

WORKING WITH XML DOCUMENTS

In this chapter

- Understanding XML 76
- Creating an XML Document 78
- Formatting an XML Document 84
- Publishing an XML Document 89
- Understanding Advanced XML Tasks 90
- Troubleshooting 94

UNDERSTANDING XML

XML—it seems to be all the rage these days. Nearly any up-to-date software product touts its XML capabilities as if you already knew what it is and why it's important to you. We can't promise that you'll understand XML completely by the time you finish this chapter, but we do hope you'll have a better idea as to what XML is and how you can use it in practical ways.

In a nutshell, *XML* is *Extensible Markup Language*, a method for defining document information and making it easy to transmit or reuse that information, without regard to proprietary software or operating systems.

WHAT'S IN XML FOR YOU

Before jumping into definitions for XML, you probably want to know whether it's necessary or even worth the effort to learn about XML.

Honestly? For most individual work, using XML is of little intrinsic value or interest. If you work on your own, and creating good-looking documents for printing is your primary use for WordPerfect, you do not need to read much further, unless, of course, you'd like to know what the fuss is all about. If you're in an environment where sharing and reusing document information is important, you probably do need to know about XML. In any case, it's quite possible that someday XML will also be useful for the rest of us, so a quick tour of this chapter might be worth your time.

One of the great challenges in our information society is the proliferation, and repetition, of information. How many times have you created a document, only to find that someone needs the same information but they use a different word processor? Sometimes, you have no common format other than the Web or a PDF version, both of which enable you to see the information but offer virtually no capability to reuse that information directly.

Consider another example. You write the Great American Novel, and, of course, you publish it first in hard-bound form. It sells well, and the publisher now wants to make it available in paperback or even electronic formats. In the past, the publisher would have to spend a great deal of time reworking the document to adapt it to the new formats: with smaller type and different page layout for a paperback version, or with links and navigation tools for an electronic book or PDA version.

XML comes to the rescue by making it possible to reuse information without having to replicate or re-create it. In XML, when you create your documents, you define the elements of the document in structured ways: book, chapters, sections, paragraphs, and so on. You then apply a definition document that interprets what those elements mean and how they should be used, and only then do you worry about how those elements are to be displayed.

Again, however, this probably isn't enough to drive you to use XML. More likely, you've come to this chapter because your company is requiring you to create XML-compliant documents. In a multiuser environment, the reuse of information in a variety of contexts is probably the best argument for understanding and using XML.

Although XML can be quite complicated, how much you need to know about XML can vary widely from those who design the Document Type Definition files (a lot), to those who create the basic document information (not too much).

NOTE

Working with XML involves new concepts and approaches to publishing a document. Before you try to learn how to use XML in WordPerfect, you should be thoroughly familiar with all WordPerfect basics. You should also be familiar with current Windows operating systems, including how to use Windows-based applications, how to browse for files, and so on.

DEFINING XML

XML stands for Extensible Markup Language, and it's a simpler, easier-to-use subset of SGML, or Standard Generalized Markup Language.

More likely you've heard of HTML, or Hypertext Markup Language. So how are these different? Consider the following snippet of code in HTML:

```
<html>
<body>
<h1>Working with XML</h1>
How are <i>HTML</i> and <i>XML</i> different?<p>
</body>
</html>
```

HTML uses *tags* to mark up the text, which then is translated by your Web browser to show such elements as heading styles (<h1>) and attributes such as italics (<i>). Anyone used to seeing WordPerfect's Reveal Codes recognizes the use of paired tags to turn on or off attributes or formatting features. HTML codes are standardized so that any Web browser can read and interpret them. Their primary purpose is to help browsers *format* content. If you need to go beyond standard HTML definitions, you might be out of luck.

Now, consider the following snippet of XML code:

```
<fruitbasket>
<apples>Granny Smith</apples>
<pears>Bartlett</pears>
<grapes>Concord</grapes>
</fruitbasket>
```

None of these codes are anything you ever learned in an HTML class. Nevertheless, you notice that this code is *well formed*. That is, each beginning tag has an ending tag, and each set of tags is properly nested within a hierarchical structure.

The problem, however, is that unless someone tells us what these tags mean, they aren't that useful. Fortunately, XML enables you to use or create definitions for its elements in separate files called *Document Type Definitions*, or *DTDs*. When an XML document is paired with a DTD, and its elements match those of the DTD, it is said to be valid. Proper XML must be both *well formed* (without syntactic errors) and *valid* (elements in the right places as determined by the DTD).

NOTE

Several government and professional organizations have created their own standardized DTDs that facilitate the exchange of information in their particular arena. These include the DocBook DTDs designed for narrative and technical books or the NITF for the newspaper industry.

