http://bit.ly/1MOGyiF

Working With Form Builder

Islandora Conference - Charlottetown, PE
August 5, 2015

Donald Moses (UPEI)/Nigel Banks (DGI)

What is the XML Form Builder?

- **XML Forms** is a collection of Drupal modules that allow for the manipulation of XML documents though Drupal forms.

- **XML Form Builder** makes it possible to create, copy, and edit ingest forms, and to affiliate them with Content Models in the repository.

XML Form Builder Features

- Works with most standards, wanna share?
- Will work with custom standards.
- Highly configurable.
- Forms are portable (Export/Import).
- Safe.
- Built upon the existing Drupal Form API.

What Form Builder is not ...

- Particularly user friendly
- Simple
- Intuitive

Metadata Standards

- You may want to join the Islandora Metadata Interest Group.
- Visualizing Standards
- Library of Congress
- Wikipedia
- JISC

Why use a standard?

- Decisions made for you
- Support from a wider community
- Interoperability between your data and others
- Leverage existing tools
  - data dictionaries
    - Digital Library Federation/Aquifer Implemen Guidelines for Shareable MODS Records
  - data extraction and transformation tools
    - http://www.loc.gov/standards/mods/mods-conversions.html

10/1/1991	ca. June 10, 1901
ca. 1920.	(ca). June 19, 1901)
(ca). 1920)	Scanned and Processed: 1998-06 01
2001.06.08 by CAD	[between 1904 and 1908]
Unknown	[ca. 1967]
1853	1918?
c1875	191-?
c1908 November 19	1870 December, c1871
[2001 or 2002]	1920, 1921, 1922, 1923, 1924, 1925, 1926, 1927, 1928, 1929
[1919?]	20th century
Halloween 1975	IIIrd century A.D. (not before 237 A.D.)
PEI's centennial	January-February, 200 A.D.
1926 Web site: 7/25/02	Series 1871 ca. 1871

How Form Builder Works

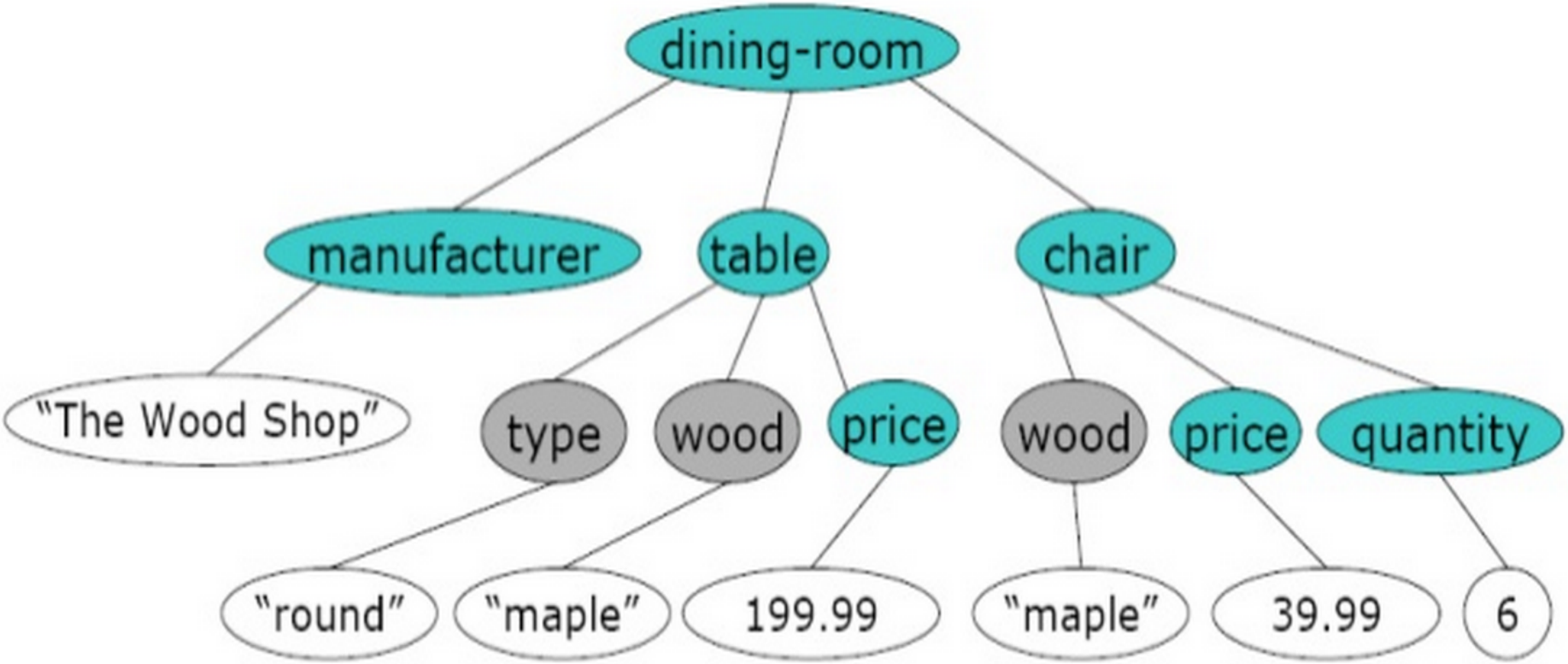
Form Builder and XPath

Form definitions are essentially about XPath.

Examples

Cheat Sheet

A good tool to help you work with XML XSLT's and XPath is Oxygen!



//price → 2 element nodes

/*/*/@wood → 2 attribute nodes

//@wood/.. → 2 nodes (table and chair)

/dining-room/*[price > 100]/@wood → 1 attribute node

//price[1]/text() → 2 text values! (both price are first)

(//price)[1]/text() → 1 text value (first from result collection)

