13

XSL: Extensible Stylesheet Language Formatting Objects

Objectives

- To become familiar with XSL formatting objects.
- To be able to use XSL Transformations to generate XSL documents.
- To be able to mark up a document with XSL formatting objects.
- To be able to use Apache's FOP processor to transform XSL documents.

Vigorous writing is concise. A sentence should contain no unnecessary words, a paragraph no unnecessary sentences.
William Strunk, Jr.

I have made this letter longer than usual, because I lack the time to make it short.
Blaise Pascal

Outline

13.1 Introduction

An XSL stylesheet can be used to transform an XML document into a variety of formats (Fig. 13.1). In Chapter 12, we used XSLT to transform XML documents into HTML, which is perhaps the most common use of XSLT. In this chapter, we introduce another aspect of XSL—called *formatting objects*—used to format XML documents for presentation. Formatting objects constitute the vast majority of XSL features. We present only a small subset of formatting-object features in this chapter. The latest XSL Working Draft can be found at **`www.w3.org/TR/xsl`**.

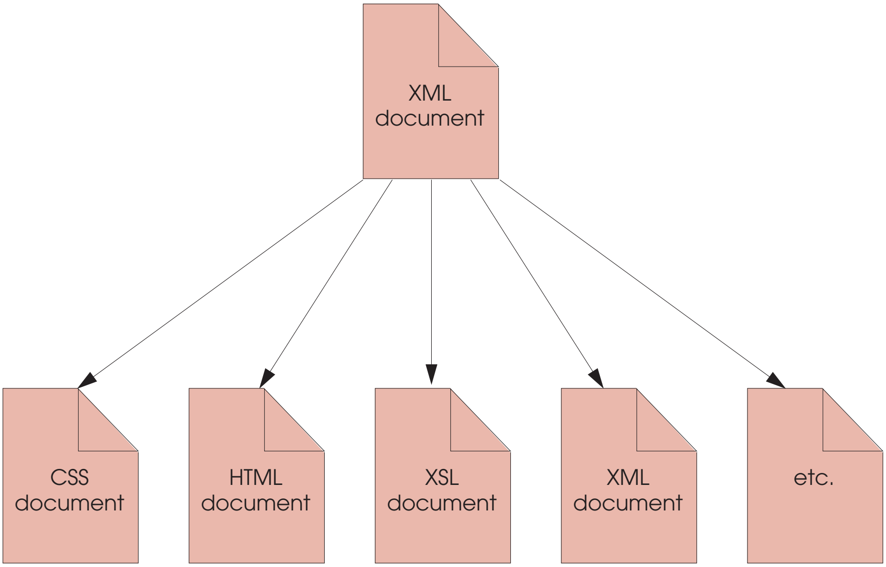


Fig. 13.1 Using XSL to transform XML into a variety of formats.

Formatting objects are typically used when the result of a transformation is for *print media* (e.g., books, magazines, etc.). An XML document is transformed into an XSL document that marks up the data using formatting objects. This XSL document can then be transformed into other formats, including *Portable Document Format* (*PDF*), a portable, proprietary format created by Adobe; a Microsoft Word document; etc.

In this chapter, we present several examples that use Apache's Java-based tool *FOP* to transform XSL documents containing formatting objects (which we call "XSL formatting-object documents") into PDF documents. Apache FOP implements only a small subset of the available formatting-object elements, but full support for the Working Draft is expected in the near future. [*Note*: RenderX's *XEP* is another major tool (or processor) for transforming XSL documents containing formatting objects. A trial version of XEP can be downloaded from **www.renderx.com**.]

13.2 Setup

In this section, we describe the software necessary to execute the chapter examples. The following software is required:

1. Java 2 Standard Edition. Download and install the Java 2 Standard Edition from **www.java.sun.com/j2se**.
2. FOP requires a Java parser. In this chapter we use Apache's Xerces parser to run FOP. Apache's Xerces parser for Java can be downloaded from **xml.apache.org/xerces-j**.
3. Apache Xalan XSLT processor. Xalan can be downloaded free of charge from **xml.apache.org/xalan**.
4. Apache FOP. We use the Java-based version available at **xml.apache.org/fop**.
5. Chapter 13 examples. Copy them from the CD-ROM that accompanies this book to your machine. Our convention is to give XSL formatting objects documents the file extension **.fo**. Because these files are XML documents, they can have other extension (e.g., **.xsl**, **.xml**, etc.) as well.

13.3 Examples of XSL Formatting-object Documents

Figure 13.2 shows an XSL document that contains a formatted description of Deitel & Associates, Inc. Formatting objects describe the physical page dimensions, fonts, etc. Apache's FOP is used to transform the document into PDF form. [*Note*: This XSL formatting-object document typically would not be written by a document author, but would be created from an XSL transformation. For simplicity, we simply show the document.]

To generate a PDF document from this XSL formatting-object document, type

```
java org.apache.fop.apps.CommandLine welcome.fo welcome.pdf
```

at the command line to transform **welcome.fo** (Fig. 13.2) to PDF form (**welcome.pdf**). The **CLASSPATH** variable must be set for the ***CommandLine*** application (i.e., FOP) and Xalan. In case you do not wish to execute this example, we have provided **welcome.pdf** in the Chapter 13 examples directory. To open the PDF file created for viewing, Adobe® Acrobat Reader™ is required. Adobe Acrobat Reader is available for download free of charge from **www.adobe.com**.

