NEW PERSPECTIVES

XML

3RD EDITION

COMPREHENSIVE

Carey :: Vodnik

New Perspectives on XML, 3rd Edition, Comprehensive

Textbook Reviewers

We are extremely grateful to the *New Perspectives on XML, 3rd Edition, Comprehensive* textbook reviewers listed below, and would like to take this opportunity to acknowledge them for their contributions in the development of this text. Their timely reviews, informed feedback, and excellent suggestions were tremendously valuable and helped us to produce an outstanding text that will meet the needs of all our New Perspectives instructors and students. Our sincere thanks to all!

Textbook Reviewers

David Doering, St. Louis Community College
Ravinder Kang, Highline Community College
Diana Kokoska, University of Maine at Augusta
Barbara Rader, University of Maryland
Sheryl Schoenacher, Farmingdale State College
Dave Sciuto, University of Massachusetts—Lowell
John Whitney, Fox Valley Technical College
Dawn Wick, Southwestern Community College

"The third edition of New Perspectives on XML takes a practical approach to teaching the foundations of XML. It provides real-world scenarios that allow students to work hands-on, applying XML concepts. The structure of each chapter affords the student, whether online or in the classroom, a variety of learning activities designed to support all learning styles. I have successfully used the New Perspectives on XML text for many years, teaching thousands of students the practical purposes for XML. Success in using the text is measurable in the number of 'Aha!' moments most students have thanks to clearly defined and explained concepts, numerous practical examples, and tutorials that challenge their working knowledge of XML. This newest edition takes vital steps toward transforming the student of XML into a working practitioner."

—Dave Sciuto,
University of Massachusetts—Lowell

NEW PERSPECTIVES ON

XML

3rd Edition

COMPREHENSIVE

Patrick Carey
Sasha Vodnik

CENGAGE Learning®

Australia • Brazil • Mexico • Singapore • United Kingdom • United States

New Perspectives on XML
3rd Edition, Comprehensive
Product Director: Kathleen McMahon
Senior Director of Development: Marah Bellegarde
Senior Product Manager: Jim Gish
Product Development Manager: Leigh Hefferon
Senior Content Developer: Kathy Finnegan
Marketing Director: Michele McTighe
Senior Marketing Manager: Eric La Scolla
Developmental Editor: Pam Conrad
Composition: GEX Publishing Services
Art Director: Marissa Falco
Text Designer: Althea Chen
Cover Designer: GEX Publishing Services
Cover Art: ©Tarek El Sombati/E+/Getty Images
Copyeditor: GEX Publishing Services
Proofreader: Vicki Zimmer
Indexer: Richard Carlson

Library of Congress Control Number: 2014952799
ISBN: 978-1-285-07582-2

Cengage Learning
20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at:
www.cengage.com/global

Cengage Learning products are represented in Canada by
Nelson Education, Ltd.

For your course and learning solutions, visit **www.cengage.com**

Purchase any of our products at your local college store or at our preferred online store **www.cengagebrain.com**

Printed in the United States of America
Print Number: 01 Print Year: 2014

Preface

The New Perspectives Series' critical-thinking, problem-solving approach is the ideal way to prepare students to transcend point-and-click skills and take advantage of all that XML has to offer.

In developing the New Perspectives Series, our goal was to create books that give students the software concepts and practical skills they need to succeed beyond the classroom. We've updated our proven case-based pedagogy with more practical content to make learning skills more meaningful to students. With the New Perspectives Series, students understand *why* they are learning *what* they are learning, and are fully prepared to apply their skills to real-life situations.

About This Book

"With the clear instructions in this text, students know exactly what code to write and where to place it. The exercises enable students to apply what they've learned using realistic business scenarios. I would recommend this text to anyone teaching XML or learning it on their own."

—Dawn Wick
Southwestern Community College

This book provides complete coverage of XML including the following:

- Using XSLT to transform XML data into HTML format
- Creating custom reports using XSLT 2.0 and XPath 2.0
- Designing database queries using XQuery

New for this edition!

- Each session begins with a Visual Overview, which includes colorful, enlarged figures with numerous callouts and key term definitions, giving students a comprehensive preview of the topics covered in the session, as well as a handy study guide.
- New ProSkills boxes provide guidance for how to use the software in real-world, professional situations, and related ProSkills exercises integrate the technology skills students learn with one or more of the following soft skills: decision making, problem solving, teamwork, verbal communication, and written communication.
- Important steps are highlighted in yellow with attached margin notes to help students pay close attention to completing the steps correctly and avoid time-consuming rework.

System Requirements

This book assumes that students have access to a current browser that supports the viewing of XML files and XML files transformed using XSLT. Current versions of the major browsers support these features of XML with the exception of Google Chrome, which does not support XML documents stored locally. The screenshots of web pages in this book were produced using Internet Explorer 10 running on Windows 7 Professional (64-bit) and Internet Explorer 11 running on Windows 8.1 (64-bit), unless otherwise noted. Students who intend to validate their XML documents in Tutorials 2 through 4 should have access to an XML validating parser, such as Exchanger XML Editor, or to an online validation service. Students who intend to transform XML documents using XSLT should have access to an XSLT processor such as Exchanger, XMLSpy or Saxon. The transformations performed in Tutorials 5 through 8 were done using Saxon-HE (home edition) available free for Java or .NET at http://saxon.sourceforge.net. Students who perform XQuery data queries in Tutorial 9 should have access to an XQuery processor. Such queries were performed in Tutorial 9 using Saxon-HE. Students who are using processors other than Saxon should consult their processor's documentation for specific installation and operation instructions.

"New Perspectives texts provide up-to-date, real-world application of content, making book selection easy. The step-by-step, hands-on approach teaches students concepts they can apply immediately."