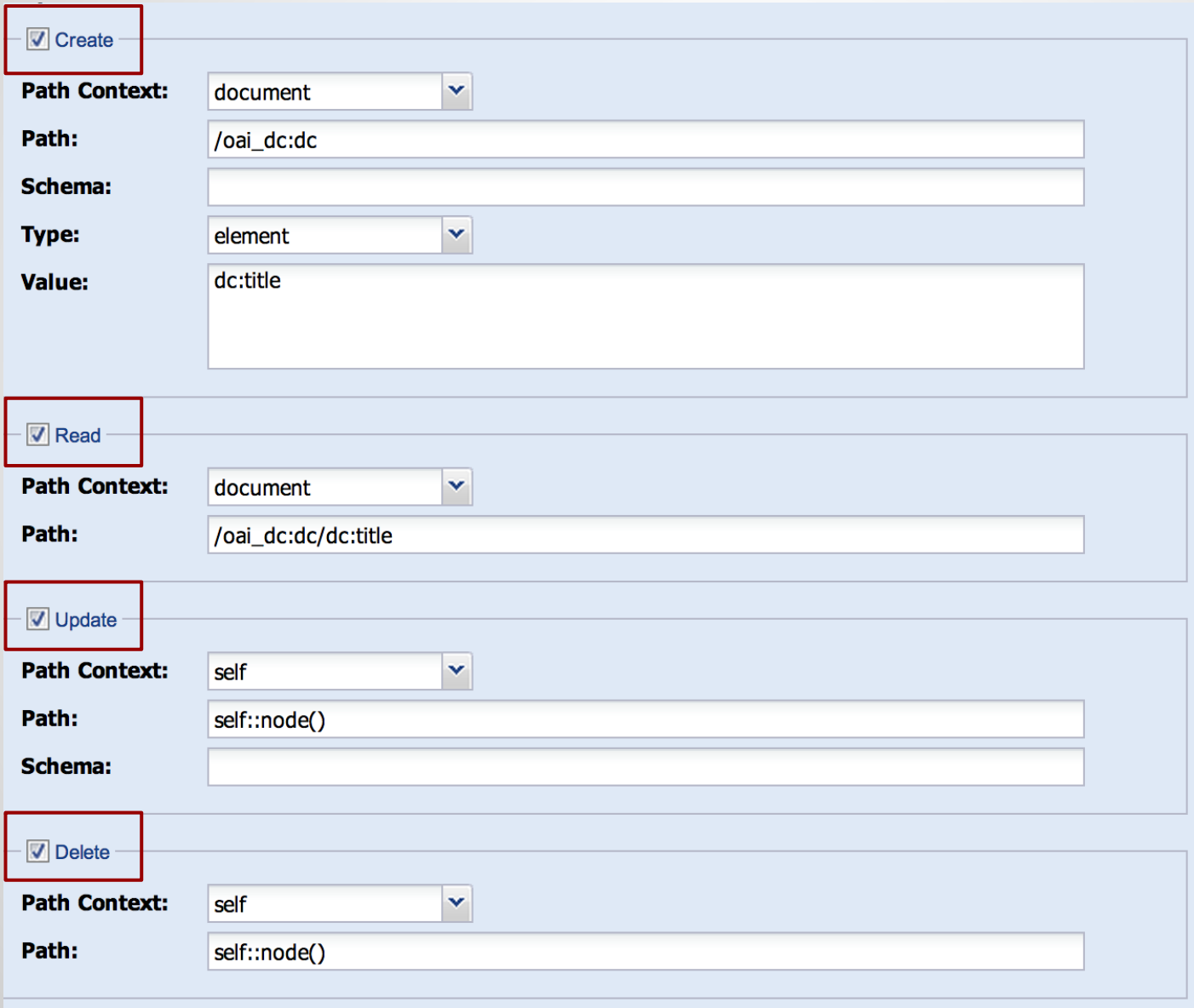
CRUD Rules

Create / Read / Update / Delete

The form treats each field in the form as a potential action, multiple actions can occur.

XPaths describe where the action will take place in the XML Document.

CRUD (Create, Read, Update, Delete)



CRUD (Read)

Populates the form with data from an XML document.

- Performed before displaying the form
- Remembers what was read for later:
  - Updating or Deletion (***self::***) Context references what was *‘read’*
- If the nothing can be read it will be **Created**

CRUD (Create)

If Read was failed.

and

If the parent XML element exists.

After the form is submitted:

We create the new element/attribute and append it to the parent XML element.

CRUD (Create) Options

- Attribute
  - eg. <name **type="personal"**>Joe</name>
- Element
  - eg. <**genre**>book</**genre**>
- XML: Snippets and the use of %value%
  - eg.

    <typeOfResource collecton=”yes”>

    %value%

    </typeOfResource>

CRUD (Update)

If Read was successful.

and

If the element/attribute exists.

After the form is submitted:

We use the value in the form field to updated the element/attribute selected.

CRUD (Delete)

If Read was successful.

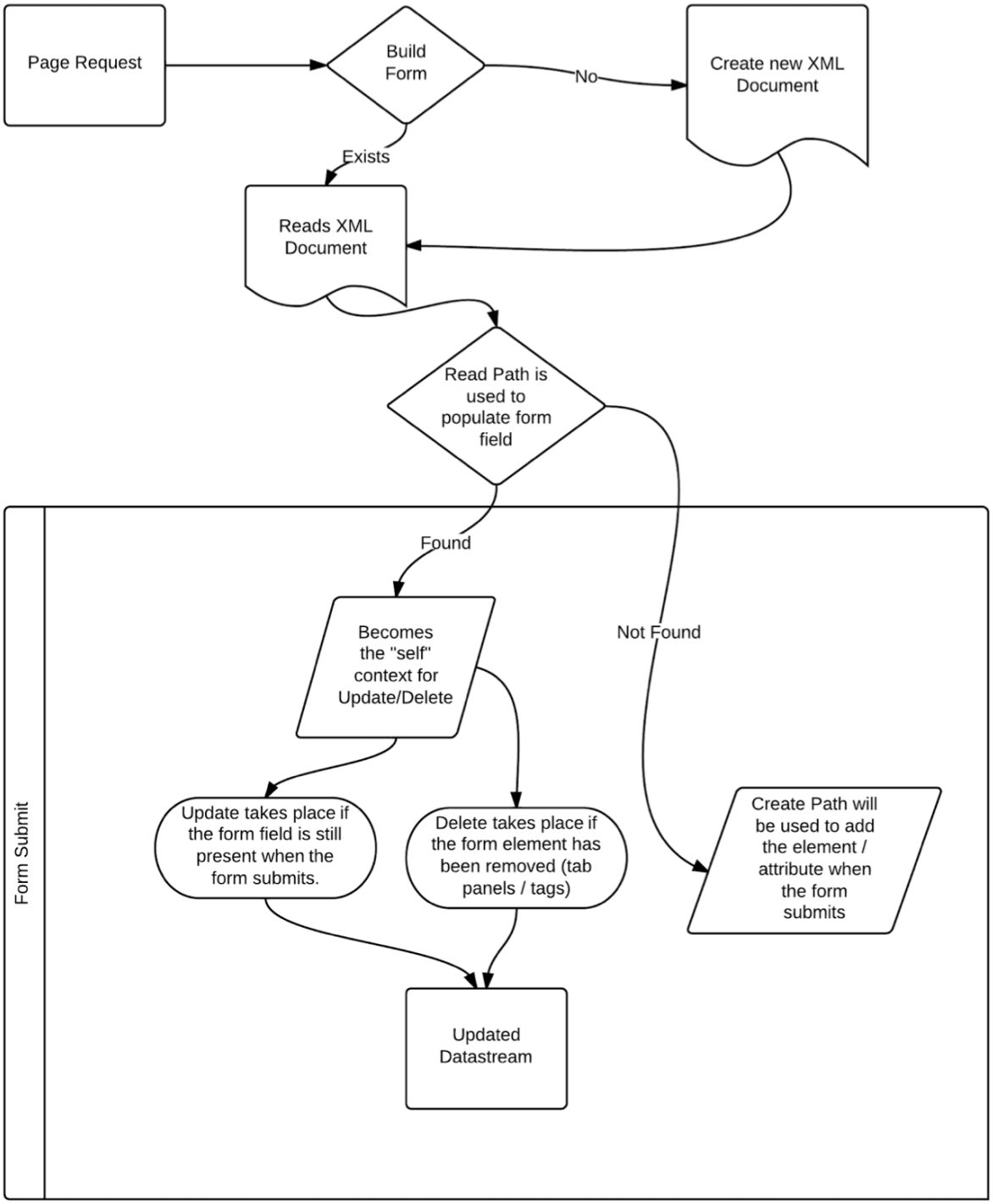
and

If the form field was **removed** from the form.

After the form is submitted:

We Delete the element/attribute selected.

Only elements which can be removed from the form can use Delete (tabpanel, tags, fieldpanel)


Page Request
Build Form
No
Create new XML Document
Exists
Reads XML Document
Read Path is used to populate form field
Found
Not Found
Form Submit
Becomes the "self" context for Update/Delete
Update takes place if the form field is still present when the form submits.
Delete takes place if the form element has been removed (tab panels / tags)
Create Path will be used to add the element / attribute when the form submits
Updated Datastream

