XML

Tutorial Two: Validating Documents with DTDs

A Guide to this Instructor's Manual:

We have designed this Instructor's Manual to supplement and enhance your teaching experience through classroom activities and a cohesive chapter summary.

This document is organized chronologically, using the same heading in **blue** that you see in the textbook.Under each heading you will find (in order): Lecture Notes that summarize the section, Figures and Boxes found in the section (if any), Teacher Tips, Classroom Activities, and Lab Activities. Pay special attention to teaching tips, and activities geared towards quizzing your students, enhancing their critical thinking skills, and encouraging experimentation within the software.

In addition to this Instructor's Manual, our Instructor's Resources also include PowerPoint Presentations, Test Banks, and other supplements to aid in your teaching experience.

For your students:

Our latest online feature, CourseCasts, is a library of weekly podcasts designed to keep your students up to date with the latest in technology news.Direct your students to http://coursecasts.course.com, where they can download the most recent CourseCast onto their 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 and sorts through and aggregates 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.

Table of Contents

Tutorial Objectives

Students will have mastered the material in Tutorial Two when they can:

Session 2.1

- Review the principles of data validation
- Create a DOCTYPE
- Declare XML elements and define their content
- Define the structure of child elements

Session 2.2

- Declare attributes
- Set rules for attribute content
- Define optional and required attributes
- Validate an XML document

Session 2.3

- Place internal and external content in an entity
- Create entity references
- Understand how to store code in parameter entities
- Create comments in a DTD
- Understand how to create conditional sections
- Understand how to create entities for non-character data
- Understand how to validate standard vocabularies

XML 68: Creating a Valid Document

LECTURE NOTES

- Explain what a DTD can do when used in conjunction with an XML parser that supports data validation
- Discuss what a DOCTYPE is and its purpose
- Review the two parts a DOCTYPE can be divided into
- Explain the reason for creating a public identifier for a DTD
- Illustrate the differences between external and internal subsets used in a DOCTYPE

BOXES

- TIP: The root value in the DOCTYPE must match the name of the XML document's root element; otherwise,parsers will reject the document as invalid.(XML 71)
- InSight: Understanding URIs (XML 72)
- ProSkills: Written Communication: Interpreting Public Identifiers (XML 73)
- Reference: Declaring a DTD (XML 74)

FIGURES

- Figure 2-1, Figure 2-2, Figure 2-3, Figure 2-4, Figure 2-5

TEACHER TIP

Provide students with a table like the one in Figure 2-1, marking which fields, if any, are optional, and ask them to devise the corresponding structure for the data in the document, like the one shown in Figure 2-3.

CLASSROOM ACTIVITIES

1. GroupActivity: Present students with scenarios and ask them to determine the kind of DOCTYPE statement the scenario would require. For example, if a company is using a standard XML vocabulary like MathML, what kind of statement is required? (Answer: A public identifier) If a company has

developed a customized XML parser to read its XML documents, what is required? (Answer: An external file with a system identifier)

2. Quick Quiz:

- A DTD can be used to ___________.
    a. ensure that all required elements are present in a document
    b. prevent undefined elements from being used in a document
    c. define default values for attributes
    d. All of these are correct.

    (Answer: D)
- A(n) ___________ subset is a set of declarations placed within the XML document. (Answer: internal)
- True/False: An internal DTD can be easily used as a common DTD among many documents, forcing them to use the same elements, attributes, and document structure. (Answer: False)

XML 75: Declaring Document Elements

LECTURE NOTES

- Explain the meaning of an element type declaration
- Illustrate the differences between elements that can contain any data, empty elements, and parsed character data

BOXES

- TIP: Element declarations must begin with <!ELEMENT in all uppercase letters and not <!Element or <!element. (XML 75)
- Reference: Specifying Types of Element Content (XML 77)

FIGURES

- Figure 2-6

TEACHER TIP

Ask students to brainstorm examples of elements that would fall under each of the five types of content models. The more examples that you can go over with them, the more it will help students understand the various types of elements.

CLASSROOM ACTIVITIES

1. GroupActivity: Present students with examples of elements that have been declared to be of a particular type and then are invalid, and ask students to explain how to correct the element so it complies with its assigned type.

