How to build an XMLForm Wizard, Step One

by Heidi Brannan

1. Step One

First we need to create our own forms in XML.

- Create a folder called "howto" in src\scratchpad\webapp\mount\xmlform.
- Copy the following xml files below and save them in the folder you created.

1.1. start.xml

Below is the first page which you can copy and save as start.xml in the folder src\scratchpad\webapp\mount\xmlform\howto

```

<?xml version="1.0" ?>

<document>
    <br/><br/><br/>
    <table align="center" width="50%" cellspacing="20">
        <tr>
            <td align="center">
                <h1>
                    Welcome to the XLMForm HowTo!
                </h1>
            </td>
        </tr>

        <tr>
            <td align="center" class="info">
                <code>
                The following form allows users to join mailing lists.
                They are given a choice of areas of interest.
                Depending on their interests they will see a selection of
                mailing lists which they can sign up to.
                </code>
            </td>
        </tr>

        <tr>
            <td align="center">
                <h3>
                    <a href="howto-wizard.html?cocoon-action-start=true">
                        Start!
                    </a>
                </h3>
            </td>
        </tr>
    </table>
</document>
```

The important part to notice is the link:

```
          <a href="howto-wizard.html?cocoon-action-start=true">
```

The text between "cocoon-action" and "=true" is passed to the method prepare of HowtoWizardAction.java which we will write soon.

Next are the actual XML pages that make up the form. These are:

- register.xml
- interest.xml
- organicGardening.xml
- cooking.xml
- smallholding.xml
- confirm.xml
- end.xml

1.2. register.xml

Register.xml lets the user register their username, password and email address so they can join the mailing lists they will next choose.

The important part of the form you will need to change in your own forms are inside the xf:form tag. The id attribute value should match the sitemaps xmlform-id parameters value:

```
       <map:parameter name="xmlform-id" value="form-feedback"/>
```

The view attribute should contain the name of the current xml file, the action attribute should contain the name of the url you are using in the sitemap.

The caption tag is the page heading. Next we have the error tags which are used if you have a validation set in your sitemap. If an error is found then this will display them when the user clicks the next button on your form.

Now we see the input options for the user, such as xf:textbox which will display a textbox. Each such option as a ref attribute which is very important as this is the value we will map to the JavaBean. If we are validating this input then it must have a violations tag inside it saying which class it belongs to.

Finally the form needs a submit tag tag. This lets the user navigate forward to the rest of the form.

```
      <?xml version="1.0" ?>


<!--

  XMLForm instance document for the Cocoon Feedback Wizard.

  Original Author: Torsten Curdt, tcurdt@dff.st, March 2002
    Author: Ivelin Ivanov, ivelin@apache.org, April 2002

-->

<document xmlns:xf="http://xml.apache.org/cocoon/xmlform/2002">


  <xf:form id="form-feedback" view="registration" action="howto-wizard.html">

    <xf:caption>Registration</xf:caption>

    <error>
      <xf:violations class="error"/>
    </error>

    <xf:textbox ref="/userName">
        <xf:caption>Last Name</xf:caption>
        <xf:violations class="error"/>
    </xf:textbox>

    <xf:textbox ref="/email">
        <xf:caption>Email</xf:caption>
        <xf:violations class="error"/>
    </xf:textbox>
```

```
      <xf:password ref="/password">
          <xf:caption>Password</xf:caption>
          <xf:violations class="error"/>
      </xf:password>

      <xf:submit id="next" class="button">
        <xf:caption>Next</xf:caption>
      </xf:submit>

    </xf:form>

  </document>
```

1.3. interest.xml

This XML page lets the user select areas of interest and depending on their answers they will be shown a selection of mailing lists they can join. The intelligence for deciding which pages to show next are in a java file which we will show you how to write later.

This page consists of tick boxes which are either true or false.

```
        <xf:selectBoolean ref="/organicGardening">
          <xf:caption>Organic Gardening</xf:caption>
      </xf:selectBoolean>
```