Getting Ready to Build a Form

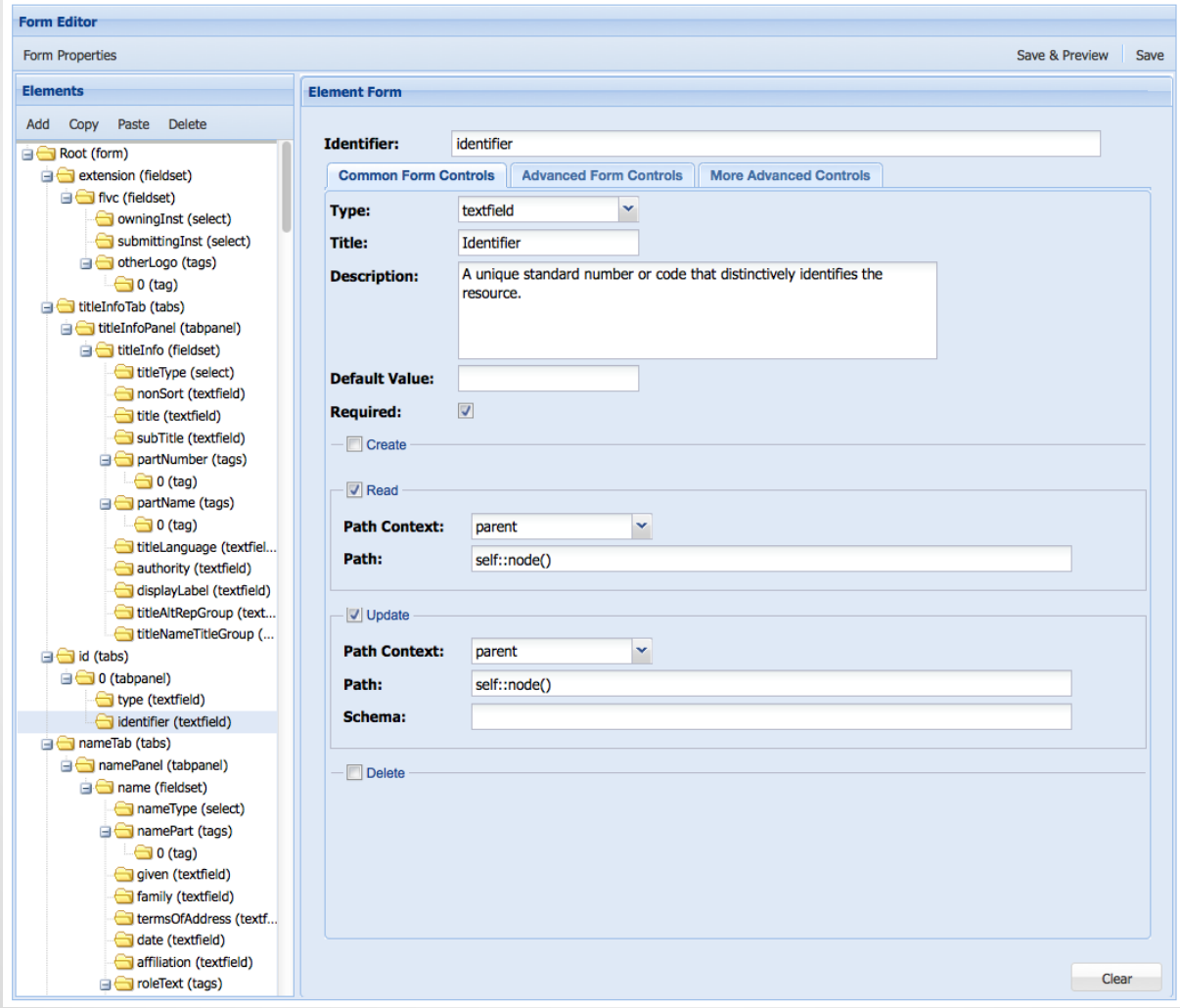
- Understand your schema
  - the xsd schema itself
  - schema documentation
  - **a good example metadata record**
  - controlled vocabularies
- eg. the DataCite Metadata Schema
  - xsd
  - documentation
  - example DataCite XML metadata record
  - vocabularies - see Appendix 1
  - resulting DataCite XML Form - dl and import
- Tools...

Form Builder Documentation

- Module documentation
  - XML Forms on the wiki
    - https://wiki.duraspace.org/display/ISLANDORA715/XML+Forms
  - XML Form Builder on the wiki
    - https://wiki.duraspace.org/display/ISLANDORA715/XML+Form+Builder
- Form Creation Tutorial
  - *How to Edit/Create Ingest Forms*

A sample Form Builder Form ... I ❤ FSU

Thank you FSU for sharing your wonderful full MODS form.



Add Form Properties

```
<?xml version="1.0" encoding="UTF-8"?>
<resource xsi:schemaLocation="http://datacite.org/schema/kern
    <identifier identifierType="DOI">10.5072/example-full</id
    <creators>
        <creator>
            <creatorName>Miller, Elizabeth</creatorName>
            <nameIdentifier schemeURI="http://orcid.org/" nam
            <affiliation>DataCite</affiliation>
        </creator>
    </creators>
    <titles>
        <title xml:lang="en-us">Full DataCite XML Example</ti
        <title xml:lang="en-us" titleType="Subtitle">Demonstr
    </titles>
    <publisher>DataCite</publisher>
    <publicationYear>2014</publicationYear>
    <subjects>
        <subject xml:lang="en-us" schemeURI="http://dewey.inf
    </subjects>
    <contributors>
        <contributor contributorType="ProjectLeader">
            <contributorName>Starr, Joan</contributorName>
            <nameIdentifier schemeURI="http://orcid.org/" nam
            <affiliation>California Digital Library</affiliat
        </contributor>
    </contributors>
    <dates>
        <date dateType="Updated">2014-10-17</date>
    </dates>
    <language>en-us</language>
    <resourceType resourceTypeGeneral="Software">XML</resourc
    <alternateIdentifiers>
        <alternateIdentifier alternateIdentifierType="URL">ht
    </alternateIdentifiers>
    <relatedIdentifiers>
        <relatedIdentifier relatedIdentifierType="URL" relati
        <relatedIdentifier relatedIdentifierType="arXiv" rela
    </relatedIdentifiers>
    <sizes>
        <size>3KB</size>
    </sizes>
```

Properties Form

Root Element

Root Element Name: resource

Namespace URI: http://datacite.org/schema/kernel-3

Schema

Name: http://schema.datacite.org/meta/kernel-3/metadata.xsd

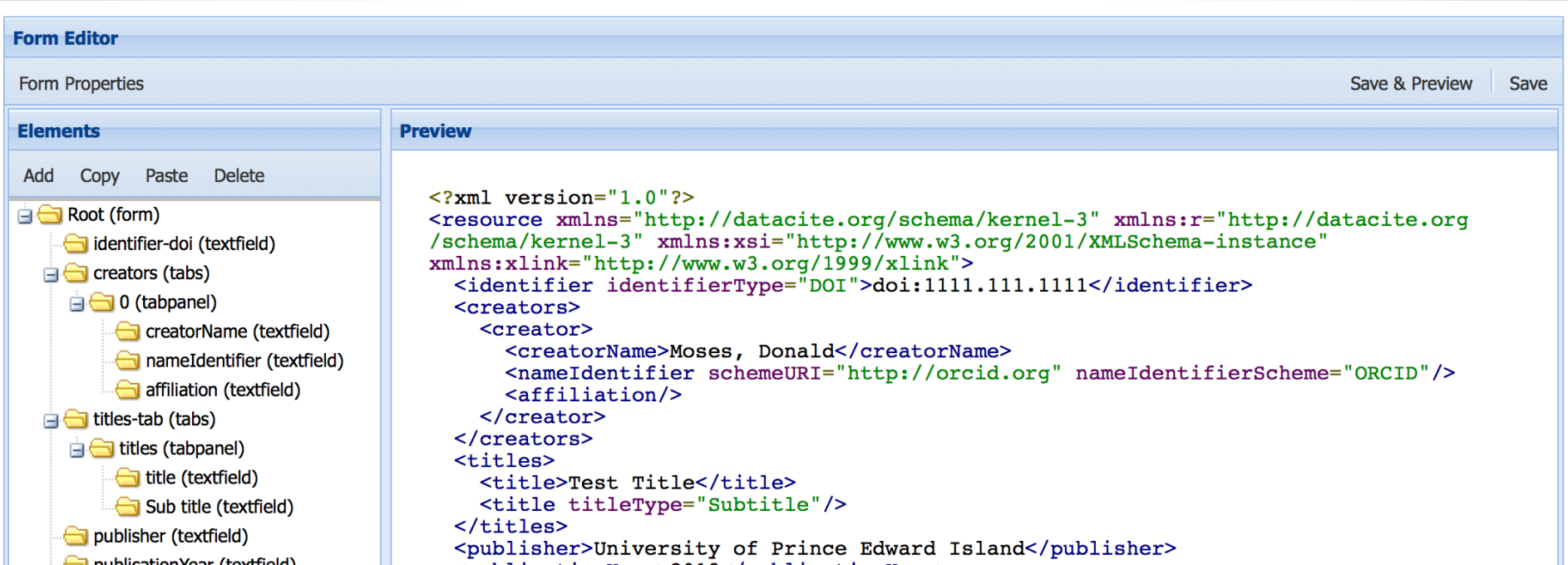
Namespaces

Add Delete

Prefix	URI
r	http://datacite.org/schema/kernel-3
xsi	http://www.w3.org/2001/XMLSchema-instance
xlink	http://www.w3.org/1999/xlink

Add new field > Submit form

- Form Builder will report issues with your form when you submit it
- After adding a new field, populate with sample data, and submit. Repeat.



