```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
  Author: Team 1
  Date:   4/10/2010

  Filename:        PicassosInventory.xsl
  Supporting Files: PicassoProducts.xml, products.xsd
-->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="4.0" />
<xsl:template match = "ChicagoStylePaper">
  <html>
   <head>
     <title>Picasso's Products Inventory</title>
     <style type="text/css">
        &lt;!--  body {
        width: 800px;
         font-family:"Times New Roman", Times, serif;
         margin: 30px;
          }


        a {
           text-decoration: none;
        }
        d {
          font-size: 140%;
        }

        p
        {
     width: 800px;
     font-size: 140%;
      line-height: 2.0;
      text-indent: 0px;
        }

       h1.break
        {
           page-break-before: always;
        }

        h3.figureheader
        {
     width: 800px;
     font-size: 140%;
     line-height: 2.0;
     text-align:left;
     text-indent: 0px;
        }

        h2
        {
     width: 800px;
     font-size: 140%;
     line-height: 2.0;
     text-align:left;
        }
```

```


              h1
              {
          width: 800px;
          font-size: 140%;
          line-height: 2.0;
          text-align: center;
              }

              p.projectmembers
              {
          width: 800px;
          text-align:center;
           text-indent: 0px;
           line-height: 1.0;
              }

              p.bibliography
              {
                  width: 800px;
                  font-family:"Courier New", Times, serif;
                  font-size: 100%;
                  text-indent: 0px;
              }

              p.bibliographyheader
              {
                  width: 800px;
                  font-family:"Courier New", Times, serif;
                  font-size: 100%;
                  text-indent: 0px;
          text-align: center;
          page-break-before: always
              }

        *.break
        {
          page-break-before: always;
        }

        li
        {
          line-height: 2.0;
          font-size: 130%;
        }
              li.bulleteditem_level
              {
           text-indent: 100px;
              }

              li.bulleteditem_level1
              {
           text-indent: 100px;
              }

              li.bulleteditem_level2
              {
```

```
        text-indent: 120px;
           }

           li.bulleteditem_level3
           {
        text-indent: 140px;
           }

           li.bulleteditem_level4
           {
        text-indent: 160px;
           }

           li.bulleteditem_level5
           {
        text-indent: 180px;
           }

           image.figure
           {
        width: 600px;
        height: 450px;
           }

           div.figuremargin
           {
               width: 750px;
               text-align: center;
               margin: 25px;
           }

           div.spacer
           {
        width: 600px;
        height: 70px;
           }

          div.titlespacer
           {
               width: 600px;
               height: 300px;
           }

           div.sectionspacer
           {
               width: 600;
               height: 30px;
           }
           --&gt;
           </style>
    </head>
    <body>
        <!--****************TITLE PAGE SECTION*****************************************-->

        <div class="titlespacer"></div>
           <xsl:variable name="lcletters">abcdefghijklmnopqrstuvwxyz</xsl:variable>
           <xsl:variable name="ucletters">ABCDEFGHIJKLMNOPQRSTUVWXYZ</xsl:variable>
        <xsl:variable name="maintitle" select="TitlePage/Title/Main"/>
```

```
<xsl:variable name="subtitle" select="TitlePage/Title/Sub"/>
<h1><xsl:value-of select="translate($maintitle, $lcletters, $ucletters)"/></h1>
<h2 class="subsectionheader"><xsl:value-of select="translate($subtitle, $lcletters, $ucletters)"/></h2>

<div class="spacer"></div>

<div class="titlepagemargin">
  <xsl:for-each select="TitlePage/Members/Member">
   <xsl:apply-templates select="TitlePage/Members/Member"/>
     <p class="projectmembers"><xsl:value-of select="."/></p>
  </xsl:for-each>
</div>

<div class="spacer"></div>
<!--****************ABSTRACT SECTION******************************************-->
 <h1 class="break"><xsl:value-of select="Abstract/AbstractHeader"/></h1>
 <xsl:variable name="fragmentsVariable" select="Abstract/Fragment"/>
 <xsl:call-template name="DoFragments">
  <xsl:with-param name="fragments" select="$fragmentsVariable"/>
  </xsl:call-template>
<!--CONTENT BODY-->
<!--INTRODUCTION-->
<div class="sectionspacer"/>
<div class="introduction">
 <h1 class="break">Introduction</h1>
  <h2>
  <xsl:value-of select="Body/Introduction/Header"/>
  </h2>
 <xsl:variable name="introductionFragments" select="Body/Introduction/Fragment"/>
 <xsl:call-template name="DoFragments">
  <xsl:with-param name="fragments" select="$introductionFragments"/>
 </xsl:call-template>
</div>
<!--SECTIONS-->
<xsl:for-each select="Body/Section">
  <xsl:apply-templates select="Body/Section"/>
   <div class="sectionspacer"/>
   <h1>
    <xsl:value-of select="./Header"/>
   </h1>
   <!--SUBSECTIONS-->
   <xsl:for-each select="./SubSection">
    <xsl:apply-templates select="./SubSection"/>