Apache FOP also provides an alternative to Acrobat Reader—*Apache's FO viewer application*—for viewing the XSL formatting object-document. To view the results of the transformation, type

```
java org.apache.fop.apps.AWTCommandLine welcome.fo
```

at the command line. The **CLASSPATH** for ***AWTCommandLine*** (i.e., Apache's FO viewer) must be set. Line 6

```
<fo:root xmlns:fo = "http://www.w3.org/1999/XSL/Format">
```

defines root element ***fo:root*** and namespace prefix ***fo*** with the URI ***http://www.w3.org/1999/XSL/Format***. Element **fo:root** is a container element only; it does not affect the document's format.

In publishing, *page masters* define a page's layout (e.g., its margins, headers, footers, etc.). Page masters provide the document author with the flexibility of changing the document's format on a page-by-page basis. Lines 8–11

```
<fo:layout-master-set>

   <fo:simple-page-master master-name = "layout1"
      page-height = "4in">
```

use container element ***fo:layout-master-set*** to group the document's page masters (i.e., *page templates*). To create a page master, element ***fo:simple-page-master*** is used. Attributes ***master-name*** and ***page-height*** specify the page name (i.e., **layout1**) and page height (i.e., **4in**), respectively. XSL formatting objects also provide attribute ***page-width***, for specifying a page's width. Although a document may contain any number of page masters, we use only one page master. In a **simple-page-master** page master, the document is divided into the five regions shown in Fig. 13.3. The header, body, footer, start and end are represented by XSL formatting elements ***fo:region-before***, ***fo:region-body***, ***fo:region-after***, ***fo:region-start*** and **fo:region-end**, respectively.

```
1  <?xml version = "1.0"?>
2
3  <!-- Fig. 13.2 : welcome.fo -->
4  <!-- Simple FO example      -->
5
6  <fo:root xmlns:fo = "http://www.w3.org/1999/XSL/Format">
7
8     <fo:layout-master-set>
9
10       <fo:simple-page-master master-name = "layout1"
11          page-height = "4in">
12
13          <fo:region-body margin-top = "1in"
14             margin-bottom = "1in" margin-left = "1.5in"
15             margin-right = "1.5in"/>
16
17          <fo:region-before extent = "1in" margin-top = "0.2in"
18             margin-bottom = "0.2in" margin-left = "0.2in"
19             margin-right = "0.2in"/>
20
21       </fo:simple-page-master>
```

Fig. 13.2 Simple FO example (part 1 of 3).

```

      <fo:page-sequence-master master-name = "run">
         <fo:repeatable-page-master-reference
            master-name = "layout1"/>
      </fo:page-sequence-master>

   </fo:layout-master-set>

   <fo:page-sequence master-name = "run">

      <fo:static-content flow-name = "xsl-region-before">

         <fo:block font-size = "10pt" line-height = "12pt"
            font-family = "sans-serif">
            page <fo:page-number/>
         </fo:block>

      </fo:static-content>

      <fo:flow flow-name = "xsl-body">

         <fo:block font-size = "36pt"
            font-family = "sans-serif" font-weight = "bold"
            space-after.optimum = "24pt" color = "blue"
            text-align = "center">Welcome!
         </fo:block>

         <fo:block font-size = "12pt"
            font-family = "sans-serif" line-height = "14pt"
            space-after.optimum = "12pt">Deitel &amp; Associates,
            Inc. is an internationally recognized corporate
            training and publishing organization specializing in
            programming languages, Internet/World Wide Web
            technology and object technology education.
         </fo:block>

         <fo:block font-size = "12pt"
            font-family = "sans-serif" line-height = "14pt"
            space-after.optimum = "12pt">Deitel &amp; Associates,
            Inc. is a member of the
            <fo:inline-sequence font-weight = "bold">World
            Wide Web</fo:inline-sequence>
            Consortium.
         </fo:block>

         <fo:block font-size = "12pt"
            font-family = "sans-serif" line-height = "14pt"
            space-after.optimum = "12pt">The company&apos;s
            clients include some of the world&apos;s largest
            computer companies, government agencies, branches
            of the military and business organizations.
         </fo:block>