Debugging Forms

- File: /var/www/drupal/htdocs/sites/all/modules/islandora_xml_forms /api/XMLDocument.incLine: 221Error: XPath query failed; either the expression is malformed or the context node is invalid.Path: self:node()Context: creator<?xml version="1.0"?>

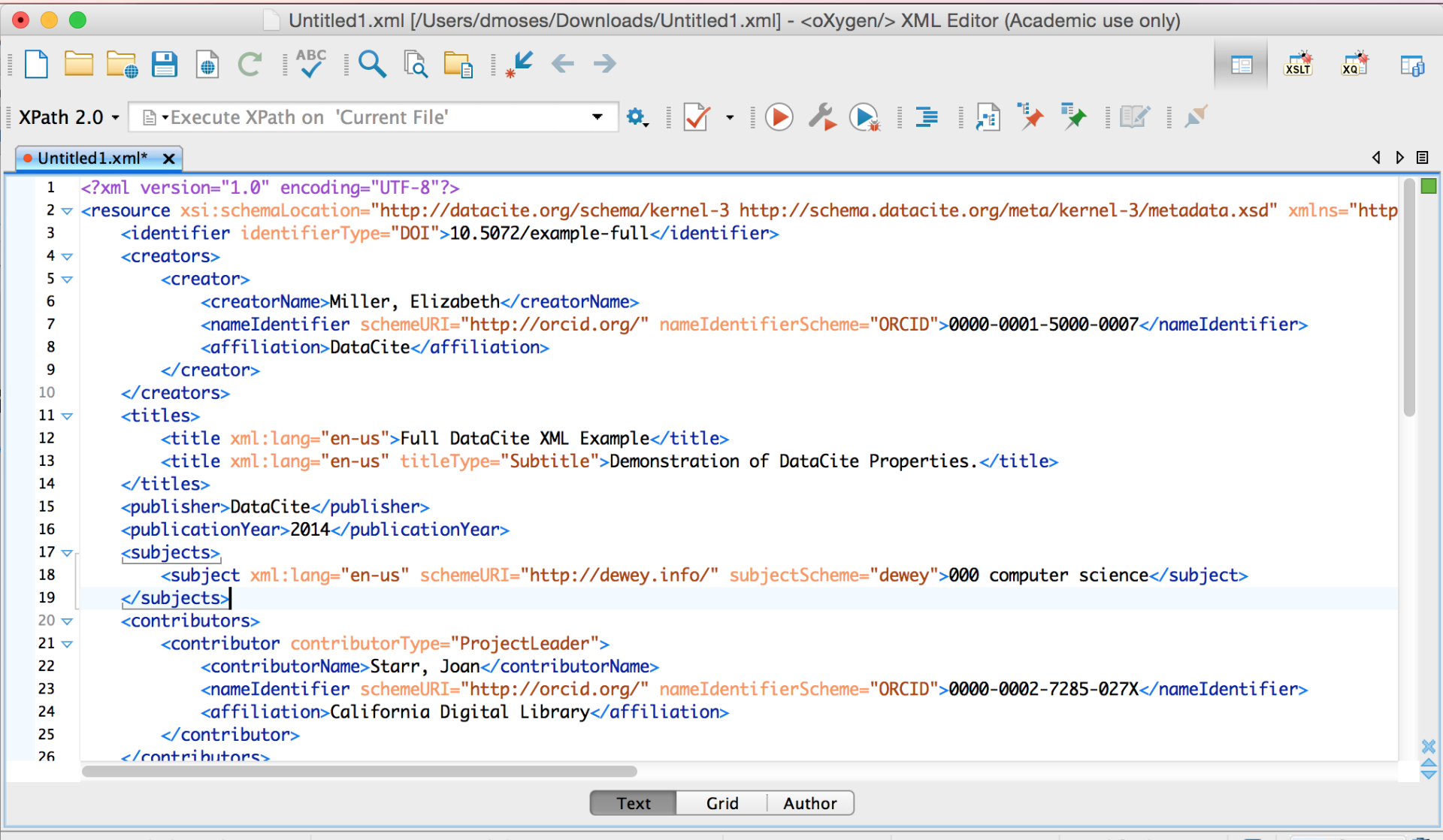
Drupal watchdog log

- admin/reports/dblog

Your apache log

- /var/log/apache2/error.log

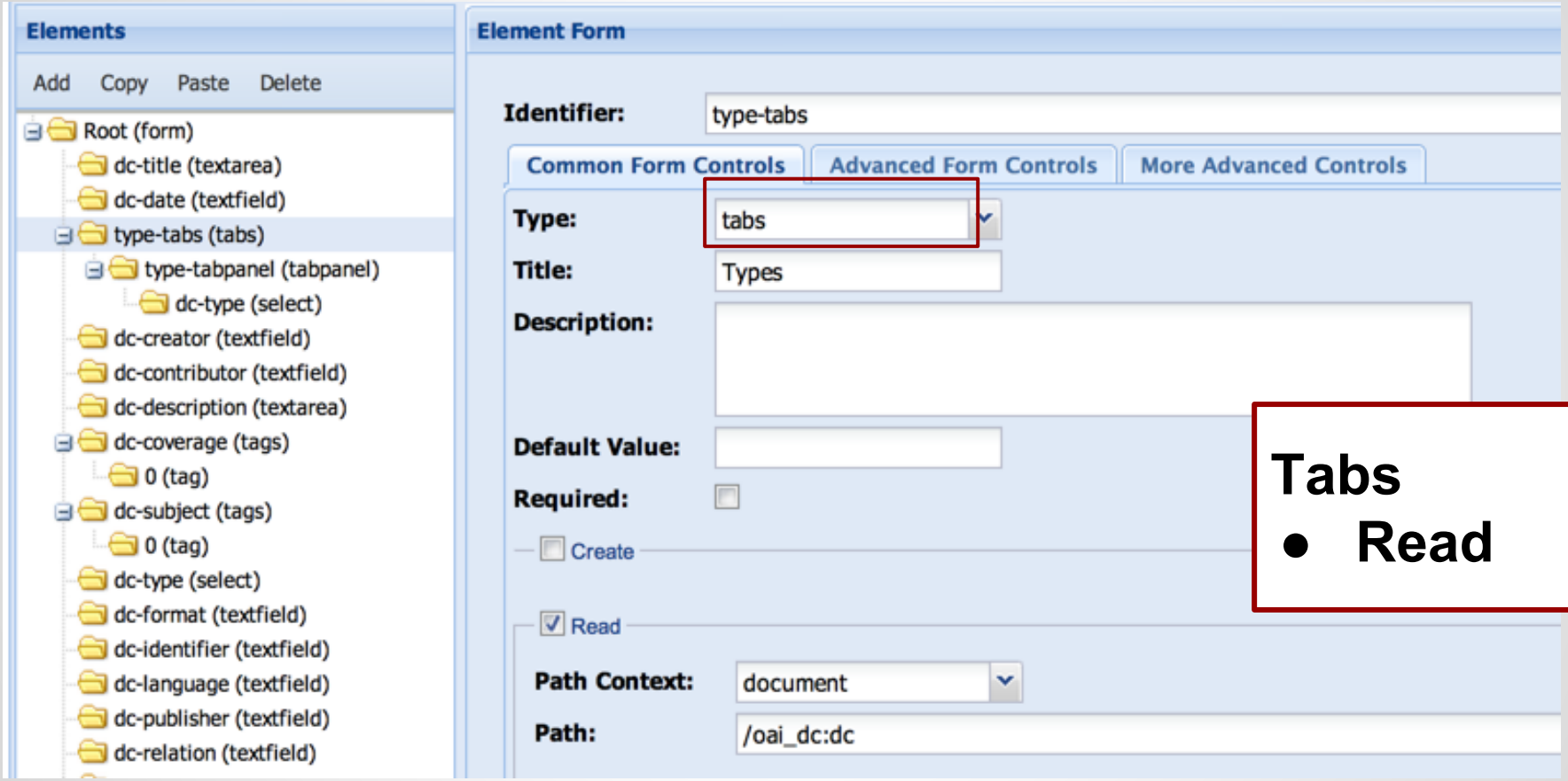
Validate your XML Output

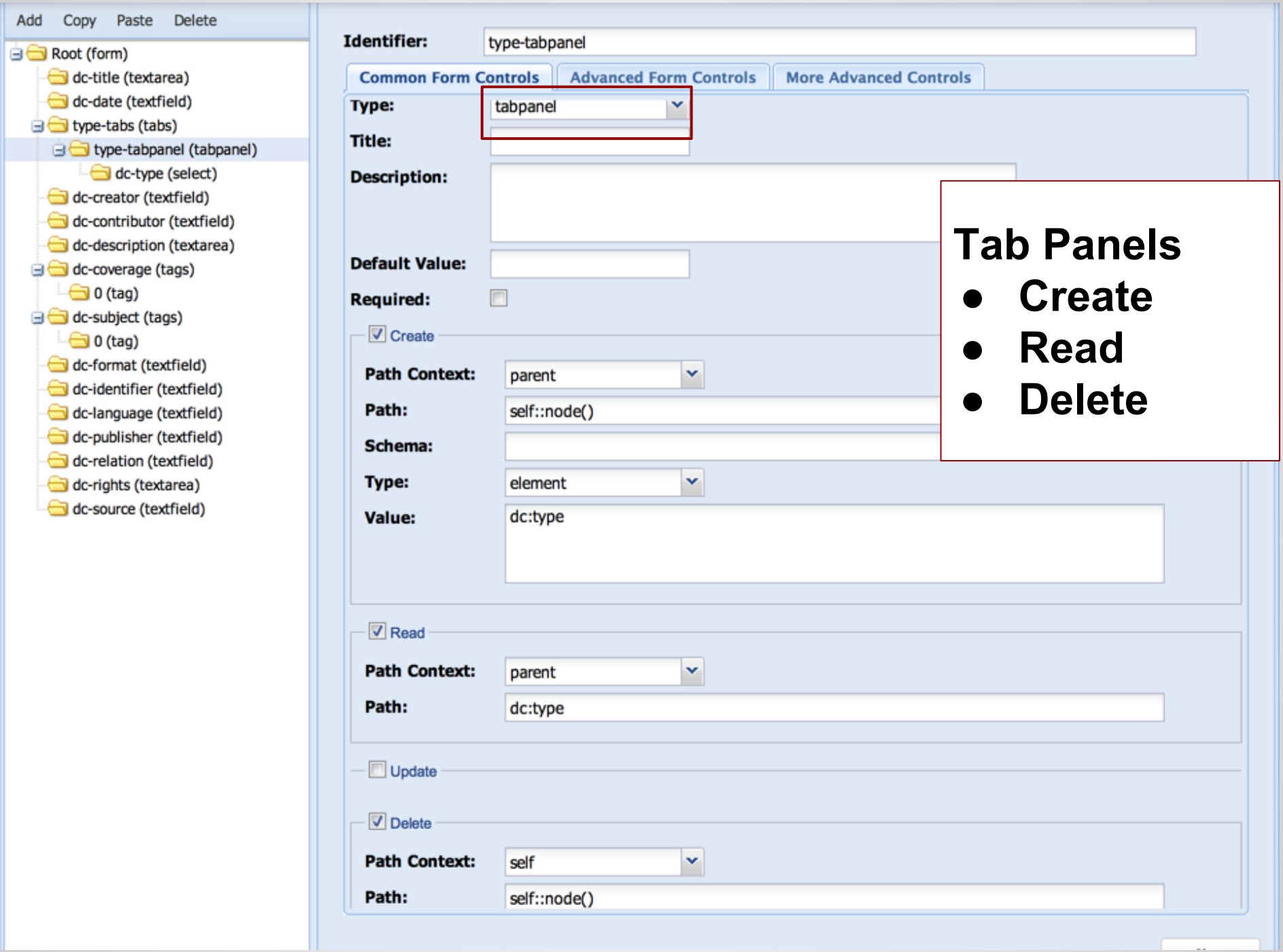


