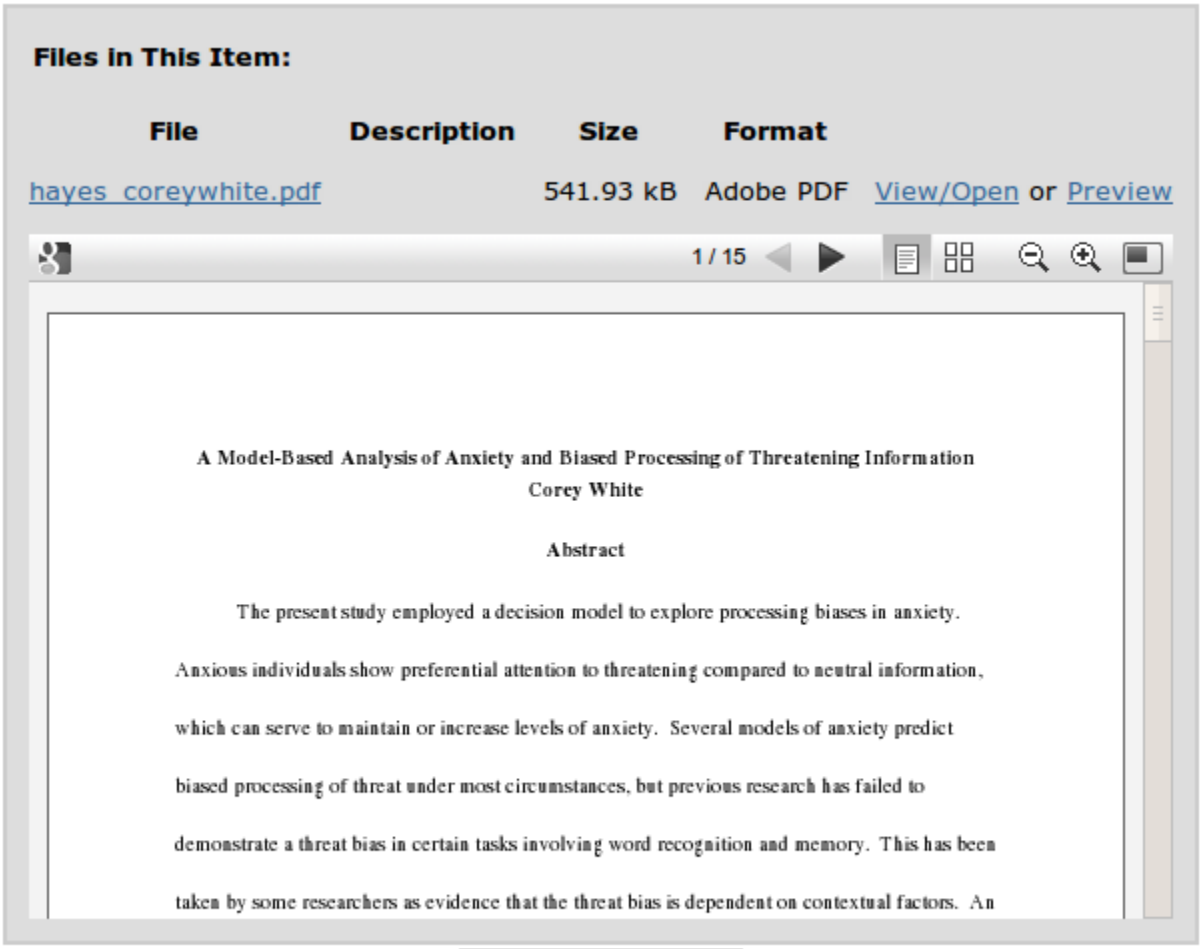
Document Preview with Google Docs viewer

What is it?:

Document Preview is embedding a document display mechanism into the item display page. It makes the distance between the item in your repository, and the end user smaller as PDF's no longer require the user to have a PDF Viewer installed on their computer. And its faster, as the document begins loading in the page.