—John Taylor
Southeastern Technical College

The New Perspectives Approach

Context
Each tutorial begins with a problem presented in a "real-world" case that is meaningful to students. The case sets the scene to help students understand what they will do in the tutorial.

Hands-on Approach
Each tutorial is divided into manageable sessions that combine reading and hands-on, step-by-step work. Colorful screenshots help guide students through the steps. **Trouble?** tips anticipate common mistakes or problems to help students stay on track and continue with the tutorial.

VISUAL OVERVIEW

Visual Overviews
New for this edition! Each session begins with a Visual Overview, a new two-page spread that includes colorful, enlarged figures with numerous callouts and key term definitions, giving students a comprehensive preview of the topics covered in the session, as well as a handy study guide.

PROSKILLS

ProSkills Boxes and Exercises
New for this edition! ProSkills boxes provide guidance for how to use the software in real-world, professional situations, and related ProSkills exercises integrate the technology skills students learn with one or more of the following soft skills: decision making, problem solving, teamwork, verbal communication, and written communication.

KEY STEP

Key Steps
New for this edition! Important steps are highlighted in yellow with attached margin notes to help students pay close attention to completing the steps correctly and avoid time-consuming rework.

INSIGHT

InSight Boxes
InSight boxes offer expert advice and best practices to help students achieve a deeper understanding of the concepts behind the software features and skills.

TIP

Margin Tips
Margin Tips provide helpful hints and shortcuts for more efficient use of the software. The Tips appear in the margin at key points throughout each tutorial, giving students extra information when and where they need it.

REVIEW
APPLY

Assessment
Retention is a key component to learning. At the end of each session, a series of Quick Check questions helps students test their understanding of the material before moving on. Engaging end-of-tutorial Review Assignments and Case Problems have always been a hallmark feature of the New Perspectives Series. Colorful bars and brief descriptions accompany the exercises, making it easy to understand both the goal and level of challenge a particular assignment holds.

REFERENCE
GLOSSARY/INDEX

Reference
Within each tutorial, Reference boxes appear before a set of steps to provide a succinct summary and preview of how to perform a task. In addition, each book includes a combination Glossary/Index to promote easy reference of material.

Our Complete System of Instruction

BRIEF

INTRODUCTORY

COMPREHENSIVE

Coverage To Meet Your Needs

Whether you're looking for just a small amount of coverage or enough to fill a semester-long class, we can provide you with a textbook that meets your needs.

- Brief books typically cover the essential skills in just 2 to 4 tutorials.
- Introductory books build and expand on those skills and contain an average of 5 to 8 tutorials.
- Comprehensive books are great for a full-semester class, and contain 9 to 12+ tutorials.

So if the book you're holding does not provide the right amount of coverage for you, there's probably another offering available. Go to our Web site or contact your Cengage Learning sales representative to find out what else we offer.

COURSECASTS

CourseCasts – Learning on the Go. Always available…always relevant.

Want to keep up with the latest technology trends relevant to you? Visit http://coursecasts.course.com to find a library of weekly updated podcasts, CourseCasts, and download them to your mp3 player.

Ken Baldauf, host of CourseCasts, is a faculty member of the Florida State University Computer Science Department where he is responsible for teaching technology classes to thousands of FSU students each year. Ken is an expert in the latest technology trends; he gathers and sorts through the most pertinent news and information for CourseCasts so your students can spend their time enjoying technology, rather than trying to figure it out. Open or close your lecture with a discussion based on the latest CourseCast.

Visit us at http://coursecasts.course.com to learn on the go!

Instructor Resources

We offer more than just a book. We have all the tools you need to enhance your lectures, check students' work, and generate exams in a new, easier-to-use and completely revised package. This book's Instructor's Manual, Cengage Learning Testing Powered by Cognero, PowerPoint presentations, data files, solution files, figure files, and a sample syllabus are all available on this text's Instructor Companion Site. Simply search for this text at login.cengage.com.

SAM

SAM: Skills Assessment Manager

Get your students workplace-ready with SAM, the premier proficiency-based assessment and training solution for Microsoft Office! SAM's active, hands-on environment helps students master computer skills and concepts that are essential to academic and career success.

Skill-based assessments, interactive trainings, business-centric projects, and comprehensive remediation engage students in mastering the latest Microsoft Office programs on their own, allowing instructors to spend class time teaching. SAM's efficient course setup and robust grading features provide faculty with consistency across sections. Fully interactive MindTap Readers integrate market-leading Cengage Learning content with SAM, creating a comprehensive online student learning environment.

Acknowledgments

I would like to thank the people who worked so hard to make this book possible. Special thanks to my developmental editor, Pam Conrad, for her excellent hard work and dedication in editing this text, and to my Content Developer, Kathy Finnegan, who has worked tirelessly in overseeing this project and made my task so much easier with her enthusiasm and good humor. Other people at Cengage Learning who deserve credit are Jim Gish, Senior Product Manager; Christian Kunciw, Manuscript Quality Assurance (MQA) Supervisor; and John Freitas, Serge Palladino, Danielle Shaw, and Susan Whalen, MQA testers.

Feedback is an important part of writing any book, and thanks go to the following reviewers for their helpful ideas and comments: David Doering, St. Louis Community College; Ravinder Kang, Highline Community College; Diana Kokoska, University of Maine at Augusta; Barbara Rader, University of Maryland; Sheryl Schoenacher, Farmingdale State College; Dave Sciuto, University of Massachusetts—Lowell; John Whitney, Fox Valley Technical College; and Dawn Wick, Southwestern Community College.