Below is the page which you can copy into the folder "howto".

```
<?xml version="1.0" ?>



<!--

  XMLForm instance document for the Cocoon Feedback Wizard.

  Original Author: Torsten Curdt, tcurdt@dff.st, March 2002
  Author: Ivelin Ivanov, ivelin@apache.org, April 2002

-->

<document xmlns:xf="http://xml.apache.org/cocoon/xmlform/2002">

  <xf:form id="form-feedback" view="interest" action="howto-wizard.html">

    <xf:caption>Areas of Interest</xf:caption>
    <xf:selectBoolean ref="/organicGardening">
        <xf:caption>Organic Gardening</xf:caption>
    </xf:selectBoolean>

    <xf:selectBoolean ref="/cooking">
        <xf:caption>Cooking</xf:caption>
    </xf:selectBoolean>

    <xf:selectBoolean ref="/smallholdingManagement">
        <xf:caption>Smallholding Management</xf:caption>
    </xf:selectBoolean>

    <xf:submit id="prev" class="button">
      <xf:caption>Prev</xf:caption>
    </xf:submit>

    <xf:submit id="next" class="button">
      <xf:caption>Next</xf:caption>
    </xf:submit>

  </xf:form>

</document>
```

1.4. organicGardening.xml

The next page is shown if the user ticked the organic gardening box in the previous page, interest.xml. This page consists a selection of mailing lists the user can chose. It is very similar to the previous page containing 3 tick boxes. The user has the choice of moving forward through the form or back to the previous page so they can alter their area of interest.

```

<?xml version="1.0" ?>



<!--

  XMLForm instance document for the Cocoon Feedback Wizard.

  Original Author: Torsten Curdt, tcurdt@dff.st, March 2002
  Author: Ivelin Ivanov, ivelin@apache.org, April 2002

-->

<document xmlns:xf="http://xml.apache.org/cocoon/xmlform/2002">

  <xf:form id="form-feedback" view="organicGardening" action="howto-wizard.html">

    <xf:caption>Organic Gardening Mailing Lists:</xf:caption>
    <xf:selectBoolean ref="/flowers">
        <xf:caption>Flowers</xf:caption>
    </xf:selectBoolean>

    <xf:selectBoolean ref="/vegetables">
        <xf:caption>Vegetables</xf:caption>
    </xf:selectBoolean>

    <xf:selectBoolean ref="/fruitTrees">
        <xf:caption>Fruit Trees</xf:caption>
    </xf:selectBoolean>

    <xf:submit id="prev" class="button">
      <xf:caption>Prev</xf:caption>
    </xf:submit>

    <xf:submit id="next" class="button">
      <xf:caption>Next</xf:caption>
    </xf:submit>

  </xf:form>

</document>
```

1.5. cooking.xml

The next page is a selection of cookery mailing lists, very similar to the organicGardening.xml page. This page will appear if the user ticked the organic gardening option on the interest.xml page.

```
<?xml version="1.0" ?>



<!--

  XMLForm instance document for the Cocoon Feedback Wizard.

  Original Author: Torsten Curdt, tcurdt@dff.st, March 2002
  Author: Ivelin Ivanov, ivelin@apache.org, April 2002

-->

<document xmlns:xf="http://xml.apache.org/cocoon/xmlform/2002">

  <xf:form id="form-feedback" view="cooking" action="howto-wizard.html">

    <xf:caption>Cooking Mailing Lists:</xf:caption>
    <xf:selectBoolean ref="/traditionalReciepes">
        <xf:caption>Traditional Reciepes</xf:caption>
    </xf:selectBoolean>

    <xf:selectBoolean ref="/soups">
        <xf:caption>Soups</xf:caption>
    </xf:selectBoolean>

    <xf:selectBoolean ref="/veganCookery">
        <xf:caption>Vegan Cookery</xf:caption>
    </xf:selectBoolean>

    <xf:submit id="prev" class="button">
      <xf:caption>Prev</xf:caption>
    </xf:submit>

    <xf:submit id="next" class="button">
      <xf:caption>Next</xf:caption>
    </xf:submit>

  </xf:form>

</document>
```

1.6. smallholdingManagement.xml

Again this page is similar to organicGardening.xml, cooking.xml as it gives the user a choice of mailing lists. This page will only appear if the user selected Smallholding Management as an interest on the interest.xml page.

```
<?xml version="1.0" ?>



<!--

  XMLForm instance document for the Cocoon Feedback Wizard.

  Original Author: Torsten Curdt, tcurdt@dff.st, March 2002
  Author: Ivelin Ivanov, ivelin@apache.org, April 2002

-->

<document xmlns:xf="http://xml.apache.org/cocoon/xmlform/2002">

  <xf:form id="form-feedback" view="smallholdingManagement" action="howto-wizard.html">

    <xf:caption>Smallholding Management Mailing Lists</xf:caption>
    <xf:selectBoolean ref="/pigKeeping">
        <xf:caption>Pig Keeping</xf:caption>
    </xf:selectBoolean>

    <xf:selectBoolean ref="/pygmyGoats">
        <xf:caption>Pygmy Goats</xf:caption>
    </xf:selectBoolean>

    <xf:selectBoolean ref="/henKeeping">
        <xf:caption>Hen Keeping</xf:caption>
    </xf:selectBoolean>

    <xf:submit id="prev" class="button">
      <xf:caption>Prev</xf:caption>
    </xf:submit>

    <xf:submit id="next" class="button">
      <xf:caption>Next</xf:caption>
    </xf:submit>

  </xf:form>

</document>
```