WORDPERFECT'S APPROACH TO XML

WordPerfect tries to make creating XML documents easier by offering the same familiar WordPerfect interface, along with a structured, yet simplified, XML editing environment.

In the XML domain, you typically must create a DTD and then create an XML document using the elements defined in the DTD. You also have to validate your XML document and then apply formatting layouts to it depending on how you want to use the document's information.

WordPerfect incorporates the same elements but a bit differently. Starting with a DTD, you compile the DTD, which becomes part of a WordPerfect template. You then create your document using a template. You can then use or create new layouts and quickly switch from one layout to another, while using the same XML document.

CREATING AN XML DOCUMENT

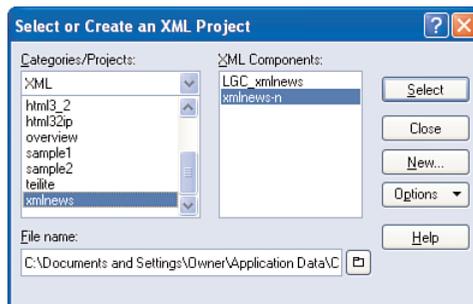
We're going to put the cart before the horse for a while. As noted previously, you need a DTD and a template before you can actually create an XML document. However, most of you won't ever need or want to worry about creating DTDs or templates. So we'll start with the assumption that someone else already has created these for you, and you're now ready to create a document.

CHOOSING AN XML TEMPLATE

Fortunately, WordPerfect comes with a few sample templates. For example, to create a newsletter article from the XML template for news, follow these steps:

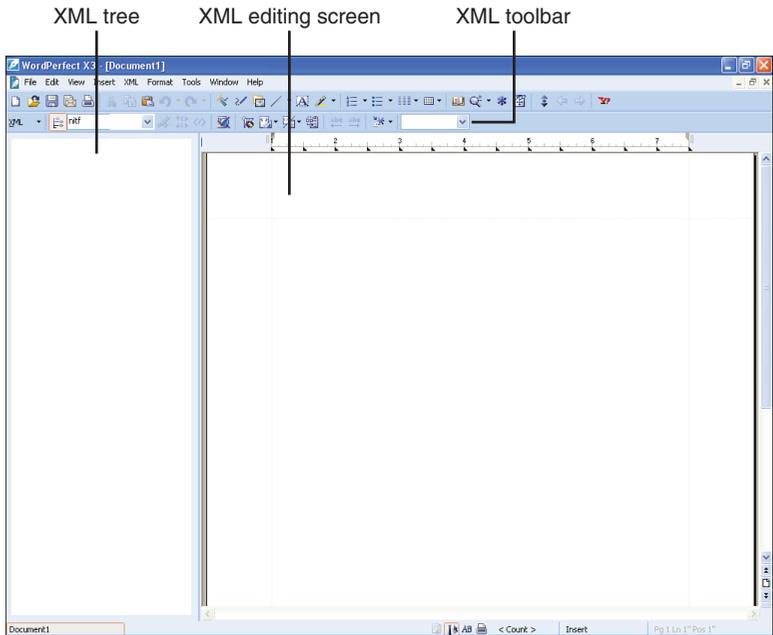
1. Choose File, New XML Document. WordPerfect displays the Select or Create an XML Project dialog box (see Figure W3.1, which shows a project already selected).

Figure W3.1
You create an XML document by selecting an XML project based on XML templates.



2. Select a category/project from the left column—for example, xmlnews.
3. In the XML Components column at the right, select the layout you want to work with. In this case, there are only two, so select xmlnews-n.
4. Click Select and WordPerfect displays a blank XML editing screen (see Figure W3.2).

Figure W3.2
Although you edit XML documents using familiar WordPerfect procedures, the screen layout, menus, and toolbars change to assist you in the XML task at hand.



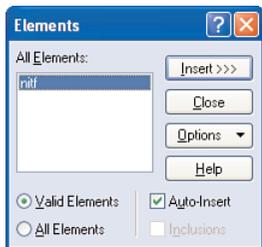
W3

CREATING XML DOCUMENT CONTENT

At this point, you could certainly begin typing as you normally do in WordPerfect, but you would create only invalid XML content. Instead, you need to insert valid XML elements and then add content within those valid elements. For example, follow these steps for adding newsletter content:

1. Choose Insert, Elements, or click the XML button on the toolbar and choose Elements. WordPerfect displays the Elements dialog box (see Figure W3.3), which provides only those elements that are valid at this point in the creation of the document.

Figure W3.3
The Elements dialog box helps you select only valid XML elements.



