Introduction to the WordPerfect XML tutorial

Introduction to the WordPerfect XML tutorial

WordPerfect provides the tools you need to create and format XML documents within a structured, manageable environment.

The lessons in this tutorial step you through the process of creating XML documents. Lesson 1 introduces the basic concepts behind XML. Lessons 2 to 5 guide you through the creation of a business letter in XML. In Lesson 6, you learn how to add a table of contents and cross-references to an XML document.

Each lesson builds on the previous lesson; so the best way to learn XML is to complete the tutorial lessons in the order presented.

Before you start the first lesson, look through all the lessons to get an idea of what WordPerfect can do. If you are new to XML, be sure to read the information in Understanding Basic XML Concepts.

If you need help as you go through the tutorial, use the WordPerfect XML online Help.

{button ,AL(`PRC Introduction to the WordPerfect XML tutorial;',0,"Defaultoverview",)} Related Topics

Before you begin

Before you begin the lessons, you should have mastered the basics of WordPerfect. You should also be familiar with Microsoft Windows 98 Second Edition, Windows ME, Windows 2000, Windows XP, or Windows NT－the operating systems under which WordPerfect runs. You should know how to open and close Windows applications, browse for files, use a mouse, and use standard Windows menus and dialog boxes. For more information about Windows, see the Microsoft Windows documentation.

{button ,AL(`PRC Introduction to the WordPerfect XML tutorial;',0,"Defaultoverview",)} Related Topics

Following the tutorial

The six lessons in this tutorial are designed to help you learn the concepts and basic skills required to work with XML. Completing the lessons, will give you the knowledge and confidence you need to create your own XML documents. Here's what you'll learn.

Lesson 1: Understanding basic XML concepts

In this lesson, you'll learn about XML, the basic components of WordPerfect that you'll use with XML, and the process involved in creating an XML document.

Lesson 2: Getting started

In this lesson, you'll learn about document type definitions (DTDs) and explore the WordPerfect DTD Compiler window. You'll also learn how to define an XML project and compile a DTD.

Lesson 3: Creating an XML document

In this lesson, you'll create an XML document. You'll explore the WordPerfect XML editing environment and learn how to insert elements and tags.

Lesson 4: Creating, editing, and saving document layout

In this lesson, you'll explore the WordPerfect XML Project Designer window, open a layout, and format elements.

Lesson 5: Using advanced layout and formatting

In this lesson, you'll modify the DTD and insert both a table and a graphic into an XML document.

Lesson 6: Managing XML documents

In this lesson, you'll create a table of contents for an existing XML document (Overview.xml) and add cross-references.

{button ,AL(`PRC Introduction to the WordPerfect XML tutorial;',0,"Defaultoverview",)} Related Topics

Lesson 1: Understanding basic XML concepts

Lesson 1: Understanding basic XML concepts

It is important to understand the basic concepts of XML. In this lesson, you'll learn about

- XML and the advantages of XML
- the components of an XML document
- the components of WordPerfect that you'll use with XML
- the process of creating an XML document

You'll need about 20 minutes to complete this lesson.

{button ,AL(`PRC Lesson 1 Understanding basic XML concepts;',0,"Defaultoverview",)} Related Topics

Understanding XML

What is XML?

XML is a simplified version of SGML; it gives you the power of SGML without the complexity. XML is an open international standard issued by the World Wide Web Consortium (W3C). Like SGML, XML provides a platform and application-independent environment for defining document architecture and document markup.

Why use XML?

There are many advantages to using XML.

- XML eliminates the problem of having to visually search a document to find a specific type of information, such as keywords or prices. With XML, complex searches can be done on minute elements of information, such as "all titles" or "the preface" in the document structure.
- XML standardizes information by formalizing and enforcing document structure and formatting rules. These rules are specified in the Document Type Definition (DTD) associated with an XML document. While a DTD is not an essential requirement for creating XML documents, the use of a DTD ensures that an XML document is valid and well-formed. This is done by validating the XML document in accordance with the rules laid out in the DTD.
- XML makes information reuse easier. For example, an order form document might contain item names, order dates, part numbers, prices, quantity ordered, the selling agent's name, and other information for a company. The shipping department needs to know order dates, part numbers, and quantity, but doesn't require the prices or the agent's name. On the other hand, the accounting department needs the name, quantity, prices, and agent's name, but has no use for the part numbers or order dates. In this case, both departments can use the same document and simply extract only the information that they require.
- XML allows information from a single source to be output in multiple ways. For example, you can print a book as a hard-bound version and as a paperback version using a smaller page size, font, and different margins without changing the content file.
- XML lets you to add meaning to document information. In contrast with HTML, which is a fixed markup language, Both SGML and XML are both flexible markup languages. A document created in HTML uses a fixed set of tags such as <h1>, <h2>, <h3>, and <p> (i.e., heading 1, heading 2, heading 3, and paragraph).

  SGML and XML, however, lets you create your own document tags. These descriptive tags, let you add meaning to the document information. For example, a document created in XML could use the tags <automobile model>, <year>, <color>, <mileage>, and <price>. This flexibility allows complex queries to be performed on specific document elements. For example, "Find all Mercury Tracers with less than 100 000 kilometers that are selling for $3000 or less." It also allows calculations to be performed on specific document elements.

{button ,AL(`PRC Lesson 1 Understanding basic XML concepts;',0,"Defaultoverview",)} Related Topics

What are the components of an XML document?

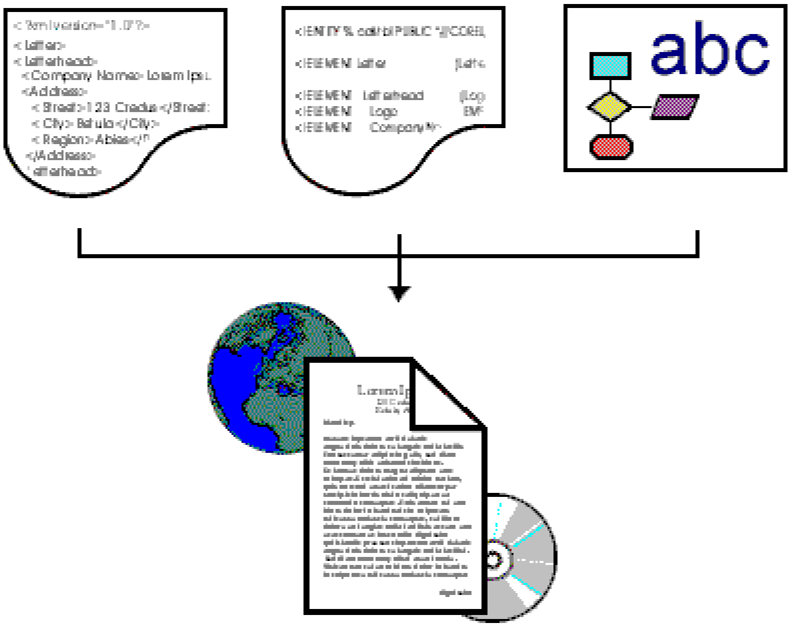
The power of XML is that it lets you identify components and define structure in a document. Once document information has been defined and structured, you can manipulate this information in a variety of ways.

Most environments using XML consist of three components:

- a document type definition (DTD)
- the XML document itself
- an application that processes the XML data

To learn about the components of an XML document

- Click any of the components in the picture below.



```
<?xml version="1.0"?>
<Letter>
<Letterhead>
  <Company Name>Lorem Ipsu
  <Address>
    <Street>123 Credus</Street>
    <City> Betula</City>
    <Region>Abies</R
  </Address>
 Letterhead>
```

```
<!ENTITY % coltbl PUBLIC "-//CORE

<!ELEMENT Letter          (Lett

<!ELEMENT  Letterhead      (Log
<!ELEMENT   Logo           EMP
<!ELEMENT   Company N
```

{button ,AL(`PRC Lesson 1 Understanding basic XML concepts;',0,"Defaultoverview",)} Related Topics

Document Type Definitions

Every valid XML document is based on a document type definition (DTD). The DTD defines the elements that make up the XML document upon which it is based. For example, a book contains elements such as a title page and chapters. The DTD defines the structure and content of these elements. For example,

```
<!ELEMENT book (titlepage, chapter+) >

<!ELEMENT titlepage (TITLE, author) >

<!ELEMENT TITLE (#PCDATA) >

<!ELEMENT author (#PCDATA) >

<!ELEMENT chapter (chaptitle, p+)>
```

In the above example, the element `book` contains a title page and more than one chapter. The element `titlepage` contains a title and the name of the author. Both `title` and `author` must consist of data, and chapter must consist of a chapter title and more than one paragraph.

XML document

An XML document consists of element tags and text. For example,

```
<book>
<titlepage>
<title>Learning XML</title>
<author>Daniel David</author>
</titlepage>
</book>
```

In the above example, each element has a start tag such as <book> and an end tag such as </book>. Some elements, known as empty elements, do not have both a start and an end tag.

Process component

XML is a markup language that allows you to identify and structure information. XML does not, however, provide a method for formatting or rendering documents. To do this, you need to use an application such as WordPerfect.

XML output

XML allows you to create documents that can be delivered across different media such as the World Wide Web, print, and CD.

Using WordPerfect and XML

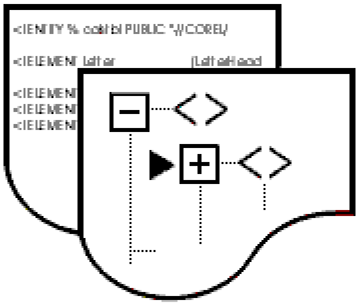
You can use WordPerfect to create, edit, retrieve, and save SGML or XML documents.

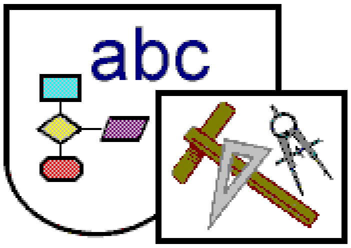
The three components of WordPerfect that you'll use with XML are:

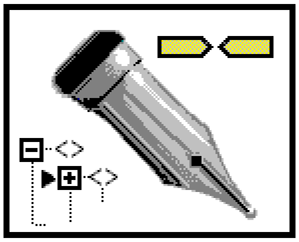
- the WordPerfect DTD Compiler
- the WordPerfect XML Project Designer
- the WordPerfect XML Editor

To learn about the components of WordPerfect that deal with XML

- Click any of the components in the picture below.







{button ,AL(`PRC Lesson 1 Understanding basic XML concepts;',0,"Defaultoverview",)} <u>Related Topics</u>

WordPerfect DTD Compiler

The WordPerfect DTD Compiler creates a binary representation of the DTD which is incorporated into a WordPerfect template. For XML documents, the catalog file is compiled into the WordPerfect template along with the DTD. For SGML documents, the SGML declaration file, the WP character mapping file, and the catalog file are compiled into the WordPerfect template along with the DTD.

The XML documents you create are based on a WordPerfect template that you have compiled and validated against the document rules contained in the template.

WordPerfect XML Project Designer

The WordPerfect XML Project Designer lets you format XML documents. Formatting XML documents involves defining rules for each document element. These element rules specify formatting such as the appearance and size of fonts, the justification of paragraphs, and page layout. Element rules can also specify formatting based on the element hierarchy.

WordPerfect XML Editor

To create XML documents, you work within the WordPerfect XML editing environment. This environment is an extension of the familiar WordPerfect word-processing environment. Menus and toolbars specific to XML are available in the WordPerfect XML Editor. Within the WordPerfect XML editing environment, you can view the structure of the XML document using the tree view. Each XML document is validated as you create it and errors are visible in the tree view.

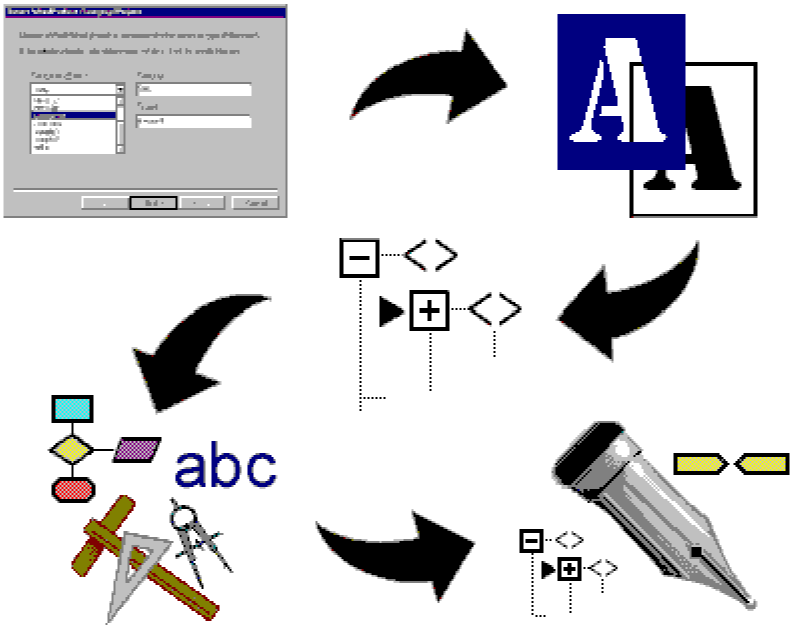
Understanding XML document creation

There are five steps involved in creating XML documents using WordPerfect:

1. Define an XML project and category.
2. Define the WordPerfect template.
3. Compile a document type definition (DTD).
4. Define the layout for the XML document.
5. Create the XML document in WordPerfect.

To learn about the steps involved in XML document creation

- Click any of the steps in the picture below.



{button ,AL(`PRC Lesson 1 Understanding basic XML concepts;',0,"Defaultoverview",)} Related Topics

Defining a category and project

Before you create an XML document, you must define a category and project. Just as in the WordPerfect word-processing environment, categories let you organize documents in a meaningful way. Each project in a category consists of a WordPerfect template upon which you create XML documents.

Defining WordPerfect templates

Every XML document is based on a WordPerfect template that defines its structure. Using templates, you can ensure that documents have the same structure. You can also customize the XML environment by including toolbars and keyboards in the WordPerfect template. To distribute an XML document, you send the XML document and its template.