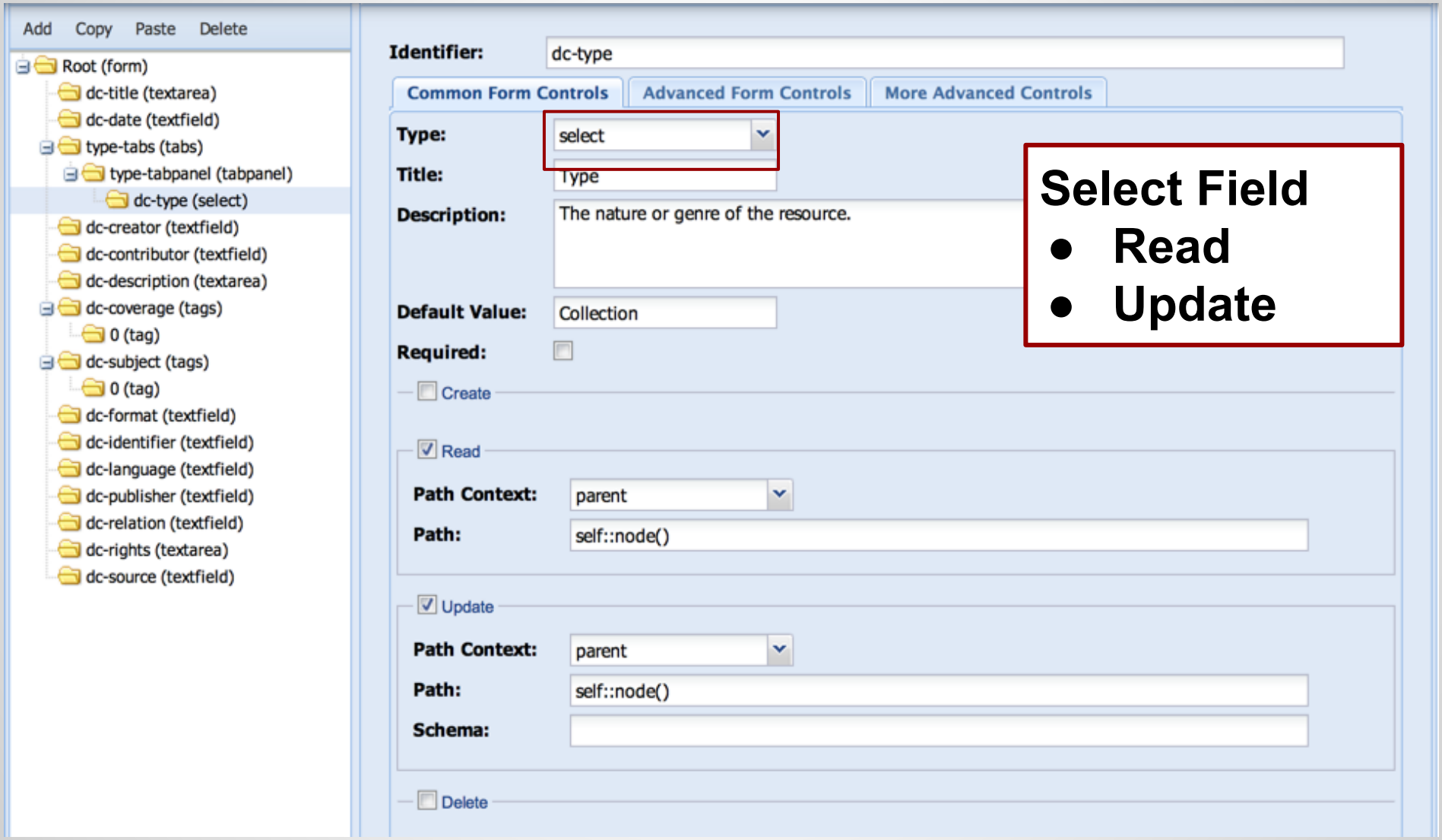
Form Building Best Practices

- Prefer **Parent** context for both **Create** and **Read**
  - *use **self::node** refers to the 'Created' or 'Read' parent form element*
- Prefer **Self** context for **Update** and **Delete**
  - *use **self::node** refers to the 'Read' or 'Created' same form element*
- Tab panel and Tag elements can be **Deleted**
- Have the hierarchy of form fields match the XML document it creates.