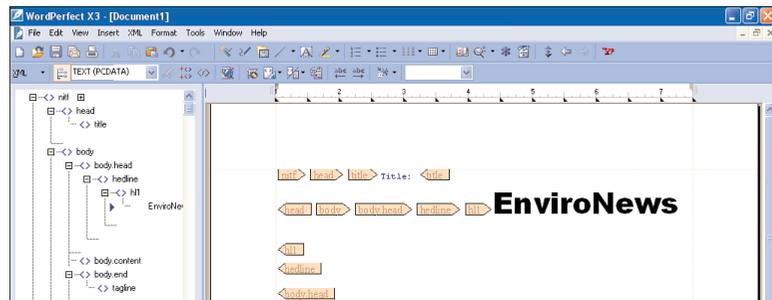
2. The only valid element is nitf (News Industry Text Format), so select it and click Insert. WordPerfect inserts a pair of nitf codes, offers to let you modify the nitf attributes, and also begins the XML tree structure in the left pane.
3. Unless you know that you want to change an attribute, you do not need to select anything. However, try selecting a version from the Attribute Name list. Click Edit Value to select it. The information appears in the actual XML code, but you do not see it in your document.
4. Click Close and WordPerfect automatically inserts the next sequence of required elements. If you want, you can type a title for the document, such as **Newsletter**. Although this isn't part of the displayed document information, it can be used as part of the structural identification of the document.
5. Click Close to close the Elements dialog box.
6. Position the cursor between the body tags.
7. If the Elements dialog box isn't already open, choose Insert, Elements. The Elements dialog box displays only those elements that are valid at this point of the document. This news article DTD allows for body.head, body.content, and body.end. Choose body.head and click Insert.

CAUTION

You can select valid elements from the Valid Elements drop-down list on the property bar. However, for some reason, the cursor tends to jump around after you select elements from this drop-down list. I recommend that you use a longer but safer method of choosing Insert, Elements and selecting elements from the Elements dialog box.

8. With the cursor between the body.head tags, you now see various valid elements, such as headline (headline), byline, and so on. Choose headline and click Insert to insert the headline tags.
9. Between the h1 (headline 1) tags, type the title of your newsletter article—for example, **EnviroNews**. Note that the font and font size are considerably larger than the regular text (see Figure W3.4). This is because a layout rule has been applied to the h1 element as part of the XML template. You'll learn about layouts later in this chapter.

Figure W3.4
Layout rules can be associated with specific XML elements.



10. Position the cursor before the closing `body.head` tag. Note that the headline element is no longer valid because you have already used it. You might want to try adding the `byline` tag and also add your name.
11. Position the cursor before the closing `body` tag and insert the `body.content` tag.
12. Position the cursor before the closing `body` tag and insert the `body.end` tag.
13. Choose File, Save, and supply a filename. WordPerfect automatically saves the document in XML UTF-8 format, but if not, choose File, Save As, and choose the XML (UTF-8) format before saving.
14. Finally, if you haven't done so already, WordPerfect asks if you want to check for errors before saving. Click OK. Then, click No to tell WordPerfect you don't need to check again. You can also close the Elements dialog box.

CAUTION

The WordPerfect XML Editor is notorious for crashing at unpredictable moments. Even more so than usual, it's a good idea to save *early* and save *often*.

You can further experiment by inserting valid `body.content` and `body.end` elements with your own text.

WORKING WITH THE XML EDITOR

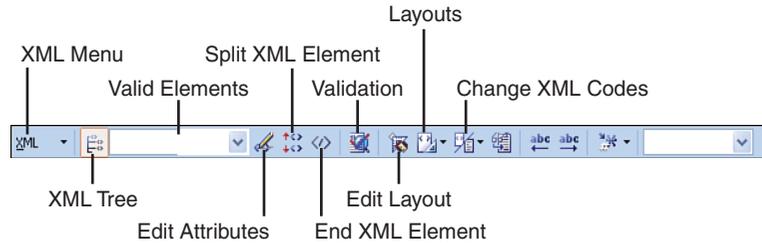
You've already discovered that the XML Editor in WordPerfect is somewhat different from the standard WordPerfect interface. Let's take some time to explore the various components of the XML Editor screen (refer to Figure W3.4).

The XML tree, at the left of the screen, shows the structure of the document and whether XML elements are valid. Although this screen can be helpful when you validate your document, it doesn't offer help much while you're creating your document. If you want more editing space, you can turn off the tree by choosing XML and deselecting Show XML Tree.

Another difference is that the Menu changes, adding XML options and removing Table options. In addition, the options under Insert and Format change dramatically to reflect XML-related tasks. Other XML Editor menu options are nearly the same as those found on standard WordPerfect menus.

Finally, the XML Editor displays the XML toolbar, which provides quick and easy access to the more important XML editing activities (see Figure W3.5).

Figure W3.5
The XML toolbar helps you quickly create valid XML documents.