Compiling document type definitions

Using the WordPerfect DTD Compiler, you create a WordPerfect template for use with XML documents. This template contains a compiled DTD and catalog files that specify references to external information for XML documents. For SGML documents, the template also contains the SGML declaration file, and the WP character mapping file.

Defining layout

Using the WordPerfect XML Project Designer, you create rules that define the formatting for each element of an XML document. You can create up to 10 different layouts for the same XML document. This allows you to output an XML document in multiple formats without changing the XML content.

Creating XML documents

You create XML documents within the WordPerfect XML editing environment by inserting elements, tags, text, and graphics.

Where to get more information

There are consultants who specialize in XML and there are many books on the subject. You can also find a lot of information on the Internet. The following site can help you get started:

The World Wide Web Consortium home page

NOTICE: This web site and its contents have been prepared by independent third parties and are not endorsed by Corel. Corel disclaims all liability regarding the content and your use of the web site. Your use of the web site may be governed by restrictions or guidelines contained therein.

{button ,AL(`PRC Lesson 1 Understanding basic XML concepts;',0,"Defaultoverview",)} Related Topics

NEXT LESSON

Summary of Lesson 1

In this lesson, you learned about

- XML and its advantages
- the components of an XML document
- the components of WordPerfect XML
- the process for creating an XML document

In Lesson 2, you'll explore the concept of a document type definition (DTD), define an XML project, learn about the compiling process, explore the WordPerfect DTD Compiler window, and compile a DTD.

{button ,AL(`PRC Lesson 1 Understanding basic XML concepts;',0,"Defaultoverview",)} Related Topics

Lesson 3: Creating an XML document

PREVIOUS LESSON

Lesson 3: Creating an XML document

In this lesson, you will create a letter in XML using the WordPerfect XML Editor. The letter will be based on the letter.wpt template you created in lesson 2.

In this lesson, you'll

- explore the WordPerfect XML Editor
- create a new XML document
- insert elements
- insert text

You'll need about 40 minutes to complete this lesson.

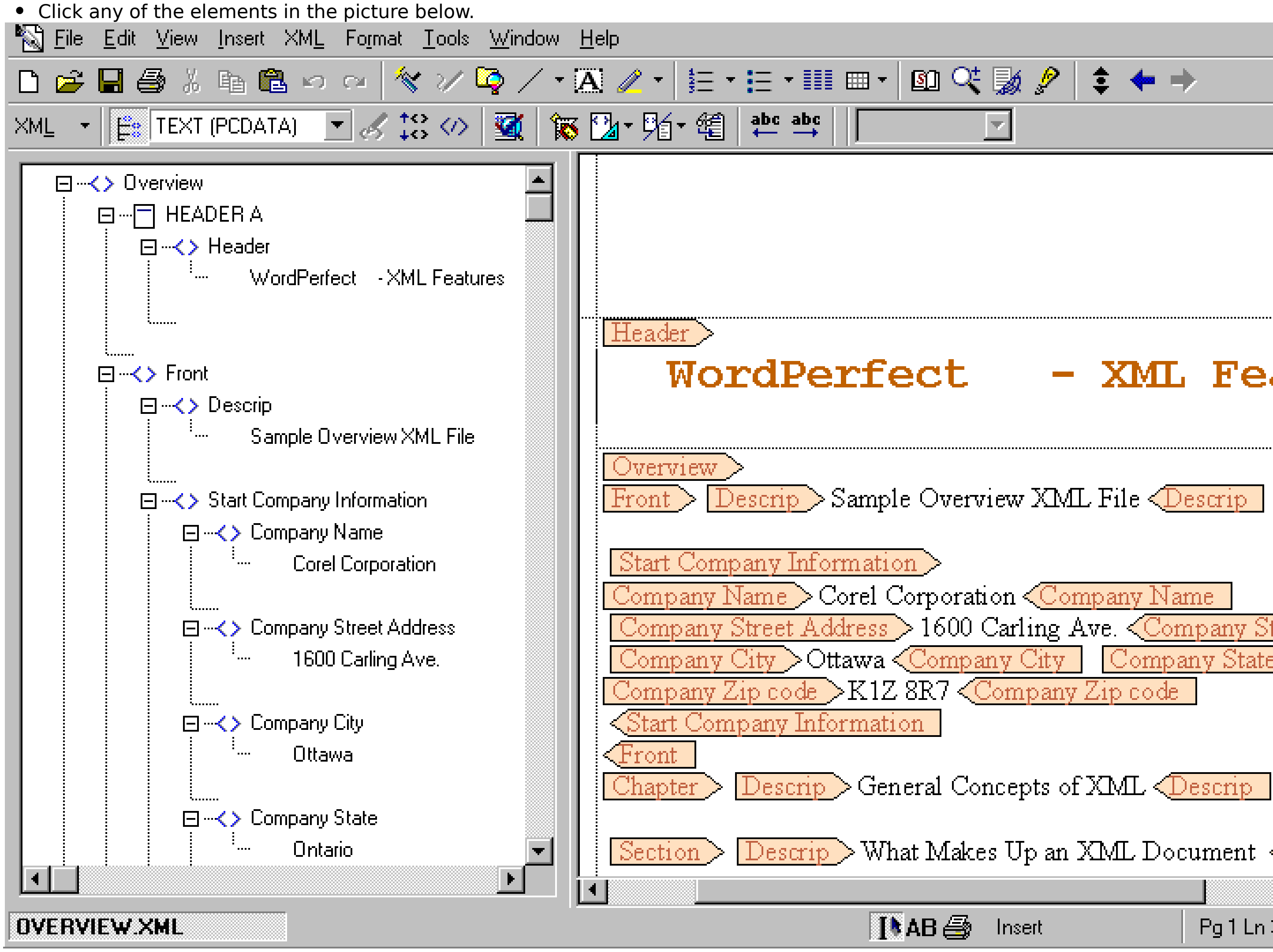
{button ,AL(`PRC Lesson 3 Creating an XML document;',0,"Defaultoverview",)} Related Topics

Exploring the WordPerfect XML Editor

You create, open, edit, and save XML documents using the WordPerfect XML Editor. The WordPerfect XML editing environment is an extension of the familiar WordPerfect word-processing environment. An XML-specific Property bar and XML-specific menu commands are available only within the WordPerfect XML editing environment.

To learn about the elements of the WordPerfect XML Editor window

- Click any of the elements in the picture below.



Overview
HEADER A
Header
WordPerfect - XML Features
Front
Descrip
Sample Overview XML File
Start Company Information
Company Name
Corel Corporation
Company Street Address
1600 Carling Ave.
Company City
Ottawa
Company State
Ontario

Header
WordPerfect - XML Fe
Overview
Front Descrip Sample Overview XML File Descrip
Start Company Information
Company Name Corel Corporation Company Name
Company Street Address 1600 Carling Ave. Company St
Company City Ottawa Company City Company State
Company Zip code K1Z 8R7 Company Zip code
Start Company Information
Front
Chapter Descrip General Concepts of XML Descrip
Section Descrip What Makes Up an XML Document

{button ,AL(`PRC Lesson 3 Creating an XML document;',0,"Defaultoverview",)} Related Topics

XML Property bar

The XML Property bar is context-sensitive and changes depending on the current position of the cursor. It provides quick access to XML commands and XML buttons.

XML Tree window

Displays a structured view of all document elements and attributes. Click the plus sign next to an element to view the attributes associated with it.

Validation errors are marked with a yellow triangle icon ⚠ and are visible in the XML Tree window. You can double-click error message icons for more information about the error.

A visual representation of where a graphic appears in the document also appears in the XML Tree window as 🎈. You can move and size the XML Tree window.

WordPerfect XML Editor window

You work within the WordPerfect XML Editor window to insert elements, text, tables, and graphics into XML documents.

Inserting elements and text into an XML document

After you exit the wizard and complete the process of creating a WordPerfect template, WordPerfect opens a blank XML document in the WordPerfect XML editing environment with the default XML Property bar.

Before you type the text of the letter, you must insert element tags. The WordPerfect Auto-Insert feature automatically inserts all the required elements defined in the DTD into your XML document . Any element defined as optional in the DTD is not automatically inserted. The generic layout created in lesson 2 presents the element tags in a more readable display.

To insert elements

1. In WordPerfect, click the XML button on the property bar, then click Elements.
2. In the Elements dialog box, click the Valid Elements option button.

   The Valid Elements list shows the elements that can be inserted at the current point in the document.
3. Click the Insert button to insert all valid elements defined in the WordPerfect template into the blank XML document.
4. When the Edit Attributes dialog box appears for the tag `Salutation`, click the Edit Value button.
5. In the Attribute Value box, choose `Miss`, then click the Select button.

   A checkmark appears in the Present column beside the attribute name `title`.
6. In the Error\Present\Attribute Name box, choose `Surname`, then click the Edit Value button.
7. In the Value box, type "Smith", click the OK button, then click the Close button to close the Edit Attributes dialog box.

Note

- To begin a new XML document based on the template created in Lesson 2, use the File, New From Project command in WordPerfect.

Tip

- To insert elements, you can also use the Valid Elements list box on the property bar or the Valid Elements command (by right-clicking in the XML document).

To insert text

1. Position the cursor between the `Company Name` start and end tags, then type "Corel Corporation."
2. Position the cursor between the element tags below and type the corresponding text.

Element	Text
Street	1600 Carling Avenue
City	Ottawa
Region	Ontario
Para	We received payment in the amount of 275 for the following items. It is a pleasure doing business with you. If there is anything else that you require, please feel free to contact me.
Adieu	Yours sincerely
Author	Alexander Day

3. Position the cursor before the end tag for `Address`. Notice that the Valid Elements list box on the Property bar shows `Postal Code` as a valid element at this point in the document. Postal Code was not inserted automatically into the document because it is defined as an optional element in letter.dtd.
4. From the Valid Elements list box on the Property bar, choose `Postal Code`.
5. Position the cursor between the `Postal Code` start and end tags, then type "K1Z 8R7".

Click here to see what the document should look like at this point.

{button ,AL(`PRC Lesson 3 Creating an XML document;',0,"Defaultoverview",)} Related Topics

Inserting a symbol

XML is based on the Unicode character set. All symbols are already mapped, so there is no need to define an external entity reference in the DTD for most symbols.

To insert a symbol

1. In the XML document, position the cursor in front of "275."
2. Click the Insert menu, Symbol.
3. From the Set button, choose Typographic Symbols.
4. Choose the symbol for the British Pound £ (number 4,11).
5. Click the Insert And Close button.

{button ,AL(`PRC Lesson 3 Creating an XML document;',0,"Defaultoverview",)} Related Topics

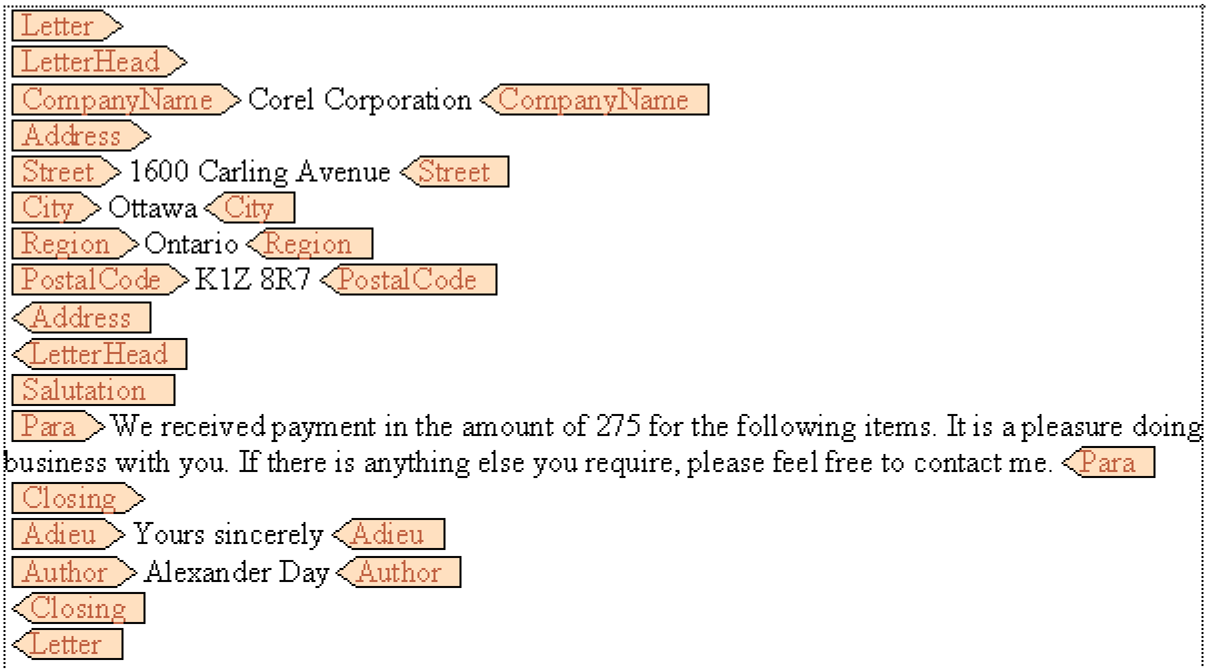
Saving an XML document

XML documents are validated by default when you save them.

To save the XML document

1. Click the File menu, Save.
2. In the File Name box, type "letter," then click the Save button.
3. In the Save As XML dialog box, click the OK button to accept the default settings for saving.
4. In the XML Validation dialog box, click No.
5. Click the File menu, Close to close letter.xml.

{button ,AL(`PRC Lesson 3 Creating an XML document;',0,"Defaultoverview",)} Related Topics



```
Letter
LetterHead
CompanyName Corel Corporation CompanyName
Address
Street 1600 Carling Avenue Street
City Ottawa City
Region Ontario Region
PostalCode K1Z 8R7 PostalCode
Address
LetterHead
Salutation
Para We received payment in the amount of 275 for the following items. It is a pleasure doing
business with you. If there is anything else you require, please feel free to contact me. Para
Closing
Adieu Yours sincerely Adieu
Author Alexander Day Author
Closing
Letter
```

NEXT LESSON

Summary of Lesson 3

In this lesson, you learned about

- the WordPerfect XML Editor
- creating a new XML document
- inserting elements
- inserting text

In Lesson 4, you'll create a document layout for the letter by formatting elements using the WordPerfect XML Project Designer.

