

where *location*, *collection*, *callno*, and *status* are the values of the location, collection, callno, and status elements. If the status element contains a return attribute, display the text (due: *return*) where *return* is the value of the return attribute.

11. Save your changes to the file.
12. Generate your result document using either an XML editor or your web browser. Verify that the layout and content of each item matches that shown in Figure 5-39.

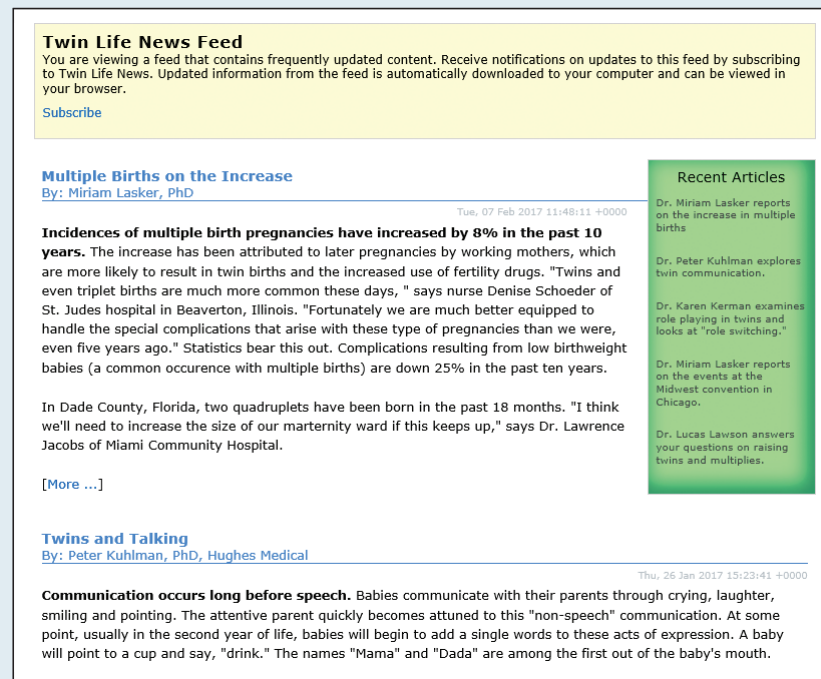
CHALLENGE

Case Problem 2

Data Files needed for this Case Problem: feedtxt.xml, newstxt.xml, +1 CSS file, +5 HTML files

Twin Life News Feed Williams Lewis manages a monthly magazine and website for parents of twins called *Twin Life News*. A feature of the website is a newsfeed to highlight current articles of interest to the site's subscribers. RSS is an XML vocabulary designed for the reporting and distribution of current news features across the Internet. William wants you to create an XSLT style sheet to transform the content of the news feed into a format that is easily readable in a web browser. Figure 5-40 shows a preview of the news feed as viewed within a browser.

Figure 5-40 News feed Web page



Each news item in the feed has a one- or two-paragraph outtake. The text of the outtake is written in HTML code embedded within a CDATA section and must be sent to the result document as text with no processing. Complete the following:

1. Using your text editor, open the **newstxt.xml** and **feedtxt.xml** files from the xml05 ► case2 folder. Enter **your name** and the **date** in the comment section of each file, and save them as **news.xml** and **feed.xml**, respectively.
2. Go to the **news.xml** file in your text editor. Review the content of the file and its structure. Note the use of different namespaces for different RSS elements. Add a processing instruction after the comment section that attaches the feed.xml style sheet to this XML document. Close the file, saving your changes.

3. Within the `stylesheet` element, insert the root template, writing the following code to the result document:

```
<html>
  <head>
    <title>title</title>
    <link href="feedstyles.css" rel="stylesheet"
type="text/css" />
  </head>
  <body>
  </body>
</html>
```

where *title* is the value of the channel title from the title element in the news.xml file using the location path `rss/channel/title`.

4. Directly below the `<body>` tag, insert the following code into the root template:

```
<div id="head">
  <h1>title</h1>
  <p>description</p>
  <p>
    <a href="link">Subscribe</a>
  </p>
</div>
```

where *title* is the channel title, *description* is the channel description, and *link* is the channel link.