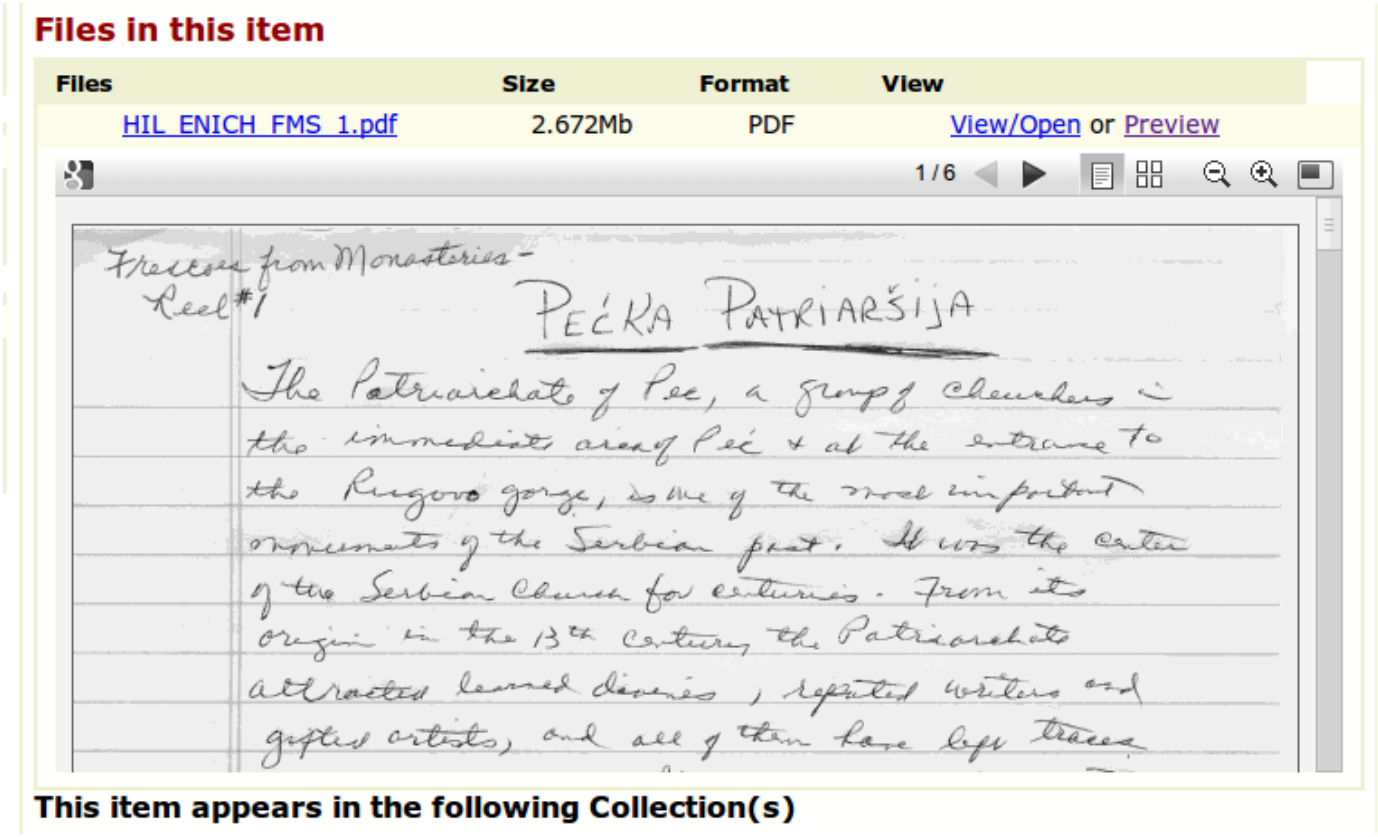
Before installing, be sure to check out Google's Terms of Service, and check that your usage is consistent with your institution's licensing policies, and is consistent with Google's usage policies. https://docs.google.com/viewer/TOS?hl=en. Additionally, there are limitations to how far this method can be customized, in which case another solution, such as @mire's document streaming would make more sense.

Document Preview in JSPUI



Show full item record

Document Preview in XMLUI



Document Preview with Google Docs viewer can be installed on either XMLUI or JSPUI, as it is just embedding a web service on your site.

