FlexPaper Integration into XMLUI Brainstorming

Please note that this third-party add on has already been developed and is described in more detail at FlexPaper Document Viewer for XMLUI.

Application Link configuration.

orainstorming for the following ticket:

Unable to locate Jira server for this macro. It may be due to

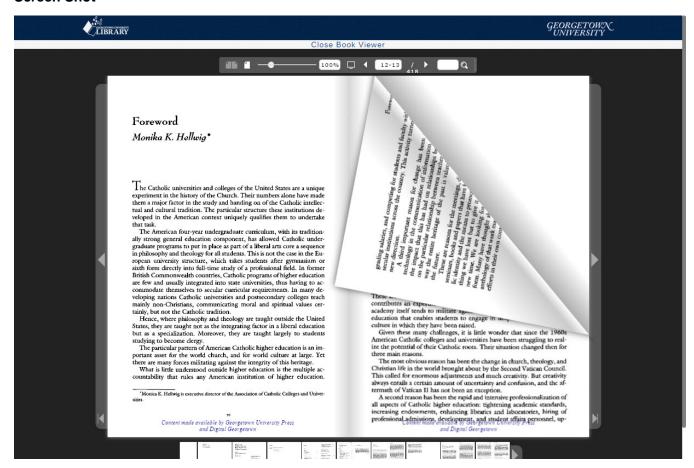
Goal

Provide an html5 PDF document viewer with page-turning capabilities.

Pre-requisites

- FlexPaper Zine Commercial License: http://flexpaper.devaldi.com/download/
- Copy css, js, and locale data into /dspace/modules/xmlui/src/main/webapp/static/flexpaper

Screen Shot



Create a flexpaper viewer landing page

/dspace/modules/xmlui/src/main/webapp/static/flexpaper/template.html

This solution described below assumes that no pre-compiled assets exist. A modified version of this script would be needed to open up a pre-compiled asset.

```
<style type="text/css" media="screen">
     html, body { height:100%; }
     body { margin:0; padding:0; overflow:auto; }
      #flashContent { display:none; }
    </style>
    <link rel="stylesheet" type="text/css" href="css/flexpaper.css?r57627" />
    <script type="text/javascript" src="js/jquery.min.js?r57627"></script>
    <script type="text/javascript" src="js/flexpaper_handlers.js?r57627"></script>
    <script type="text/javascript">
    $(document).ready(function(){
     var loc = document.location.search;
      //Pull the path attribute from the URL
     loc = loc.replace(/.*path=/,"");
      //Load the dynamic html5 viewer from FlexPaper.
      //The full pdf must be downloaded to the browser before the document will be rendered
     $('#documentViewer').FlexPaperViewer(
        { config : {
         PDFFile : loc,
         RenderingOrder : 'html5',
         key : '<<License key goes here>>',
       }}
      );
      //define a callback to be run once the document viewer has loaded
      $('#documentViewer').bind('onDocumentLoaded',function(e,totalPages){
       //this callback is executed before the rendering has completed
        //{
m therefore} insert a short time delay before hiding the message at the top of the screen
       setTimeout(function(){
         $("#message").hide();
       }, 1000);
     });
    });
    </script>
   </head>
   <body>
     <h4 id="message">The document viewer is being prepared. Please be patient while your document is loaded.</h4>
     <div id="documentViewer" class="flexpaper_viewer" style="position:absolute;;width:100%;height:100%;</pre>
background-color:#222222;;"></div>
   </body>
 </html>
```

XMLUI Theme Change

Modify the following template rule to include a link to the book viewer <xsl:template match="mets:file">.

```
<xsl:variable name="bitstreamurl" select="mets:FLocat[@LOCTYPE='URL']/@xlink:href"/>
<xsl:variable name="bitstreamid" select="@ID"/>
<xsl:choose>
   <xsl:when test="contains($bitstreamurl, 'isAllowed=n')"/>
    <xsl:when test="@MIMETYPE='application/pdf'">
      <a target="bookView">
       <xsl:attribute name="href">
         <xsl:text>/static/flexpaper/template.html?path=</xsl:text>
          <xsl:value-of select="$bitstreamurl"/>
       </xsl:attribute>
       <xsl:text>Book View</xsl:text>
     </a>
    </xsl:when>
  </rd></rd></rd>
 <xsl:text>&#160;</xsl:text>
```

Sitemap.xmap changes

The following changes may be needed to ensure that static resources within the flexpaper folder are loaded.

Pre-compilation Process

For very large PDF's that will be accessed frequently, it might make sense to pre-compile the document viewer.

This solution presumes that pre-compiled assets have been made accessible under the following path.

/bookview/<handle>/<seq>/index.html

```
Add the following code to template.xml (above)
//Isolate the handle and sequence number from the bitstream path
var matchArr = /\/bitstream\/handle\/([\d\.]+)\/([\d\.]+)\/.+\.pdf\?sequence=(\d+)/.exec(loc);
if (matchArr != null) {
  if (matchArr.length > 3) {
    //pre-compiled assets must conform to the following structure in order to be opened as an
    //alternative
    var precomp = "/bookview/" + matchArr[1] + "/" + matchArr[2] + "/" +
         matchArr[3] + "/index.html";
    //\mbox{{\sc make}} ajax call to determine if a pre-compiled asset exists.
    //If so, redirect to that page (which will terminate the load of the dynamic viewer)
    //Otherwise the client-side html5 viewer will load as normal.
    $.get(precomp, function(){
      //if the get operation is successful, the following callback will be executed
      //note that the remainder of this script will process
      document.location=precomp;
    });
  }
```