{button ,AL(`PRC Lesson 3 Creating an XML document;',0,"Defaultoverview",)} Related Topics

Lesson 4: Creating, editing, and saving the document layout

PREVIOUS LESSON

Lesson 4: Creating, editing, and saving the document layout

Now that you've created the letter, you need to create a layout for it. In this lesson, you'll

- explore the WordPerfect XML Project Designer window
- format elements

You'll need about 30 minutes to complete this lesson.

{button ,AL(`PRC Lesson 4 Creating editing and saving the document layout;',0,"Defaultoverview",)} Related Topics

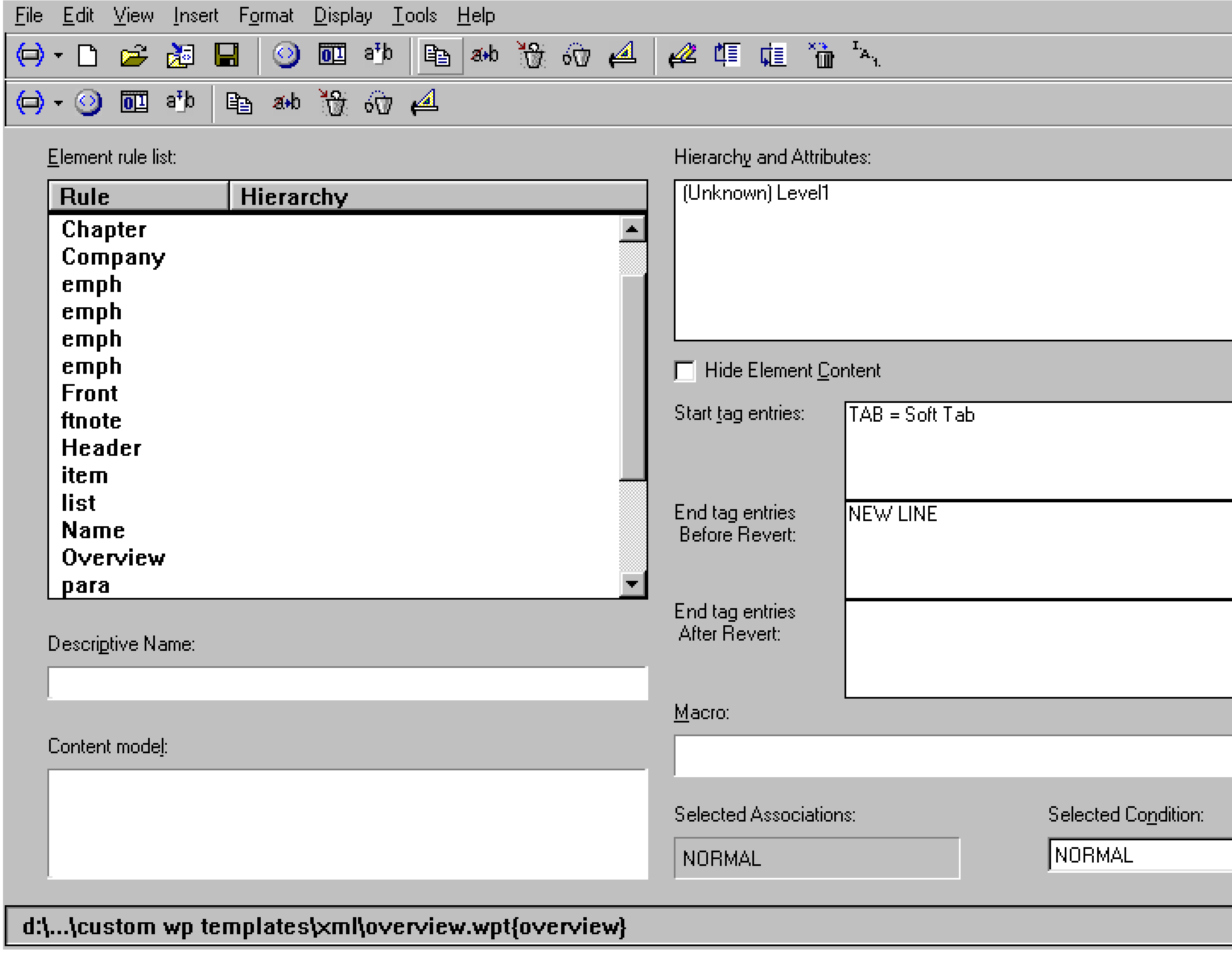
Exploring the WordPerfect XML Project Designer window

The XML Project Designer window displays the layout settings specified for the current XML project. When you create a new project, all fields are empty.

The picture below shows you the default Corel XML Project Designer window and toolbars.

To learn about the elements of the WordPerfect XML Project Designer window

- Click any of the elements in the picture below.



Element rule list:

Rule	Hierarchy
Chapter	
Company	
emph	
emph	
emph	
emph	
Front	
ftnote	
Header	
item	
list	
Name	
Overview	
para	

Descriptive Name:

Content model:

Hierarchy and Attributes: (Unknown) Level1

□ Hide Element Content

Start tag entries: TAB = Soft Tab

End tag entries Before Revert: NEW LINE

End tag entries After Revert:

Macro:

Selected Associations: NORMAL

Selected Condition: NORMAL

d:\...\custom wp templates\xml\overview.wpt{overview}

{button ,AL(`PRC Lesson 4 Creating editing and saving the document layout;',0,"Defaultoverview",)} <u>Related Topics</u>

Menus

The WordPerfect XML Project Designer has seven XML-specific menus. Use the

- File menu to access the WordPerfect DTD Compiler, open and create XML documents, or save and close XML documents
- Edit menu to define elements, counters, and rules
- View menu to display the status line, toolbar or property bar
- Insert menu to add cross-references, hard spaces, horizontal lines, and to revert to previous settings for an element
- Format menu to add formatting to your XML document
- Display menu to show attribute values, counters, styles, and text
- Tools menu to view and edit toolbars and Property bars

XML Property Bar

The XML Property bar is context-sensitive and changes depending on the current position of the cursor. It provides quick access to XML commands and XML buttons.

Element Rule List box

Displays all the elements that have layout codes specified for them. Double-click an element in the Element Rule List to edit the layout codes associated with it.

Content Model box

Displays the content model, as defined in the document type definition for the selected element.

Hierarchy and Attributes box

Displays all the elements specified as parent elements and the attributes specified for the selected element. Double-click the element to change attributes and parent elements.

Hide Element Content check box

Hides the content of an element in WordPerfect. Start tag and End tag entries are unavailable when you enable this checkbox.

Start Tag Entries box

Displays the element rules that take effect at the start tag for the element.

End Tag Entries (Before Revert) box

Displays the element rules that take effect at the end tag before reverting to the next element. For example, the layout at the start of the element invokes columns. At the end of the element the columns end. Any layout information specified in "Before Revert" appears inside the columns. Layout information specified in "After the Revert" appears after the columns end.

End Tag Entries (After Revert) box

Displays the element rules that take effect at the end tag for an element after reverting to the next element. For example, the layout at the start of the element invokes columns. At the end of the element the columns end. Any layout information specified in "Before Revert" appears inside the columns. Layout information specified in "After the Revert" appears after the columns end.

Macro list box

Displays the macro associated with the selected element.

Selected Associations box

Displays the WordPerfect feature associated with the selected element. When the selected element is inserted in the WordPerfect XML Editor, the WordPerfect feature is automatically opened.

Selected Condition list box

Displays the Condition for the selected element. You can specify whether you want the layout applied for every occurrence of the element or only the first occurrence.

Formatting elements

Using the XML Project Designer, you can create formatting rules for each element in a document. You must first open the letter.xml document you created in lesson 3.

To format elements

1. In WordPerfect, on the Property bar, click the XML button, then choose Edit Layout.
2. From the Element Rule List box, choose `Company Name`.
3. Click the Edit menu, Edit Rule.
4. Click the Format menu, Font.
5. In the Font Properties dialog box, choose Times New Roman from the Face box.
6. From the Size box, choose 16, and enable the Bold check box, then click OK.
7. In the Element Rule dialog box, click OK to accept the formatting.

To add text to the format for the Adieu and City elements

1. From the Element Rule List box, choose `Adieu`, then click the Edit menu, Edit Rule.
2. From the Rule Entry for End Tag (Before Revert) box, choose NEW LINE, then click the Delete button.
3. Click in the Rule Entry for End Tag (After Revert) box, then click the Display menu, Text.
4. In the Text box, type a comma and a space, click OK, then click the Line menu, New Line, and repeat three more times.
5. Click the OK button to return to the WordPerfect XML Project Designer.
6. From the Element Rule List box, choose `City`, then click the Edit menu, Edit Rule.
7. Repeat steps 2 and 3, and in the Text box, type a comma and a space, then click OK.
8. Click the OK button to return to the WordPerfect XML Project Designer.

To add text to the format for the Salutation element

1. From the Element Rule List box, choose `Salutation`, then click the Edit menu, Edit Rule.
2. From the Rule Entry for Start Tag box, choose NEW LINE, then click the Delete button.
3. Click the Display menu, Text and in the Text box, type "Dear" followed by a space, then click OK.
4. Click the Display menu, Attribute Value, and from the Attribute box, choose `title`, then click OK.
5. Click the Display menu, Text, and in the Text box, type a period followed by a space, then click OK.
6. Click the Display menu, Attribute Value, and from the Attribute box, choose `surname`, then click OK.
7. Click the Display menu, Text, and in the Text box, type a colon, then click OK.
8. Click the Line menu, New Line, and repeat a second time.
9. Click the OK button to return to the WordPerfect XML Project Designer.

To save the layout

1. Click the File menu, Save.
2. Click the File menu, Exit to return to WordPerfect.

Click here to see what the document should look like at this point.

{button ,AL(`PRC Lesson 4 Creating editing and saving the document layout;',0,"Defaultoverview",)} Related Topics

NEXT LESSON

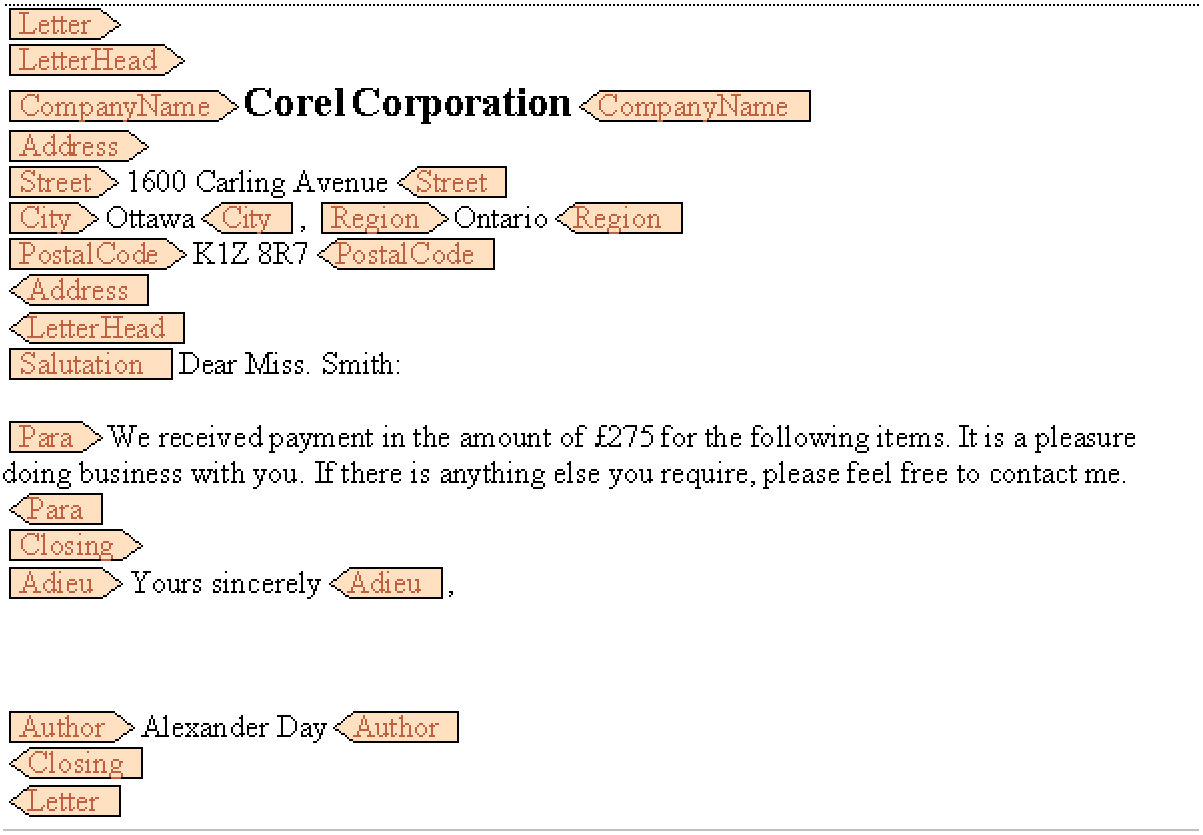
Summary of Lesson 4

In this lesson, you learned

- about the WordPerfect XML Project Designer
- how to format elements

In Lesson 5, you'll learn how to add tables, graphics, and bulleted lists to an XML document.

{button ,AL(`PRC Lesson 4 Creating editing and saving the document layout;',0,"Defaultoverview",)} Related Topics



```
Letter>
LetterHead>
CompanyName> Corel Corporation <CompanyName
Address>
Street> 1600 Carling Avenue <Street
City> Ottawa <City , Region> Ontario <Region
PostalCode> K1Z 8R7 <PostalCode
<Address
<LetterHead
Salutation Dear Miss. Smith:

Para> We received payment in the amount of £275 for the following items. It is a pleasure
doing business with you. If there is anything else you require, please feel free to contact me.
<Para
Closing>
Adieu> Yours sincerely <Adieu ,



Author> Alexander Day <Author
<Closing
<Letter
```

Lesson 5: Using advanced layout and formatting

PREVIOUS LESSON

Lesson 5: Using advanced layout and formatting

In this lesson, you will become familiar with special design features such as tables, graphics, and numbered lists.

