

Migrating Books from Older Versions of Islandora

If you are upgrading to Islandora 6.x-13.1.x from the previous release Islandora 6.x-12.2.x, these steps are not necessary.

Migrate Your Books with the Migrate Books Module

The Migrate Books module is packaged with the Books Solution Pack. You can only use this module as the Admin user (user 1). Once it has been installed and enabled a menu link will appear in your Navigation menu: Add new Islandora RDF relationships to existing books. Clicking this link will rewrite the RDF relationships for all existing books in your repository to include the necessary relationships for the new Book Solution Pack.

This process is destructive (i.e. it will overwrite existing RDF Datastreams), so any customizations will be lost. In most cases there will not be any customizations, however. This process can be potentially time consuming, especially for large book collections.

Updating GSearch and Solr

1. Locate and edit **demoFoxmlToSolr.xslt** (usually located in **/usr/local/fedora/tomcat/webapps/fedoragsearch/WEB-INF/classes/config/index/gsearch_solr/**)

2. Add the following lines to index the RELS-EXT Datastream:

```
<xsl:for-each select="foxml:datastream[@ID='RELS-EXT']/foxml:datastreamVersion[last()]/foxml:xmlContent//rdf:Description/*">
  <field>
    <xsl:attribute name="name">
      <xsl:value-of select="concat('rels.', local-name())"/>
    </xsl:attribute>
    <xsl:choose>
      <xsl:when test="@rdf:resource"><!-- Deal with URIs/resources -->
        <xsl:value-of select="@rdf:resource"/>
      </xsl:when>
      <xsl:otherwise><!-- otherwise, assume it's a literal -->
        <xsl:value-of select="text()"/>
      </xsl:otherwise>
    </xsl:choose>
  </field>
</xsl:for-each>
```