```

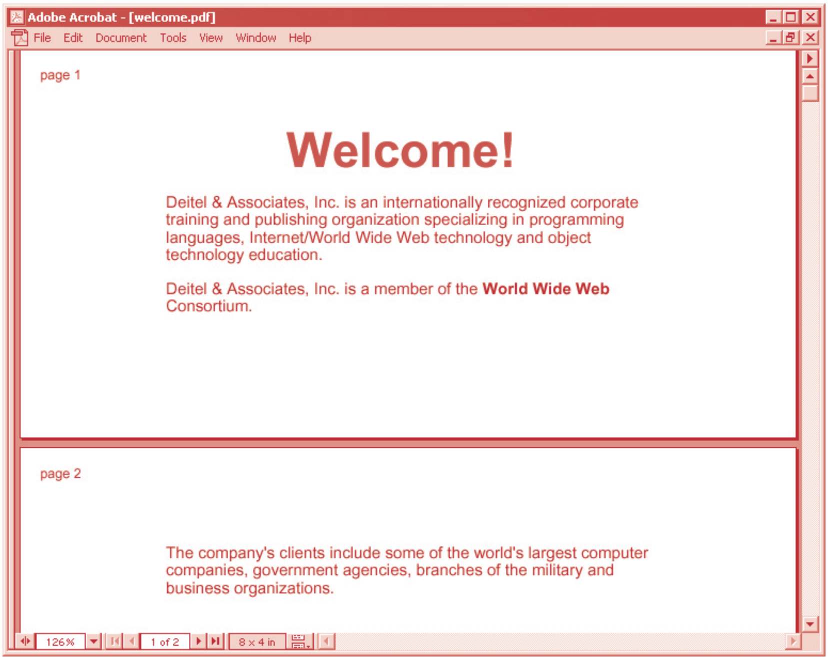
Fig. 13.2 Simple FO example (part 2 of 3).

```
         </fo:flow>

      </fo:page-sequence>

   </fo:root>
```

Fig. 13.2 Simple FO example (part 3 of 3).


Adobe Acrobat - [welcome.pdf]
File Edit Document Tools View Window Help
page 1
page 2
126%
1 of 2
8 x 4 in

Welcome!

Deitel & Associates, Inc. is an internationally recognized corporate training and publishing organization specializing in programming languages, Internet/World Wide Web technology and object technology education.

Deitel & Associates, Inc. is a member of the **World Wide Web** Consortium.

The company's clients include some of the world's largest computer companies, government agencies, branches of the military and business organizations.

Fig. 13.2 Adobe Acrobat displaying **`welcome.pdf`**. (Adobe and Acrobat Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.)

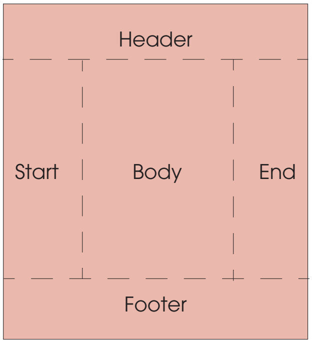


Fig. 13.3 Regions in a **`simple-page-master`** document.

Lines 13–15

```
<fo:region-body margin-top = "1in"
   margin-bottom = "1in" margin-left = "1.5in"
   margin-right = "1.5in"/>
```

use element **fo:region-body** to define the document's body size as having top and bottom margins of 1 inch, and left and right margins of 1.5 inches. Attributes ***margin-top***, ***margin-bottom***, ***margin-left*** and ***margin-right*** represent the top, bottom, left and right values, respectively.

Lines 17–19

```
<fo:region-before extent = "1in" margin-top = "0.2in"
    margin-bottom = "0.2in" margin-left = "0.2in"
    margin-right = "0.2in"/>
```

use element **fo:region-before** to set the page header's margins to 1 inch in height, with a 0.2-inch margin on all sides. The page header is contained within the top margin of the body.

Lines 23–26

```
<fo:page-sequence-master master-name = "run">
   <fo:repeatable-page-master-reference
      master-name = "layout1"/>
</fo:page-sequence-master>
```

use element ***fo:page-sequence-master*** to specify the order in which master pages will be created for the ***master-name*** **run**. We use element ***repeatable-page-master-reference*** to indicate that the **simple-page-master layout1** can be repeated as many times as necessary to contain the document's content. Notice that the second page (in Fig. 13.2) has the same format as the first page.

Lines 30–77 define the pages of the document. Lines 32–39

```
<fo:static-content flow-name = "xsl-region-before">

   <fo:block font-size = "10pt" line-height = "12pt"
      font-family = "sans-serif">
      page <fo:page-number/>
   </fo:block>

</fo:static-content>
```

use element ***fo:static-content*** to specify text that appears on each document page. Attribute ***flow-name*** is assigned the value ***xsl-region-before***, indicating that text will appear in each document page's "header." The text (i.e., the page number) is formatted using an ***fo:block*** element. Attribute ***line-height*** sets the line height to 12 points, and attribute ***font-family*** sets the text's font to ***sans-serif***. When the document is transformed, empty element ***fo:page-number*** is replaced with the page number. Page numbers begin at 1 by default.

Lines 41–75 denote the contents of the pages, with the text being placed in the region set by **fo:region-body**.

The first block, on lines 43–47

```
<fo:block font-size = "36pt"
   font-family = "sans-serif" font-weight = "bold"
   space-after.optimum = "24pt" color = "blue"
   text-align = "centered">Welcome!
</fo:block>
```

marks up blue, bold text that has a font size of 36 points and is centered in the page. We also set to 24 points the optimal amount of space that should follow the text by using attribute ***space-after.optimum***. XSL formatting objects also provide attribute ***space-before.optimum***, to specify the optimum amount of space preceding the text.

The three blocks of text on lines 49–73 each have a font size of 12 points, a line height of 14 points and an optimal spacing of 12 points following each block of text.