Your form can have multiple types … you can use tab panels.




Add
Copy
Paste
Delete
Root (form)
dc-title (textarea)
dc-date (textfield)
type-tabs (tabs)
type-tabpanel (tabpanel)
dc-type (select)
dc-creator (textfield)
dc-contributor (textfield)
dc-description (textarea)
dc-coverage (tags)
0 (tag)
dc-subject (tags)
0 (tag)
dc-format (textfield)
dc-identifier (textfield)
dc-language (textfield)
dc-publisher (textfield)
dc-relation (textfield)
dc-rights (textarea)
dc-source (textfield)
Identifier:
type-tabpanel
Common Form Controls
Advanced Form Controls
More Advanced Controls
Type:
tabpanel
Title:
Description:
Default Value:
Required:
Create
Path Context:
parent
Path:
self::node()
Schema:
Type:
element
Value:
dc:type
Read
Path Context:
parent
Path:
dc:type
Update
Delete
Path Context:
self
Path:
self::node()
Tab Panels
Create
Read
Delete


Add
Copy
Paste
Delete
Root (form)
dc-title (textarea)
dc-date (textfield)
type-tabs (tabs)
type-tabpanel (tabpanel)
dc-type (select)
dc-creator (textfield)
dc-contributor (textfield)
dc-description (textarea)
dc-coverage (tags)
0 (tag)
dc-subject (tags)
0 (tag)
dc-format (textfield)
dc-identifier (textfield)
dc-language (textfield)
dc-publisher (textfield)
dc-relation (textfield)
dc-rights (textarea)
dc-source (textfield)
Identifier:
dc-type
Common Form Controls
Advanced Form Controls
More Advanced Controls
Type:
select
Title:
Type
Description:
The nature or genre of the resource.
Default Value:
Collection
Required:
Create
Read
Path Context:
parent
Path:
self::node()
Update
Path Context:
parent
Path:
self::node()
Schema:
Delete
Select Field
Read
Update

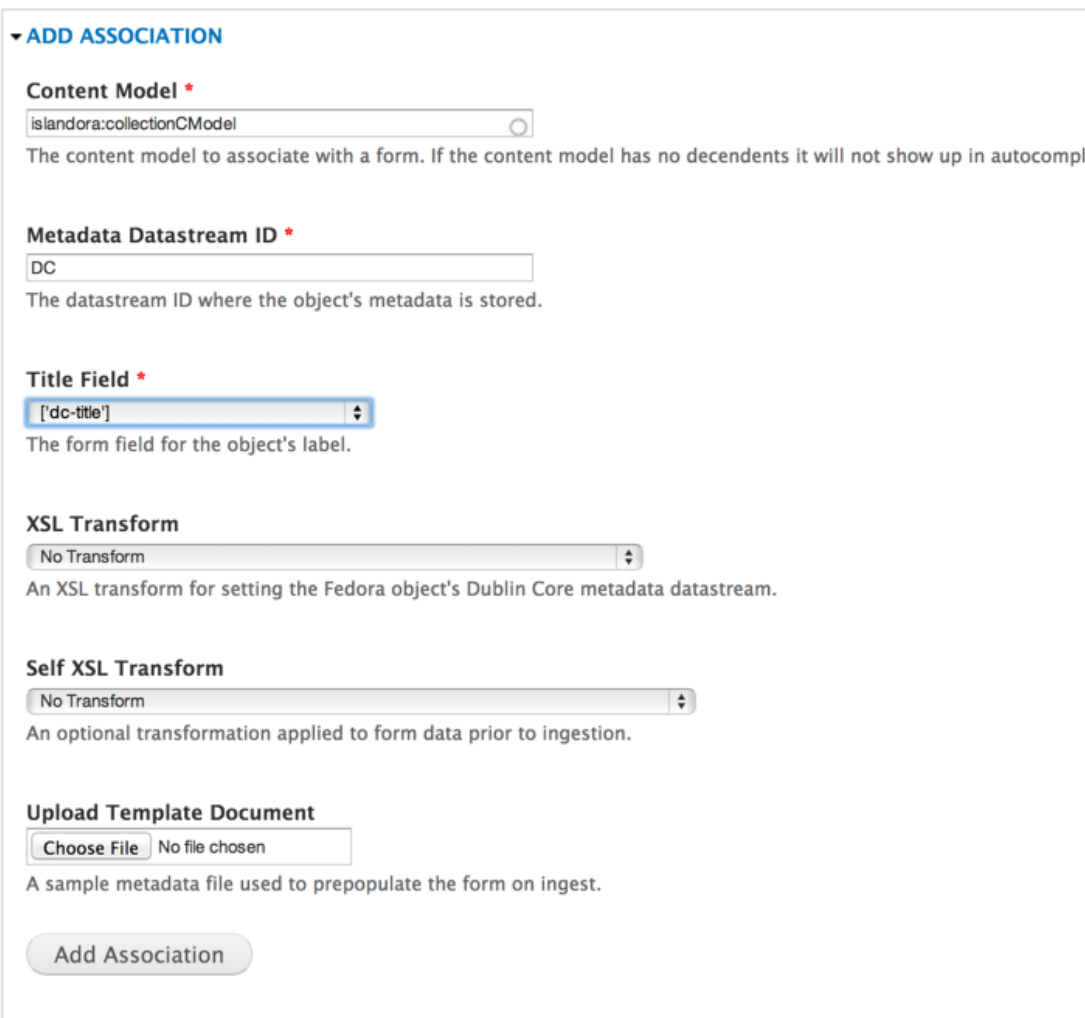
Associate the Form

admin/islandora/xmlform

and locate your form and select the Associate option.

Select the appropriate Content Model.

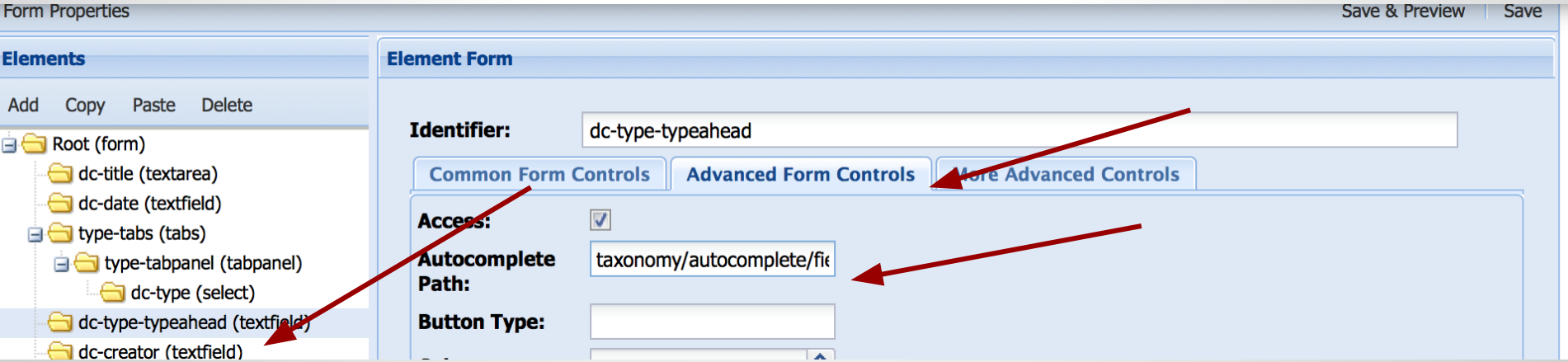
Specify the DSID of the metadata.



