The DSpace Course - Metadata Input Customisation

Lewis, Stuart

http://hdl.handle.net/10673/73

Downloaded from DSpace Repository, DSpace Institution's institutional repository
Module: Metadata Input Customisation

Module overview:
This module initially looks at the submission process in further detail, specifically input-forms.xml. It details what the input-forms.xml file is and how it can be used to customize the DSpace submission process. The module will conclude with a practical on a simple modification to the DSpace submission process.

This module builds on topics covered in modules ‘Repository Structure’ and ‘Introduction to Items’. It is advised that you have completed these modules before continuing with this module.

Module objectives:
By the end of this module you will:

1. Understand which part of the submission process can be customised
2. Understand about input-forms.xml specifically:
   a. Field Definitions
   b. Field Types
3. Have modified the DSpace submission process

Note

For the practical exercise, please refer to your sheet ‘Local instructions’ for details of the following:

- How to launch a terminal window and a web browser
- What the path to [dspace] is
- How to restart Tomcat
- What the URL of your DSpace installation is
The Submission Process

- In module ‘Introduction to Items’ an item was submitted using the default submission process
- This submission process can be customised allowing:
  - Adding additional metadata to be collected about an item
  - Individual collections to have unique metadata for their items
\textit{Input-forms.xml}

- The forms used during a submission of an item via the UI are created based on
  \[ \text{[dspace]/config/input-forms.xml} \]
- Detailed information can be found in
  \[ \text{[dspace]/docs/submission.html} \]

- The input-forms.xml consists of the top level element \textit{<input-forms>} which contains 3 elements:
  - \textit{<form-map>} Mapping defined forms to collections
  - \textit{<form-definitions>} Definition of single forms
  - \textit{<form-value-pairs>} Definition of selection lists
Form Map

- The form-map maps collection handles to forms
- Each `<form-map>` has 2 attributes:
  - `collection-handle="collectionHandle"` ("default" if not selected)
  - `form-name="formName"` (e.g. `<form name="myForm">`)

```
<input-forms>
<form-map>
  <name-map collection-handle="default" form-name="traditional" />
  <name-map collection-handle="110" form-name="one" />
</form-map>
</input-forms>
```

- Collection 123456789/110 has a form called 'one' defined
Form Definitions

- The form-definitions map lays out the detailed definition of all the submission forms which contain the metadata field entries.
- Each separate form set has a unique name as an attribute. This name matches one of the names in the form-map.
- A form called “traditional” has been defined in this form definition.

```
<input-forms>
  <!-- Form Set Definitions -->
  <form-definitions>
    <form name="traditional">
      ...
    </form>
  </form-definitions>
</input-forms>
```
Form Definitions

- A form is created with `<form></form>`
- Each form has the attribute name e.g. `<form name="myForm">`
- This name is used to map a form to a collection.
- A form may consist of up to 3 elements:
  - `<page></page>`
  - `<field></field>`
  - `<page number="1">`
Form Definitions - Fields

A `<field>` consists of:

<table>
<thead>
<tr>
<th><code>&lt;field&gt;</code> Name</th>
<th>Mandatory?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;dc-schema&gt;</code></td>
<td>True</td>
<td>metadata schema name</td>
</tr>
<tr>
<td><code>&lt;dc-element&gt;</code></td>
<td>True</td>
<td>metadata element</td>
</tr>
<tr>
<td><code>&lt;dc-qualifier&gt;</code></td>
<td></td>
<td>metadata qualifier</td>
</tr>
<tr>
<td><code>&lt;repeatable&gt;</code></td>
<td>True</td>
<td>Values: true or false Mark a field as repeatable.</td>
</tr>
<tr>
<td><code>&lt;label&gt;</code></td>
<td>True</td>
<td>Label used in UI</td>
</tr>
<tr>
<td><code>&lt;input-type&gt;</code></td>
<td>True</td>
<td>Type of input field</td>
</tr>
<tr>
<td><code>&lt;hint&gt;</code></td>
<td>True</td>
<td>Hint used in UI</td>
</tr>
<tr>
<td><code>&lt;required&gt;</code></td>
<td>True</td>
<td>If completed, the field is marked as mandatory.</td>
</tr>
</tbody>
</table>

Form Definitions - Fields
Form Definitions - Fields

- The input-type:

<table>
<thead>
<tr>
<th>Input-type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>onebox</td>
<td>Single boxes for short text</td>
</tr>
<tr>
<td>twobox</td>
<td>A pair of single boxes</td>
</tr>
<tr>
<td>textarea</td>
<td>A text area</td>
</tr>
<tr>
<td>name</td>
<td>A pair of text boxes for personal names</td>
</tr>
<tr>
<td>date</td>
<td>At least a year must be given</td>
</tr>
<tr>
<td>dropdown</td>
<td>A selection list</td>
</tr>
<tr>
<td>quadrop-value</td>
<td>A selection list combined with a single textbox</td>
</tr>
</tbody>
</table>

1. Title: Enter the main title of the item.
2. Series/Report No.: Enter the series name and report or paper number.
3. Abstract: Enter the abstract of the item below.
4. Authors: Last name, e.g. Smith and first name(s) + "Jr", e.g. Donald Jr.
5. Date Issued: Enter the date in the format Month/No Month/Day/Year.
6. Language: Select the language from the options provided.

Form Definitions - Fields
Form Value Pairs

- Form-value-pairs populate dropdown and qualdrop-value lists.
- The form-value-pairs element holds child elements named 'value-pairs'.
- A form value pair has two attributes:
  - value-pairs-name
  - dc-term

```xml
<form-value-pairs>
  <value-pairs name="common_identifiers" dc-term="identifier">
    <pair>
      <displayed-value>ISSN</displayed-value>
      <stored-value>issn</stored-value>
    </pair>
    ...
  </value-pairs>
</form-value-pairs>
```

Form Value Pairs

- Form-value-pairs populate dropdown and qualdrop-value lists.
- The form-value-pairs element holds child elements named 'value-pairs'.
- A form value pair has two attributes:
  - value-pairs-name
  - dc-term
Practical: Metadata Customisation

1. Create a new collection called ‘Metadata’ as specified in the module ‘Repository Structure’

2. Once created, navigate into the collection, click edit on the ‘admin tools’ menu and note its handle id i.e. 123456789/422

3. Open a terminal window

4. Navigate to [dspace]/config

5. Open the input-forms.xml using gedit: type ‘gedit input-forms.xml’

6. Locate the lines:

7. Add an additional line setting:
   - collection-handle=“<Your collection ID>”
   - form-name=“one”

8. This forces any submission into this collection to use a different input form

9. Restart Tomcat

10. Submit an item to your newly created collection as specified in the module ‘Introduction to Items’

11. The submission form input should now modified
Credits

- These notes have been produced by:
  - Stuart Lewis & Chris Yates
  - Repository Support Project
    - http://www.rsp.ac.uk/
  - Part of the RepositoryNet
  - Funded by JISC
    - http://www.jisc.ac.uk/