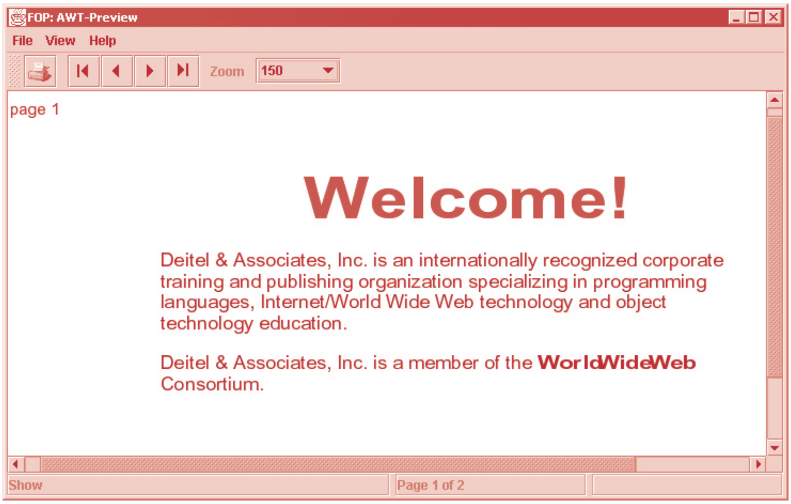
Lines 62 and 63

```
<fo:inline-sequence font-weight = "bold">World
Wide Web</fo:inline-sequence>
```

use element ***inline-sequence*** to change the format of **World Wide Web** to bold.

We present an XSL document (Fig. 13.5) for transforming Chapter 5's XML business letter (Fig. 5.6) into an XSL formatted-object document. We then transform the document to PDF format (Fig. 13.6) using Apache FOP. To generate a PDF document from this XSL formatting-object document, type (on a single line)

```
java -classpath C:\xalan\xerces.jar;C:\xalan\xalan.jar
   org.apache.xalan.xslt.Process -IN letter.xml
   -XSL letter.xsl -OUT letter.fo
```



Welcome!

Deitel & Associates, Inc. is an internationally recognized corporate training and publishing organization specializing in programming languages, Internet/World Wide Web technology and object technology education.

Deitel & Associates, Inc. is a member of the **WorldWideWeb** Consortium.

Courtesy of XML project of the Apache Software Foundation; `xml.apache.org`.

Fig. 13.4 Viewing with Apache's FO viewer application.

to transform the document using Xalan to an XSL formatted-object document. Next, type

```
java org.apache.fop.apps.CommandLine letter.fo welcome.pdf
```

at the command line to transform **letter.fo** (Fig. 13.2) to PDF format (**welcome.pdf**). [*Note*: the Apache FOP JAR file must be in the **CLASSPATH**.] For more information about the PDF, visit the Web site **www.adobe.com/products/acrobat**.

```
<?xml version = "1.0"?>

<!-- Fig. 13.5 : letter.xsl       -->
<!-- Formatting a business letter -->

<xsl:stylesheet version = "1.0"
   xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
   xmlns:fo = "http://www.w3.org/1999/XSL/Format">

   <xsl:template match = "/">

      <fo:root xmlns:fo = "http://www.w3.org/1999/XSL/Format">

         <fo:layout-master-set>

            <fo:simple-page-master master-name = "first">

               <fo:region-body margin-top = "1.75in"
                  margin-bottom = "1in" margin-left = "1.25in"
                  margin-right = "1.25in"/>

               <fo:region-before extent = "1.5in"
                  margin-top = "0.5in" margin-bottom = "0.2in"
                  margin-left = "4.25in"
                  margin-right = "1.25in"/>

            </fo:simple-page-master>

            <fo:simple-page-master master-name = "other">

               <fo:region-body margin-top = "1in"
                  margin-bottom = "1in" margin-left = "1.25in"
                  margin-right = "1.25in"/>

            </fo:simple-page-master>

            <fo:page-sequence-master master-name = "run1">

               <fo:single-page-master-reference
                  master-name = "first"/>

               <fo:repeatable-page-master-reference
                  master-name = "other"/>

```

Fig. 13.5 XSLT document for transforming an XML document into an XSL formatted-object document (part 1 of 4).

```
         </fo:page-sequence-master>

      </fo:layout-master-set>

      <fo:page-sequence master-name = "run1">
         <xsl:apply-templates/>
      </fo:page-sequence>

   </fo:root>

</xsl:template>

<xsl:template match = "letter">

   <fo:static-content flow-name = "xsl-region-before">

      <xsl:apply-templates
         select = "contact[@type = 'from']"/>

   </fo:static-content>

   <fo:flow flow-name = "xsl-body">

      <fo:block font-size = "10pt"
         font-family = "monospace" line-height = "10pt">

         <xsl:apply-templates
            select = "contact[@type = 'to']"/>

      </fo:block>

      <fo:block font-size = "12pt"
         font-family = "monospace" line-height = "14pt"
         space-before.optimum = "18pt"
         space-after.optimum = "18pt">
         <xsl:value-of select = "salutation"/>
      </fo:block>

      <xsl:apply-templates select = "paragraph"/>

      <fo:block font-size = "12pt"
         font-family = "monospace" line-height = "14pt">
         <xsl:value-of select = "closing"/>,
      </fo:block>

      <fo:block font-size = "12pt"
         font-family = "monospace" line-height = "14pt">
         <xsl:value-of select = "signature"/>
      </fo:block>

   </fo:flow>