Tips and Tricks

Integrating Drupal Taxonomies

- add your taxonomy terms
- add the field to a Drupal content type
- add the field to the form's Autocomplete path
- works with both textfield and tag(?) fields
- After a Google post, Jennifer Eustis wrote up "Create Taxonomy for XML Form".



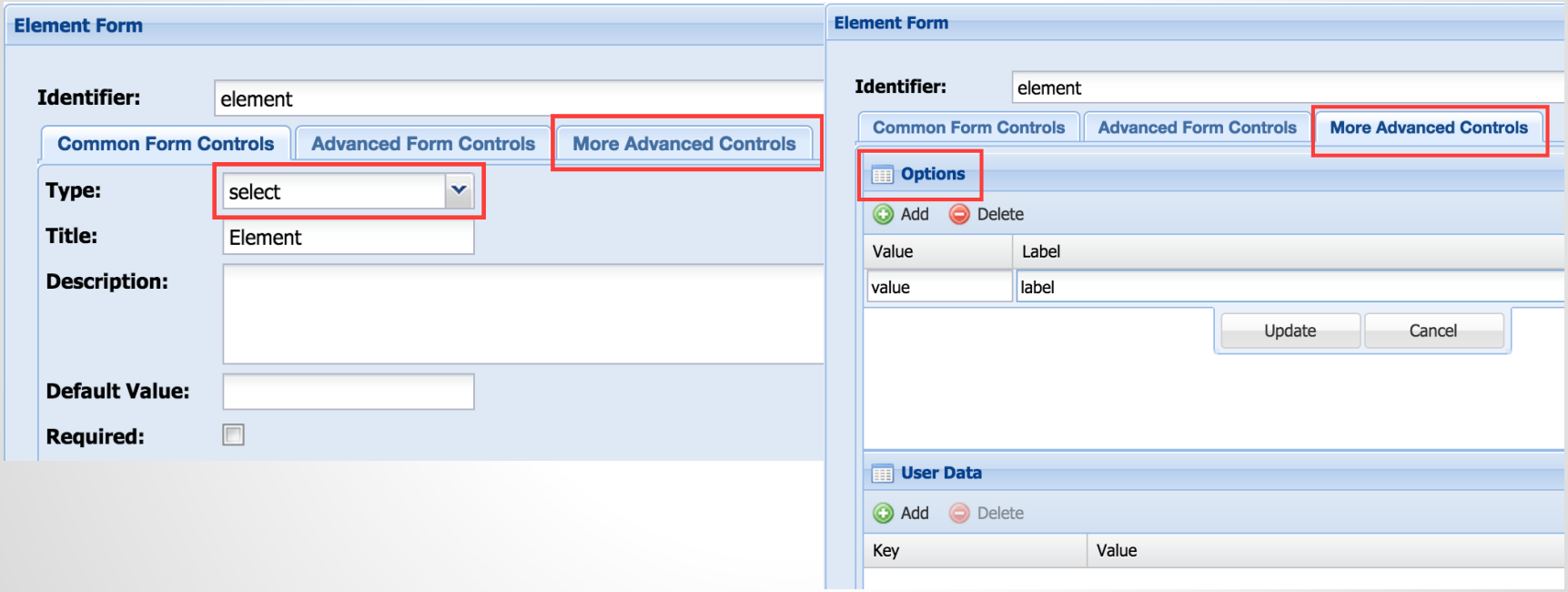
Modifying Parent Value

- Modify parent element
- Repeating element / attribute pair

https://github.com/Islandora-Labs/islandora_ingest_forms/blob/master/Examples/Repeating_Element_with_Attributes_Example_Form.xml

Populating Select Field

For a few setting them manually is fine:



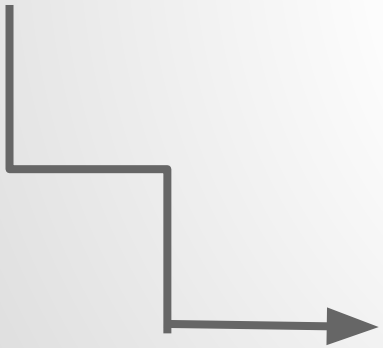
Populating Select Field

For large numbers, export and edit directly in Oxygen.

```
<children>
  <element name="element">
    <properties>
      <type>select</type>
      <access>TRUE</access>
      <collapsed>FALSE</collapsed>
      <collapsible>FALSE</collapsible>
      <disabled>FALSE</disabled>
      <executes_submit_callback>FALSE</executes_submit_callb
      <multiple>FALSE</multiple>
      <options>
        <index key="abr">Abridger</index>
        <index key="act">Actor</index>
        <index key="adp">Adapter</index>
        <index key="rcp">Addressee</index>
        <index key="anl">Analyst</index>
        <index key="anm">Animator</index>
        <index key="ann">Annotator</index>
        <index key="apl">Appellant</index>
        <index key="ape">Appellee</index>
        <index key="app">Applicant</index>
        <index key="arc">Architect</index>
        <index key="arr">Arranger</index>
        <!-- ... -->
      </options>
```

Dealing with Order Based Schemas

Create a Self XSL Transform



[MADS example XSLT]

ADD ASSOCIATION

Content Model *

islandora:personCModel

The content model to associate with a form. If the content model has no decendents it will not show up in autocomplete.

Metadata Datastream ID *

MADS

The datastream ID where the object's metadata is stored.

Title Field

['authority']['display_name']

The form field for the object's label.

XSL Transform

sites/all/modules/islandora_xml_forms/builder/transforms/mads_to_dc.xsl

An XSL transform for setting the Fedora object's Dublin Core metadata datastream.

Self XSL Transform

sites/all/modules/islandora_xml_forms/builder/self_transforms/cleanup_and_order_mads.xsl

An optional transformation applied to form data prior to ingestion.

Upload Template Document

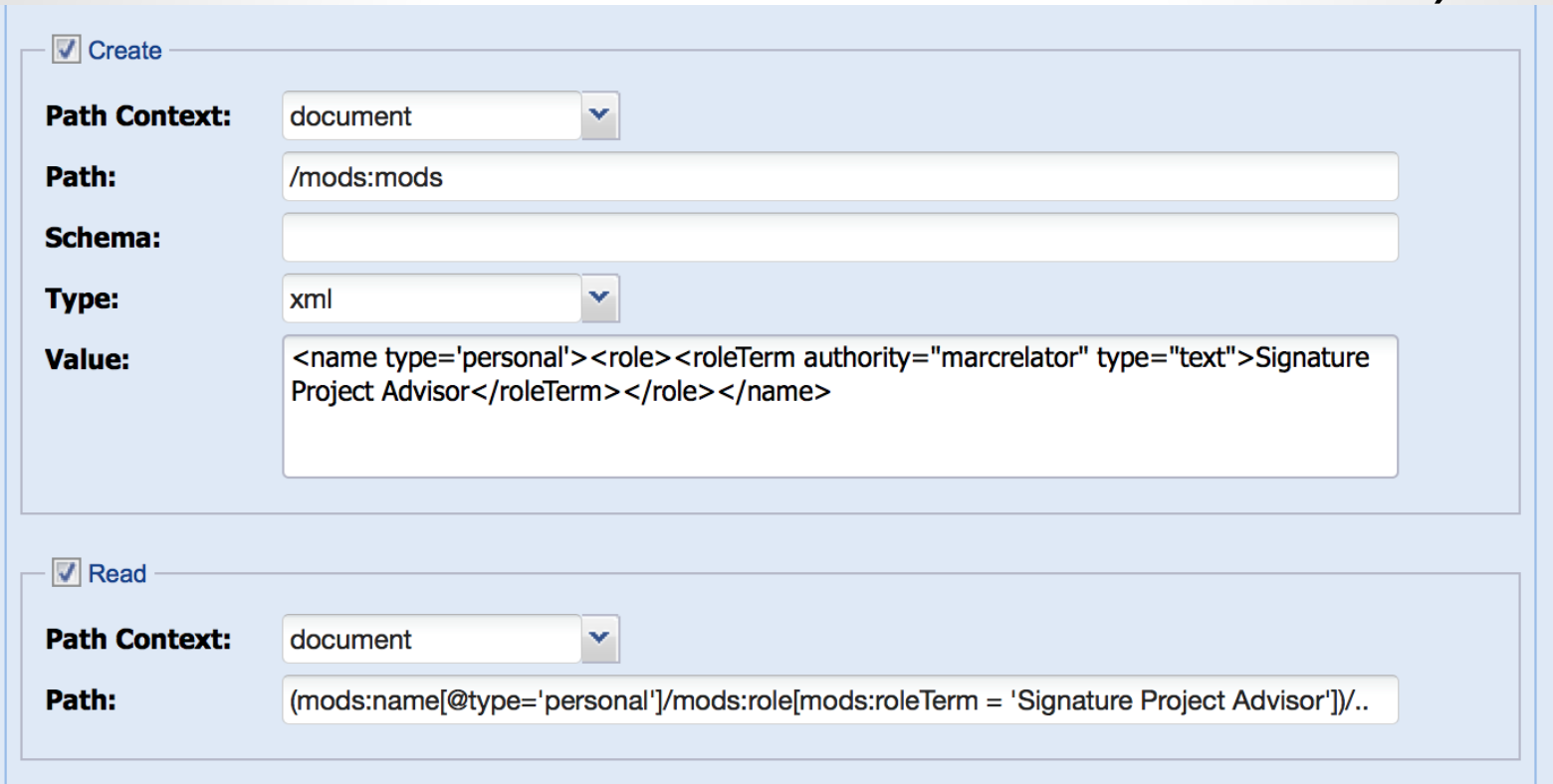
Browse… No file selected.

