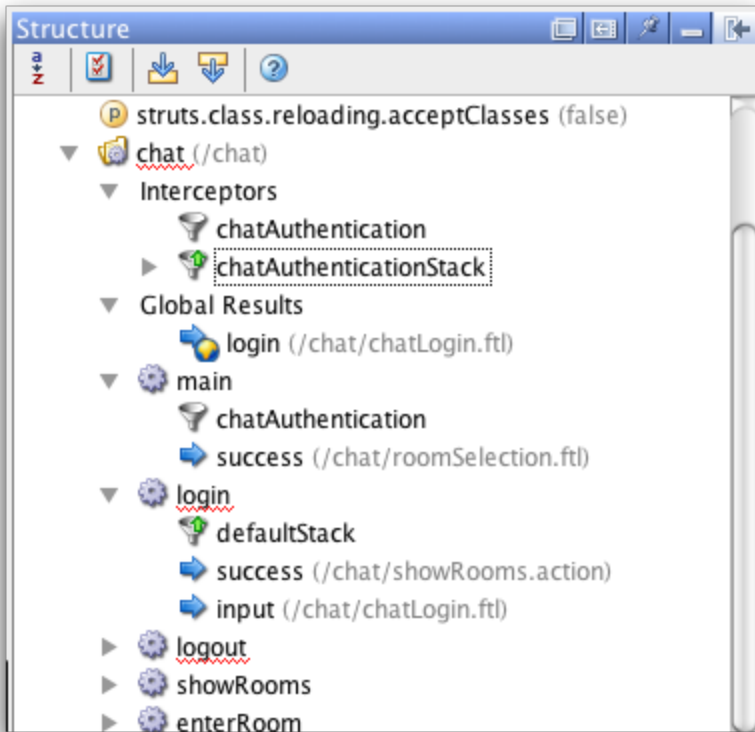
Completion will only show resources relevant for the given (or inherited) result type.

+ Configuration errors

Highlight common configuration errors or possible problems on-the-fly - while you're editing. Catch the bugs before time-consuming deploy & manual-test.

✔ Structure View

Visually groups all relevant info with Autoscroll from/to source. Hint: searching for something in the tree? Just start typing its name to find it.



✔ struts.xml: Pageflow graph

IDEA 8+

Graph of all actions and results makes it easy to understand the structure of your application.



## validation.xml

Please see this screencast to see it in action: [Validator integration](#)

- ✓ Navigation to/from corresponding Action-class

IDEA 11+

Autocompletion/Navigation to Action-class properties in `<field>` "name"  
"Navigate -> Related File" from Action-class to corresponding `validation.xml` file(s).

- ✗ Show `validation.xml` node(s) under corresponding Action-class (like GUI-Designer forms)

- ✓ Create corresponding `validation.xml`-template from Action

IDEA 11+

Intention to create new `ActionClassName-validation.xml` from file template.

- ✓ Structure View

- ✓ I18N-support

IDEA 11+

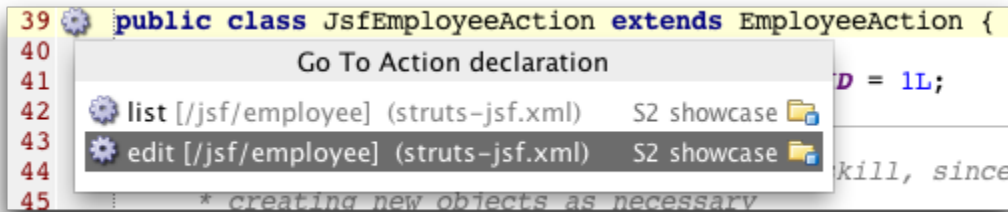
Support for `.properties`-based I18N for `<message>` "key" (can be disabled in S2 facet, tab "Features").

## Actions (JAVA/Groovy)

- ✓ Navigation to Action-declaration in `struts.xml`

Simply click on the action icon in the left editor bar to quickly open up the corresponding mapping(s) in `struts.xml`:

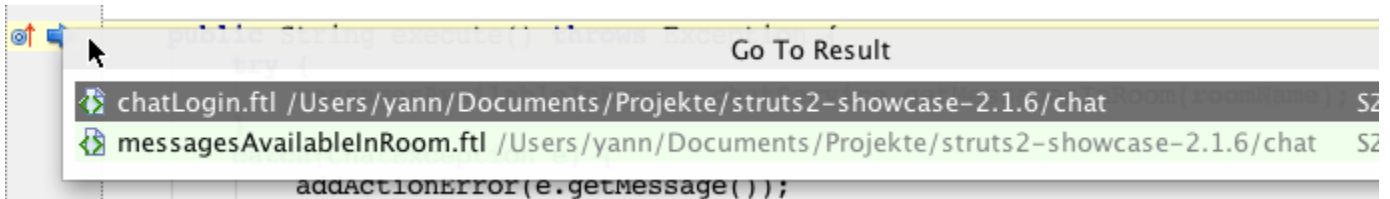
IDEA 8.1.1+: Groovy support added



- ✓ Navigate to result

IDEA 10.5+

Use the gutter icon (or use Navigate->Related File) to directly navigate to any result(s) from your action method:



## JSP

- ✓ Navigation to Action-declaration/class

Click on the Action-class icon to quickly open the corresponding Action-class or Ctrl+click on the "action"-attribute open its `struts.xml`-definition:

```
17
18
19 Action "standardTags" [struts-freemarker.xml]
20 <s:url var="url" action="standardTags" namespace="/freemarker" />
21
```

