Chapter 10 - Enabling Form Creation with the XML Forms Modules

XML Forms is a collection of Drupal modules that allow for the manipulation of XML documents though Drupal forms. The Islandora Form Builder (XML_Forms modules) makes it possible for users to create, copy, and edit ingest forms, and to affiliate them with Content Models in the repository.

On this page:

Modules Overview

The following Modules are installed as part of the XML Forms package. See inline documentation for additional details.

- Objective Forms
- Islandora XML Form API
- Islandora XML Schema API
- Islandora XML Form Elements
- Islandora XML Form Builder
- Islandora XML Forms
- Islandora Content Model Forms

Objective Forms

This module allows for the creation of Object Based Drupal Forms. It provides a number of functions and class for processing/populating forms.

Some important notes.

- Each form element is assigned a unique hash Form Property to identify it, **#hash**.
- Each form element that is created is stored in a registry and it will persist though out the lifetime of the form even if it's removed from the form. Ancestry of Form Elements is stored so if a Form Element is cloned we will be able to determine the Form Element that is was cloned from.
- Form Properties can be objects. To define new Form Properties implement the hook **objectify_properties**.
- Form's will be auto-populated from $form_states['values'].
- There is a **FormStorage** class that can be used to store any persistent data.

Islandora XML Form API

The core of the library this module provides functions for processing XML files through forms.

In essence this module models.

- The form to be processed.
- The form properties needed to manipulate XML
- The XML document to be manipulated
- The actions required to generate repeating form elements (tabs,tags) from the XML document
- The schema needed to determine the insert locations of elements and the validation requirements. (included via the **Islandora XML Schema API**)

Islandora XML Schema API

This module provides functions for processing Schema files. It's used to determine where to insert XML Nodes, and how to validate them.

Islandora XML Form Elements

This module defines custom Drupal Form Elements along with AHAH callbacks.

Elements

tabs/tabpanels

These Form Elements are used to model XML Nodes that can repeat and contain other XML Nodes.

Example:

```
<cntaddr> <!-- tabpanel one -->

<addrtype>undefined</addrtype>

<city>undefined</city>

<!-- ... -->

</cntaddr>

<cntaddr> <!-- tabpanel two -->

<addrtype>one</addrtype>

<!-- ... -->

</cntaddr>
```

tags/tag

These Form Elements are used to model XML Nodes that can repeat and contain only character data.

Example:

```
<origin>test</origin> <!-- tag #1 -->

<origin>undefined</origin> <!-- tag #2 -->

<origin>testing</origin> <!-- tag #3 -->
```

Islandora XML Form Builder

This module allows for importing/exporting/creating/editing/deleting form definitions. The form definitions are defined in XML and stored in the Drupal Database.

The form builder can be reached by following the “XML Form Builder” link in the **Admin Content Management** section.

From the main interface, users can:

- Create a new form, by selecting the Create button.
- Select a form to Copy/Edit/View/Export/Delete via the drop down field.
- Copy the selected form
- Edit the selected form
- View the Selected form
- Export the select form as an **XML Form Defintition**.
- Delete the selected form from the site.

XML Form Edit GUI

This GUI models the XML Definition, allowing the user to add/remove/edit form elements, as well as change their order.

- Common Form Elements
    - Implemented by all elements.
    - XML CRUD Actions (Create, Read, Update, Delete)
- Advanced Form Elements
    - Only implemented by some elements.
    - In most cases they will not be modified

XML CRUD Actions are very important. They define how to create/read/update/delete XML nodes using the form elements value and properties.

All actions define the Path and Path Context properties. These define where approximately in the XML document will the action be performed.

Path is an XPath that defines where the action will take place.

Path Context defines what context the path can be executed in. It can be one of three values:

1. **Document - The XPath is executed at the root of the document.**
2. *Parent - The XPath is executed at its parents context. This is only valid if the form element is embedded in another form element like a tab panel or fieldset, and that tab panel or fieldsets Read action selected an XML node.*
3. *Self - The XPath is executed in the context selected by the Read action. This can be a bit tricky to understand. Read is the first action performed on the form element before rendering the element. Its used to populate the form element with a value. The node in which it gets its value from is set to be the Self context.*

XML Node Actions

Create

- Path
  - XPath to the parent node where this new element will be inserted
- Path Context
  - The context in which Path is executed.
- Schema
  - The XPath to this elements to create's definition.
- Type
  - One of three values: **Element**, **Attribute**, **XML**. This is the type of XML node that will be created. **XML** stands for a XML snippet.

Element

- Value
  - The name of the type to create if it is **Element** or**Attribute.** If it is **XML** then it will be a XML snippet where the form fields submitted value will replace **%value%** in the snippet.

Read

- Path
  - XPath to the element we want to use to populate this form field.
- Path Context
  - The context in which Path is executed.

Update

- Path
  - XPath to the element to update. This will often be **self::node()**.
- Path Context
  - The context in which Path is executed, this is often **self** which means it refers to the node selected by **Read**.
- Schema
  - Schema path to the elements definition, this is used for validating the submitted form values.

Delete

- Path
  - XPath to the element to delete. This will often be **self::node()**.
- Path Context
  - The context in which Path is executed. This is often **self** which means it refers to the node selected by **Read**.

Islandora XML Forms

This module exists to set up the proper namespace and requirements for including all other modules.

Islandora Content Model Forms

Overview

Create associations between content models and forms. A content model can be associated with any number of forms, and meta-data data-streams.

When this form is submitted it modifies the existing meta-data data-stream adding/removing/modifying elements according to what has changed in the form.

Pre-installation software checklist

Ensure that Islandora is installed and working, and that you have downloaded and enabled the XML Forms modules from http://islandora.ca/downloads. There will be a number of dependencies listed in the XML Forms section of your Drupal modules page – these must be downloaded and enabled before XML Forms will work.

Installation Steps

1. Download XML_Forms from http://islandora.ca/downloads and unzip in your modules directory

2. Install all dependencies

3. Activate the modules

4. Set permissions associated with Form Builder

6. Navigate to Administer>Content management>XML Form Builder to access the Form Builder and Administer>Content management>Form Associations to administer Form associations.