I want to thank my wife Joan for her support during this project and for my six children to whom this book is dedicated.
– Patrick Carey

Many thanks to everyone who helped in this revision. Pam Conrad, my sharp-eyed developmental editor, suggested improvements and asked a lot of important questions that helped me immeasurably in tightening up the material. The good advice of Kathy Finnegan, my Content Developer, kept me focused on the important aspects of the revision process, and she sweated a lot of the small stuff so I didn't have to. I'm also grateful to Jim Gish, the Senior Product Manager, for keeping the faith during the evolution of this revision. The staff at GEX Publishing Services made it all look amazing. And MQA testers Serge Palladino, Danielle Shaw, and Susan Whalen read everything through, completed all the steps, and gave smart feedback that removed many roadblocks for future users. Finally, thanks to my husband, Jason Bucy, for encouraging me to balance diving deep into XML with stepping away from the computer, getting outside, and enjoying the world with him.
– Sasha Vodnik

BRIEF CONTENTS

XML

TABLE OF CONTENTS

TUTORIAL 1

Creating an XML Document

Developing a Document for SJB Pet Boutique

OBJECTIVES

Session 1.1

- Describe the history of XML and the uses of XML documents
- Understand XML vocabularies
- Define well-formed and valid XML documents, and describe the basic structure of an XML document
- Create an XML declaration
- Work with XML comments
- Work with XML parsers and understand how web browsers work with XML documents

Session 1.2

- Create XML elements and attributes
- Work with character and entity references
- Describe how XML handles parsed character data, character data, and white space
- Create an XML processing instruction to apply a style sheet to an XML document
- Declare a default namespace for an XML vocabulary and apply the namespace to an element

Case | *SJB Pet Boutique*

SJB Pet Boutique in Delafield, Wisconsin, creates beautiful jewelry and clothing accessories "for pets and their humans." The boutique's top two best-selling items are holiday pet costumes, and matching pet collar and human necklace pendants.

During the past year, the boutique has received more requests for custom work. The owners would like to further develop this aspect of their business by making it available on the SJB Pet Boutique website. Patricia Dean manages the boutique's website. She has been investigating using Extensible Markup Language to organize information about the boutique's product line and the custom work offered. **Extensible Markup Language (XML)** is a markup language that can be extended and modified to match the needs of the document author and the data being recorded. XML has some advantages in presenting structured content such as descriptions of available customizations. Data stored in an XML document can be integrated with the boutique's website. Through the use of style sheets, Patricia can present XML data in a way that would be attractive to potential customers.

The boutique's website already takes advantage of many of the latest web standards, including HTML5 and CSS. Patricia would like to gradually incorporate XML into the website and increase the use of style sheets. As a first step, she has asked for your help in creating a document that will display a small part of the boutique's inventory using XML.