✔ Autocompletion/navigation for form properties

IDEA 11+

Form properties will be autocompleted/resolved in "surrounding" Action class. Ofcourse, rename refactoring and finding usages is supported here as well - no more manual search and replace necessary when updating your forms!

```
<s:action var="skillAction" namespace="/skill" name="edit">
  <s:textfield name="skillName"/>
</s:action>
```

actionErrors	Collection<String>
actionMessages	Collection<String>
currentSkill	Skill
fieldErrors	Map<String, List<String>>
skillDao	SkillDao
skillName	String
toDelete	String[]

+ Autocompletion/navigation for other tags

Autocompletion is available for (almost) all attributes with static values.

```
<s:url var="url"
  action="standardTags"
  namespace="/freemarker"
  includeParams="|"/>
```

all
get
none

Did you know th

✔ JavaScript support

(plugin "JavaScript Support" must be enabled)

Enjoy the power of IDEA's powerful JavaScript-support within all S2 UI-tags.

```
<s:form action="save" function alert(sMessage)
  onkeydown="alert('Full JavaScript support!'); return false;">
```

✔ CSS support

(IDEA 8+: plugin "CSS Support" must be enabled)

Define inline-styles (`cssStyle`) or reference existing CSS classes (`cssClass`).



```
<s:textfield label="%{getText('skill.name')}}"
            name="currentSkill.name"
            cssStyle="color: ; text-align: right; width: 250px;"/>
```

✔ I18N-support

IDEA 10+

Support for .properties-based I18N for <s:text> (can be disabled in S2 facet, tab "Features").

```
<?if test="currentEmployee!=null">
  <s:text var="title" name="item.edit"><s:param><s:text name="employee"/></s:param>
</s:if>
<s:else>
  <s:text var="title" name="item.edit"><s:param><s:text name="employee"/></s:param>
</s:else>
<html>
<head>
  <title><s:property value="title"/>
  <s:head/>
  <sx:head/>
</head>
<body>
```

+ OGNL (Object Graph Navigation Library) Support

IDEA 11+

Provides editing, highlighting and autocompletion support for OGNL expressions - not just for Struts 2 applications.

Please see Struts 2 Plugin OGNL Support for details.

```
<s:textfield disabled="%{ id == 5 ? true : }"/>
```

## Plugins/3rd Party Extensions

Make sure to include struts-plugin.xml from corresponding plugin JAR to your Struts 2 fileset (configured in S2 facet in module settings).

✔ Spring plugin

(plugin "Spring Support" must be enabled and Spring facet setup properly)

Please see this [screencast](#) to see Spring integration "in action".

Reference Spring beans where supported (e.g. in <action> "class", <constant> "value").

```

<action name="login" class="chatLoginAction">
  <interceptor-ref name="chainingInterceptor" />
  <result type="redirect" />
  <result name="input" type="chatLogoutAction" />
</action>

```

+ Convention plugin

IDEA 9+

Sophisticated configuration properties support

Autocompletion/navigation/usage search in annotations (WIP):

```

@ParentPackage("myPackage")
public class MyClass {
}

```

✓ Tiles 2 plugin

Blog post

(plugin "Struts 1.x" must be enabled)

IDEA 8.1.1+

struts2-tiles-plugin.jar must be present as well as Tiles libraries. Configured tiles.xml-files will be set automatically as described in <http://struts.apache.org/2.x/docs/tiles-plugin.html>.

```

<action name="test">
  <result name="success" type="tiles">showcase.index</result>
</action>

```

✓ Freemarker

(plugin "FreeMarker support" must be enabled)

IDEA 8+

All supported S2 taglibs are automatically available:

```

<div valign="top">
  <p align="right">
    <@s.url id="url" value="/hangman/images/choose.png" />
    <@s.accesskey, cssClass, cssErrorClass, cssErrorStyle, cssErrorText />
  </p>
  <@s.action(executeResult, flush, id, ignoreContextParams, ... />
  <@s.actionerror(accesskey, cssClass, cssErrorClass, cssErrorText />
  <@s.actionmessage(accesskey, cssClass, cssErrorClass, cssErrorText />
  <@s.append(id, var) />
  <@s.bean(id, name, var) />
  <@s.checkbox(accesskey, cssClass, cssErrorClass, cssErrorText />
  <@s.checkboxlist(accesskey, cssClass, cssErrorClass, cssErrorText />
  <@s.combobox(accesskey, cssClass, cssErrorClass, cssErrorText />
  <@s.component(accesskey, cssClass, cssErrorClass, cssErrorText />
  <@s.date(format, id, name, nice, var) />

```

Global variables as well as properties provided from corresponding Action-class will be resolved:

```
<#if hangman.characterGuessedBefore(currentCharacter)>
  <@s.s.p charactersAvailable List<Character>
  <@s.u.p vocab Vocab
  <img.p class class java.lang.Class<?>
  characterGuessedBefore(Character character) boolean
<#else>
  equals(Object) boolean
  <@s.u gameEnded() java.lang.Boolean
  <img getCharactersAvailable() java.util.List<java.lang.Character>
  getClass() java.lang.Class<?>
</#if>
s.iterator
  guess(Character character) void
```

+ Velocity

(plugin "Velocity support" must be enabled)  
IDEA 10.x+  
Add all global macros (struts.vm in struts2-core.jar) automatically.  
Autocomplete all global variables.  
[Blog post with screen-shots](#)

+ struts2-jQuery plugin

IDEA 9+  
Provides autocomplete, navigation for JSP tags.  
[Blog post showing off some features](#)  
[struts-jQuery plugin](#)