```

Fig. 13.5 XSLT document for transforming an XML document into an XSL formatted-object document (part 2 of 4).

```
   </xsl:template>

   <xsl:template match = "contact[@type = 'from']">

      <fo:block font-size = "10pt"
         font-family = "monospace" line-height = "12pt">
         <xsl:value-of select = "name"/>
      </fo:block>

      <fo:block font-size = "10pt"
         font-family = "monospace" line-height = "12pt">
         <xsl:value-of select = "address1"/>
      </fo:block>

      <fo:block font-size = "10pt"
         font-family = "monospace" line-height = "12pt">
         <xsl:value-of select = "address2"/>
      </fo:block>

      <fo:block font-size = "10pt"
         font-family = "monospace" line-height = "12pt">
         <xsl:value-of select = "city"/>,
         <xsl:value-of select = "state"/>
         <xsl:text> </xsl:text>
         <xsl:value-of select = "zip"/>
      </fo:block>

      <fo:block font-size = "10pt"
         font-family = "monospace" line-height = "12pt">
         <xsl:value-of select = "phone"/>
      </fo:block>

   </xsl:template>

   <xsl:template match = "contact[@type = 'to']">

      <fo:block font-size = "12pt"
         font-family = "monospace" line-height = "14pt">
         <xsl:value-of select = "name"/>
      </fo:block>

      <fo:block font-size = "12pt"
         font-family = "monospace" line-height = "14pt">
         <xsl:value-of select = "address1"/>
      </fo:block>

      <fo:block font-size = "12pt"
         font-family = "monospace" line-height = "14pt">
         <xsl:value-of select = "address2"/>
      </fo:block>

```

Fig. 13.5 XSLT document for transforming an XML document into an XSL formatted-object document (part 3 of 4).

```
148          <fo:block font-size = "12pt"
149             font-family = "monospace" line-height = "14pt">
150             <xsl:value-of select = "city"/>,
151             <xsl:value-of select = "state"/>
152             <xsl:text> </xsl:text>
153             <xsl:value-of select = "zip"/>
154          </fo:block>
155
156          <fo:block font-size = "12pt"
157             font-family = "monospace" line-height = "14pt">
158             <xsl:value-of select = "phone"/>
159          </fo:block>
160
161       </xsl:template>
162
163       <xsl:template match = "paragraph">
164
165          <fo:block font-size = "12pt" font-family = "monospace"
166             line-height = "14pt" space-after.optimum = "18pt">
167             <xsl:apply-templates/>
168          </fo:block>
169
170       </xsl:template>
171
172       <xsl:template match = "bold">
173
174          <fo:inline-sequence font-weight = "bold">
175             <xsl:value-of select = "."/>
176          </fo:inline-sequence>
177
178       </xsl:template>
179
180    </xsl:stylesheet>
```

Fig. 13.5 XSLT document for transforming an XML document into an XSL formatted-object document (part 4 of 4).

Lines 16–27 create a page master named **`first`** that contains two regions. The first region (lines 18–20) is the document's body, while the second region (lines 22–25) is the document's header that contains the letter's return address. Attribute ***`extent`*** specifies the size of a region (i.e., **`fo:region-before`**, **`fo:region-after`**, **`fo:region-start`** and **`fo:region-end`**). Region **`fo:region-body`** does not have an **`extent`** attribute and is given the remaining area after the other four regions are sized.

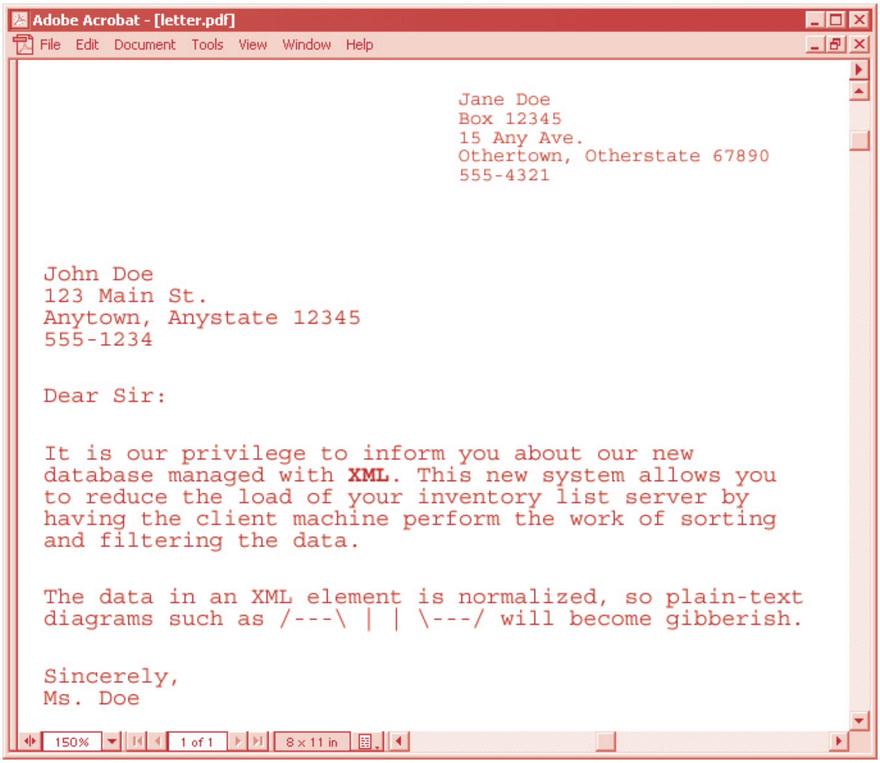
Lines 29–35 create a second page master named **`other`** that will define the layout for successive pages (after the first). Lines 37–45

```
<fo:page-sequence-master master-name = "run1">

   <fo:single-page-master-reference
      master-name = "first"/>

   <fo:repeatable-page-master-reference
      master-name = "other"/>

</fo:page-sequence-master>
```


