

# Creating a Resource Type, Part 4: Editing

## Table of contents

1 Adding the Relax NG Schema.....	2
2 Enabling the One-Form Editor.....	2

In this section we make our person documents editable.

## 1. Adding the Relax NG Schema

When we allow to edit documents, we have to make sure that the resulting XML conforms to the FoaF specification. For this purpose we provide a [Relax NG](http://relaxng.org) (<http://relaxng.org>) schema for the FoaF person XML format. Here's a simple schema, created with [Trang](http://www.thaiopensource.com/relaxng/trang.html) (<http://www.thaiopensource.com/relaxng/trang.html>), which - according to the resource type declaration - has to be located at `$MODULE_HOME/resources/schemas/foaf.rng`:

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
  <start>
    <element name="rdf:RDF">
      <element name="foaf:Person">
        <attribute name="rdf:ID"><data type="NCName"/></attribute>
        <element name="foaf:title"><text/></element>
        <element name="foaf:givenname"><text/></element>
        <element name="foaf:family_name"><text/></element>
        <element name="foaf:mbox">
          <attribute name="rdf:resource">
            <data type="anyURI"/>
          </attribute>
        </element>
        <element name="foaf:phone">
          <attribute name="rdf:resource">
            <data type="NMTOKEN"/>
          </attribute>
        </element>
        <element name="foaf:workplaceHomepage">
          <attribute name="rdf:resource">
            <data type="anyURI"/>
          </attribute>
        </element>
      </element>
    </element>
  </start>
</grammar>
```

## 2. Enabling the One-Form Editor

In the first step we don't want to invest much time, so we just enable the one-form editor. We have to add the corresponding menu item to our `menu.xsp` file. But since we want to enable it only for person documents, we have to test if the currently displayed document has the *person* resource type. Here's the extended `menu.xsp` file - note that a new namespace declaration (`xmlns:input`) has been added:

```
<?xml version="1.0"?>
<xsp:page
  language="java"
  xmlns:xsp="http://apache.org/xsp"
  xmlns:i18n="http://apache.org/cocoon/i18n/2.1"
  xmlns:uc="http://apache.org/cocoon/lenya/usecase/1.0"
  xmlns:input="http://apache.org/cocoon/xsp/input/1.0"
  xmlns="http://apache.org/cocoon/lenya/menuubar/1.0">

  <xsp:structure>
    <xsp:include>org.apache.lenya.cms.publication.Document</xsp:include>
  </xsp:structure>

  <menu>
    <menus>
```

```
<menu il8n:attr="name" name="File">
  <block areas="site authoring">
    <item uc:usecase="sitemanagement.create" href="?doctype=person">
      <il8n:translate>
        <il8n:text>New ... Document</il8n:text>
        <il8n:param>Person</il8n:param>
      </il8n:translate>
    </item>
  </block>
</menu>

<menu il8n:attr="name" name="Edit">
  <xsp:logic>
    try {
      Object doc = <input:get-attribute module="page-envelope"
        as="object" name="document"/>;
      if (doc instanceof Document &&& ((Document) doc).exists()) {
        String doctype = <input:get-attribute module="page-envelope"
          as="string" name="document-type"/>;
        if ("person".equals(doctype)) {
          <block areas="authoring">
            <item uc:usecase="editors.oneform" href="?">
              <il8n:text>With one Form</il8n:text>
            </item>
          </block>
        }
      }
    } catch (Exception e) {
      throw new ProcessingException("Error during menu generation: ", e);
    }
  </xsp:logic>
</menu>

</menus>
</menu>
</xsp:page>
```

Now you should be able to edit your person document with the one-form editor.

In the [next section](#) ([../..../docs/2\\_0\\_x/tutorials/resourcetype/part5.html](#)) we explain how to make person documents editable with the BXE WYSIWYG editor.