2. Quick Quiz:

- An element declaration can specify all of the following except ___________.
    a. the reserved symbols that the element name can contain
    b. the element’s name
    c. what kind of content the element can contain
    d. the order in which elements appear in the document

    (Answer: A)
- An element declared as ___________ cannot store any content. (Answer: empty)

- True/False: Generally, elements are declared as being of type ANY or #PCDATA. (Answer: False)

XML 78: Working with Child Elements

LECTURE NOTES

- Explain how to declare an element that contains child elements
- Discuss the two ways to declare an element that has multiple child elements
- Define a modifying symbol and describe its uses

BOXES

- Reference: Specifying Child Elements (XML 80)
- TIP: You can specify that an element contains a minimum number of a child element by entering duplicate elements equal to the minimum number and adding a + to the last one. (XML 80)
- Reference: Applying Modifying Symbols (XML 81)
- InSight: DTDs and Mixed Content (XML 83)
- Review:Session 2.1 Quick Check (XML 83)

FIGURES

- Figure 2-7

TEACHER TIP
Present students with examples of elements that have been declared to be of a particular type and then are invalid, and ask students to explain how to correct the elements so they comply with their assigned type. This will help them to understand child elements better.

CLASSROOM ACTIVITIES

1. GroupActivity: Describe in words the child elements a parent element must contain (for example,“a citizen must have a state or a province code, but not both”) and ask students to write the element declarations that correspond to the examples presented. Another exercise would be to present a list of child elements (e.g., name, company) and ask what element declaration allows the given list of child elements, making sure to remind students that modifying symbols add extra possible solutions.
2. Quick Quiz:
   - True/False: For content that involves multiple child elements, you can specify the elements in either a sequence or a choice of elements. (Answer: True)
   - A __________ is a list of elements that follow a defined order.
     a. choice
     b. source
     c. sequence
     d. series
     (Answer: C)

- A(n) __________ symbol specifies the number of occurrences of each child element. (Answer: modifying)

LAB ACTIVITY

Have students begin to construct a DTD for the following situation:

- Students who have student ID, first name, last name, email, phone
- Courses which have course ID, description, and number of credits
- Grade records which have record ID, student ID, course ID, and grade

They should create the main elements and then specify the children. Students may have more than one phone number and email address.

XML 86: Declaring Attributes

LECTURE NOTES

- Review the various purposes of an attribute list declaration
- Discuss the different ways to declare elements with multiple attributes

BOXES

- Reference: Declaring Attributes in a DTD (XML 87)

FIGURES

- Figure 2-8, Figure 2-9

TEACHER TIP

Stress that right now the attribute declarations would be rejected since the attribute types are not specified. As with the other sections, go through several examples to help students understand the process.

CLASSROOM ACTIVITIES

1. Class Discussion: Ask students to come up with attributes for a course element. What attributes would they have? How would they be specified in a DTD?
2. Quick Quiz:
   - True/False: If a parser encounters more than one declaration for the same attribute, it only recognizes the second statement and ignores the first. (Answer: False)
   - Which of the following is true of attribute-list declarations?
     a. They must be located at the beginning of the DTD.
     b. They must be located at the end of the DTD.
     c. They must be located adjacent to the declaration for the element with which they are associated.
     d. They can be located anywhere within the DTD.

     (Answer: D)
   - To enforce attribution properties on a document, you must add a(n) __________ to the document's DTD. (Answer: attribute-list declaration)

XML 89 Working with Attribute Types

LECTURE NOTES

- Discuss different data types that DTDs support for attribute values
- Explain the differences in character data and enumerated types of attribute values

- Discuss the four types of tokens that DTDs support

BOXES

- TIP: You'll learn more about default attribute values later in this tutorial. (XML 90)
- TIP: Because an ID must be valid XML names, it cannot begin with a number. Commonly used identifiers, such as Social Security numbers, must be prefaced with one or more alphabetical characters, such as SS123-45-6789. (XML 93)

FIGURES

- Figure 2-10, Figure 2-11, Figure 2-12, Figure 2-13

TEACHER TIP

Point out the fact that quantities, to which students would expect to assign a data type of "integer" or "number," are expressed as being of type "CDATA." If they have programmed at all before, they may be confused about this. Caution students that the use of the IDREF token requires that there be a matching attribute to be cross-referenced, or the document will be rejected as invalid.

