

Exception Configuration

Exception mappings is a powerful feature for dealing with an Action class that throws an Exception. The core idea is that an Exception thrown during the Action method can be automatically caught and mapped to a predefined Result. This declarative strategy is especially useful for frameworks, like Hibernate and Acegi, that throw RuntimeExceptions.

As with many other parts of the framework, an Interceptor is needed to activate the exception mapping functionality. Below is a snippet from `struts-default.xml` which has the exception mapping already activated.

snippet of struts-default.xml

```
...
<interceptors>
  ...
  <interceptor name="exception" class="com.opensymphony.xwork.interceptor.ExceptionMappingInterceptor"/>
  ...
</interceptors>

<interceptor-stack name="defaultStack">
  <interceptor-ref name="exception"/>
  <interceptor-ref name="alias"/>
  <interceptor-ref name="servlet-config"/>
  <interceptor-ref name="prepare"/>
  <interceptor-ref name="i18n"/>
  <interceptor-ref name="chain"/>
  <interceptor-ref name="debugging"/>
  <interceptor-ref name="profiling"/>
  <interceptor-ref name="scoped-model-driven"/>
  <interceptor-ref name="model-driven"/>
  <interceptor-ref name="fileUpload"/>
  <interceptor-ref name="checkbox"/>
  <interceptor-ref name="static-params"/>
  <interceptor-ref name="params"/>
  <interceptor-ref name="conversionError"/>
  <interceptor-ref name="validation">
    <param name="excludeMethods">input,back,cancel,browse</param>
  </interceptor-ref>
  <interceptor-ref name="workflow">
    <param name="excludeMethods">input,back,cancel,browse</param>
  </interceptor-ref>
</interceptor-stack>
...
```

To use exception mapping, we simply need to map Exceptions to specific Results. The framework provides two ways to declare an exception mapping `<exception-mapping/>` - globally or for a specific action mapping. The exception mapping element takes two attributes, `exception` and `result`.

When declaring an exception mapping, the Interceptor will find the closest class inheritance match between the Exception thrown and the Exception declared. The Interceptor will examine all declared mappings applicable to the action mapping, first local and then global mappings. If a match is found, the Result is processed, just as if it had been returned by the Action.

i This process follows the same rules as a Result returned from an Action. It first looks for the Result in the local action mapping, and if not found, it looks for a global Result.

Below is an example of global and local exception mappings.

snippet from struts.xml

```
<struts>
  <package name="default">
    ...
    <global-results>
      <result name="login" type="redirect">/Login.action</result>
      <result name="Exception">/Exception.jsp</result>
    </global-results>

    <global-exception-mappings>
      <exception-mapping exception="java.sql.SQLException" result="SQLException"/>
      <exception-mapping exception="java.lang.Exception" result="Exception"/>
    </global-exception-mappings>
    ...
    <action name="DataAccess" class="com.company.DataAccess">
      <exception-mapping exception="com.company.SecurityException" result="login"/>
      <result name="SQLException" type="chain">SQLExceptionAction</result>
      <result>/DataAccess.jsp</result>
    </action>
    ...
  </package>
</struts>
```

In the example above, here is what happens based upon each Exception:

- A `java.sql.SQLException` will chain to the `SQLExceptionAction` (action mapping not shown)
- A `com.company.SecurityException` will redirect to `Login.action`
- Any other exception that extends `java.lang.Exception` will return the `/Exception.jsp` page

Exception Values on the ValueStack

By default, the `ExceptionMappingInterceptor` adds the following values to the Value Stack:

exception	The exception object itself
exceptionStack	The value from the stack trace

Sample JSP using Error and Exception Values

```
<h2>An unexpected error has occurred</h2>
<p>
  Please report this error to your system administrator
  or appropriate technical support personnel.
  Thank you for your cooperation.
</p>
<hr/>
<h3>Error Message</h3>
<s:actionerror/>
<p>
  <s:property value="%{exception.message}"/>
</p>
<hr/>
<h3>Technical Details</h3>
<p>
  <s:property value="%{exceptionStack}"/>
</p>
```

Exception in constructors

Global exception mappings are designed to be used with exceptions thrown by action methods (like `execute`). exceptions thrown from constructors will **not** be handled by global exception mappings.