A sample metadata file used to prepopulate the form on ingest.

Add Association

Selecting Specific Elements

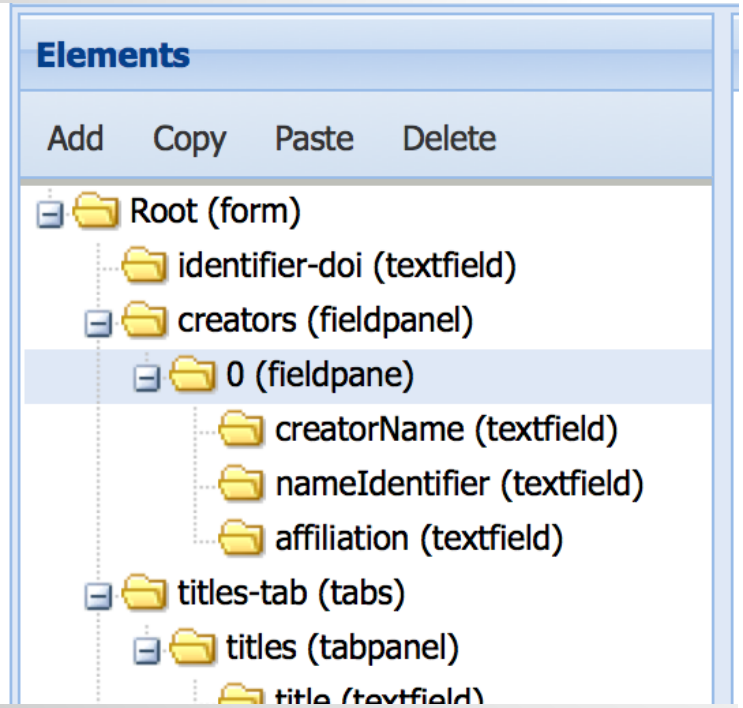
If you have a number of names(tabpanel), but they have different roles (eg. Signature Project Advisor, Examination Committee Member)

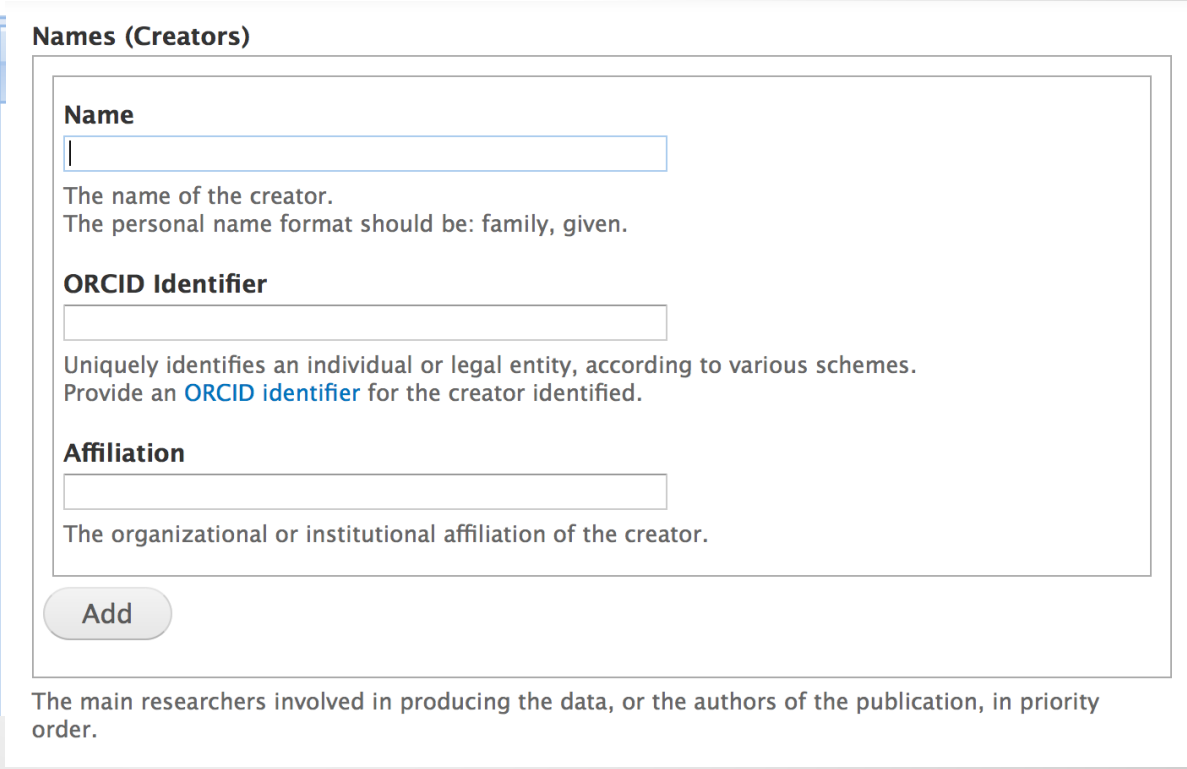


Contrib Modules

Islandora Form Fieldpanel

- An alternative to the tabpanels/tabs from the form builder.
  - https://github.com/Islandora/islandora_form_fieldpanel





Islandora Authority

This module adds **two form fields** and a **menu path** used for autocompletion used for autocompletion on one of them.

islandora_authority_textfield: A text field which can autocomplete on a Solr field.

islandora_authority_hidden: A hidden element which can be autocompleted, as a result of a search.

https://github.com/discoverygarden/islandora_authority

What XML Form Builder Can’t Do

- checkboxes don’t work
- radio buttons don’t work
- date doesn’t work, but datepicker does … so use datepicker.
- hierarchical selects
- and more …

What’s safe: textfield, textarea, tabs, tabpanel, tags, tag, select, fieldset, markup, hidden, datepicker...

Drupal Form API (Form Controls)

If you hover over any of the fields in the formbuilder they'll provide some documentation about them, which can be found here:

http://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7

All the fields presented correspond to settings provided by Drupal.

The Future of Form Builder in i7.2/F4

- *The Future of Forms in Islandora*
- F4IG meeting 9 minutes
  - Summary: do we port what exists now? do we kill and create something new? Or make a hybrid thing?
- XML Form use cases document

Questions?