CLASSROOM ACTIVITIES

1. GroupActivity: Ask students to brainstorm scenarios in which enumerated-type attributes would be useful because the values must be limited to a known, bounded set.
2. Quick Quiz:
    - The simplest form for attribute text is __________.
        a. CDATA
        b. ID
        c. enumerated list
        d. NMTOKEN

        (Answer: A)
    - Attributes that are limited to a set of possible values are known as __________ types. (Answer: enumerated)
    - True/False: Tokens are used when an attribute value refers to a file containing nontextual data, like a graphic image or a video clip. (Answer: False)

XML 95: Working with Attribute Defaults

LECTURE NOTES

- Review the four types of attribute defaults and explain their uses
- Demonstrate how to declare each of the four types

BOXES

- TIP: If you specify a default value for an attribute, omit #REQUIRED and #IMPLIED from the attribute declaration so parsersdon't reject the DTD. (XML 95)
- Reference: Specifying the Attribute Default (XML 96)

FIGURES

- Figure 2-14, Figure 2-15

TEACHER TIP

In studying XML, students often worry that there are too many options to remember. Provide more examples and ask students to contribute a few; thiswill help students to remember the various options.

CLASSROOM ACTIVITIES

1. Class Discussion: Ask students to break into teams. Have them devise scenarios that allow them to provide examples of the uses of each of the attribute defaults in Figure 2-14. Have them discuss their results with each other.
2. Quick Quiz:
    - You can indicate that a given element must always have a particular attribute by adding the ___________ value to the attribute declaration.
        a. #FIXED
        b. #REQUIRED
        c. #IMPLIED
        d. #MANDATED
        (Answer: B)
    - The ___________ value for an attribute indicates that the use of this attribute is optional. (Answer: #implied)

XML 97: Validating an XML Document

LECTURE NOTES

- Discuss the process of validating an XML document
- Explain how to correct common errors

BOXES

- TIP: You can also press F7 (Windows) of fn+F7(Mac) to validate a document in Exchanger XMLEditor. (XML 98)
- ProSkills: Problem Solving: ReconcilingDTDs and Namespaces (XML 102)
- Review:Session 2.2 Quick Check (XML 103)

FIGURES

- Figure 2-16, Figure 2-17, Figure 2-18, Figure 2-19, Figure 2-20

TEACHER TIP
As the text example shows, go through a document with errors and let students see how to correct them. This really helps them prepare for when they are on their own and validating their work.

CLASSROOM ACTIVITIES

1. GroupActivity: If you have a computer with a projection device, use it to open valid XML code and make changes to it, creating some errors on purpose, in order to illustrate the kinds of errors students will become familiar with as they do more coding in XML.
2. Quick Quiz:
    - True/False: You can use Internet Explorer's MSXML parser to validate your XML document yourself. (Answer: False)

LAB ACTIVITY

- Students should now add attributes to their DTD. For “phone,” they should add a phoneType attribute that identifies the phone as home, work, or cell. For “email,” there should be an emailType attribute that identifies the email address as home, work, or school. For “first name,” they can add an attribute of title. Also, the record “no” for grade should include semester and year attributes, which will be used to indicate which semester and year the grade was given for a course. The grade record attributes are required. Have the students specify default values where appropriate.
- Students should create an XML file to go along with this (or you can provide one if you like), and then they should validate their work.

XML 106: Introducing Entities

LECTURE NOTES

- Identify the five built-in entities supported by XML
- Discuss the usefulness of creating customized entities

FIGURES

- Figure 2-21

TEACHER TIP
Stress that entities can be used to avoid data entry errors. An analogy from outside programming is macros, such as Microsoft Word’s AutoText feature, which allows a few keystrokes to represent a longer character string.

CLASSROOM ACTIVITIES

1. Class Discussion: Can students think of scenarios when they might use a customized entity?
2. Quick Quiz:
    - Which of the following is not a built-in XML entity?
        a. &lt;
        b. &amp;
        c. &quot;
        d. &posit;
        (Answer: D)
    - True/False: Using entities can help you to avoid data errors. (Answer: True)

XML 106: Working with General Entities

LECTURE NOTES

- Identify the differences between general, external, and internal entities
- Explain that the content referenced by an entity can be either parsed or unparsed
- Explain how to create a parsed entity
- Illustrate how to reference a general entity

BOXES

- TIP: For a long text string that will be repeated throughout an XML document, avoid data entry errors by placing the text string in its own entity.(XML 106)

- TIP: Including markup tags in an entity value lets you create a section of XML code and content, and insert it once or multiple times into a document. (XML 107)
- Reference: Declaring and ReferencingParsed Entities (XML 108)

FIGURES

- Figure 2-22, Figure 2-23,Figure 2-24,Figure 2-25,Figure 2-26,Figure 2-27

TEACHER TIP
Caution students that if they use an external file, it must contain well-formed XML content and no XML declaration. Emphasize that students must handle the & and % symbols with careful consideration.