In this lesson, you'll add the following elements to the letter you created in Lessons 2 and 3:

- a table
- a graphic
- a bulleted list

You'll need about 50 minutes to complete this lesson.

{button ,AL(`PRC Lesson 5 Using advanced layout and formatting;',0,"Defaultoverview",)} Related Topics

Adding table declarations to the DTD

Before you can insert a table into the letter, you must modify the document type definition by adding a PUBLIC identifier that references a DTD with table support, in this case CALS_XML.dtd. This DTD has been created to represent WordPerfect CALS support. You can use Notepad to modify the DTD.

To add table declarations to the DTD

1. In Notepad, open the letter.dtd file located in the XML\Tutorial folder.
2. Position the cursor on the first blank line after the last element (<!ELEMENT ENC) in the letter.dtd file.
3. Type the following text:

```
<!ENTITY % calsXML PUBLIC "-//COREL//DTD WordPerfect XML CALS table support//EN"
"cals_xml.dtd">
```

4. On the next blank line, type `%calsXML;`.
5. Click the File menu, Save to save the letter.dtd file.

Note

- XML is case sensitive. Make sure you type the text exactly as printed, including uppercase and lowercase letters.

{button ,AL(`PRC Lesson 5 Using advanced layout and formatting;',0,"Defaultoverview",)} Related Topics

Creating an entry in your catalog file

Catalog files provide a method for locating external information referenced from a DTD or document. DTDs reference information such as markup declarations for elements, entities, attributes, etc. Documents often reference external blocks of text, or non-textual information such as graphics and sound. Two catalog files are supplied with WordPerfect: Compile.cat and Import.cat.

- Compile.cat is the file that the DTD Compiler will try to use if the user does not supply a different one.
- Catalog.cat is the default catalog file used when importing SGML/XML documents. It contains entries that match up a DTD's PUBLIC/SYSTEM identifiers and/or root element name to the appropriate WordPerfect Template file.

For more information about catalogs, see Working with catalog files.

After you add the appropriate table commands to your DTD, you usually need to create an entry in the catalog file that indicates the table model for your XML document. However, when you compiled the letter.dtd in Lesson 2, the default catalog file (Compile.cat) was used. Compile.cat already contains the required catalog entries for tables.

The following entity declarations appear in Compile.cat:

```
-- ============================================================ --
-- = Table DTD fragments = --
-- ============================================================ --
PUBLIC "-//COREL//DTD WordPerfect CALS table support//EN"
".\cals_tbl.dtd"

PUBLIC "-//COREL//DTD WordPerfect XML CALS table support//EN"
".\cals_xml.dtd"

PUBLIC "-//COREL//DTD WP 6.0 Minimal Table Declarations//EN"
".\wp60min.dtd"

PUBLIC "-//COREL//DTD WP 6.0 Table Declarations//EN"
".\wp60tbl.dtd"

PUBLIC "-//COREL//DTD WP 5.1 Table Declarations//EN"
".\wptable.dtd"
```

{button ,AL(`PRC Lesson 5 Using advanced layout and formatting;',0,"Defaultoverview",)} Related Topics

Recompiling letter.dtd

After adding the necessary commands for creating graphics and tables in your DTD, you need to recompile letter.dtd.

To recompile the letter.dtd

1. On the Windows taskbar, click the Start button, then click WordPerfect Office 11, Utilities, WordPerfect XML Project Designer.
2. Click the File menu, DTD Compile.
3. Click the folder icon next to the DTD (Input) File box, then locate letter.dtd.
4. Click the folder icon next to the WP Template (Output) File box, then locate letter.wpt.
5. Enable the Compile DTD As XML check box.
6. Click the Compile button to recompile letter.dtd.
7. Click the Close button, then click the File menu, Exit.

Tip

- You can also access the WordPerfect XML Project Designer from within the WordPerfect XML editing environment by clicking File, Edit Layout.

{button ,AL(`PRC Lesson 5 Using advanced layout and formatting;',0,"Defaultoverview",)} <u>Related Topics</u>

Inserting a table

After modifying and recompiling the DTD, you can insert a table into the letter and tag it appropriately.

To insert a table

1. In WordPerfect, click the File menu, then choose letter.xml from the Recently Used File List.
2. Position the cursor after the text "We received payment in the amount of 275 for the following items."
3. Click the Insert menu, Table.
4. In the Columns box, type 2, and in the Rows box, type 5.
5. Click the Create button.
6. Type the following text and numbers in the table cells:

Items returned	Quantity
keyboard trays	10
mouse pads	10
white boards	5
dry erase markers	10 packages

7. Position the cursor in the table, then click the XML menu, Tag Tables.
8. Click the Tag Current button, then click the Close button.
9. Click anywhere outside the table.

Tip

- Click the XML Tree button on the toolbar to see the `Table` element in the structured view of the document.

{button ,AL(`PRC Lesson 5 Using advanced layout and formatting;',0,"Defaultoverview",)} Related Topics

Adding a graphic

In the letter.dtd, the `Logo` element is defined as an empty element with the attribute `graphic` associated with it. The `graphic` attribute is defined as an entity. When you insert a graphic into an XML document, WordPerfect automatically creates a file reference that defines the location and format of the graphic entity.

To add a graphic, you must first associate the element attributes defined for the `Logo` element with WordPerfect functionality. In doing so, you ensure that changes you make in WordPerfect to the horizontal position, the vertical position, the width, and the height of the graphic are reflected in the attribute values associated with the `Logo` element.

To associate element attributes with WordPerfect functionality

1. On the Property Bar, click the XML button, then choose Edit Layout.
2. From the Element Rule List box, choose the `Logo` element.
3. Click the Edit menu, Element Associate, then from the Element box, choose `Logo`.
4. From the Association box, choose `Image Box and f`rom the Graphic Reference box, choose `graphic`.
5. From the Horizontal Position box, chose `horzpos`.
6. From the Vertical Position box, choose `vertpos`.
7. From the Height box, choose height.
8. From the Width box, choose width.
9. Click the OK button, save the document layout, then exit the WordPerfect XML Project Designer.

To define the file reference

1. On the Property Bar, click the XML button, then choose File References.
2. Click the New button, then type "Logo" in the Name box.
3. Click the folder icon next to the File Path box and locate the corelsym.bmp stored in the XML/Tutorial folder.
4. Choose NDATA from the Data Type list box, then choose bmp from the Notation list box.
5. Click the OK button.
6. Enable the User Declared option button, then choose `Logo` in the File Reference box.
7. Click the Close button.

To add a graphic

1. On the Property Bar, click the XML button, then choose Layouts.
2. From the Layouts box, choose "letter", then click the Select button.
3. Position the cursor before the Company Name start tag.
4. From the Valid Elements list box on the XML Property bar, choose `Logo`.
5. Click the folder icon next to the Look In list box, then locate the corelsym.bmp in the XML\Tutorial folder.
6. From the Look In list box, choose corelsym.bmp, then click the Insert button.
7. With the graphic selected, click the XML menu, Tag Box.

Tip

- You can see the tags surrounding the graphic in Reveal Codes.

To size and position the graphic

1. Select the graphic.
2. Right-click, then choose Size.
3. In the Width section, enable the Set option button, then type 0.6 in the Width box.
4. In the Height section, enable the Set option button, then type 0.6 in the Height box.
5. Click the OK button.
6. Right-click the graphic, then choose Position.
7. In the Attach Box To box, choose Page.
8. In the Horizontal section, type -0.8, then choose from Left Margin.
9. With the graphic selected, click the XML menu, Tag Box.

Notes

- Do not drag the graphic to a new position. Dragging the graphic moves the box tag and results in a validation error.
- Every time you make changes to the size or position of the graphic, you must retag it.

Click here to see what the document should look like at this point.

{button ,AL(`PRC Lesson 5 Using advanced layout and formatting;',0,"Defaultoverview",)} Related Topics

Creating a bulleted list

Creating a bulleted list is a two step process. First, create a style just as you would when doing word processing. Then, associate that style to an element that is defined in the associated DTD.

To create a style for bulleted lists

1. In WordPerfect, click the Format menu, Styles.
2. Click the Create button.
3. In the Style Name box, type "bulleted list."
4. Click in the Contents box.
5. Click the Insert menu, Outline/Bullets and Numbering.
6. Click the Bullets tab, then click the OK button to accept the default bullet style.
7. In the Styles Editor dialog box, click the OK button.
8. In the Styles dialog box, click the Close button.

To associate a style with an element

1. On the Property bar, click the XML button, then choose Edit Layout.
2. From the Element Rule List box, choose `ListItem`, then click the Edit menu, Edit Rule.
3. Click the Format menu, Styles.
4. In the Available Styles box, choose `bulleted list`, then click the OK button.

   Note that the rule STYLE=bulleted list appears in the Rule Entry For Start Tag box.
5. Click the OK button.
6. Click the File menu, Exit, then click Yes to save changes to the layout.

To insert the bulleted list

1. Position the cursor after the `table` end tag.
2. On the Property bar, choose `BulletList` from the Valid Elements list box.
3. Position the cursor before the `BulletList` start tag, then type "When you return items, please include:".
4. Position the cursor before the `ListItem` end tag, then type "the original bill of sale."
5. Position the cursor before the `BulletList` end tag.
6. On the Property bar, choose `ListItem` from the Valid Elements list box, then type "all the original packaging."
7. Click File, Save to save letter.xml.

Click here to see what the letter should look like at this point.

{button ,AL(`PRC Lesson 5 Using advanced layout and formatting;',0,"Defaultoverview",)} Related Topics

NEXT LESSON

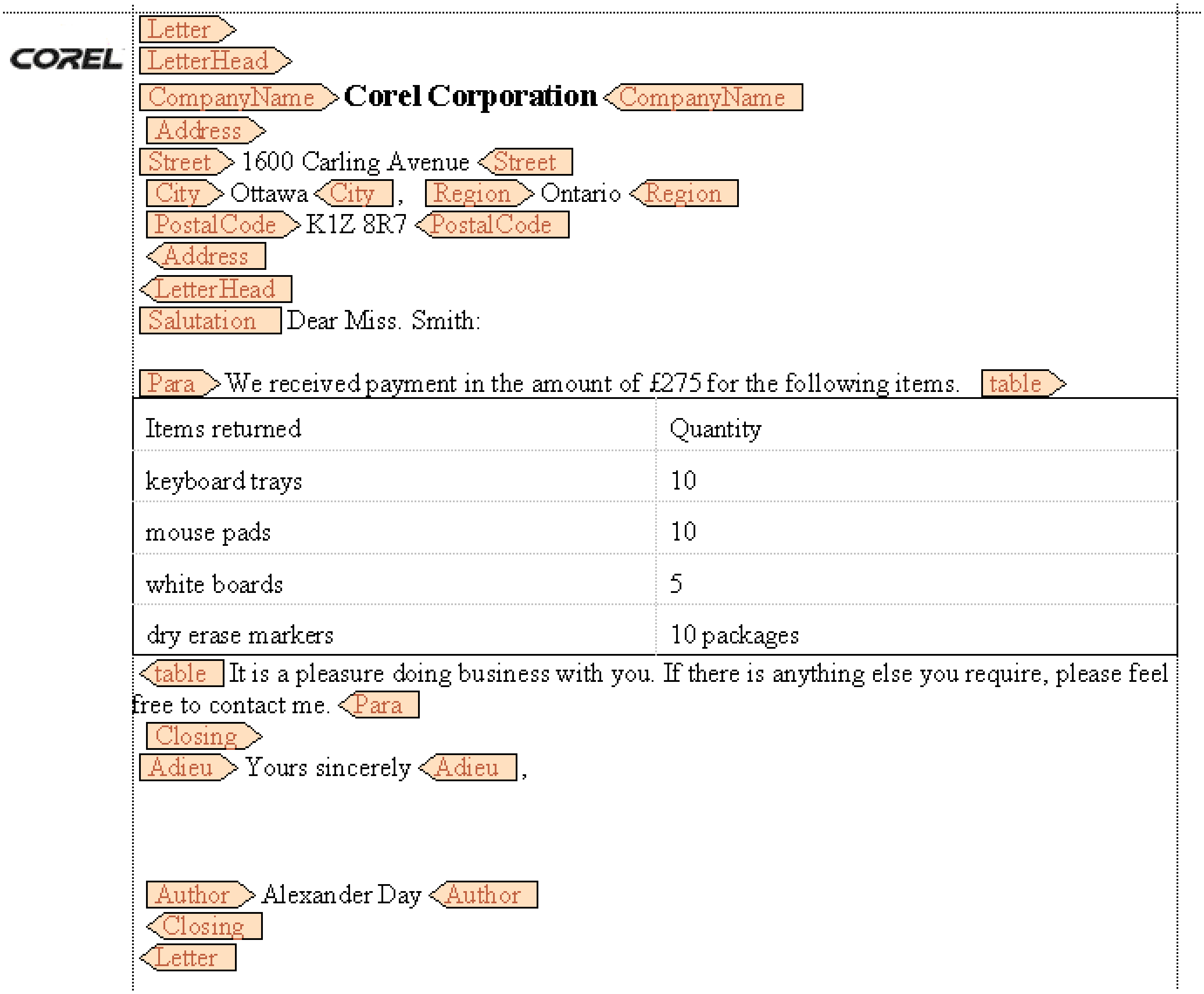
Summary of Lesson 5

In this lesson, you learned how to

- add a table
- insert a graphic
- add a bulleted list

In Lesson 6, you'll learn how to create a table of contents and cross-references.