STARTING DATA FILES

tutorial
sjbCustomOrders.xml
sjbpet.css

review
resale.css
resale.txt

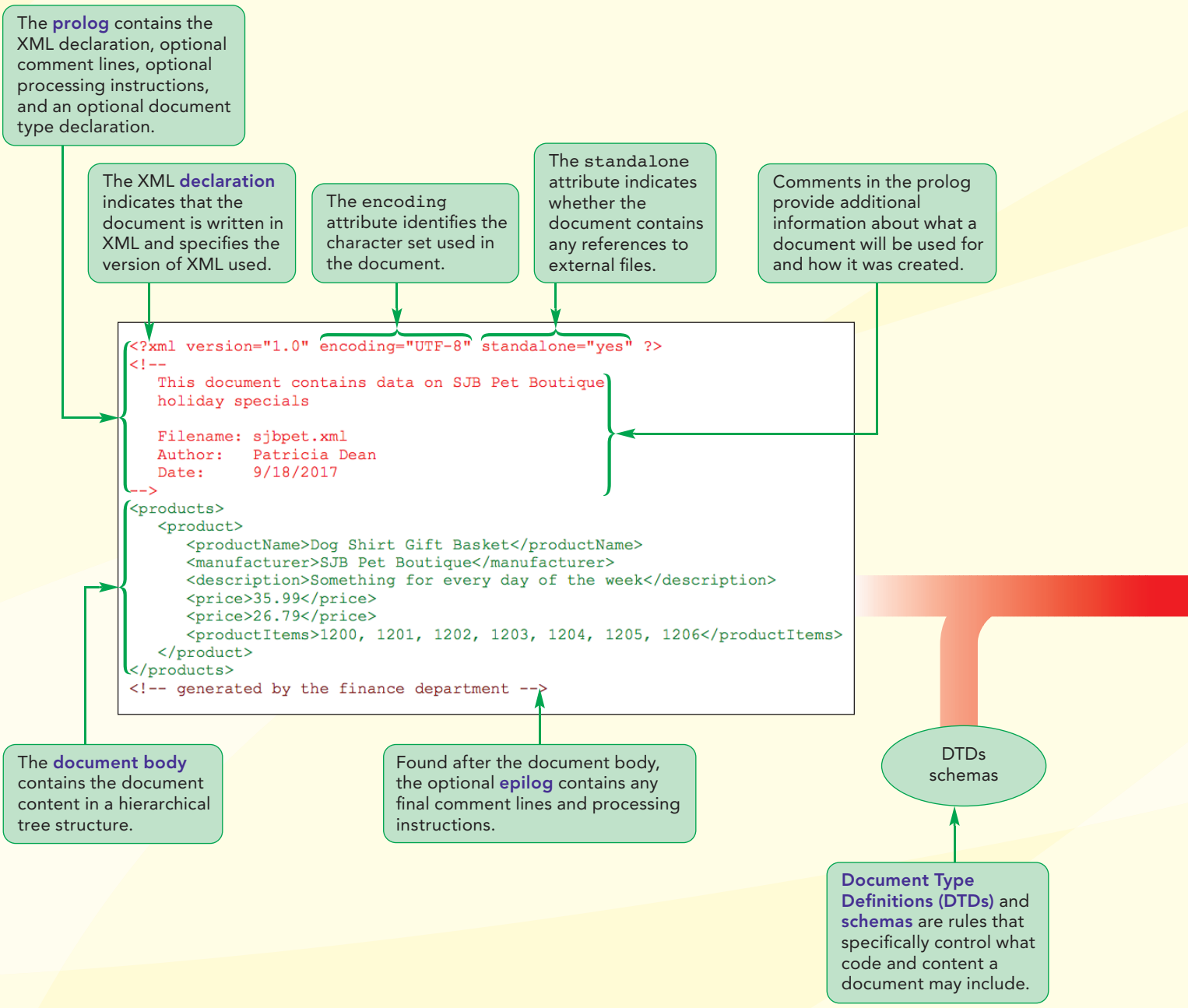
case1
membership.css
membership.txt

case2
chester.css
chester.txt

case3
recipe.txt
weekendFun.css
weekendFuntxt.xml

case4
rycheBooks.css
rycheBooks.txt
rycheBooksPrint.css

Session 1.1 Visual Overview:

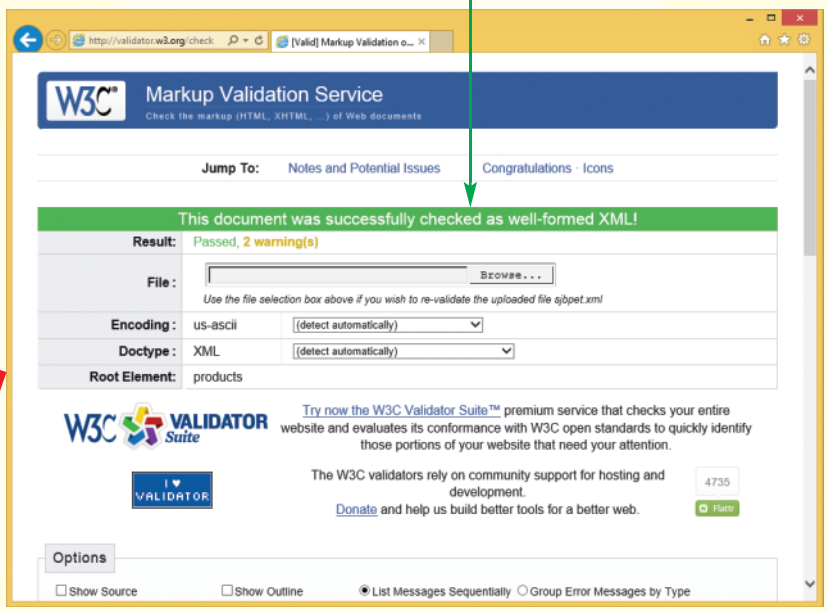


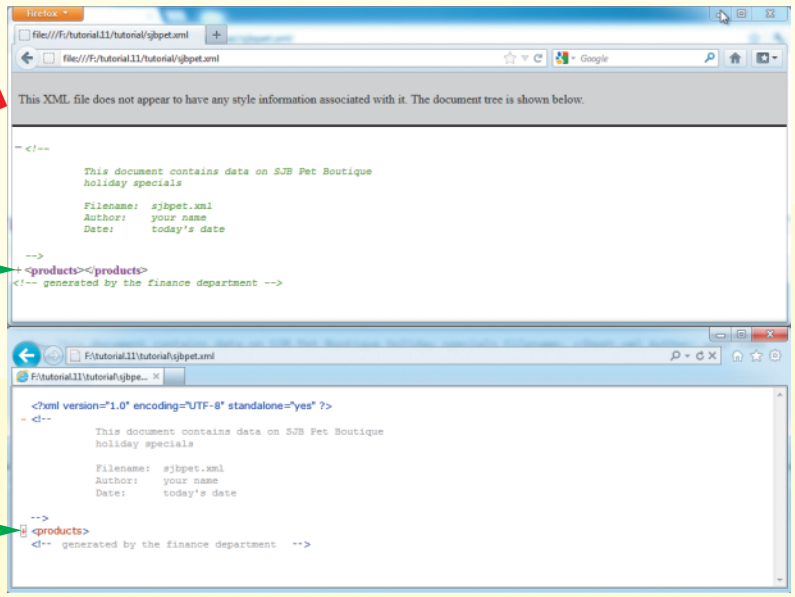
XML Overview

A **well-formed document** has no syntax errors and satisfies the general specifications for XML code defined by the World Wide Web Consortium (W3C).

An **XML parser** or **XML processor** interprets a document's code and verifies that it satisfies the W3C specifications.

XML parser





Most major browsers display XML content in a hierarchical format by default.

Introducing XML

The following short history lesson may help you better understand how XML fits in with today's technologies.

The Roots of XML

XML has its roots in **Standard Generalized Markup Language (SGML)**, a language introduced in the 1980s that describes the structure and content of any machine-readable information. SGML is device-independent and system-independent. In theory, this means that documents written in SGML can be used on almost any type of device under almost any type of operating system. SGML has been the chosen vehicle for creating structured documents in businesses and government organizations of all sizes.

Even though SGML provides tools to manage enormous projects, it is a difficult language to learn and to apply because of its power, scope, and flexibility. XML can be thought of as a lightweight version of SGML. Like SGML, XML is a language used to create vocabularies for other markup languages, but it does not have SGML's complexity and expansiveness. XML is a markup language that is extensible, so it can be modified to match the needs of the document author and the data being recorded. The standards for XML are developed and maintained by the **World Wide Web Consortium (W3C)**, an organization created in 1994 to develop common protocols and standards for sharing information on the World Wide Web. When the W3C started planning XML, it established a number of design goals for the language. The syntax rules of XML are easy to learn and easy to use, as shown in Figure 1-1.