CLASSROOM ACTIVITIES

1. GroupActivity: Present students with a series of types of content and ask them to characterize what type of entity could be used to reference them (e.g., a paragraph of text inside the DTD would be an internal parsed entity; a video file would be an unparsed external entity).
2. Quick Quiz:
    - True/False: You cannot use the hyphen character in an entity's value because this is the symbol used for inserting parameter entities. (Answer: False)
    - Which of the following entity types would reference a video file?
        a. unparsed internal
        b. parsed internal
        c. unparsed external
        d. parsed external
        (Answer: C)
    - A(n) __________ entity is an entity that references content to be used within an XML document. (Answer: general)

XML 113: Working with Parameter Entities

LECTURE NOTES

- Explain that a parameter entity is used to insert content into the DTD because it can be used to break it into modules
- Demonstrate how to declare parameter entities and then reference them

BOXES

- TIP: Note that when declaring a parameter entity, you include a space after the %; but when referencing a parameter entity, there is no space between the % and the entity name. (XML 114)
- Reference: Declaring and ReferencingParameter Entities (XML 115)

FIGURES

- Figure 2-28

TEACHER TIP
Make sure to point out that not all browsers support external entities in combination with DTDs.

CLASSROOM ACTIVITIES

1. GroupActivity: Have students come up with scenarios when parameters may be used, and then have themcreate declarations for the scenarios.
2. Quick Quiz:
   - You use a(n) __________ to insert content into a DTD. (Answer: parameter entity)
   - True/False: Firefox browsers allow external entities in combination with DTDs. (Answer: False)

XML115: Inserting Comments into a DTD

LECTURE NOTES

- Explain how to put comments into a DTD
- Give examples of how documenting code in comments is beneficial

BOXES

- ProSkills: Teamwork: Documenting Shared Code with Comments (XML 115)

FIGURES

- Figure 2-29

TEACHER TIP
Let students know that comments can also be useful to explain the browser compatibility issues.

CLASSROOM ACTIVITIES

1. Class Discussion: Have students brainstorm ways to use comments in their DTDs.

XML 116: Creating Conditional Sections

LECTURE NOTES

- Explain how to create a conditional section in an external DTD to either include or ignore sections of the declaration
- Discuss the best ways to use conditional sections

TEACHER TIP
Remind students thatconditional sections cannot be applied to internal DTDs.

CLASSROOM ACTIVITIES

1. Class Discussion: If you have a computer with a projection device, show a couple of examples of conditional sections. Can students think of other uses?
2. Quick Quiz:
   - True/False: One effective way of creating IGNORE sections is to create a parameter entity that defines when the section should be included or not. (Answer: True)

XML 117: Working with Unparsed Data

LECTURE NOTES

- Discuss the steps involved when working with unparsed data
- Demonstrate how to declare a notation to identify the data type of the unparsed data
- Explain how to create an unparsed entity that references specific items that use a notation

BOXES

- TIP: As an alternative to notations, you can place a URL that lists a resource for nontextual content in an element or attribute, and then allow your application to work with that element or attribute value directly. (XML 118)
- Reference: Declaring an Unparsed Entity (XML 119)

TEACHER TIP
Stress that the notation will not necessarily enable an application to open. Current Web browsers may not support notation mechanisms to display files from helper applications.

CLASSROOM ACTIVITIES

1. GroupActivity: Ask students to brainstorm scenarios in which unparsed data would be useful to an XML document.
2. Quick Quiz:
    1. True/False: Current Web browsers support mechanisms for validating and rendering unparsed data declared in the DTDs of XML documents. (Answer: False)
    2. True/False: An alternate to notations is to place a URL in an element or attribute. (Answer: True)

XML 119: Validating Standard Vocabularies

LECTURE NOTES

- Discuss how to validate a standard XML vocabulary by specifying an external DTD
- Review the different DOCTYPEs for standard vocabularies

BOXES

- InSight: Advantages and Disadvantages of DTDs (XML 121)
- Review:Session 2.3 Quick Check (XML 121)

FIGURES

- Figure 2-30, Figure 2-31

TEACHER TIP
The online validator by W3C is a great tool. Show how to use it to validate and fix errors so that students can become familiar with common errors.