Adobe Acrobat - [letter.pdf]
File Edit Document Tools View Window Help
150%
1 of 1
8 x 11 in

Jane Doe
Box 12345
15 Any Ave.
Othertown, Otherstate 67890
555-4321

John Doe
123 Main St.
Anytown, Anystate 12345
555-1234

Dear Sir:

It is our privilege to inform you about our new database managed with **XML**. This new system allows you to reduce the load of your inventory list server by having the client machine perform the work of sorting and filtering the data.

The data in an XML element is normalized, so plain-text diagrams such as /---\ | | \---/ will become gibberish.

Sincerely,
Ms. Doe

Fig. 13.6 PDF output of the business letter. (Adobe and Acrobat Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.)

uses element **`fo:page-sequence-master`** to set the order in which the page masters occur. Page master **`first`** (line 16) occurs first and can occur only once, because it is marked up as ***`fo:single-page-master-reference`***. Page master **`other`** (line 29) follows the **`first`** page master and can occur any number of times. Lines 59–64

```
<fo:static-content flow-name = "xsl-region-before">

   <xsl:apply-templates
      select = "contact[@type = 'from']"/>

</fo:static-content>
```

place the letter's return address in the document's header (i.e., **`xsl-region-before`**). The remainder of the document uses templates to match the text for the document's body.

13.4 Lists

XSL formatting objects also provide the capabilities to format lists of items. Apache FOP provides support for *lists* and *tables*. In this section, we provide an example (Fig. 13.7) that formats data into a list. The rendered output is rendered in Fig. 13.8.

```
<?xml version = "1.0"?>

<!-- Fig. 13.7 : topic_list.fo -->
<!-- List example              -->

<fo:root xmlns:fo = "http://www.w3.org/1999/XSL/Format">

   <fo:layout-master-set>

      <fo:simple-page-master master-name = "layout1">

         <fo:region-body margin-top = "1in"
         margin-bottom = "1in" margin-left = "1.5in"
         margin-right = "1.5in"/>
      </fo:simple-page-master>

      <fo:page-sequence-master master-name = "run">
         <fo:repeatable-page-master-reference
            master-name = "layout1"/>
      </fo:page-sequence-master>

   </fo:layout-master-set>

   <fo:page-sequence master-name = "run">

      <fo:flow>

         <fo:block font-size = "36pt"
            font-family = "sans-serif" font-weight = "bold"
            space-after.optimum = "12pt" color = "yellow"
            background-color = "black"
            text-align = "center" line-height = "42pt">
            Deitel Book Topics
         </fo:block>

         <fo:block font-size = "12pt"
            font-family = "sans-serif" line-height = "14pt"
            space-after.optimum = "12pt">Here are some topics
            that have been covered:
         </fo:block>

         <fo:list-block>

            <fo:list-item>

               <fo:list-item-label>
                  <fo:block>-</fo:block>
               </fo:list-item-label>

               <fo:list-item-body>
                  <fo:block>Java</fo:block>
               </fo:list-item-body>

```

Fig. 13.7 List supported by Apache's FOP (part 1 of 2).

```
54              </fo:list-item>
55
56              <fo:list-item>
57
58                 <fo:list-item-label>
59                    <fo:block>-</fo:block>
60                 </fo:list-item-label>
61
62                 <fo:list-item-body>
63                    <fo:block>C / C++</fo:block>
64                 </fo:list-item-body>
65
66              </fo:list-item>
67
68              <fo:list-item>
69
70                 <fo:list-item-label>
71                    <fo:block>-</fo:block>
72                 </fo:list-item-label>
73
74                 <fo:list-item-body>
75                    <fo:block>HTML</fo:block>
76                 </fo:list-item-body>
77
78              </fo:list-item>
79
80              <fo:list-item>
81
82                 <fo:list-item-label>
83                    <fo:block>-</fo:block>
84                 </fo:list-item-label>
85
86                 <fo:list-item-body>
87                    <fo:block>XML</fo:block>
88                 </fo:list-item-body>
89
90              </fo:list-item>
91
92           </fo:list-block>
93
94        </fo:flow>
95
96     </fo:page-sequence>
97
98  </fo:root>
```

Fig. 13.7 List supported by Apache's FOP (part 2 of 2).