{button ,AL(`PRC Lesson 5 Using advanced layout and formatting;',0,"Defaultoverview",)} Related Topics



Letter>
LetterHead>
CompanyName> **Corel Corporation** <CompanyName
Address>
Street> 1600 Carling Avenue <Street
City> Ottawa <City , Region> Ontario <Region
PostalCode> K1Z 8R7 <PostalCode
<Address
<LetterHead
Salutation Dear Miss. Smith:

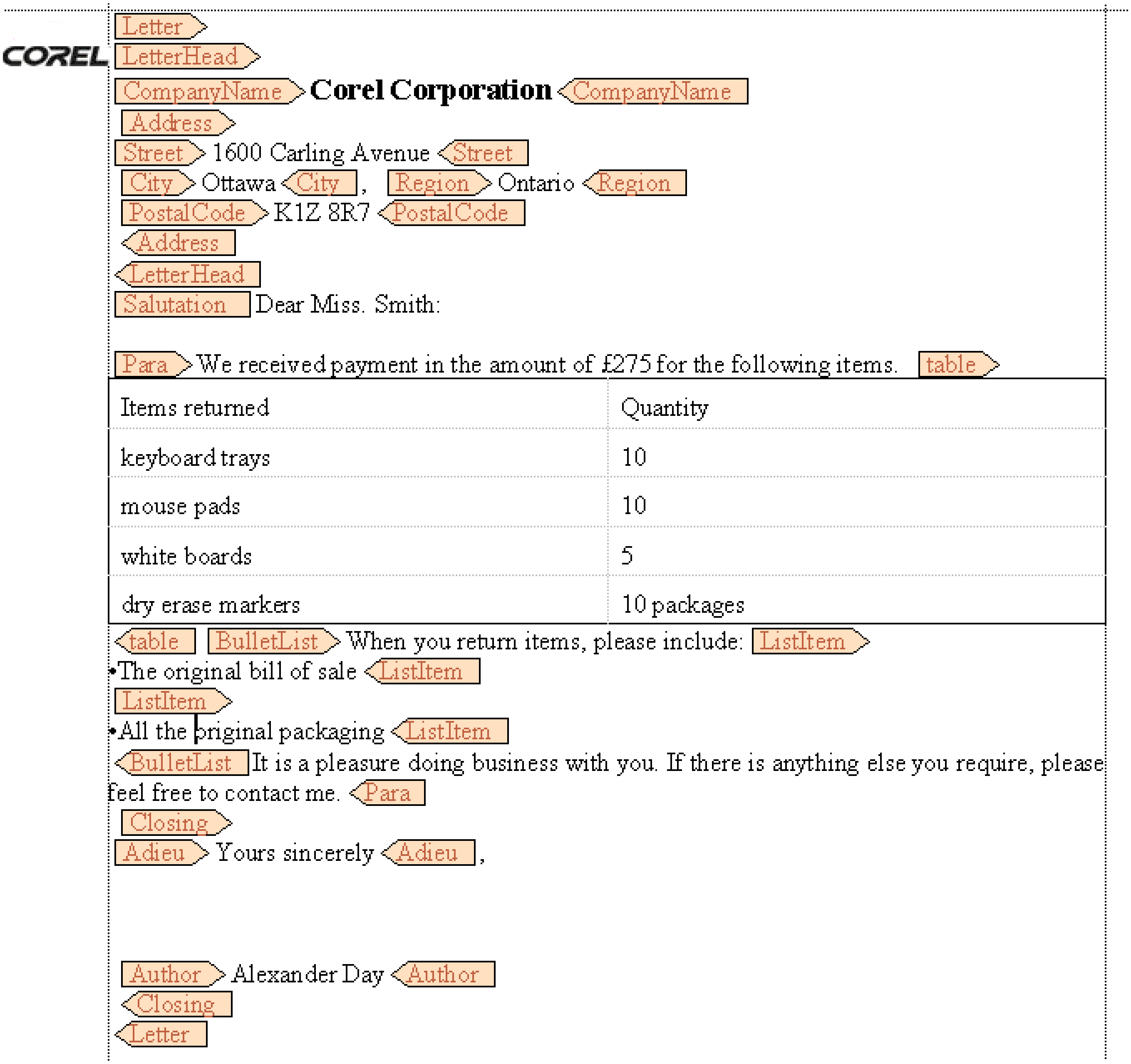
Para> We received payment in the amount of £275 for the following items. table>

Items returned	Quantity
keyboard trays	10
mouse pads	10
white boards	5
dry erase markers	10 packages

<table It is a pleasure doing business with you. If there is anything else you require, please feel free to contact me. <Para

Closing>
Adieu> Yours sincerely <Adieu ,

Author> Alexander Day <Author
<Closing
<Letter



Letter>
LetterHead>
CompanyName> **Corel Corporation** <CompanyName
Address>
Street> 1600 Carling Avenue <Street
City> Ottawa <City , Region> Ontario <Region
PostalCode> K1Z 8R7 <PostalCode
<Address
<LetterHead
Salutation Dear Miss. Smith:

Para> We received payment in the amount of £275 for the following items. table>

Items returned	Quantity
keyboard trays	10
mouse pads	10
white boards	5
dry erase markers	10 packages

<table BulletList> When you return items, please include: ListItem>
•The original bill of sale <ListItem
ListItem>
•All the original packaging <ListItem
<BulletList It is a pleasure doing business with you. If there is anything else you require, please feel free to contact me. <Para
Closing>
Adieu> Yours sincerely <Adieu ,

Author> Alexander Day <Author
<Closing
<Letter

Lesson 6: Managing XML documents

PREVIOUS LESSON

Lesson 6: Managing XML documents

In this lesson, you'll work with an existing XML document (Overview.xml) and learn how to add special design features to manage XML documents.

In this lesson, you'll create

- a table of contents
- a cross-reference

You can use the formatting features of WordPerfect to create an index for your XML document. For more information, see Generating and updating indexes in the WordPerfect Help.

Creating a table of contents

Creating a table of contents for an XML document is very similar to creating a table of contents for a WordPerfect document. You must first define a table of contents by specifying the number of levels and the numbering format just as you would when creating a WordPerfect document. Then, you mark entries for the table of contents. These entries can include section, heading, subheading, and chapter entries.

For more information about creating a table of contents for WordPerfect documents, see Generating and updating tables of contents in the WordPerfect Help.

Creating a cross-reference

In this lesson, you will create a reference for the `XRef` element. The target for the `XRef` element is a specified `Descrip` element which contains a description of a particular chapter.

Creating cross-references in XML documents is very similar to creating cross-references in WordPerfect documents except that in XML documents you mark elements, not text, and you use attributes, not identifying names, to link a reference with a target.

For more information about creating cross-references for WordPerfect documents, see Generating and updating cross-references in the WordPerfect Help.

You'll need about 50 minutes to complete this lesson.

{button ,AL(`PRC Lesson 6 Managing XML documents;',0,"Defaultoverview",)} Related Topics

Creating a table of contents

To add a table of contents, you first define the table of contents, then you will use the WordPerfect XML Project Designer to add table of contents markings to elements. A generated table of contents is not saved with the XML document but can be used when the document is saved in a WordPerfect format.

To define a Table of Contents

1. In WordPerfect, open the Overview.xml document located in the \XML\Tutorial folder.
2. On the Property bar, click the XML button, then choose, Edit Layout.
3. In the Element Rule List box, choose the `TOC` element, then click the Edit menu, Edit Rule.
4. Click the Insert menu, Table of Contents, Define.
5. In the Number Of Levels box, type 2, then click the OK button.
6. In the Element Rule dialog box, click the OK button.

To define elements for inclusion in the Table of Contents

1. From the Element Rule List box, choose `Descrip`, then click the Edit menu, Copy Rule.
2. With the first `Descrip` element selected, click the Edit menu, Edit Rule.
3. Click the Hierarchy button, then click the Show All Elements option button.
4. Double-click the `Chapter` element in the Element Parents box, click the OK button to return to the Element Rule dialog box, then click the OK button again.
5. From the Element Rule List box, choose the second `Descrip` element, then click the Edit menu, Edit Rule.
6. Click the Hierarchy button, then click the Show All Elements option button.
7. Double-click the `Section` element in the Element Parents box, click OK to return to the Element Rule dialog box, then click OK again.

To mark elements for inclusion in the Table of Contents

1. From the Element Rule List box, choose the first `Descrip` element, then click the Edit menu, Edit Rule.
2. Click in the Rule Entry For Start Tag box, then click the Insert menu, Table Of Contents, Mark 1, Begin.
3. Click in the Rule Entry For End Tag (After Revert) box, then click the Insert menu, Table Of Contents, Mark 1, End.
4. Click the OK button.
5. From the Element Rule List box, choose the second `Descrip` element, then click the Edit menu, Edit Rule.
6. Click in the Rule Entry For Start Tag box, then click the Insert menu, Table Of Contents, Mark 2, Begin.
7. Click in the Rule Entry For End Tag (After Revert) box, then click the Insert menu, Table Of Contents, Mark 2, End.
8. Click the OK button.
9. Save the layout file, then exit the WordPerfect XML Project Designer.

To create a Table of Contents

1. Position the cursor before the `Front` end tag.
2. On the XML Property Bar, choose the `Table of Contents` element from the Valid Elements list box.
3. Click the Tools menu, Reference, Table Of Contents.
4. On the Table of Contents Property bar, click the Generate button.
5. In the Generate dialog box, click the OK button.

Note

- To save the Table of Contents information, select the Table of Contents and insert WP DATA codes around it using the WPDATA command from the Insert menu, then save the document in a .wpd format.

{button ,AL(`PRC Lesson 6 Managing XML documents;',0,"Defaultoverview",)} Related Topics

Marking references and targets for cross-references

Cross-references are linked through the attributes of the target element and the reference element. Therefore, only elements with attributes can be cross-referenced.

To mark a reference

1. In WordPerfect, open the Overview.xml document located in the \XML\Tutorial folder.
2. On the Property bar, click the XML button, then choose, Edit Layout.
3. Click the Tools menu, Create Generic Rules.

   The XML document will be easier to read with a generic layout applied.
4. On the Property bar, click the Elements button, then choose `XRef`.
5. Click in the Rule Entry For Start Tag box, then click the Insert menu, Cross Reference.
6. From the Attribute Target box, choose `idref`.
7. Click the Mark Reference button, then click Close.

8 Click OK to return to the WordPerfect XML Project Designer.

To mark a target

1. From the Element Rule List box, choose `Descrip`.
2. Click Edit, Edit Rule.
3. Click Insert, Cross Reference.
4. Click in the Rule Entry For Start Tag box, then click the Insert menu, Cross Reference.
5. From the Attribute Target box, choose `id`.
6. Click the Mark Target button, then click Close.
7. Click OK to return to the WordPerfect XML Project Designer.
8. Click the File menu, Exit and then click Yes to save the changes.

{button ,AL(`PRC Lesson 6 Managing XML documents;',0,"Defaultoverview",)} <u>Related Topics</u>

Marking and generating cross-references

To mark cross-references, you type an attribute value for the reference and a matching attribute value for the target. Cross-references are created between matching attribute values.

To mark a cross-reference

1. In WordPerfect, position the cursor after the `Descrip` start tag located in front of "What Makes Up An XML Document."
2. On the Property bar, click the XML button, then choose Edit Attributes.
3. Click the Edit Value button, type "a1" in the Value box, then click OK.
4. Click the Close button.
5. Click the Edit menu, Find And Replace, then type "Table Tagging" and click Find Next.
6. Position the cursor after the end tag for `table`, then type "For information about the contents of an XML document see".
7. On the Property bar, from the Valid Elements list box, choose `XRef`.
8. Click the Edit Value button, then type "a1", then click OK.
9. Click the Close button.

To generate cross-references

1. Click Tools, Reference, Generate.
2. In the Generate dialog box, click OK.
3. Click the File menu, Save to save the updated Overview.xml.

Notes

- To set up a cross-reference, you must have a cross-reference element (such as XRef) defined in your DTD.
- Until you use the Generate command, a question mark appears where you have marked references. When you generate, the reference number is inserted or updated.

{button ,AL(`PRC Lesson 6 Managing XML documents;',0,"Defaultoverview",)} Related Topics

⎘ Copies the selected element and its associated attributes.

Displays the Counter Numbering dialog box where you can create, edit, or delete counters and set the value for counters.

✎ Opens the dialog box associated with the formatting code where you can edit the selected code.

Launches the WordPerfect XML Project Designer.

Displays the Element Rule dialog box where you can modify formatting for the selected element.

🚪 Exits the WordPerfect XML Project Designer.

Creates a new layout based on the active WordPerfect template.

a→b Lets you rename an existing element rule.

Retrieves an existing layout file into another layout file.

Displays and hides the XML Tree.

Restores your last deleted rule.

Displays the list of elements defined in the associated WordPerfect template.

Lets you change the appearance of XML codes.

Lets you tag the selected box.

Lets you specify the elements to associate with the box, content, and caption for the selected graphic.

Lets you automatically tag tables in an XML document.

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

Alias
Alias file
Any listed element
Attribute
Attribute list definitions

B

BASESET mapping files

C

Cascading Style Sheets
Catalog file
Content model

D

Default.dec
DTD compiler
Document Type Definition
Document type declaration

E

Elements

Empty elements

Entity files

Entity declarations

Entity references

Error lists

External entities

'External parameter entities

F

Find errors

G

General entity

Group connectors

H

Hard Page

Hard Space

Hyphen

Hyphen Soft Return

I

Invalid elements

J

K

L

Layout file

Listed element

Log file

Logic file

M

Markup declarations

Marked sections

Markup languages

N

Notations

O

Occurrence Indicators

OLE Object

Optional elements

P
Parameter entities
Parent
Processing instructions
Projects
PUBLIC identifier

R
References
Root element

S
SGML
SGML declaration file
SGML entity declaration files
Soft Hyphen
SYSTEM identifier

T
Tags
Target
Target Name

U
Unicode character set
Unknown elements

V
Valid documents

W
Well-formed documents
WordPerfect Mapping Files
WordPerfect DTD Compiler

X
XML character reference
XML Project Designer

Z

Alias

An alias is a descriptive name given to make element names more meaningful.