Figure 1-1 Highlights of XML syntax rules

Syntax Rule	Application
Every XML element must have a closing tag.	Every element must have a closing tag. A self-closing tag is permitted.
XML tags are case sensitive.	Opening and closing tags (or start and end tags) must be written with the same case.
XML elements must be properly nested.	All elements can have child (sub) elements. Child elements must be in pairs and be correctly nested within their respective parent element.
Every XML document must have a root element.	Every XML document must contain a single tag pair that defines the root element. All other elements must be nested within the root element.
XML elements can have attributes in name-value pairs.	Each attribute name within the same element can occur only once. Each attribute value must be quoted.
Some characters have a special meaning in XML.	The use of certain characters is restricted. If these characters are needed, entity references or character references may be used. References always begin with the character "&" (which is specially reserved) and end with the character ";".
XML allows for comments.	Comments cannot occur prior to the XML Declaration. Comments cannot be nested.

XML Today

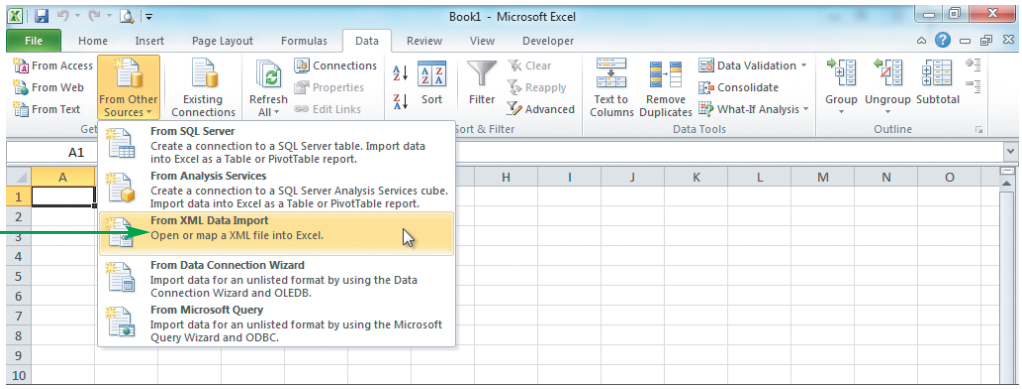
XML was originally created to structure, store, and transport information. Today, XML is still used for that purpose and has become the most common tool for data transmission among various applications. XML is used across a variety of industries, including accounting, banking, human resources, medical records, information technology, and insurance. Generally, it is used in all major websites, including major web services.

XML with Software Applications and Languages

Currently, many software applications such as Microsoft Excel and Microsoft Word, and server languages such as Java, .NET, Perl, and PHP, can read and create XML files. As of the 2007 releases of Microsoft Office and OpenOffice, users can exchange data among Office applications and enterprise systems using XML and file compression technologies. Not only are the documents universally accessible, but the use of XML also reduces the risk of damaged files. Figure 1-2 shows Microsoft Excel's built-in mechanism for importing an XML file into an Excel spreadsheet.

Figure 1-2 **Importing XML data into Microsoft Excel**

feature in Microsoft Excel to import XML data from an external source



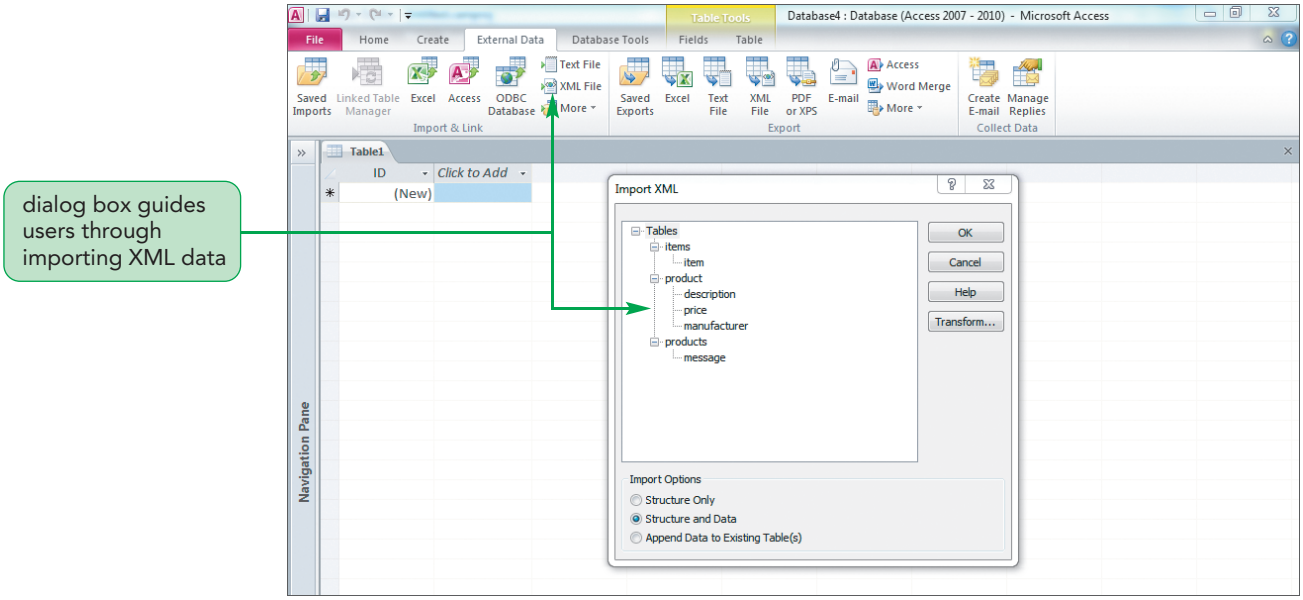
XML and Databases

Databases store data, and XML is widely used for data interchange. All major databases, including Microsoft Access, Oracle, and MySQL, can read and create XML files. The fact that XML isn't platform-dependent gives the language flexibility as technologies change.