3. Next, add the following lines to index the MODS from parent books:

```
<!-- get the mods of the parent book if this object is a pageCModel object-->

<xsl:variable name="thisCurrentCModel">
  <xsl:value-of select="foxml:datastream[@ID='RELS-EXT']/foxml:datastreamVersion
[last()]/foxml:xmlContent//rdf:Description/fedora-model:hasModel/@rdf:resource"/>
</xsl:variable>
<xsl:if test="$thisCurrentCModel = 'info:fedora/islandora:pageCModel'">
  <!-- get the parent pid-->
  <xsl:variable name="Book_Pid">
    <xsl:value-of select="foxml:datastream[@ID='RELS-EXT']/foxml:
datastreamVersion[last()]/foxml:xmlContent//rdf:Description/rel:isMemberOf/@rdf:
resource"/>
  </xsl:variable>
  <field>
    <xsl:attribute name="name">PARENT_pid</xsl:attribute>
    <xsl:value-of select="substring($Book_Pid,13)"/>
  </field>
  <xsl:variable name="PARENT_MODS"
    select="islandora-exts:getXMLDatastreamASNodeList(substring($Book_Pid,13),
$REPOSITORYNAME, 'MODS', $FEDORASOAP, $FEDORAUSER, $FEDORAPASS, $TRUSTSTOREPATH,
$TRUSTSTOREPASS)"/>

    <!--<xsl:variable name="PARENT_MODS" select="document(concat($PROT, '://',
$LOCALFEDORAUSERNAME, ':', $LOCALFEDORAPASSWORD, '@',
  $HOST, ':', $PORT, '/fedora/objects/', substring($Book_Pid,13), '
/datastreams/', 'MODS', '/content'))"/>-->
    <xsl:for-each select="$PARENT_MODS//mods:title">
      <field>
        <xsl:attribute name="name">
          <xsl:value-of select="concat('PARENT_', local-name())"/>
        </xsl:attribute>
        <xsl:value-of select="normalize-space(text())"/>
      </field>
    </xsl:for-each>
    <xsl:for-each select="$PARENT_MODS//mods:originInfo/mods:dateIssued">
      <field>
        <xsl:attribute name="name">
          <xsl:value-of select="concat('PARENT_', local-name())"/>
        </xsl:attribute>
        <xsl:value-of select="normalize-space(text())"/>
      </field>
    </xsl:for-each>
    <xsl:for-each select="$PARENT_MODS//mods:genre">
      <field>
```

```

        <xsl:attribute name="name">
            <xsl:value-of select="concat('PARENT_', local-name())"/>
        </xsl:attribute>
        <xsl:value-of select="normalize-space(text())"/>
    </field>
</xsl:for-each>
<xsl:for-each select="$PARENT_MODS//mods:subject/mods:name/mods:namePart/*">
    <xsl:if test="text() [normalize-space(.) ]">
        <!--don't bother with empty space-->
        <field>
            <xsl:attribute name="name">
                <xsl:value-of select="concat('PARENT_', 'subject')"/>
            </xsl:attribute>
            <xsl:value-of select="normalize-space(text())"/>
        </field>
    </xsl:if>
</xsl:for-each>
<xsl:for-each select="$PARENT_MODS//mods:topic">
    <xsl:if test="text() [normalize-space(.) ]">
        <!--don't bother with empty space-->
        <field>
            <xsl:attribute name="name">
                <xsl:value-of select="concat('PARENT_', 'topic')"/>
            </xsl:attribute>
            <xsl:value-of select="normalize-space(text())"/>
        </field>
    </xsl:if>
</xsl:for-each>

<xsl:for-each select="$PARENT_MODS//mods:geographic">
    <xsl:if test="text() [normalize-space(.) ]">
        <!--don't bother with empty space-->
        <field>
            <xsl:attribute name="name">
                <xsl:value-of select="concat('PARENT_', 'geographic')"/>
            </xsl:attribute>
            <xsl:value-of select="normalize-space(text())"/>
        </field>
    </xsl:if>
</xsl:for-each>
<xsl:for-each select="$PARENT_MODS//mods:subject/mods:name/mods:namePart/*">
    <xsl:if test="text() [normalize-space(.) ]">
        <!--don't bother with empty space-->
        <field>
            <xsl:attribute name="name">
                <xsl:value-of select="concat('PARENT_', 'subject')"/>
            </xsl:attribute>
            <xsl:value-of select="normalize-space(text())"/>
        </field>
    </xsl:if>
</xsl:for-each>

```

```

        </field>
    </xsl:if>

</xsl:for-each>
<xsl:for-each select="$PARENT_MODS//mods:part/mods:detail/*">
    <xsl:variable name="TYPE">
        <xsl:value-of select="..@type"/>
    </xsl:variable>
    <field>
        <xsl:attribute name="name">
            <xsl:value-of select="concat('PARENT_', $TYPE)"/>
        </xsl:attribute>
        <xsl:value-of select="."/>
    </field>
</xsl:for-each>
</xsl:if>
<!--*****END PAGE PARENT MODS
*****-->

```

The following namespace declarations should be present at the top of the XSLT. If not, you will need to add them:

```

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rel="info:fedora/fedora-system:def/relations-external#"
xmlns:mods="http://www.loc.gov/mods/v3"
xmlns:islandora-exts="xalan://ca.upei.roblib.DataStreamForXSLT"
xmlns:fedora-model="info:fedora/fedora-system:def/model#"
xmlns:fedora="info:fedora/fedora-system:def/relations-external#"

```

4. Locate and edit **schema.xml** (usually located in **/usr/local/fedora/gsearch_solr/solr/conf/**)

5. Verify that OCR is being both indexed and stored. The line should look something like this:

```

<field name="OCR.OCR" type="text" indexed="true" stored="true" multiValued="true"/>

```

6. Restart GSearch (or just restart Fedora) and reindex by going to [*http://your.site:8080/fedoragsearch/rest/*](http://your.site:8080/fedoragsearch/rest/*) (http://your.site:8080/fedoragsearch/rest/*)

a. Click `updateIndex`

b. Click `updateIndex` from `FoXMLFiles`

You should now be able to search within books.