Alias file (.LNM)

The Alias file specifies descriptive names for the elements in the DTD. This is useful because many times the element names in a DTD are cryptic and most DTDs are written in English. By specifying more descriptive names in an alias file, end users of multiple languages can all use the same DTD.

Any listed elements

Any Listed Elements allows the Find and Replace feature to limit the search to text that appears only within specified elements. The elements do not have to occur in any specific order.

For example, if you specified the elements <CAPTION> and <TITLE>, WordPerfect only searches the text located between the opening and closing tags of these elements, <CAPTION> and </CAPTION> and <TITLE> and </TITLE>.

Attributes

An attribute is a qualifier that provides further information about an element. Attributes give information that applies to a specific occurrence of an element. An attribute usually does not apply to all occurrences of that element. Attributes are defined in the Document Type Definition (DTD).

Attribute list declarations

An element may be defined further by associating attributes to it. For example, in the following attribute list declaration the element `Memo` has the attribute `status` associated with it. The status of the memo can be either `confidential` or `public`. The default attribute for the element `Memo` is `public`.

```
<!ATTLIST Memo status (confiden | public) public>
```

BASESET mapping files

BASESET mapping files are activated in an SGML declaration and are not used in XML. A BASESET mapping file is an ASCII file that describes a BASESET of characters in terms of WordPerfect Characters. This file contains one or more BASESET mapping entries. Each BASESET mapping entry corresponds to a particular character in the BASESET being described and specifies the following two fields of information:

- The decimal number representing the character's position in the BASESET.
- The character set and character number of the WordPerfect Character that most closely matches the BASESET character.

For example, the following three entries are from ECMA94.SET, the BASESET mapping file describing the ECMA-94 Latin 1 BASESET:

161	4,7	--Inverted Exclamation Point--
162	4,19	--Cent--
163	4,11	--Pound/Sterling--

Any character not described by a BASESET mapping entry is assumed to be mapped to WordPerfect Character 0,0 and is considered UNUSED.

BASESET mapping entries and their individual fields may be separated by any combination of spaces, tabs, carriage returns, line feeds, or SGML comments.

You can use the *.SET files (BASESET mapping files) to map 8-bit ASCII characters to WordPerfect Characters (also called Corel Symbols). The CP*.SET files correspond to IBM code pages. ECMA94.SET corresponds to the ECMA-94 Latin 1 BASESET defined by the European Computer Manufacturers Associations and used in the sample SGML declaration in ISO 8879.

CP333.SET

CP437.SET

CP850.SET

CP851.SET

CP8510.SET

CP852.SET

CP853.SET

CP860.SET

CP8600.SET

CP8601.SET

CP861.SET

CP863.SET

CP865.SET

CP866.SET

CP8660.SET

CP895.SET

CP897.SET

ECMA94.SET

Note

- If a WordPerfect Character is mapped to both a BASESET character and an entity name, the BASESET mapping will take precedence when a document is validated or saved to SGML. In other words, the WordPerfect Character is treated as an 8-bit SGML character rather than an entity reference.

Cascading Style Sheets

Cascading Style Sheets is a W3C recommendation that specifies how styles should be defined for HTML documents.

Catalog files

Catalog files provide a method for locating external information referenced from a DTD or document. DTDs reference information such as markup declarations for elements, entities, attributes, etc. Documents often reference external blocks of text, or non-textual information such as graphics and sound.

The format of the catalog file is defined by the Organization for the Advancement of Structured Information Standards (OASIS). The most common catalog file entries map a PUBLIC or SYSTEM identifier to a file on your system. WordPerfect uses the proprietary keywords WPT_PUBLIC, WPT_SYSTEM and WPT_ROOT to locate the appropriate WordPerfect template file using the information from a document's DOCTYPE declaration.

For more information about catalog files, see "Working with catalog files."

Content model

The content model consists of rules that define the content of each element in the DTD. For example, the element `table` is defined in the content model as `(title?, tgroup)`. This means that the element `table` may contain a title (but doesn't have to) but will contain the element `tgroup`.

Default.dec

Default.dec is an SGML declaration file included with WordPerfect. Default.dec matches the SGML capabilities of WordPerfect. It is different from SGML's reference concrete syntax because it does not support short reference delimiters. If you want to use a different SGML declaration as your default, you can modify DEFAULT.DEC, copy your SGML declaration file to Default.dec, or specify the name of your SGML declaration file in the SGML Declaration File text box.

Document Type Definition File (DTD)

A file that defines the elements and the structured relationship between the elements. The DTD also defines entities and attributes. The DTD defines all valid elements, the order in which elements can be used, and how many times a particular element can appear in a document. When compiled by the WordPerfect DTD Compiler, the DTD is integrated in a WordPerfect Template along with the .LSI (Layout Specification Instance) file and the .LNM (Alias) file.

Document type declaration

A Document Type Declaration is a markup declaration that starts with "<!DOCTYPE". The Document Type Declaration begins an SGML document. This declaration includes and/or references the markup declarations for elements, entities, etc. that make up the Document Type Definition. The root element is specified in the Document Type Declaration.

Elements

An element is a specific part of a document, delimited by a start tag and (usually) an end tag. For example, the start tag for the element `title` is <title>and the end tag is </title>. The forward slash in the end tag distinguishes it from the start tag.

While humans can visually identify different parts of a book such as titles, paragraphs and footnotes, the same is not true of computer programs. Marking elements with start and end tags makes it easy for a program to recognize the beginning and ending of each element without relying on the visual format of the document.

Empty elements

Empty elements do not have any content, and therefore cannot have an end tag. They are declared in the DTD, and may have attributes assigned to them. In SGML, elements declared as EMPTY are represented by a normal start tag such as <graphic>. In XML, the tag itself serves as an indicator that an element has no content: <graphic/>. This style of tag may be used in XML both for elements declared as EMPTY in the DTD, and for non-EMPTY elements that happen to contain no content in the current situation.

Entity files

An entity file is a file that represents the content of an external entity reference declared in a document type definition (DTD). Entity files can contain items such as entity declarations, text, or graphics.

Entity declarations

An entity declaration specifies the name of an entity and information about its content or replacement text. An entity can be declared as either a general entity or a parameter entity.

Entity references

Not all characters can be typed by all keyboards. For example, the character £ is not available on the standard keyboard, or in the standard SGML character set.

To specify £ or other non-standard characters in your SGML document, use entity references. An entity reference must be defined in the DTD you are using. The accented letters and special symbols of modern European languages (found in ISO 8879) are already defined in many DTDs.

A DTD can also declare entity references that specify commonly used text. For example, an entity reference might specify the company name or a standard legal disclaimer required by all documents using that DTD.

SGML entity references appear as characters in the document window. However, when the document is saved in SGML format, the characters are translated into the entity reference.

Error lists

You can specify whether you want Error List to display a list of Logic Errors or a list of ID/IDREF errors. Error List checks the document to the location of the insertion point then displays a list of all the errors found to that point.

- Using Logic Errors, the markup is compared to the selected WordPerfect document type. The list displays errors found in the placement of tags and the contents of tag attributes.
- Using ID/IDREF, the list displays an error if the document contains an ID that does not have a corresponding IDREF attribute.

External entities

An entity is declared as "external" by specifying a PUBLIC and/or SYSTEM identifier instead of a literal replacement string. The content of an external entity is contained in a separate file. External general entities are used in documents to reference external files such as graphics. For example:

```
<!DOCTYPE book SYSTEM "book.dtd" [

<!ENTITY graphic1 SYSTEM "windmill.bmp">

]]>

<book><para>This is a picture of a windmill: &graphic1;</para></book>
```

External parameter entities are used to split a DTD into separate modules, as is done with the ISO character sets. For example:

```
<!DOCTYPE book [

<!ELEMENT book (para+)>

<!ELEMENT para (#PCDATA)>

<!ENTITY % ISOlat1 PUBLIC "ISO 8879:1986//ENTITIES Added Latin 1//EN">

%ISOlat1;

]]>
```

In the WordPerfect editing environment, external entities are referred to as "File References". Non-external entities are called "Text References".

External parameter entities

An external parameter entity is a place holder for a particular type of object. For example, an external parameter entity retains the size and file-type information for a graphic (general entity.) When you create the DTD, you don't need to specify which graphic picture will occur there (that would be an entity.) Instead, the notation indicates that a graphic would be appropriate in that location, and that the graphic will be of a certain file-type, with a particular color array, etc.

Find errors

Find Errors validates the contents of the document as well as the structure. Find Errors checks tags and attributes, exactly as Error Lists does, but it also parses (reads) the text in your document to check entity references and to warn you about non-valid characters. Find Errors starts checking the document from the top of the document and stops and displays a message at the first error it finds.

General entities

The simplest general entity is one which has a literal replacement text string:

```
<!ENTITY WP011 "WordPerfect Office11">
```

Such an entity would be referenced in an XML or SGML document as shorthand for a commonly used phrase.

Group Connectors

A comma (,) indicates that the element must occur in the specified order.

An ampersand (&) indicates that the elements may occur in any order.

A pipe (|) indicates that only one of the elements can occur.

Hard Page

A Hard Page forces a new page.

Hard Space

A Hard Space keeps two words together on the same line.

Hyphen

A hyphen always displays as a '-' character, but also acts as a wrapping point at the end of a line.

Hyphen Soft Return

A hyphen soft return is a never-displayed marker placed in a word to indicate where the word should be split when it falls at the end of a line.

Invalid elements

Elements with invalid parent elements, attributes, or attribute values appear as invalid elements in the Element list (for example, [Invalid]).

Layout Specification Instance File (.LSI)

The Layout Specification Instance (.LSI) file specifies formatting information (bold, underline, etc.) for the start and end tags in a WordPerfect XML template. When compiled by the WordPerfect DTD Compiler, the LSI is integrated in a WordPerfect Template along with the DTD file and the .LNM (Alias) file. If a layout file has been created for the WP XML template you are using, you can view the format of your XML document in the document window. Layout files are created using the XML Project Designer.

Listed element order

Listed Element Order allows the Find and Replace feature to limit its search to text that appears only within a specific element contained within another specific element.

For example, if Elements to Search In 1 specifies the element <BOOK> and Elements to Search In 2 specifies the element <TITLE>, WordPerfect searches for a <TITLE> element contained within a <BOOK> element, then searches the text until it encounters the </TITLE> end tag. It searches text again when it locates another <TITLE> element within a <BOOK> element.

Log file

The log file contains the details of the DTD compiling process.

Logic file (.LGC)

A binary version of the DTD created in WordPerfect.

Marked sections

In an SGML document, marked sections are selected areas of a document that have been tagged as CDATA, RDATA, IGNORE, or INCLUDE. Use:

- IGNORE when you do not want the content of the marked information to be part of the document.
- INCLUDE when you want the content of the marked section to be included in the document
- CDATA when you want to indicate that the content of the marked section contains no markup
- RCDATA when you want to indicate that the only kinds of markup recognized in the content of the marked section are references and character references

You can also create marked sections that combine several or all of these tags.

The IGNORE tag has priority over all other tags. If there is no IGNORE tag, the INCLUDE tag is assumed regardless of whether it is stated or not. The CDATA tag overrides RCDATA.

In XML documents, the only type of marked section that is valid is CDATA. The Marked Sections dialog is not accessible when you are working with an XML document.

Markup declarations

Markup declarations are the components of a DTD. For example, elements defined in the DTD are known as element declarations.

Markup languages

A markup language consists of tags that identify groups of information. SGML, HTML, XML are all markup languages. For example, the title of a book could be marked up with a start tag and an end tag such as <title>Work in the 21st Century</title>.

Notations

Notations may be used:

- for external entities which contain data that is not XML or SGML character data or markup. In the declaration for such an entity, a notation name is used to identify the type of data contained in the entity, and to notify the XML or SGML processor not to process it.
- as an attribute associated with an element. When a value is specified for such an attribute in the element's start tag, it means that the element's content is to be interpreted in a non-standard way by an external process. The XML/SGML processor does not try to process the element's content in the normal way.
- as the "target" of a processing instruction in XML only. This identifies an external process which is intended to act on the characters appearing in the processing instruction.

In SGML, attributes can be associated with a notation name using an attribute list declaration in a DTD. In XML, notations cannot have associated attributes.

Occurrence indicators

A plus sign (+) indicates that the element must occur at least once in the document.

A question mark (?) indicates that the element is optional and does not need to occur, but if the element does occur, it can occur only one time in the document.

An asterisk (*) indicates that the element is optional and can occur multiple times in the document.

OLE Objects

Information from a file created in one program (the source program) that has been inserted into a file in another program (the destination program). For example, you can embed a picture created in CorelDRAW into a WordPerfect document.

Optional elements

Elements that are defined in the SGML content model as optional.

Parameter entities

Parameter entities are most often used within a DTD as shorthand for one or more markup declarations. In SGML DTDs, they can also be used as shorthand for single tokens in a markup declaration For example:

XML and SGML

```
<!ENTITY % DataElmt "<!ELEMENT data (#PCDATA)>">
%DataElmt;
```

SGML only

```
<!ENTITY %data "#PCDATA">
<!ELEMENT para (%data; | quote | emphasis | graphic)>
```

Parent

A parent is an element that contains another element. Whether or not an element can contain another element is specified in the element's content model in the DTD.

Processing instructions

A processing instruction contains data which is not considered part of a document or DTD. This data may take the form of formatting commands, or a command line for an external process, or any number of other formats. XML or SGML processors do not act directly on such data, but may pass it on to its intended target. In XML, the first word in each processing instruction is the name of the intended "target", which must match the name of a notation declared in the DTD.

Projects

Each WordPerfect project is grouped within a defined category and associated with a WordPerfect template. This allows you to organize XML documents in a meaningful way and structure XML documents within the same category. You must define a WordPerfect project before you begin creating an XML document.

PUBLIC identifiers

A PUBLIC identifier provides an indirect method of representing the contents of an external entity. In general, PUBLIC identifiers are descriptive phrases that are agreed upon by several persons or organizations. An entry in a catalog file maps a PUBLIC identifier to an actual file containing the data represented by the PUBLIC identifier.

References

In cross-references, this is the place in your document from which you direct a reader to other parts of the document.