XML and relational databases are tightly woven together in most web applications. However, the two use distinctly different models to structure data. The relational model used by relational databases is based on two-dimensional tables, which have no hierarchy and no significant order. By contrast, XML is based on hierarchical trees in which order is significant. In the relational model, neither hierarchy nor sequence may be used to model information. In XML, hierarchy and sequence are the main methods used to represent information. This is one of the more fundamental differences between the two models, but there are more.

On web pages, XML is very useful because the structure of XML closely matches the structure used to display the same information in HTML. Both HTML and XML use tags in similar ways, often creating distinctly hierarchical structures to present data to users. Most of the data for web pages comes from relational databases and it must be converted to appropriate XML hierarchies for use in web pages. For these reasons, it makes more sense to see XML as a tool that works in conjunction with databases, rather than as a competitor to them. Major databases support easy-to-use integration with XML. For instance, Figure 1-3 shows how Access has incorporated easy XML importing and exporting of data.

Figure 1-3 **Importing XML-formatted data into Access**



XML and Mobile Development

It is highly doubtful that when members of the W3C got together to discuss XML, they even considered mobile device development and the importance that XML would play in this area. In fact, mobile device platforms such as Google's Android and Apple's iOS use XML in a variety of ways.

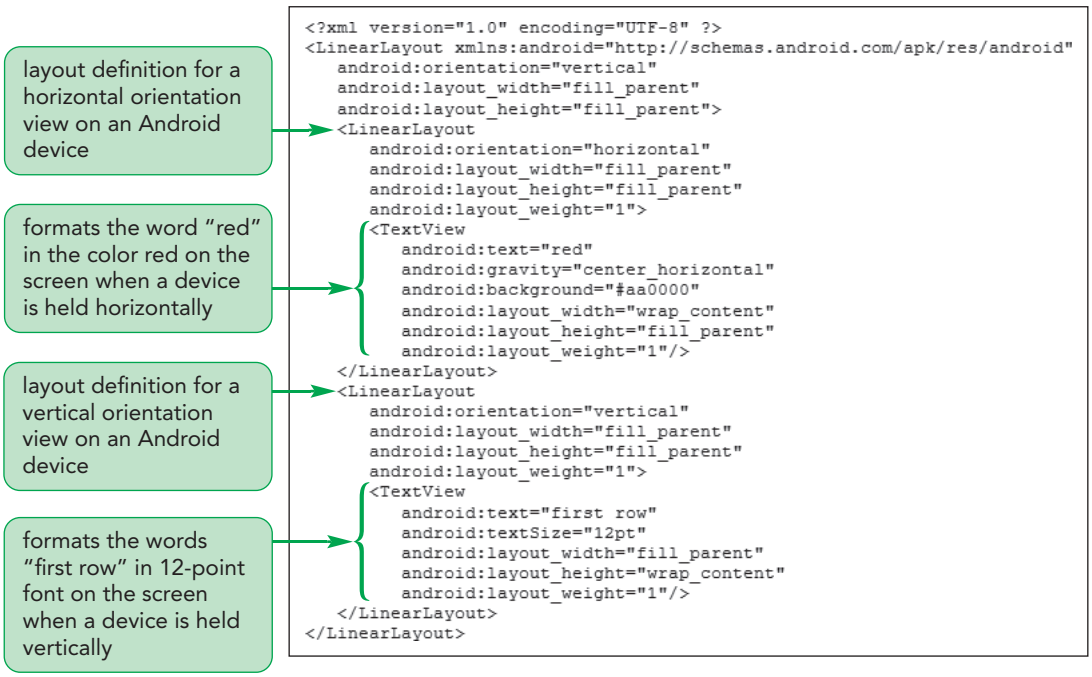
In iOS, Apple has built in the ability to import and export data classes in XML format. This makes it very easy to transfer information via XML. A popular use of XML in the iPhone is in a preference list or property list—commonly abbreviated as p-list—to organize data into named properties and lists of values, as shown in Figure 1-4.

Figure 1-4 **iOS p-list file written in XML**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
         "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
   <dict>
      <key>Accessories</key>
      <string>Collar Tag</string>
      <key>Shirt</key>
      <string>Week Day</string>
      <key>Bowls</key>
      <string>Ceramic 2 Holder</string>
   </dict>
</plist>
```

Android uses XML for screen layout and for working with data. Android provides a straightforward XML vocabulary for laying out content on the screen, allowing creation of XML layouts for different screen orientations, different device screen sizes, and different languages. Declaring an Android layout in XML makes it easier to visualize the structure of a user interface. Figure 1-5 shows an example of how Android uses XML to lay out the screen.

Figure 1-5 Android layout definitions written in XML



Creating an XML Vocabulary

HTML is an SGML application and is the foundation of web development. Like SGML, XML can be used to create **XML applications** or **vocabularies**, which are markup languages tailored to contain specific pieces of information. If Patricia wanted to create a vocabulary for the items in the SJB Pet Boutique product catalog, she might use XML to store the product information in the following format:

```
<productName>Dog Shirt Gift Basket</productName>
<manufacturer>SJB Pet Boutique</manufacturer>
<description>Something for every day of the week
</description>
<price currency="USD">$35.99</price>
<price currency="EUR">€26.79</price>
<productItems>1200, 1201, 1202, 1203, 1204, 1204, 1205, 1206
</productItems>
```

You'll explore the structure and syntax of this document further in the next session, but you can already infer a lot about the type of information this document contains even without knowing much about XML. You can quickly see that this file contains data on a product named "Dog Shirt Gift Basket," including its manufacturer, its description, its two selling prices, and the product numbers of the items it includes.

The `productName`, `manufacturer`, `description`, `price`, and `productItems` elements in this example do not come from any particular XML specification; rather, they are custom elements that Patricia might create specifically for one of the SJB Pet Boutique documents.

Patricia could create additional elements describing things such as the product number, the seller name, and the quantity on hand. In this way, Patricia could create her own XML vocabulary that deals specifically with product, manufacturer, and inventory data.

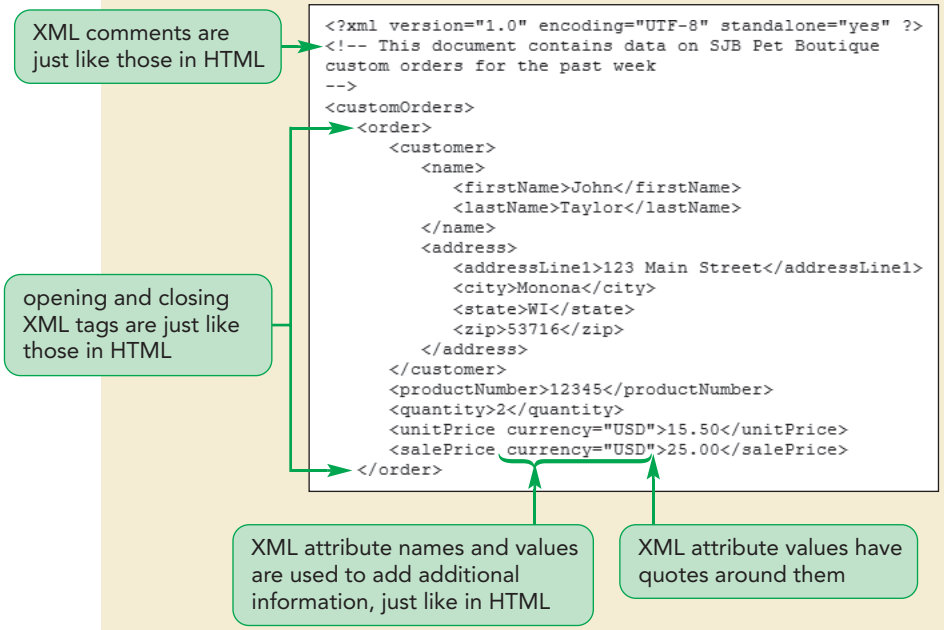
You'll start your work for Patricia by examining an XML document and comparing the similarities between HTML and XML documents.

Like HTML documents, XML documents can be created and viewed with a basic text editor such as Notepad or TextEdit. More sophisticated XML editors are available, and using them can make it easier to design and test documents. However, you can complete the project in this tutorial with a basic text editor.

To open an XML document in a text or XML editor:

1. In a text editor or XML editor, open **sjbCustomOrders.xml** from the xml01 ▸ tutorial folder where your data files are located. Figure 1-6 shows the contents of the sjbCustomOrders.xml document for the first order.

Figure 1-6 Opening an XML document




```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!-- This document contains data on SJB Pet Boutique
custom orders for the past week
-->
<customOrders>
   <order>
      <customer>
         <name>
            <firstName>John</firstName>
            <lastName>Taylor</lastName>
         </name>
         <address>
            <addressLine1>123 Main Street</addressLine1>
            <city>Monona</city>
            <state>WI</state>
            <zip>53716</zip>
         </address>
      </customer>
      <productNumber>12345</productNumber>
      <quantity>2</quantity>
      <unitPrice currency="USD">15.50</unitPrice>
      <salePrice currency="USD">25.00</salePrice>
   </order>
```

2. Examine the code, noting the similarities between an XML document and an HTML document, such as comments and opening and closing tags. You'll explore each aspect of an XML document's structure in the next session.
3. Close the file.

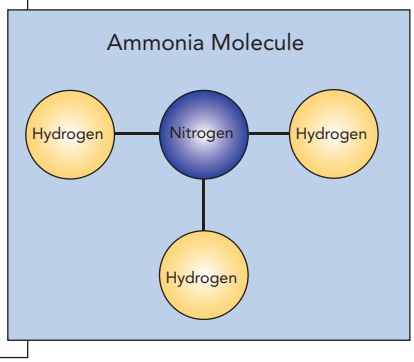
Standard XML Vocabularies

If Patricia wanted to share the vocabulary that she uses for SJB Pet Boutique with other companies, she might use a standard vocabulary that is accepted throughout the industry. You can think of a **standard vocabulary** as a set of XML tags for a particular industry or business function. As XML has grown in popularity, standard vocabularies continue to be developed across a wide range of disciplines.

For example, chemists need to describe chemical structures containing hundreds of atoms bonded to other atoms and molecules. To meet this need, an XML vocabulary called **Chemical Markup Language (CML)** was developed, which codes molecular information. Figure 1-7 shows an example of a CML document used to store information on the ammonia molecule.

Figure 1-7 **Ammonia molecule described using CML**

```
<molecule id="Ammonia">
   <formula>N H3</formula>
   <atomArray>
      <atom id="a1">
         <string builtin="elementType">N</string>
      </atom>
      <atom id="a2">
         <string builtin="elementType">H</string>
      </atom>
      <atom id="a3">
         <string builtin="elementType">H</string>
      </atom>
      <atom id="a4">
         <string builtin="elementType">H</string>
      </atom>
   </atomArray>
   <bondArray>
      <bond id="b1">
         <string builtin="atomRef">a1</string>
         <string builtin="atomRef">a2</string>
      </bond>
      <bond id="b2">
         <string builtin="atomRef">a1</string>
         <string builtin="atomRef">a3</string>
      </bond>
      <bond id="b3">
         <string builtin="atomRef">a1</string>
         <string builtin="atomRef">a4</string>
      </bond>
   </bondArray>
</molecule>
```



One of the more important XML vocabularies on the Internet is **Really Simple Syndication (RSS)**, which is the language used for distributing news articles or any content that changes on a regular basis. Subscribers to an RSS feed can receive periodic updates using a software program called a **feed reader** or an **aggregator**. Most current browsers contain some type of built-in feed reader to allow users to retrieve and view feeds from within the browser window. Most RSS feeds contain just links, headlines, or brief synopses of new information. Because an RSS file is written in XML, the RSS code follows the conventions of all XML documents. Figure 1-8 shows a segment of an RSS document.

Figure 1-8 **RSS document**

the item element contains news story information

the title element defines the title of the story

the description element contains a brief description of the news story

the link element defines the hyperlink to the story