JSPUI Instuctions

Files:

- dspace-jspui/dspace-jspui-api/src/main/java/org/dspace/app/webui/jsptag/ItemTag.java (Item Display Page)
- dspace/modules/jspui/src/main/webapp/utils.js (Custom Javascript)

Instructions:

1. Patch ItemTag.java to add the preview button and the hidden row that contains the iframe.

```
--- dspace-jspui/dspace-jspui-api/src/main/java/org/dspace/app/webui
/jsptag/ItemTag.java
+++ dspace-jspui/dspace-jspui-api/src/main/java/org/dspace/app/webui
/jsptag/ItemTag.java
@@ -927,10 +980,24 @@
                                                        .
getLocalizedMessage(

pageContext,
                                                                "org.
dspace.app.webui.jsptag.ItemTag.view")
-                                               + "</a></td></tr>");
+                                               + "</a>");
```

```
+                                                   String bsUrl = "
/bitstream/"
+                                                           + item.
getHandle()
+                                                           + "/" +
bitstreams[k].getSequenceID()
+                                                           + "/" +
UIUtil.encodeBitstreamName(bitstreams[k].getName(), Constants.
DEFAULT_ENCODING);
+                                                   out.print(" or <a
href=\"#preview\" onclick=\"setPreviewSource('"+bsUrl+"');\">Preview<
/a>");
+                                                   out.print("</td><
/tr>");
                                                }
                                            }
                                        }
+                                   // END, can put the doc viewer here.
+                                   out.println("<tr><td colspan=5>");
+                                   out.println("<a name='preview'");
+                                   out.println("<div id=\"preview\" style=\"
display:none;\"> " +
+                                                                   "<iframe
id=\"embed\" src=\"\" width=\"100%\" height=\"342px\" style=\"border:
none;\"></iframe>" +
+                                                                   "<
/div>");
+                                   out.println("</td></tr>");
+
                               }

                               out.println("</table>");
```

2. Copy utils.js from dspace-jspui/dspace-jspui-webapp/src/main/webapp/utils.js to dspace/modules/jspui/src/main/webapp/utils.js
3. Patch utils.js to add some additional needed functions.