Element ***fo:flow*** (line 26) specifies content that can flow from one page to the next. A common example of **fo:flow** is text on a book's page that naturally flows to the next page. The primary difference between **fo:flow** and **fo:static-content** is that **fo:static-content** is duplicated on each page. For example, a book title and page number (e.g., **fo:static-content**) generally appear on every page in a book. However, the text of a paragraph (e.g., **fo:flow**) in a novel is not duplicated on every page.

Lines 28–34 define a block that formats the text **Deitel Book Topics**. Attribute ***background-color*** sets the background color to black. Attribute ***text-align*** aligns the text. In this particular case, we **center** the text.

Lines 42–92 set the ***fo:list-block*** element, which contains the list items. Lines 46–48

```
<fo:list-item-label>
   <fo:block>-</fo:block>
</fo:list-item-label>
```

use element ***fo:list-item-label*** to mark up the text that precedes each item in the list. In this particular case, we mark up a hyphen, **-**.

Lines 50–52

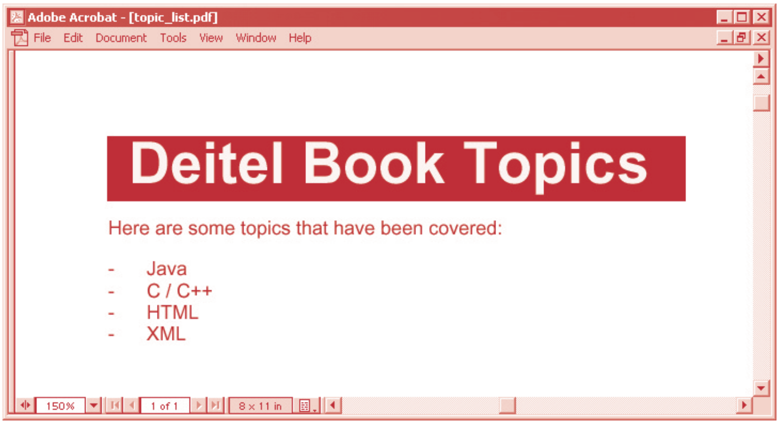
```
<fo:list-item-body>
   <fo:block>Java</fo:block>
</fo:list-item-body>
```

use element ***fo:list-item-body*** to mark up an individual list item's text (i.e., **Java**). The remaining list items are marked up in a similar manner.

13.5 Internet and World Wide Web Resources

www.xml.com/pub/Guide/XSL_FO's
XSL Formatting Object vocabulary links.

www.renderx.com/Tests/validator/fo2000.dtd.html
DTD for the last version of XSL FO.


Adobe Acrobat - [topic_list.pdf]
File Edit Document Tools View Window Help
150%
1 of 1
8 x 11 in

Deitel Book Topics

Here are some topics that have been covered:

- Java
- C / C++
- HTML
- XML

Fig. 13.8 Adobe Acrobat displaying **topic_list.pdf**. (Adobe and Acrobat Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.)

xml.apache.org/fop
Home page of FOP, an open-source XSL formatter-renderer, from Apache.

www.renderx.com
Provides XEP, a commercial XSL FO processor.

www.arbortext.com
Arbortext Web site where a commercial XSL FO processor is available.

www.xmlsoftware.com
Provides links to many different XML-related pieces of software—including XSL processors.

SUMMARY

- XSL formatting objects format XML documents for presentation. Formatting objects constitute the vast majority of XSL features.
- Formatting objects are typically used when the result of a transformation is for print media (e.g., books, magazines, etc.).
- Apache's Java-based tool, FOP, transforms XSL documents containing formatting objects into PDF documents.
- XSL formatting-object documents have root element **fo:root** and namespace prefix **fo**. Namespace URI **http://www.w3.org/1999/XSL/Format** is used with formatting objects. Element **fo:root** is a container element only; it does not affect the document's format.
- In publishing, page masters define a page's layout (e.g., its margins, headers, footers, etc.). Page masters provide the document author with the flexibility of changing the document's format on a page-by-page basis.
- Element **fo:layout-master-set** groups the document's page masters (i.e., page templates). To create a page master, element **fo:simple-page-master** is used. Attributes **master-name** and **page-height** specify the page name and page height, respectively. XSL formatting objects also provide attribute **page-width**, for specifying a page's width.
- In a **simple-page-master** page master, the document is divided into five regions: header, body, footer, start and end. Each of these regions are represented by XSL formatting elements **fo:region-before**, **fo:region-body**, **fo:region-after**, **fo:region-start** and **fo:region-end**, respectively.
- Attributes **margin-top**, **margin-bottom**, **margin-left** and **margin-right** represent the top, bottom, left and right margin values, respectively.
- Element **fo:page-sequence-master** specifies the order in which master pages are created for a page master. Element **repeatable-page-master-reference** indicates that a **simple-page-master** can be repeated as many times as necessary in order to contain the document's content.
- Element **fo:static-content** specifies text that appears on each document page. Attribute **flow-name**, when assigned the value **xsl-region-before**, indicates that text will appear in each document page's header.
- Attribute **line-height** sets the line height, and attribute **font-family** sets the text's font. Empty element **fo:page-number** represents the page number.
- XSL formatting objects provide attributes **space-before.optimum** and **space-after.optimum** to specify the optimum amount of space preceding and following the text, respectively.
- Element **inline-sequence** changes the format of text inside a block.