Root element

The root element is the base element for the XML document from which validation starts. All elements are nested within the root element. An XML document cannot be created without a root element.

SGML declaration file

The SGML declaration file controls information such as which character set is used in the document, how long element names can be, and other important information. WordPerfect provides a declaration file called Default.dec. This file can be edited in an ASCII editor. You can also edit Default.dec in WordPerfect, but you must save it as an ASCII DOS text file. The declaration file may also be included (in-line) in your DTD.

SGML entity declaration files (*.ENT)

Use the ISO*.ENT files to access the public ISO character entity sets described in section D.4 of the SGML standard, ISO 8879-1986(E). The WP*.ENT files can be used to access the WordPerfect Character entity sets included with WordPerfect. You can use these files directly or as examples for creating your own external entity files.

ISOAMSA.ENT

ISOAMSB.ENT

ISOAMSC.ENT

ISOAMSN.ENT

ISOAMSO.ENT

ISOAMSR.ENT

ISOBOX.ENT

ISOCYR1.ENT

ISOCYR2.ENT

ISODIA.ENT

ISOGRK1.ENT

ISOGRK2.ENT

ISOGRK3.ENT

ISOGRK4.ENT

ISOLAT1.ENT

ISOLAT2.ENT

ISONUM.ENT

ISOPUB.ENT

ISOTECH.ENT

WPSET1.ENT

WPSET3.ENT

WPSET4.ENT

WPSET5.ENT

WPSET6.ENT

WPSET7.ENT

WPSET8.ENT

WPSET9.ENT

Soft Hyphen

A soft hyphen is a wrapping point which is not displayed unless it is at the end of a line, in which case it is displayed as a '-' character.

Standard Generalized Markup Language (SGML)

SGML is an open international standard issued by the International Organization for Standardization (ISO). The SGML standard provides a platform and application-independent environment for defining document architecture and document markup.

Because SGML isolates the formatting from the document content, you can transfer both the document text and its associated formatting across platforms and between applications. In SGML, elements form the structural building blocks of a document. For example, the structure of a book can be broken down into elements such as a title page, a table of contents, chapters, appendices, a glossary, and an index.

The SGML standard provides a method for defining each element in the content of a document. This allows advanced formatting to be applied to the SGML document so that the same content can be printed multiple ways without changing or converting the content. For example, you can print a book as a hard bound version and as a paperback version, which uses a smaller page size, font and different margins without changing the content file.

Supported Table Types

The following table DTD fragments and the HTML DTD are included with WordPerfect. The HTML DTD contains a table fragment that can be used in other DTDs.

DTD	Explanation
Aap_tbl.dtd	Defined by Association of American Publishers (AAP).
Cals_tbl.dtd	Table type defined by the Cals.dtd.
Cals_xml.dtd	CALS table information considered relevant by the SGML features of WordPerfect.
Wptable.dtd	Support for WordPerfect 5.x-level tables.
Wp60tbl.dtd	Full-featured table DTD designed to work with WP 6.x, 7 and 8 tables.
Wp60min.dtd	Simplified version of Wp60tbl.dtd.
Html_tbl.dtd	HTML 3.2 level DTD that contains a table fragment that can be used in other DTDs.

SYSTEM identifier

A SYSTEM identifier provides an indirect method of representing the contents of an external entity. An entry in a catalog file maps a SYSTEM identifier to an actual file containing the data represented by the SYSTEM identifier.

Tags

A tag is an XML or SGML markup code. XML or SGML elements are generally identified with a start tag and an end tag. For example, the start tag for the element Name is `<Name>` and the end tag is `</Name>`. The end tag is identified with a backslash.

Target

For cross-references, this is where the cross-reference leads.

Target Name

Ties a reference and target together. When the program generates cross-references, it uses the target name to match a reference with a target. The target name is used only when you generate cross-references. It is not printed.

Unicode Character Set

Unicode is an international standard developed by the Unicode Consortium. The Unicode character set is the 16-bit base character set used by XML. It includes a very large international character set.

Unknown elements

Elements defined in the layout file you retrieve that are not defined in the current compiled DTD appear as unknown elements in the Element list (for example, [Unknown] Memo).

Valid documents

A valid XML document is one that is both "well-formed" and that follows the pattern defined by the associated DTD.

Well-formed documents

A well-formed XML document is one that complies with the well-formed constraints given in the XML specification issued by the World Wide Web Consortium (W3C).

WordPerfect Character Mapping Files (*.WPC)

The *.WPC files contain the WordPerfect Character entity sets and ISO character entity sets. You can use these files directly or as examples for creating your own WordPerfect Character entity sets.

ALLCHARS.WPC
ISO8879.WPC
ISOAMSA.WPC
ISOAMSB.WPC
ISOAMSC.WPC
ISOAMSN.WPC
ISOAMSO.WPC
ISOAMSR.WPC
ISOBOX.WPC
ISOCYR1.WPC
ISOCYR2.WPC
ISODIA.WPC
ISOGRK1.WPC
ISOGRK2.WPC
ISOGRK3.WPC
ISOGRK4.WPC
ISOLAT2.WPC
ISONUM.WPC
ISOPUB.WPC
ISOTECH.WPC
WPCHARS.WPC

WordPerfect DTD Compiler

You use the WordPerfect DTD Compiler to create a binary version of your Document Type Definition files (DTDs). The resulting WordPerfect template contains the compiled DTD, the .LSI, and the Alias file. WordPerfect uses the information in the template to create and validate your XML documents. You must create a template to use the XML features in WordPerfect.

XML Character References

Many characters have a special meaning in XML. For example, the ampersand and less than symbols (&,<) are used to begin the delimiters for tags and references. If you want to use one of these characters in the text of your document, you can insert the character as a character reference so it is not interpreted as markup.

In WordPerfect, click Insert ▸ Symbol to insert character references. XML character references appear as characters in the document window. However, when the document is saved in XML format, the character is translated into a character entity reference.

The WordPerfect XML Project Designer

You use the WordPerfect XML Project Designer to create files that automatically render SGML or XML documents in WordPerfect. A layout file (.LSI) assigns formatting to different elements in a document, such as larger fonts for headings, italics for references, or bold for emphasis. You can create different layouts for the same DTD so you can print the finished document to different paper sizes or to an online format.

Field 1: WordPerfect Character Number or WordPerfect Function Name

Field 1 of a WordPerfect Character mapping entry has three components: the number of the WordPerfect Character set (or Corel Symbol set) that contains the character, a comma (,), and the number of the character within the set.

For example, á is character number 27 in character set 1, so its value in field 1 is 1,27.

Entities that are mapped to characters 0-31 from WordPerfect character set 0, and entities that are mapped to WordPerfect Characters which have already been mapped, are not handled by the WordPerfect Character mapping feature. Instead these entity names appear in the SDATA entities list in the Text References dialog box. Select the entities from that list in order to include them in your document.

Field 2: Entity Name

Field 2 of a WordPerfect Character mapping entry is the entity name. This is the name that will be used for any matching entity references when the file is saved in SGML format.

Lesson 2: Getting started

Lesson 2: Getting started

In this lesson, you'll start working through the XML document creation process. Using the wizard, you will complete the first step in XML document creation: defining an XML project and compiling a document type definition (DTD).

In this lesson, you'll

- explore the concept of a DTD
- learn about compiling
- explore the WordPerfect DTD Compiler window
- compile a DTD

You'll need about 40 minutes to complete this lesson.

{button ,AL(`PRC Lesson 2 Getting started;',0,"Defaultoverview",)} Related Topics

What is a document type definition?

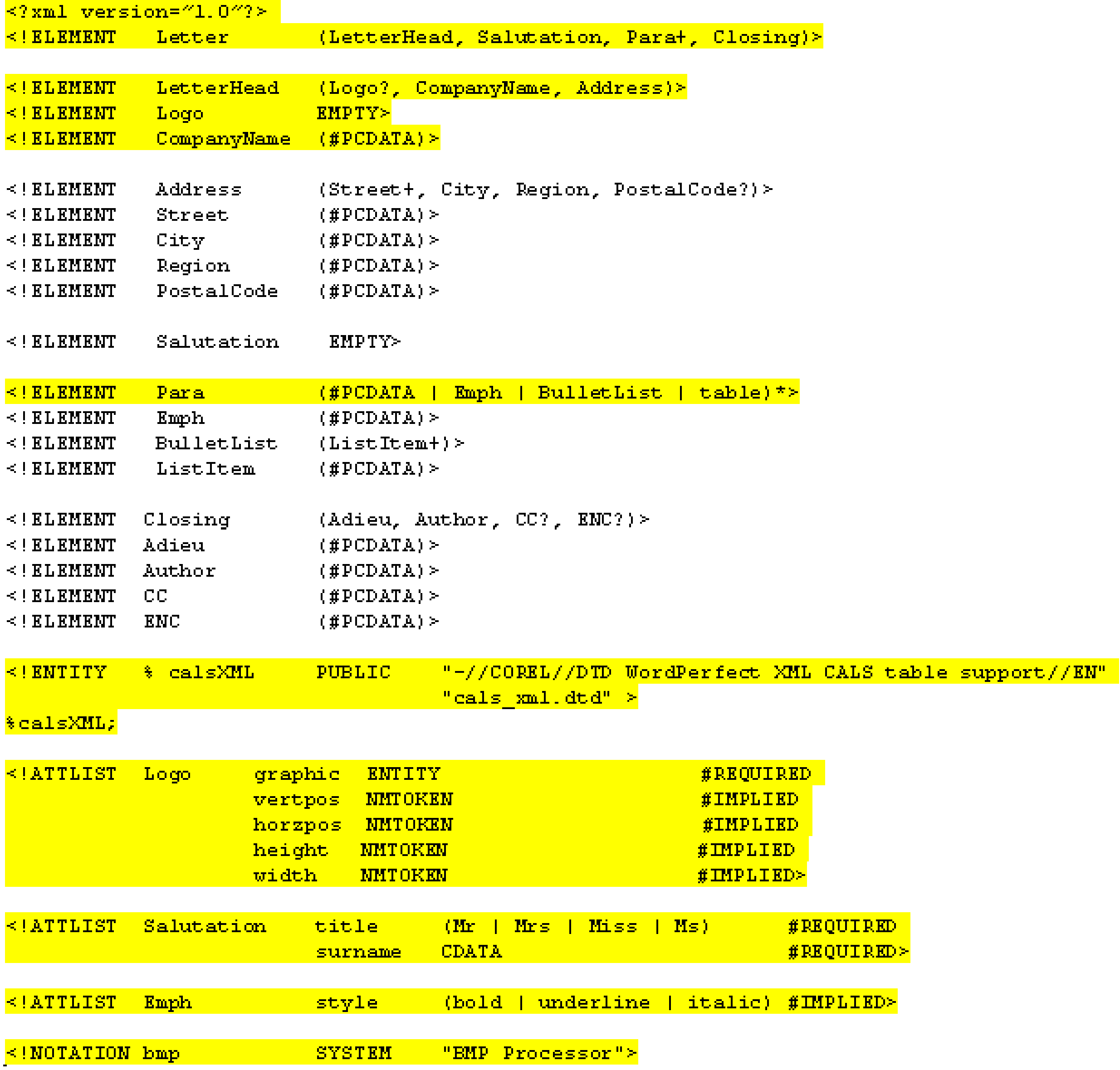
Document Type Definitions (DTDs) define document components and structure. Some of the components (or markup declarations) that are defined in a DTD include

- elements
- attributes
- notations
- entities

WordPerfect uses the DTD to validate the associated XML document. During the validation process, the contents of the DTD elements are compared with the associated XML document to ensure that the XML document follows the structure and content defined in the DTD.

To learn about the markup declarations defined in a DTD

- Click any of the highlighted markup declarations defined in the picture below.



```
<?xml version="1.0"?>
<!ELEMENT   Letter       (LetterHead, Salutation, Para+, Closing)>

<!ELEMENT   LetterHead   (Logo?, CompanyName, Address)>
<!ELEMENT   Logo         EMPTY>
<!ELEMENT   CompanyName  (#PCDATA)>

<!ELEMENT   Address      (Street+, City, Region, PostalCode?)>
<!ELEMENT   Street       (#PCDATA)>
<!ELEMENT   City         (#PCDATA)>
<!ELEMENT   Region       (#PCDATA)>
<!ELEMENT   PostalCode   (#PCDATA)>

<!ELEMENT   Salutation    EMPTY>

<!ELEMENT   Para         (#PCDATA | Emph | BulletList | table)*>
<!ELEMENT   Emph         (#PCDATA)>
<!ELEMENT   BulletList   (ListItem+)>
<!ELEMENT   ListItem     (#PCDATA)>

<!ELEMENT  Closing       (Adieu, Author, CC?, ENC?)>
<!ELEMENT  Adieu         (#PCDATA)>
<!ELEMENT  Author        (#PCDATA)>
<!ELEMENT  CC            (#PCDATA)>
<!ELEMENT  ENC           (#PCDATA)>

<!ENTITY   % calsXML     PUBLIC    "-//COREL//DTD WordPerfect XML CALS table support//EN"
                                   "cals_xml.dtd" >
%calsXML;

<!ATTLIST  Logo     graphic  ENTITY                     #REQUIRED
                    vertpos  NMTOKEN                    #IMPLIED
                    horzpos  NMTOKEN                    #IMPLIED
                    height   NMTOKEN                    #IMPLIED
                    width    NMTOKEN                    #IMPLIED>

<!ATTLIST  Salutation    title     (Mr | Mrs | Miss | Ms)       #REQUIRED
                         surname   CDATA                        #REQUIRED>

<!ATTLIST  Emph          style     (bold | underline | italic) #IMPLIED>

<!NOTATION bmp           SYSTEM    "BMP Processor">
```

{button ,AL(`PRC Lesson 2 Getting started;',0,"Defaultoverview",)} Related Topics

XML declaration

The XML declaration is a processing instruction that identifies a document as an XML document and specifies the version of XML that was used to author the document. XML does not require an XML declaration.