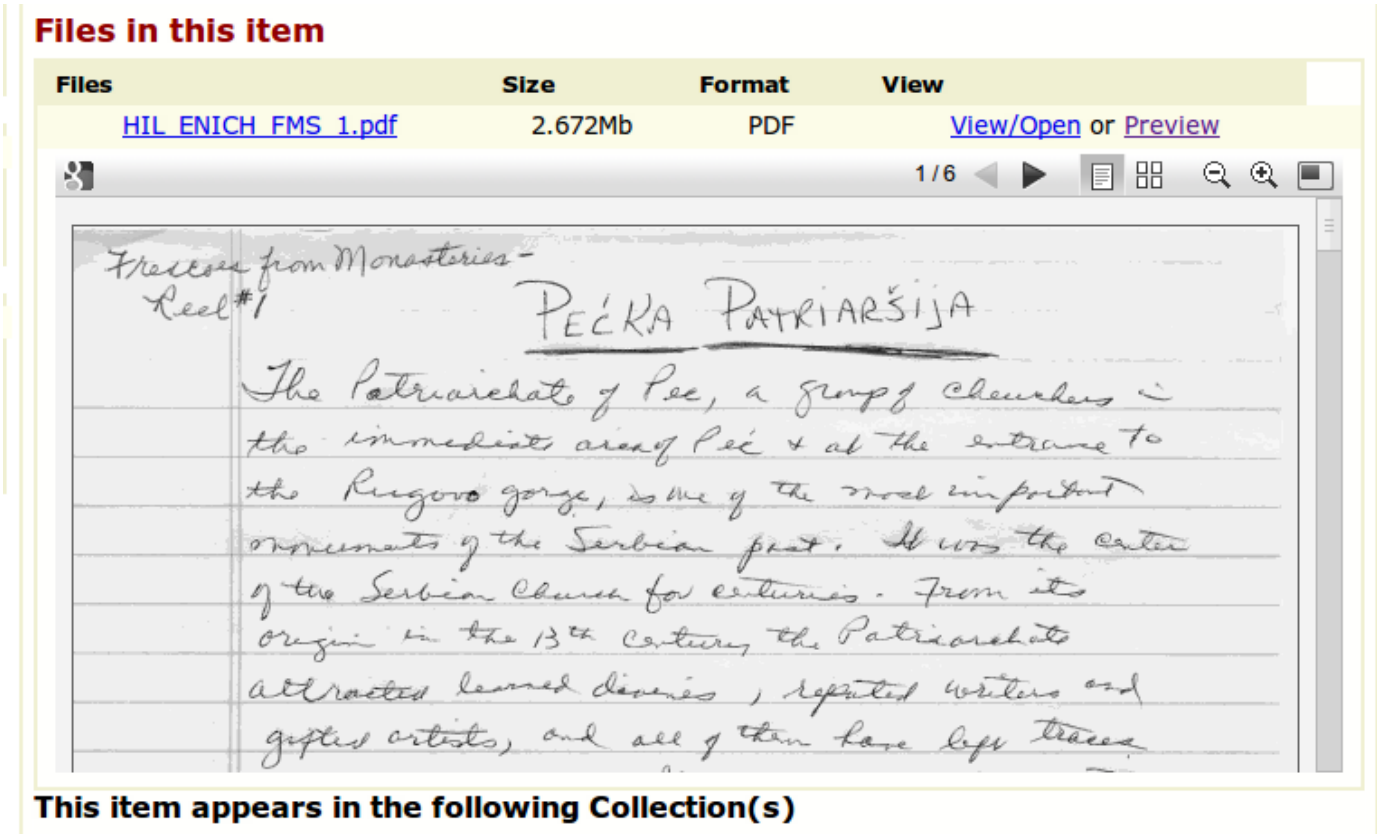
```
--- dspace/modules/jspui/src/main/webapp/utils.js
+++ dspace/modules/jspui/src/main/webapp/utils.js
@@ -301,3 +307,36 @@



+function showHideToggle(divID) {
+       if (document.getElementById && !document.all) {
+               previewBox = document.getElementById(divID);
+               var state = previewBox.style.display;
+               if (state == 'block') {state = 'none';} else {state
= 'block';}
+               previewBox.style.display = state;
+       }
+}
+
```

```
+function urlencode(str) {
+    return escape(str).replace(/\+/g,'%2B').replace(/%20/g, '+').
replace(/\*/g, '%2A').replace(/\//g, '%2F').replace(/@/g, '%40');
+}
+
+String.prototype.endsWith = function(str)
+{
+    return (this.match(str+"$")==str)
+}
+
+function setPreviewSource(source) {
+    if(false) {
+        // if you need to change bitstream link to use a web
viewable server (used in development)
+        source = "https://kb.osu.edu/dspace" + source;
+    }
+
+    if(source.endsWith(".pdf") || source.endsWith(".doc")) {
+        source = urlencode(source);
+        source = "http://docs.google.com/viewer?url="+source+"
&embedded=true";
+    }
+
+    $("#embed").attr("src", source); //requires jQuery
+    showHideToggle('preview');
+    // and toggle
+}
```

1. The javascript functions do make minor use of jQuery, which can easily be added:

```
<script type="text/javascript" src="http://www.google.com/jsapi"><
/script>
<script type="text/javascript">
        google.load("jquery", "1.4");
        google.setOnLoadCallback(function() {
            // Place init code here instead of $(document).ready()
        });
</script>
```

XMLUI Instructions



Best practices are to put customizations into a theme directory, and not to modify the dri2xhtml files.

Additional Javascript function, note that you will need to use jQuery.
Add to dspace/webapps/themes/your-theme/lib/yourtheme.js

```
/**
 * Creates an iframe with the specified source
 * @REQUIRES jQuery
 * @REQUIRES that there exists an element with ID preview-embed
 */
function embeddedPreview(source) {
    if($("#embed").length > 0) {
        //set the source to the location asked for
        $("#embed").attr("src", source);
    } else {
        //Create the embed iframe
        $("#preview-embed").append("<iframe id='embed' src='"+source+"'
width='100%' height='342px' style='border:none;'/>"); //requires jQuery
    }
}
```

