

Configuration

by Don Brown

Advanced configuration

Table of contents

1 Overview.....	2
2 ActionForm Configuration.....	2
3 XML Model Definition.....	2
3.1 The XML Models File.....	2
3.2 XMLForm.org XForm File.....	3
4 Validation.....	5
5 Apache Axis Configuration.....	5

1. Overview

To take full advantage of XML forms, there are three areas that need to be configured: ActionForm configuration, XML model definitions, and validation. In several cases, the configuration is different depending whether the XML forms will be used with XMLForm.org's XForms implementation or not. Finally, the same XML form can be exposed as SOAP-based web service using Axis.

2. ActionForm Configuration

To use an XML form as a Struts ActionForm, define it as a form bean using the class of the XML form in `struts-config.xml`. This is an example of a form that uses the DOM XML form:

```
<form-bean name="userForm" type="com.oroard.stxx.xform.DOMForm" />
```

If XMLForm.org's XForms implementation will be used to display the form, the location of the XForm needs to be defined. Since Struts doesn't support ActionForm parameters, the dynaform configuration elements are used. This is an example of an XML Form that uses XMLForm's XForms implementation:

```
<form-bean name="userForm"
  type="com.oroard.stxx.xform.DOMForm">
  <form-property name="xml" type="userForm.xml" />
</form-bean>
```

3. XML Model Definition

The XML the form will populate must be defined before the XML form can be used either by a file or set in a Struts Action. Therefore, a form will populate the XML's existing attributes and text nodes. The easiest way to define the XML form's XML model is to let stxx load it either from one file that holds all XML models or, if XMLForm's XForms implementation is used, in each XForm XML.

3.1. The XML Models File

To define all XML models in one file, the following property must be set in `struts-config.xml`:

Name	Description	Default
xmlform-models	Location of the XML file	If not specified, no XML form

	containing all XML models.	models will be pre-loaded.
--	----------------------------	----------------------------

Table 1: XML models file property in struts-config.xml

The XML models file can contain one or more XML form models. This is an example of the XML models file that defines the model for the XML form named "userForm":

```
<document>
  <model name="userForm">
    <user>
      <names firstname="jim" lastname="" displayname="Jim bob jr." />
      <internet>
        <email />
        <email1 />
      </internet>
      <phone>
        <workphone />
        <homephone />
        <fax />
      </phone>
    </user>
  </model>
</document>
```

3.2. XMLForm.org XForm File

If the XML form is to be displayed by XMLForm.org's implementation of XForms, you can define the XML model right in the XForm XML. This is an example of the XML model defined in an XForm form:

```
<document xmlns:xf="http://www.xmlform.org/2003">
  <model>
    <user>
      <names firstname="jim" lastname="" displayname="Jim bob jr." />
      <internet>
        <email />
        <email1 />
      </internet>
      <phone>
        <workphone />
        <homephone />
        <fax />
      </phone>
    </user>
  </model>
  <xf:form id="requestForm" view="userIdentity" action="xformExample.do"
method="GET">
    <xf:label>Personal Information</xf:label>
    <error>
      <xf:violations class="error"/>
    </error>
  </xf:form>
</document>
```

```

</error>
<xf:group ref="names">
  <xf:label>Names</xf:label>
  <xf:input ref="/firstname">
    <xf:label>First</xf:label>
    <xf:violations class="error"/>
  </xf:input>
  <xf:input ref="/lastname">
    <xf:label>Last</xf:label>
    <xf:violations class="error"/>
  </xf:input>
  <xf:input ref="/displayname">
    <xf:label>Display</xf:label>
    <xf:violations class="error"/>
  </xf:input>
</xf:group>
<xf:group ref="internet">
  <xf:label>Internet</xf:label>
  <xf:input ref="/email">
    <xf:label>Email</xf:label>
    <xf:help>Please check this carefully</xf:help>
    <xf:violations class="error"/>
  </xf:input>
  <xf:input ref="/email1">
    <xf:label>Alternate Email</xf:label>
    <xf:violations class="error"/>
  </xf:input>
</xf:group>
<xf:group ref="phone">
  <xf:label>Phone</xf:label>
  <xf:input ref="/workphone">
    <xf:label>Work</xf:label>
    <xf:violations class="error"/>
  </xf:input>
  <xf:input ref="/homephone">
    <xf:label>Home</xf:label>
    <xf:help>Home phone number is required</xf:help>
    <xf:violations class="error"/>
  </xf:input>
  <xf:input ref="/fax">
    <xf:label>Fax</xf:label>
    <xf:violations class="error"/>
  </xf:input>
</xf:group>
<xf:selectBoolean ref="debug">
  <xf:label>Debug?</xf:label>
</xf:selectBoolean>
  <xf:submit id="submit" class="button">
    <xf:label>Submit</xf:label>
    <xf:hint>Add the contact</xf:hint>
  </xf:submit>
</xf:form>
</document>

```

4. Validation

XML form support multiple methods of validation. To determine which validation technique to use, stxx uses the schema namespace property. Also, stxx needs the schema file itself. These properties are set as stxx plugin properties in `struts-config.xml`.

Name	Description	Default
xmlform-schema	Location of the schema file containing validation rules.	If not schema specified, no validation will be performed.
xmlform-schemaNS	The validation schema namespace used to determine how to validate the XML.	<code>http://www.ascc.net/xml/schematron</code>

Table 1: XML form validation properties in struts-config.xml

The error messages reported by validation are checked against the current locale's message resources, if available. This is an example of a validation rule using the default validation method, Schematron:

```
<rule context="/user/names/@firstname">
  <assert test="string-length(.) &gt; 0">err.firstname.none</assert>
</rule>
```

5. Apache Axis Configuration

[Apache Axis](#) uses a [Web Service Deployment Descriptor \(WSDD\)](#) to configure how a web service will be handled. Typically, a web service will have a WSDD, which will then be sent to the Axis installation via the AdminClient (more info in the [Axis User's Guide](#)). If you want your application to be bundled with Axis, you can integrate this configuration into Axis's `server-config.wsdd`, the main WSDD.

stxx requires several stxx-specific parameters in the WSDD:

Name	Description	Required
xmlFormClass	The XMLForm-implementing class	Yes
xmlFormName	The form name	No
xmlFormPhase	The validation phase to use	Required for validation
xmlFormSchema	The validation schema file	Required for validation
xmlFormSchemaNS	The validation schema	No, defaults to

	namespace	http://www.ascc.net/xml/schematron
--	-----------	------------------------------------

Table 1: stxx parameters in the Axis WSDD

This is an example of a WSDD entry:

```
<service name="AddAddressService" provider="java:MSG"
  style="message" use="literal">
  <parameter name="allowedMethods" value="process"/>
  <parameter name="scope" value="session"/>
  <parameter name="className" value="com.oroada.stxx.xform.XMLFormService"/>

  <!-- custom parameters used by stxx -->
  <parameter name="xmlFormClass" value="foo.AddAddressForm" />
  <parameter name="xmlFormSchema" value="/WEB-INF/address-schema.xml" />
  <parameter name="xmlFormPhase" value="all" />
</service>
```

For more information, please consult the [Apache Axis website](#).