- Attribute **extent** specifies the size of a region (i.e., **fo:region-before**, **fo:region-after**, **fo:region-start** and **fo:region-end**). Region **fo:region-body** does not have an **extent** attribute and is given the remaining area after the other four regions are sized.
- A page master occurs first and can occur only once when it is marked up as **fo:single-page-master-reference**.
- XSL formatting objects also provide the capabilities to format lists of items.
- Element **fo:flow** specifies content that can flow from one page to the next. The primary difference between **fo:flow** and **fo:static-content** is that **fo:static-content** is duplicated on each page.
- Attribute **background-color** sets the background color of text. Attribute **text-align** aligns text.
- Element **fo:list-block** marks up a list of items.
- Element **fo:list-item-label** marks up the text that precedes an item in a list.
- Element **fo:list-item-body** mark ups an individual list item's text.

TERMINOLOGY

Adobe Portable Document Format (PDF) document
Apache FOP
Apache Xalan
background-color attribute
color attribute
document layout scheme
document pages
extent attribute
.fo filename extension
fo namespace prefix
fo:block element
fo:flow element
fo:layout-master-set element
fo:list-block element
fo:list-item element
fo:list-item-body element
fo:list-item-label element
fo:page-sequence-master element
fo:region-after element
fo:region-before element
fo:region-body element
fo:region-end element
fo:region-start element
fo:root element
fo:repeatable-page-master-reference element
fo:simple-page-master element
fo:single-page-master-reference element
fo:static-content element
font-family attribute
font-size attribute
formatting object
line-height attribute
margin-bottom attribute
margin-left attribute
margin-right attribute
margin-top attribute
master-name attribute
page masters
page-height attribute
page-width attribute
space-after.optimum attribute
space-before.optimum attribute
text-align attribute
XSL formatting-object document

SELF-REVIEW EXERCISES

13.1 State whether the following are *true* or *false*. If *false*, explain why.

a) Apache's FOP provides support for lists and tables.
b) Extensible Stylesheet Language was created to provide formatting for CSS documents.
c) Element **fo:root** is the root element for an XSL formatting-object document.
d) Attribute **text-align** specifies how text is aligned.

e) Each page sequence in a styled XML document is defined using element **fo:sequence-specification**, which contains elements **fo:static-content** and **fo:flow**.
f) If multiple single-page specifiers are used in sequence definitions, then each page of content is mapped to a page master.
g) The **fo:block** element is usually used to format paragraphs, titles, captions and other textual objects.
h) The set of page masters for a page sequence is defined using the element **fo:page-order**.
i) The **fo:static-content** element must precede any **fo:flow** elements.
j) The repeating page specifier does not allow for setting the page master for the first page. Another page master must be used to do this.

13.2 Fill in the blanks in each of the following statements.
a) XSL formatting objects are usually given the __________ namespace prefix.
b) Element **fo:page-sequence** defines a sequence of pages that use a specific __________.
c) Attribute __________ defines a region's width or height.
d) A single-page __________ instantiates a single-page master.
e) The __________ page specifier sets the page masters for __________ and odd pages.
f) Element **fo:**__________ is used to define static content for a page region in the pages of a page __________.
g) Element **fo:flow** is used to hold the content for a __________ of a page, which can __________ multiple pages.
h) Element __________ is used within elements **fo:static-content** and **fo:flow**.
i) Element **fo:inline-**__________ formats inline objects.
j) **fo:list-block** formats __________ and contains elements **fo:list-**__________.

ANSWERS TO SELF-REVIEW EXERCISES

13.1 a) True. b) False. The Extensible Stylesheet Language was created to provide formatting for XML documents. c) True. d) True. e) False. Each page sequence is defined using element **fo:page-sequence**, which contains element **fo:sequence-specification**, **fo:static-content** and **fo:flow**. f) True. g) True. h) False. The set of page masters for a page sequence is defined using the element **fo:sequence-specification**. i) True. j) False. The repeating page specifier allows for setting the page master for the first page and another page master for all pages following the first.

13.2 a) **fo**. b) page master. c) **extent**. d) specifier. e) **alternating**, even. f) **static-content**, sequence. g) region, span. h) **fo:block**. i) **sequence**. j) lists, **item**.

EXERCISES

13.3 Write a simple XSLT document that would transform **intro.xml** (Fig. 5.1) into an FO document. Process the FO document to obtain a PDF document by using Apache's FOP. Render the message element in different colors and font sizes. Each page in the PDF document should contain a page number on its top-right corner.

13.4 Write code that would create a PDF document using **usage.xml** (Fig. 5.5). The PDF document generated should contain the title of the book, appearing in bold fonts, followed by the author of the book. It should also contain a table with two columns, consisting of page numbers and corresponding titles (preface, chapter and appendix). [*Hint:* Tables can be drawn using the **fo:table** element. Columns are declared with the **fo:table-column** element. Rows are represented by the **fo:table-row** element that contains the **fo:table-cell** element.]