XSL overrides. This adds the "or Preview" link to the item downloads section, as well as an empty div hiding in the table.
Add to dspace/webapps/themes/yourtheme/yourtheme.xls

```
<!-- Document Preview with Google Docs viewer -->
    <!-- Generate the bitstream information from the file section -->
```

```
    <xsl:template match="mets:fileGrp[@USE='CONTENT']">
        <xsl:param name="context"/>
        <xsl:param name="primaryBitstream" select="-1"/>

        <h2><i18n:text>xmlui.dri2xhtml.METS-1.0.item-files-head</i18n:
text></h2>
        <table class="ds-table file-list">
            <tr class="ds-table-header-row">
                <th><i18n:text>xmlui.dri2xhtml.METS-1.0.item-files-file<
/i18n:text></th>
                <th><i18n:text>xmlui.dri2xhtml.METS-1.0.item-files-size<
/i18n:text></th>
                <th><i18n:text>xmlui.dri2xhtml.METS-1.0.item-files-format<
/i18n:text></th>
                <th><i18n:text>xmlui.dri2xhtml.METS-1.0.item-files-view<
/i18n:text></th>
                <!-- Display header for 'Description' only if at least one
bitstream contains a description -->
                <xsl:if test="mets:file/mets:FLocat/@xlink:label != ''">
                    <th><i18n:text>xmlui.dri2xhtml.METS-1.0.item-files-
description</i18n:text></th>
                </xsl:if>
            </tr>
            <xsl:choose>
                <!-- If one exists and it's of text/html MIME type, only
display the primary bitstream -->
                <xsl:when test="mets:file[@ID=$primaryBitstream]
/@MIMETYPE='text/html'">
                    <xsl:apply-templates select="mets:file
[@ID=$primaryBitstream]">
                        <xsl:with-param name="context" select="$context"/>
                    </xsl:apply-templates>
                </xsl:when>
                <!-- Otherwise, iterate over and display all of them -->
                <xsl:otherwise>
                    <xsl:apply-templates select="mets:file">
                            <xsl:sort data-type="number" select="boolean(.
/@ID=$primaryBitstream)" order="descending" />
                        <xsl:sort select="mets:FLocat[@LOCTYPE='URL']
/@xlink:title"/>
                        <xsl:with-param name="context" select="$context"/>
                    </xsl:apply-templates>
                </xsl:otherwise>
            </xsl:choose>
<!-- Add the document previewer window -->
            <tr>
                <td colspan='5'>
                    <a name="preview"></a>
                    <div id="preview-embed"/>
                </td>
            </tr>
        </table>
    </xsl:template>
```

```
    <!-- Build a single row in the bitsreams table of the item view page --
>
    <xsl:template match="mets:file">
        <xsl:param name="context" select="."/>
        <tr>
            <xsl:attribute name="class">
                <xsl:text>ds-table-row </xsl:text>
                <xsl:if test="(position() mod 2 = 0)">even </xsl:if>
                <xsl:if test="(position() mod 2 = 1)">odd </xsl:if>
            </xsl:attribute>
            <td>
                <a>
                    <xsl:attribute name="href">
                        <xsl:value-of select="mets:FLocat[@LOCTYPE='URL']
/@xlink:href"/>
                    </xsl:attribute>
                    <xsl:attribute name="title">
                        <xsl:value-of select="mets:FLocat[@LOCTYPE='URL']
/@xlink:title"/>
                    </xsl:attribute>
                    <xsl:choose>
                        <xsl:when test="string-length(mets:FLocat
[@LOCTYPE='URL']/@xlink:title) > 50">
                            <xsl:variable name="title_length" select="
string-length(mets:FLocat[@LOCTYPE='URL']/@xlink:title)"/>
                            <xsl:value-of select="substring(mets:FLocat
[@LOCTYPE='URL']/@xlink:title,1,15)"/>
                            <xsl:text> ... </xsl:text>
                            <xsl:value-of select="substring(mets:FLocat
[@LOCTYPE='URL']/@xlink:title,$title_length - 25,$title_length)"/>
                        </xsl:when>
                        <xsl:otherwise>
                            <xsl:value-of select="mets:FLocat
[@LOCTYPE='URL']/@xlink:title"/>
                        </xsl:otherwise>
                    </xsl:choose>
                </a>
            </td>
            <!-- File size always comes in bytes and thus needs conversion
-->
            <td>
                <xsl:choose>
                    <xsl:when test="@SIZE &lt; 1000">
                        <xsl:value-of select="@SIZE"/>
                        <i18n:text>xmlui.dri2xhtml.METS-1.0.size-bytes<
/i18n:text>
                    </xsl:when>
                    <xsl:when test="@SIZE &lt; 1000000">
                        <xsl:value-of select="substring(string(@SIZE div
1000),1,5)"/>
                        <i18n:text>xmlui.dri2xhtml.METS-1.0.size-kilobytes<