5. Directly after the closing `</div>` tag, insert:


```
<div id="sidebar">
  <h1>Recent Articles</h1>
  <p>
    <a href="#guid">subhead</a>
  </p>
  <p>
    <a href="#guid">subhead</a>
  </p>
  ...
</div>
```

where `<p>subhead</p>...` list each subhead of every item element in the document formatted as a paragraph with a hypertext link. Use the value of the `guid` element to identify the link. (Hint: Use the `for-each` instruction to go through the list of item elements.)

6. Directly below the `</div>` tag you added in the previous step, apply the template for the location path `/rss/channel/item`.
7. Create the template for the item element that will write the following code to the result document:

```
<div class="article">
  <h1 id="guid">title</h1>
  <h2>By: author</h2>
  <h3>pubDate</h3>
</div>
```

where *guid* is the value of the `guid` element, *title* is the item title, *author* is the item author, and *pubDate* is the item publication date.

-  **EXPLORE 8.** Between the closing `</h3>` and the closing `</div>` tags, insert the following code:

```
<div class="outtake">
  description
</div>
<p>[<a href="link">More ...</a>]</p>
```

where *description* contains the HTML code for the item description and *link* is the link for the news item. Disable escaping for the value of the description element so that the XSLT processor will submit the description value directly as text.

9. Save your changes to the file.
10. Generate your result document using either an XML editor or your web browser. Verify that the page layout and format resembles Figure 5-40. Also confirm that all of the recent article links listed in the box on the right margin point to the article headings and that by clicking the [More ...] link you can view a page containing the complete news article.

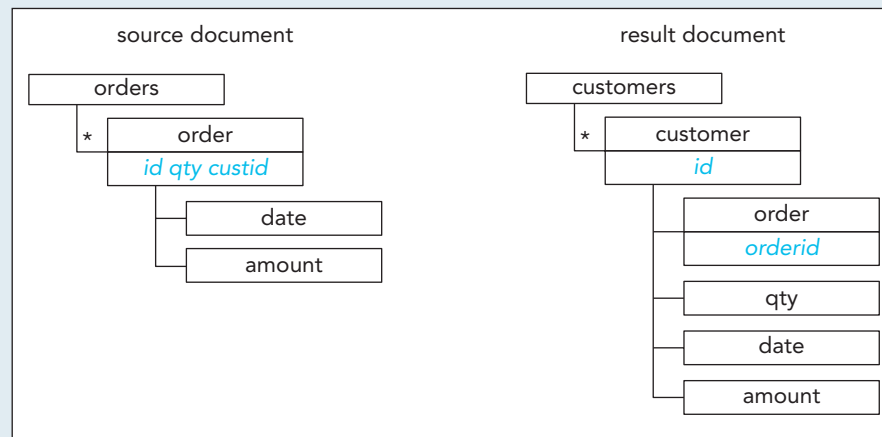
CHALLENGE

Case Problem 3

Data Files needed for this Case Problem: *clisttxt.xml*, *orderstxt.xml*, +1 DTD file

The Spice Bowl Sharon Krueger manages accounts for *The Spice Bowl*, an online seller of spices in bulk quantities for restaurants and grocery stores. One of Sharon's colleagues has created an order report in an XML file detailing some of the orders placed at the Spice Bowl over the last few days. Rather than organizing the data by orders, Sharon would prefer to have an XML document that organizes the data by customer ID number. The layout of the file she was given and the file she would like to have are displayed in Figure 5-41.

Figure 5-41 Structure of the source and result document



She's asked you to write an XSLT style sheet to transform the layout of her file into the new structure. To create this style sheet, you'll work with the XSLT commands to create elements and attributes. Note that you must have access to an XML editor to generate the result document. Complete the following:

1. Using your text editor, open the **orderstxt.xml** and **clisttxt.xml** files from the xml05 ► case3 folder. Enter **your name** and the **date** in the comment section of each file, and save them as **orders.xml** and **clist.xml**, respectively.
2. Go to the **orders.xml** file in your text editor. Add a processing instruction after the comment section that attaches the **clist.xml** style sheet to this XML document. Close the file, saving your changes.
3. Go to the **clist.xml** file in your text editor. Insert the root **stylesheet** element and namespace.
4. Add an **output** element to the style sheet that instructs the XSLT processor that the result document is an XML document, that the version is 1.0, that the text is indented, and that the XML document should contain a DOCTYPE system declaration pointing to the **customers.dtd** file.