    <div class="sectionspacer"/>
    <div class="subsectiondiv">
     <h2>
      <xsl:value-of select="./SubSectionHeader"/>
     </h2>
     <!--FRAGMENTS-->
         <xsl:variable name="subsectionFragments" select="./Fragment"/>
         <xsl:call-template name="DoFragments">
          <xsl:with-param name="fragments" select="$subsectionFragments"/>
          </xsl:call-template>
    </div>
   </xsl:for-each>
</xsl:for-each>

<!-- CONCLUSION-->
```

```
  <div class="sectionspacer"/>
  <div>
   <h1>
    <xsl:value-of select="Body/Conclusion/ConclusionHeader"/>
    </h1>
   <xsl:variable name="conclusionFragments" select="Body/Conclusion/Fragment"/>
   <xsl:call-template name="DoFragments">
    <xsl:with-param name="fragments" select="$conclusionFragments"/>
   </xsl:call-template>
  </div>
  <!--Bibliography-->
  <p class="bibliographyheader">
       Bibliography
  </p>
  <xsl:for-each select="Body/Bibliography/Reference">
   <xsl:apply-templates select="Body/Bibliography/Reference"/>
    <p class="bibliography">
      <xsl:for-each select="./ReferenceSegment">
        <xsl:apply-templates select="./ReferenceSegment"/>
      <xsl:choose>
       <xsl:when test="@italic = 'true' and @bold = 'true'">
         <i><b><xsl:value-of select="."/></b></i>
       </xsl:when>
        <xsl:when test="@italic = 'true'">
         <i><xsl:value-of select="."/></i>
       </xsl:when>
       <xsl:when test="@bold = 'true'">
        <b><xsl:value-of select="."/></b>
       </xsl:when>
       <xsl:otherwise>
                 <xsl:value-of select="."/>
       </xsl:otherwise>
      </xsl:choose>
      </xsl:for-each>
    </p>
  </xsl:for-each>
 </body>
</html>
</xsl:template>
<xsl:template name="DoFragments">
 <xsl:param name="fragments"/>
 <xsl:for-each select="$fragments">
    <xsl:variable name="paragraph" select="./Paragraph"/>
   <xsl:variable name="figure" select="./Figure"/>
   <xsl:variable name="bulletedlist" select="./BulletedList"/>

  <xsl:if test="$paragraph">
   <p>
    <xsl:variable name="paragraphWithOpeningItalic">
       <xsl:call-template name="replace-string">
            <xsl:with-param name="text" select="./Paragraph"/>
            <xsl:with-param name="replace" select="'[ITALIC]'"/>
            <xsl:with-param name="with" select="'&lt;i&gt;'"/>
           </xsl:call-template>
     </xsl:variable>
     <xsl:variable name="paragraphWithOpeningBold">
       <xsl:call-template name="replace-string">
            <xsl:with-param name="text" select="$paragraphWithOpeningItalic"/>
```

```
                           <xsl:with-param name="replace" select="'[BOLD]'"/>
                           <xsl:with-param name="with" select="'&lt;b&gt;'"/>
                        </xsl:call-template>
            </xsl:variable>
            <xsl:variable name="paragraphWithClosingBold">
                <xsl:call-template name="replace-string">
                           <xsl:with-param name="text" select="$paragraphWithOpeningBold"/>
                           <xsl:with-param name="replace" select="'[/BOLD]'"/>
                           <xsl:with-param name="with" select="'&lt;/b&gt;'"/>
                        </xsl:call-template>
            </xsl:variable>
            <xsl:variable name="finalParagraph">
                <xsl:call-template name="replace-string">
                           <xsl:with-param name="text" select="$paragraphWithClosingBold"/>
                           <xsl:with-param name="replace" select="'[/ITALIC]'"/>
                           <xsl:with-param name="with" select="'&lt;/i&gt;'"/>
                        </xsl:call-template>
                    </xsl:variable>

          <xsl:value-of select="$finalParagraph" disable-output-escaping="yes"/>
        </p>
      </xsl:if>
      <xsl:if test="$figure">
          <div class="figuremargin">
              <div>
                  <img class="figure" alt="{Figure/FileName}" src="./images/{./Figure/FileName}"></img>
            </div>
          <h3 class="figureheader"><xsl:value-of select="./Figure/Caption"/> </h3>
        </div>
      </xsl:if>
      <xsl:if test="$bulletedlist">
        <ul>
          <xsl:for-each select="./BulletedList/Item">
            <xsl:apply-templates select="./BulletedList/Item"/>
              <li class="bulleteditem_level{@level}">
                  <xsl:value-of select="."/>
              </li>
          </xsl:for-each>
        </ul>
      </xsl:if>
     </xsl:for-each>
   </xsl:template>
   <!-- brace-escaping to guard against interpretation as AVTs in the
output got this template from http://www.dpawson.co.uk -->
  <xsl:template name="replace-string">
    <xsl:param name="text"/>
    <xsl:param name="replace"/>
    <xsl:param name="with"/>
    <xsl:choose>
      <xsl:when test="contains($text,$replace)">
        <xsl:value-of select="substring-before($text,$replace)"/>
        <xsl:value-of select="$with"/>
        <xsl:call-template name="replace-string">
          <xsl:with-param name="text"
select="substring-after($text,$replace)"/>
          <xsl:with-param name="replace" select="$replace"/>
          <xsl:with-param name="with" select="$with"/>
        </xsl:call-template>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="$text" disable-output-escaping="yes"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```