CLASSROOM ACTIVITIES

1. GroupActivity: If you have a computer with a projection device, use it to show a portion of the XHTML 1.0 strict DTD from W3C.org for another element or elements.
2. Quick Quiz:
    - True/False: Most standard vocabularies make their DTDs available online for inspection. (Answer: True)
    - True/False:The W3C does not provide an online validator.(Answer: False)

LAB ACTIVITY

Students are going to modify the grades section of their XML document to include a reference to the professor. They should modify their DTD to add the professor element. They should create elements for five professors called prof1, prof2, prof3, prof4, and prof5. The elements should contain the professor's name and title (i.e., John Smith, Associate Professor). Then, in the XML, they should just reference the entity when referring to the professor. They should then add appropriate comments to their work. Finally, have the students validate their work again to make sure it is correct.

End of Tutorial Material

- **SAM Assessment, Training, and Projects:** This text is available with SAM Assessment, Training, and Projects that map directly to the learning objectives covered in each chapter. SAM's active, hands-on training and skill-based assessment help you master Microsoft Office skills. SAM Projects let you apply skills in real-world scenarios using the actual Microsoft Office applications. Immediate feedback and comprehensive study guides give you the practice and support you need to succeed. To obtain a SAM account, visit www.cengagebrain.com or contact your instructor or bookstore for additional information.
- **Review Assignments:** Review Assignments provide students with additional practice of the skills they learned in the tutorial using the same tutorial case, with which they are already familiar. These assignments are designed as straight practice only and should not include anything of an exploratory nature.
- **Case Problems:**A typical NP tutorial has four Case Problems following the Review Assignments. Short tutorials can have fewer Case Problems (or none at all); other tutorials may have five Case Problems. The Case Problems provide further hands-on assessment of the skills and topics presented in the tutorial, but with new case scenarios. There are four types of Case Problems:
    - **Apply**. In this type of Case Problem, students apply the skills that they have learned in the tutorial to solve a problem. "Apply" Case Problems can include "Explore" steps, which go a bit beyond what was presented in the tutorial, but should include only 1 or 2 Explore steps if any at all.
    - **Create**. In a "Create" Case Problem, students are either shown the end result, such as a finished Word document, and asked to create the document based on the figure provided; or, students are asked to create something from scratch in a more free-form manner.
    - **Challenge**.A "Challenge" Case problem involves 3 or more Explore steps. These steps challenge students by having them go beyond what was covered in the tutorial, either with guidance in the step or by using online Help as directed.
    - **Research**.In this type of Case Problem, students need to go to the Web to find information that they will incorporate somehow in their work for the Case Problem.

  A tutorial does not have to include each of the four types of Case Problems; rather, the tutorial's content should dictate the types of exercises written. It's possible, therefore, that some tutorials might have three Case Problems of one type and only one Case Problem of a different type. To the extent possible, the first Case Problem in a tutorial should be an "Apply," so that the Case Problems progress in degree of difficulty.
- **ProSkills Exercises:**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. The goal of these exercises is to enhance students'

understanding of the soft skills and how to apply them appropriately in real-world, professional situations that also involve software application skills. ProSkills exercises are offered at various points throughout a text, encompassing the concepts and skills presented in a standalone tutorial or a group of related tutorials.

Glossary of Key Terms

- #IMPLIED (XML 85)
- #PCDATA (XML 67)
- #REQUIRED (XML 85)
- attribute declaration (XML 84)
- attribute-list declaration (XML 84)
- CDATA data type (XML85)
- conditional section (XML 116)
- DOCTYPE (XML71)
- document type declaration (XML71)
- element declaration (XML66)
- element type declaration (XML66)
- enumerated type (XML 85)
- external entity (XML106)
- external subset (XML 71)
- formal public identifier (XML 72)
- general entity (XML 104)
- ID token (XML 92)
- ID token type (XML 85)
- IDREF token (XML 93)
- IDREF token type (XML 85)
- internal entity (XML 104)
- internal subset (XML 71)
- mixed content (XML 83)
- modifying symbol (XML 67)
- module (XML 113)
- NMTOKEN (XML94)
- NMTOKENS (XML95)
- notation (XML 92) (XML 117)
- parameter entity (XML106) (XML 113)
- parsed entity (XML 104)
- public identifier (XML 72)
- sequence (XML 67)
- system identifier (XML 71)
- tokenized type (XML 92)
- token (XML 92)
- unparsed entity (XML106)

Top of Document