1.7. confirm.xml

This page shows the user the data that has been collected from them. The JavaBean that stores all the data is queried and the information collected is displayed. This is done using the xf:output tag, specifying which value to display with the ref attribute. If they wish to change any of the information they can do this now by clicking the previous button and altering their answers. Otherwise they can click the finish button which will take them to the end page, end.xml.

```

<?xml version="1.0" ?>

<!--

  XMLForm instance document for the Cocoon Feedback Wizard.

  Original Author: Torsten Curdt, tcurdt@dff.st, March 2002
  Author: Ivelin Ivanov, ivelin@apache.org, April 2002

-->
```

```
<document xmlns:xf="http://xml.apache.org/cocoon/xmlform/2002">


  <xf:form id="form-feedback" view="confirm" action="howto-wizard.html">


    <xf:caption>Confirm Input</xf:caption>


    <!-- from page1 -->

    <xf:output ref="/userName">
        <xf:caption>User Name</xf:caption>
    </xf:output>

    <xf:output ref="/email">
        <xf:caption>Email</xf:caption>
    </xf:output>

    <xf:output ref="/password">
        <xf:caption>Password</xf:caption>
    </xf:output>

    <!-- from page2 -->
    <xf:output ref="/organicGardening">
        <xf:caption>Organic Gardening</xf:caption>
    </xf:output>

    <xf:output ref="/cooking">
        <xf:caption>Cooking</xf:caption>
    </xf:output>

    <xf:output ref="/smallholdingManagement">
        <xf:caption>Smallholding Management</xf:caption>
    </xf:output>

    <!-- from page3 -->
    <xf:output ref="/flowers">
        <xf:caption>Flowers</xf:caption>
    </xf:output>

    <xf:output ref="/vegetables">
        <xf:caption>Vegetables</xf:caption>
    </xf:output>

    <xf:output ref="/fruitTrees">
        <xf:caption>Fruit Trees</xf:caption>
    </xf:output>

    <!-- from page4 -->
    <xf:output ref="/traditionalReciepes">
        <xf:caption>Traditional Reciepes</xf:caption>
    </xf:output>
    <xf:output ref="/soups">
        <xf:caption>Soups</xf:caption>
    </xf:output>

    <xf:output ref="/veganCookery">
        <xf:caption>Vegan Cooking</xf:caption>
    </xf:output>

    <!-- from page5 -->
    <xf:output ref="/pigKeeping">
        <xf:caption>Pig Keeping</xf:caption>
    </xf:output>

    <xf:output ref="/pygmyGoats">
        <xf:caption>Pygmy Goats</xf:caption>
    </xf:output>

    <xf:output ref="/henKeeping">
        <xf:caption>Hen Keeping</xf:caption>
    </xf:output>

    <!-- submit -->

    <xf:submit id="prev" class="button">
      <xf:caption>Prev</xf:caption>
    </xf:submit>

    <xf:submit id="next" class="button">
      <xf:caption>Finish</xf:caption>
    </xf:submit>

  </xf:form>

</document>
```

1.8. end.xml

This page displays the final page. It tells the reader they have succesfully filled in the form and gives them the option to return to the start.

```

     <?xml version="1.0" ?>
<document>
    <br/><br/><br/>
    <table align="center" width="50%" cellspacing="20">
        <tr>
            <td align="center">
                <h1>
                    You have reached the last page of the How To Form example!
                </h1>
            </td>
        </tr>

        <tr>
            <td align="center" class="info">
                <code>
                    Your feedback form was processed successfully.
                </code>
            </td>
        </tr>

        <tr>
            <td align="center">
                <h3>
                    <a href="howto-wizard.html">Go to home page.</a>
                </h3>
            </td>
        </tr>
    </table>
</document>
```

Step 2: Validation

2. Revisions

Find a problem with this document? Consider contacting the mailing lists or submitting your own revision. For instructions, read the How To Submit a Revision.

- Revision, 2003-09-14 10:56