```
<channel>
    <title>SJB Pet Boutique</title>
    <link>http://www.sjbpetboutique.com</link>
    <description>Everything for a pet and their owner</description>
    <item>
        <title>Holiday Fun</title>
        <link>http://www.sjbpetboutique.com/holiday</link>
        <description>New holiday products in stock</description>
    </item>
    <item>
        <title>Specials</title>
        <link>http://www.sjbpetboutique.com/specials</link>
        <description>New specials from SJB</description>
    </item>
    <item>
        <title>Clearance</title>
        <link>http://www.sjbpetboutique.com/clearance</link>
        <description>Clearance specials from SJB</description>
    </item>
</channel>
```

Figure 1-9 lists a few of the many vocabularies that have been developed using XML.

Figure 1-9 XML vocabularies

XML Vocabulary	Description
Bioinformatic Sequence Markup Language (BSML)	Coding of bioinformatic data
Extensible Hypertext Markup Language (XHTML)	HTML written as an XML application
Mathematical Markup Language (MathML)	Presentation and evaluation of mathematical equations and operations
Music Markup Language (MML)	Display and organization of music notation and lyrics
Weather Observation Definition Format (OMF)	Distribution of weather observation reports, forecasts, and advisories
Really Simple Syndication (RSS)	Distribution of news headlines and syndicated columns
Synchronized Multimedia Integration Language (SMIL)	Editing of interactive audiovisual presentations involving streaming audio, video, text, and any other media type
Voice Extensible Markup Language (VoiceXML)	Creation of audio dialogues that feature synthesized speech, digitized audio, and speech recognition
Wireless Markup Language (WML)	Coding of information for smaller-screened devices, such as PDAs and cell phones

TIP

You can learn more about several standard XML vocabularies at the W3C site, www.w3.org/XML/.

One of the more important XML vocabularies is **XHTML (Extensible Hypertext Markup Language)**, which is a reformulation of HTML as an XML application. You'll examine some properties of XHTML as you learn more about XML in the upcoming tutorials. Don't worry if you find all of these acronyms and languages a bit overwhelming.

DTDs and Schemas

For different users to share a vocabulary effectively, rules must be developed that specifically control what code and content a document from that vocabulary might contain. This is done by attaching either a Document Type Definition (DTD) or a schema to the XML document containing the data. Both DTDs and schemas contain rules for how data in a document vocabulary should be structured. A DTD defines the structure of the data and, very broadly, the types of data allowable. A schema more precisely defines the structure of the data and specific data restrictions.

For example, Patricia can create a DTD or schema to require her documents to list the name, the manufacturer, a description, a list of prices, and a list of product items for each product in the SJB Pet Boutique inventory. DTDs and schemas are not required, but they can be quite helpful in ensuring that your XML documents follow a specific vocabulary. The standard vocabularies listed in Figure 1-9 all have DTDs to ensure that people in a given industry or area all work from the same guidelines.

To create a DTD or a schema, you simply need access to a basic text editor. You'll explore how to create DTDs and schemas in later tutorials.

Well-Formed and Valid XML Documents

To ensure a document's compliance with XML rules, it can be tested against two standards—whether it's well formed, and whether it's valid. A well-formed document contains no syntax errors and satisfies the general specifications for XML code as laid out by the W3C. At a minimum, an XML document must be well formed or it will not be readable by programs that process XML code.

If an XML document is part of a vocabulary with a defined DTD or schema, it also must be tested to ensure that it satisfies the rules of that vocabulary. A well-formed XML document that satisfies the rules of a DTD or schema is said to be a **valid document**. In this tutorial, you'll look only at the basic syntax rules of XML to create well-formed documents. You'll learn how to test documents for validity in later tutorials.

PROSKILLS

Problem Solving: Designing for Efficiency and Effectiveness

Although XML can do many different things, it is used most effectively to communicate data. In this respect, XML and databases go hand-in-hand—XML communicates data, and databases store data. XML delivers structured information in a generic format that's independent of how that information is used. As a result, the data does not rely on any particular programming language or software. XML developers have the freedom to work with a wide range of applications, devices, and complementary languages. A much larger benefit to the structural and logical markup is the ability to reuse portions of the information easily in any context where the information is structurally valid. Because XML focuses on communicating the data, the overall structure is simple and easy to design and maintain. This approach allows for a high level of efficiency and effectiveness, which in the long term reduces the amount of time and money spent on development and maintenance.

Creating an XML Document

Now that you're familiar with the history and theory of XML, you're ready to create your first XML document.

The Structure of an XML Document

An XML document consists of three parts—the prolog, the document body, and the epilog. The prolog includes the following parts:

- **XML declaration**: indicates that the document is written in the XML language
- **Processing instructions** (optional): provide additional instructions to be run by programs that read the XML document
- **Comment lines** (optional): provide additional information about the document contents
- **Document type declaration (DTD)** (optional): provides information about the rules used in the XML document's vocabulary

Figure 1-10 illustrates the structure of a prolog.