Entity declaration

An entity declaration specifies the name of an entity and information about its content or replacement text. An entity can be declared as either a general entity or a parameter entity. This entity declaration defines table support for the documents created using this DTD.

DOCTYPE declaration

A document type declaration is a markup declaration that starts with a DOCTYPE statement. The document type declaration begins an SGML document. This declaration includes or references the markup declarations for elements, entities, etc. that make up the Document Type Definition (DTD). The root element is specified in the document type declaration. If no document type declaration is included, you are prompted to supply the name of the root element and optional PUBLIC and SYSTEM identifiers.

ELEMENT Letter

The element `Letter` is the root element for this DTD and must contain the elements `LetterHead`, `Salutation`, `Para`, and `Closing`. The plus sign after `Para` indicates that the element `Letter` can contain more than one paragraph.

ELEMENT LetterHead

The element `LetterHead` can contain a logo and must contain the elements `CompanyName` and `Address`. The question mark after `Logo` indicates that `Logo` is an optional element.

ELEMENT Logo

The element `Logo` is an empty element. It acts as a marker that contains no markup or text. The element Logo has an attribute assigned to it in the attribute list below (!ATTLIST Logo). The attribute associates the element `Logo` with a graphic file such as Corelsym.bmp.

ELEMENT CompanyName

The element `CompanyName` must contain text.

ELEMENT Para

The element `Para` must contain text, `Emph`, `BulletList` element or `Table` element. The vertical line indicates an OR operation and the asterisk indicates that the elements within `Para` can appear any number of times.

ENTITY % calsXML

This entity declaration subset contains the CALS table information that is considered relevant by the SGML features of WordPerfect. The CALS_XML.dtd is NOT a complete DTD, and cannot be compiled into an XML template file without modification.

Although the CALS_XML.dtd can be included in an arbitrary DTD to add some level of CALS table support, it does not include all of the elements, entities, and attributes found in the complete DTDs in MIL-M-28001. It is intended primarily for your information, and is NOT for use in CALS-compliant environments.

Sample invocation:

```
<!ENTITY WP-CALS PUBLIC "-//COREL//DTD WordPerfect XML

CALS table support//EN">

%WP-CALS;
```

ATTLIST Logo

The attribute for the element `Logo` is called `graphic` and is required. `Graphic` references an external file that is defined in the ENTITY statement.

ATTLIST Salutation

The attribute list for the element `Salutation` includes two required attributes: `title` and `surname`. The attribute `title` must contain one of the following: Mr, Mrs, Miss or Ms. The attribute `surname` must contain text.

ATTLIST Emph

The attribute list for the element `Emph` includes an implied attribute called `style`. The `style` attribute can contain either bold, underline, or italic. An implied attribute is optional.

NOTATION

This notation specifies that the format of the unparsed entity `graphic` is bitmap. Notations can identify the format of unparsed entities, the format of elements with notation attributes, or the application to which a processing instruction is addressed.

Understanding the DTD compilation process

WordPerfect uses document type definition (DTD) information to tag and validate documents. You must compile a DTD in order for WordPerfect to access DTD information.

The WordPerfect DTD Compiler creates a WordPerfect template using the compiled DTD, a layout file and an alias file. Including all required components in one template makes it easy to distribute XML documents: you send the XML document and the associated WordPerfect template.

When you use the WordPerfect DTD Compiler, you must specify a catalog file if external entities are referenced in the DTD you want to compile. The catalog file specifies which files on your system correspond to the external entities in the DTD. For more information about catalogs, see Working with catalog files .

As part of the compilation process, WordPerfect checks the DTD for syntax errors. The WordPerfect DTD Compiler will not create a WordPerfect template if it finds any errors in the DTD. Compilation errors are displayed in the WordPerfect DTD Compiler Log window. For more information about an error, double-click the error in the Log window.

By default, the WordPerfect DTD Compiler creates WordPerfect templates in the \Template\Custom WP Templates\XML folder. To access the WP templates files in WordPerfect, you can specify this location as the default folder in the file settings. For more information about file settings, see "Compiling a DTD".

{button ,AL(`PRC Lesson 2 Getting started;',0,"Defaultoverview",)} Related Topics

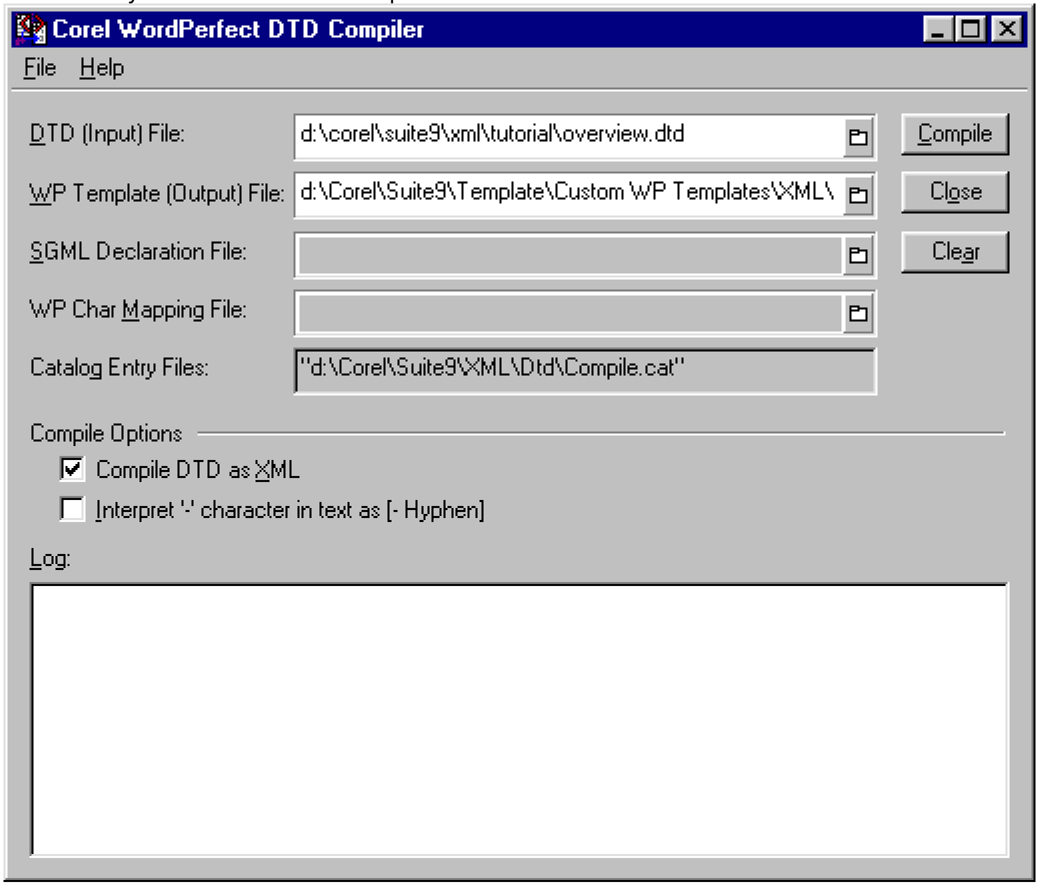
Exploring the WordPerfect DTD Compiler window

The WordPerfect DTD Compiler window displays the settings specified for the current WordPerfect template. When you create a new template, all fields are empty.

The picture below shows you the default WordPerfect DTD Compiler window.

To learn about the elements of the WordPerfect DTD Compiler window

- Click any of the elements in the picture below.



DTD (Input) File: d:\corel\suite9\xml\tutorial\overview.dtd

WP Template (Output) File: d:\Corel\Suite9\Template\Custom WP Templates\XML\

SGML Declaration File:

WP Char Mapping File:

Catalog Entry Files: "d:\Corel\Suite9\XML\Dtd\Compile.cat"

Compile Options

☑ Compile DTD as XML

□ Interpret '-' character in text as [- Hyphen]

Log:

{button ,AL(`PRC Lesson 2 Getting started;',0,"Defaultoverview",)} Related Topics

DTD (Input) File

Specify the Document Type Definition (DTD) you want to compile as a WordPerfect template.

WP Template (Output) File

Specify a filename for the WordPerfect template. The template incorporates the compiled DTD file (or logic file), the layout specification file (.LSI) and the Alias file (.LNM). By default, the WordPerfect DTD Compiler uses the name of the DTD input file as the filename for the WP template output file, but adds a .WPT extension.

SGML Declaration File

Specify the SGML declaration file you want to use with the DTD you are compiling. If you do not specify an SGML declaration file, and the DTD you are using does not contain an embedded SGML declaration file, the WordPerfect DTD Compiler uses the SGML declaration file specified in the file settings. For information about file settings, see "Customizing the WordPerfect DTD Compiler".

If no declaration file is specified in the file settings, the WordPerfect DTD Compiler uses the SGML declaration in DEFAULT.DEC. The DEFAULT.DEC declaration matches the WordPerfect SGML support, which means it does not support short reference delimiters as defined in SGML's reference concrete syntax. DEFAULT.DEC is usually located in the folder \Program Files\Corel\WordPerfect Office 11\XML\DTD on the drive where WordPerfect Office 11 is installed.

If you enter a path and filename in the SGML Declaration File text box, the WordPerfect DTD Compiler searches for the SGML declaration only in the path you specify. If you enter only the filename, the WordPerfect DTD Compiler searches the current working folder, then searches the Program Files\Corel\WordPerfect Office 11\XML\DTD folder.

This text box is not accessible when you enable the XML check box.

Note

- If the document type definition you specify contains an embedded SGML declaration, the declaration contained in the DTD takes precedence over any declaration specified in the SGML Declaration text box.

WP Char Mapping File

Specify the WordPerfect character mapping file that defines which WordPerfect characters correspond to the character entities defined in the document type definition. If you do not include a folder for the WP character mapping file, the WordPerfect DTD Compiler searches the current working folder. If the mapping file is not found, the WordPerfect DTD Compiler searches the current working folder, and then searches the folder Program Files\Corel\WordPerfect Office 11\XML\DTD\Mapfiles.

This field is not accessible when you enable the XML check box.

For more information about the WordPerfect character mapping file, see "Mapping to WordPerfect template files.

Catalog Files

Specify the catalog file that defines entity references. If the document type definition you are compiling contains references to external entities, you need to include at least one catalog file in the Active Catalog Files list box of the Catalog File Manager.

For information about catalogs, see Working with catalog files.

Note

- When an external entity is referenced in a DTD, the WordPerfect DTD Compiler attempts to match the entity's declaration to an entry in the catalog files. The WordPerfect DTD Compiler searches the catalog files for an entry that matches the entity being referenced. If it finds a matching entry, the filename specified in the matching entry is used as the name of the entity file. If the filename for a matching entry includes a path, the WordPerfect DTD Compiler searches for the entity file at the specified path. If no path is included in the entry, the WordPerfect DTD Compiler searches for the file in the folder containing the catalog file. If a relative path is included in the entry, it is considered to be relative to the folder containing the catalog file.

Compile DTD as XML

Enable this check box to compile the DTD as XML.

Interpret '-' character in text as [-Hyphen]

Enable this check box to specifies that the '-' character is interpreted as a hyphen rather than a non-wrapping dash character.

Log window

The Log window displays errors and warnings resulting from the compilation process. For more information about errors or warnings double-click the error or warning in the Log window.

Compiling letter.dtd

When you create a new XML document, the first step is to compile a document type definition and create a WordPerfect template. In the following procedure, you will compile the letter.dtd and create a WordPerfect template that you will use to create a business letter in XML. WordPerfect uses the resulting letter.wpt template to validate the XML letter you create.

After you compile letter.dtd, you will create a generic layout for the letter. You will modify this generic layout in Lesson 4.

You must then specify a catalog file. Catalog files contain references to external information referenced from the DTD or the XML document.

To compile letter.dtd

1. Start WordPerfect, then click the File menu, New XML Document.
2. Click the New button, from the Categories/Projects list box select the XML category, then in the Project box, type "letter" then click the Next button.

   You have now created a new project under the XML category that you can re-use to create new XML documents that conform to letter.dtd.
3. Click the Next button to accept the default name for the WordPerfect Template (letter).

   A WordPerfect template called letter.wpt will be created that contains the rules governing the structure of the letter you will create.
4. Enable the Compile DTD And Create Layout option button, click the Next button, then click the Compile A DTD button.
5. Enable the Compile DTD As XML checkbox.
6. Click the folder icon next to the DTD (Input) File box, choose letter.dtd located in the /Corel/WordPerfect Office 11/XML/Tutorial folder, click the Select button, then click the Compile button.
7. In the Document Type Declaration dialog box, click OK to accept `Letter` as the root element for the letter you are creating and to accept the system path to letter.dtd.
8. In the Compile Status dialog box, click OK, then click the Close button to exit the WordPerfect DTD Compiler.
9. Click the Next button.

To create a generic layout

1. Enable the Create Generic Layout When Project Designer Invoked checkbox, then click the Project Designer button.

   The WordPerfect XML Project Designer automatically creates generic layout rules for all the elements defined in letter.dtd.
2. Click the File menu, Save.
3. In the Layout Name box, type "letter," then click the Save button.
4. Click the File menu, Exit.
5. Click the Next button.

To specify a catalog file and continue in the wizard

1. Click the Next button and accept the default catalog file (Import.cat), then click the Finish button.
2. From the Categories/Projects list box, choose the XML category and the `letter` project, then click the Select button.

   A blank XML document opens in the WordPerfect XML Editor.

Note

- To create a generic layout, WordPerfect creates rules based on the element's content model and the content model of its possible parents. If neither an element nor its parents contain text, a rule with a new line for the start tag is created. If an element's parents do not contain text, then a rule with a new line for the end tag is also created.

{button ,AL(`PRC Lesson 2 Getting started;',0,"Defaultoverview",)} Related Topics

Summary of Lesson 2

In this lesson, you learned about

- document type definitions (DTDs)
- the compilation process
- the WordPerfect DTD Compiler window
- compiling a DTD

In Lesson 3, you'll create a new XML document, and insert elements and text.

{button ,AL(`PRC Lesson 2 Getting started;',0,"Defaultoverview",)} Related Topics