```

```
/i18n:text>
                    </xsl:when>
                    <xsl:when test="@SIZE &lt; 1000000000">
                        <xsl:value-of select="substring(string(@SIZE div
1000000),1,5)"/>
                        <i18n:text>xmlui.dri2xhtml.METS-1.0.size-megabytes<
/i18n:text>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:value-of select="substring(string(@SIZE div
1000000000),1,5)"/>
                        <i18n:text>xmlui.dri2xhtml.METS-1.0.size-gigabytes<
/i18n:text>
                    </xsl:otherwise>
                </xsl:choose>
            </td>
            <!-- Lookup File Type description in local messages.xml based
on MIME Type.
                In the original DSpace, this would get resolved to an
application via
                the Bitstream Registry, but we are constrained by the
capabilities of METS
                and can't really pass that info through. -->
            <td>
              <xsl:call-template name="getFileTypeDesc">
                <xsl:with-param name="mimetype">
                  <xsl:value-of select="substring-before(@MIMETYPE,'/')"/>
                  <xsl:text>/</xsl:text>
                  <xsl:value-of select="substring-after(@MIMETYPE,'/')"/>
                </xsl:with-param>
              </xsl:call-template>
            </td>
            <td>
                <xsl:choose>
                    <xsl:when test="$context/mets:fileSec/mets:fileGrp
[@USE='THUMBNAIL']/
                        mets:file[@GROUPID=current()/@GROUPID]">
                        <a class="image-link">
                            <xsl:attribute name="href">
                                <xsl:value-of select="mets:FLocat
[@LOCTYPE='URL']/@xlink:href"/>
                            </xsl:attribute>
                            <img alt="Thumbnail">
                                <xsl:attribute name="src">
                                    <xsl:value-of select="$context/mets:
fileSec/mets:fileGrp[@USE='THUMBNAIL']/
                                        mets:file[@GROUPID=current()
/@GROUPID]/mets:FLocat[@LOCTYPE='URL']/@xlink:href"/>
                                </xsl:attribute>
                            </img>
                        </a>
                    </xsl:when>
                    <xsl:otherwise>
```

```
                        <a>
                            <xsl:attribute name="href">
                                <xsl:value-of select="mets:FLocat
[@LOCTYPE='URL']/@xlink:href"/>
                            </xsl:attribute>
                            <i18n:text>xmlui.dri2xhtml.METS-1.0.item-files-
viewOpen</i18n:text>
                        </a>
                        <xsl:choose>
                            <xsl:when test="@MIMETYPE='application/pdf'">
                                <xsl:text> or </xsl:text>
                                <a>
                                    <xsl:attribute name="href">
                                        <xsl:text>#preview</xsl:text>
                                    </xsl:attribute>
                                    <xsl:attribute name="onclick">
                                        <xsl:text>embeddedPreview("</xsl:
text>
                                        <xsl:text>http://docs.google.com
/viewer?url=</xsl:text>
<!--Google can't reach a local development machines, so you may want to
code in your production server, where Google downloads the file from-->
                                        <!--<xsl:text>http://example.edu/<
/xsl:text>-->
<!-- Getting bitstreams by ID -- XMLUI: /xmlui/bitstream/id  JSPUI: /jspui
/retrieve -->
                                        <xsl:text>/xmlui/bitstream/id/<
/xsl:text>
                                        <xsl:value-of select="substring
(@ID,6)"/>
                                        <xsl:text>&amp;embedded=true</xsl:
text>
                                        <xsl:text>");</xsl:text>
                                    </xsl:attribute>
                                    Preview
                                </a>
                            </xsl:when>
                        </xsl:choose>
                    </xsl:otherwise>
                </xsl:choose>
            </td>
            <!-- Display the contents of 'Description' as long as at least
one bitstream contains a description -->
            <xsl:if test="$context/mets:fileSec/mets:fileGrp/mets:file
/mets:FLocat/@xlink:label != ''">
                <td>
                    <xsl:value-of select="mets:FLocat[@LOCTYPE='URL']
/@xlink:label"/>
                </td>
            </xsl:if>

        </tr>
    </xsl:template>
```

Essentially this works by "Preview" making a JavaScript call to create an iframe with a specified source (of your pdf bitstream).

NOTE: I originally posted this concept at Document Preview in DSpace, using Google Docs Viewer