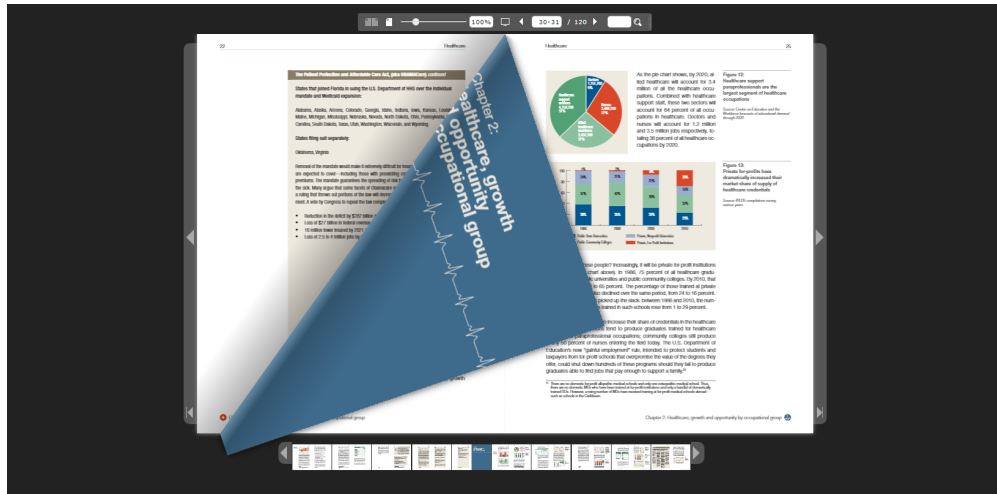


FlexPaper Document Viewer for XMLUI

- 1 What is the FlexPaper Document Viewer for XMLUI
- 2 Technical Implementation Details

What is the FlexPaper Document Viewer for XMLUI

FlexPaper is a web-based PDF Viewer that can be compiled into html4, html5, and flash based assets. Among the [Document Viewer Integration](#) options documented on the DSpace site, this solution is unique in providing an html5 solution with page-turning animation.



This solution requires a licensed copy of FlexPaper Zine. This page will outline the steps required to integrate FlexPaper Zine into the XMLUI user interface.

Example: Dynamic Book Viewer

 [DigitalGeorgetown Example: Dynamic Book Viewer](#)

FlexPaper offers a dynamic viewer and a viewer that utilizes pre-compiled assets (several assets per page). This solution describes how to integrate the dynamic viewer into DSpace.

The dynamic viewer requires no pre-processing of assets which simplifies deployment. End users will need to wait for a PDF file to download fully before the viewer will render.

Use case and high level benefits

The FlexPaper Document Viewer will allow an end user to view a digitized book as a book. This feature is particularly useful for rendering digitized books and digitized rare books.

Installation

- Purchase a FlexPaper Zine Commercial License.
- Download your licensed copy of the software. <http://flexpaper.devaldi.com/download/>
- Create a folder named flexpaper in /dspace/modules/xmlui/src/main/webapp/static
- From your FlexPaper download, copy the css, js, and locale data folders into /dspace/modules/xmlui/src/main/webapp/static/flexpaper
- Create the following in /dspace/modules/xmlui/src/main/webapps/static/:
 - template.html
 - bookview-banner.html,
 - bookview-banner.js
 - Source for these files
 - <https://gist.github.com/terrywbrady/487acdb7d5bf779036bb>
 - Replace \$\$\$\$YOUR_KEY_HERE\$\$\$ with your license key
 - Customize bookview-banner.html to match your theme
- Add the following to /dspace/modules/xmlui/src/main/webapp/sitemap.xmap

```

<map:match pattern="static/**/*.js">
  <map:read type="ConcatenationReader" src="static/{1}.js">
    <map:parameter name="requestQueryString" value="{request:queryString}" />
  </map:read>
</map:match>

<map:match pattern="static/**/*.css">
  <map:read type="ConcatenationReader" src="static/{1}.css">
    <map:parameter name="requestQueryString" value="{request:queryString}" />
  </map:read>
</map:match>

<map:match pattern="*.txt">
  <map:read src="static/{1}.txt" />
</map:match>

```

- Update the template for mets:file in your custom theme. Add a hypertext link to your FlexPaper template
 - See `dspace/dspace-xmlui/src/main/webapp/themes/dri2xhtml/General-Handler.xsl`

```

<xsl:if test="@MIMETYPE='application/pdf'">
  <xsl:variable name="bitstreamurl" select="mets:FLocat[@LOCTYPE='URL']/@xlink:href"/>
  <xsl:variable name="bitstreamid" select="@ID"/>

  <td>
    <xsl:choose>
      <xsl:when test="contains($bitstreamurl, 'isAllowed=n')">
        <xsl:text>Restricted Access</xsl:text>
      </xsl:when>
      <xsl:otherwise>
        <a target="bookView">
          <xsl:attribute name="href">
            <xsl:text>/static/flexpaper/template.html?path=</xsl:text>
            <xsl:value-of select="$bitstreamurl"/>
          </xsl:attribute>
          <!-- TODO: internationalize the image label -->
          <xsl:text>Book View</xsl:text>
        </a>
      </xsl:otherwise>
    </xsl:choose>
    <xsl:text>&#160;</xsl:text>
  </td>

```

Technical Implementation Details

This solution has been designed with a loose integration with XMLUI rendered only at item display.

Rendering Large PDF Files

Large PDF's will take some time to render in the viewer.

You might wish to develop a solution to pre-compile high-demand assets using FlexPaper Desktop or the FlexPaper server compilation components. The sample file template.html contains commented out code that suggests a mechanism to detect the presence of pre-compiled components.

Example: Pre-compiled Book Viewer

 [DigitalGeorgetown Example: Pre-compiled Book Viewer](#)

Rendering Documents with Inconsistent Page Sizes

The dynamic FlexPaper viewer does not gracefully handle documents with inconsistent page sizes. The FlexPaper desktop flash component handles inconsistent page sizes more gracefully than the dynamic viewer.

Example: Pre-compiled Book Viewer with Inconsistent Page Sizes

 [DigitalGeorgetown Example: Pre-compiled Book Viewer with Inconsistent Page Sizes](#)