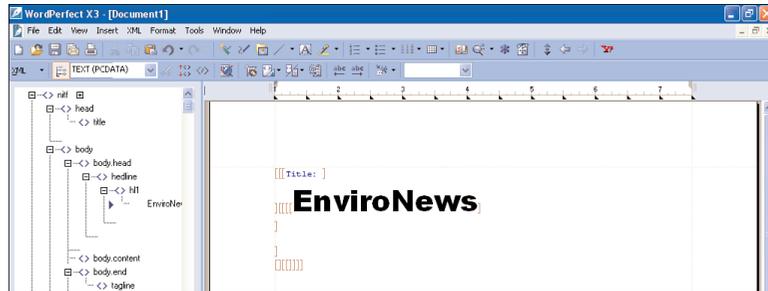


The buttons on the XML toolbar specific to editing XML documents include

- **XML Menu**—Most of the elements found under the XML menu, or represented by XML toolbar buttons, can also be found under the XML drop-down menu. Clicking the XML button can be a handy way to remember what options are available, although using a toolbar button directly might be quicker and easier when you're familiar with its use.
- **XML Tree**—Click this button to turn the XML tree view on or off.
- **Valid Elements**—This drop-down list displays elements you can use at the current position of your cursor. You can also insert valid elements from this list by clicking the element from the drop-down list.
- **Edit Attributes**—This option is reserved for those elements that require further definition, such as the root (first) element of the document. It does not apply to most elements and has nothing to do with formatting attributes.
- **Split XML Element**—Clicking this button splits the current element, in effect replicating the element. This can be particularly useful when creating a series of paragraphs. For example, with the cursor in a current paragraph, you click the button and a new pair of paragraph tags appears, without having to position the cursor outside the paragraph tags and then insert a new set of paragraph tags.
- **End XML Element**—Clicking this button usually moves the cursor past the closing tag. Because the WordPerfect XML Editor automatically creates pairs for most elements, this option isn't required to create closing element tags.
- **Validation**—At any time, but certainly before saving or publishing an XML document, you can check the validity of the elements you've used.
- **Edit Layout**—Clicking this button opens the WordPerfect XML Project Designer where you can create or modify layouts for the current XML document.
- **Layouts**—If you have more than one layout assigned to the current XML document, clicking this button enables you to choose from among them.
- **Change XML Codes**—This button's name is somewhat misleading because it doesn't actually change XML Codes. It does change how tags are displayed in your XML document. By default, XML tags display fully (refer to Figure W3.4). However, you can also choose to display no tags at all or to display markers (see Figure W3.6). As you begin

working with XML documents, you'll probably want to display the full XML tags. As you gain more experience, or if you want to view how the document will look, choose one of the other options.

Figure W3.6
You can use markers to indicate XML elements instead of using descriptive tags or not using anything at all.



VALIDATING AN XML DOCUMENT

The two requirements for an XML document are that it must be well formed and that its elements must be valid. As you create the document, WordPerfect makes it difficult to use elements incorrectly. However, as you edit a document, you might inadvertently add or delete elements in such a way that they are no longer well formed or valid.

At any time, but certainly before saving an XML document, you should validate your document. Otherwise, you defeat the purpose of an XML document because others won't be able to use or reuse your document because they won't understand its structure and elements.

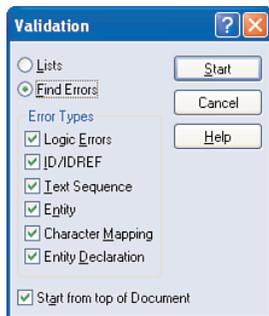
WordPerfect makes it easy to quickly determine what does and doesn't meet these criteria, using either of two methods:

- The first is to use the XML Tree view. If you turned this view off, turn it back on by choosing XML, Show XML Tree, or click the XML Tree button on the toolbar. In the XML Tree, problems are noted by an inverted yellow triangle. If you hover the mouse pointer over the triangle, WordPerfect displays a brief description of the problem.
- The second method is to validate the entire document by choosing XML, Validation, or by clicking the Validation button on the toolbar. WordPerfect displays the Validation dialog box (see Figure W3.7). By default WordPerfect tries to find all possible errors, starting at the beginning of the document. Click Start and WordPerfect quickly moves through all of the document's elements, pausing only to report problems it finds (see Figure W3.8).

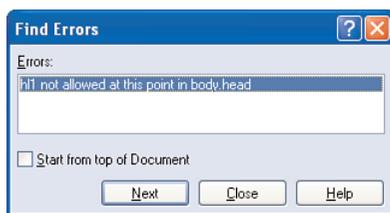
Unfortunately, until you become more experienced creating valid XML code, some of the descriptions offered during validation might be difficult to interpret. For example, if the message tells you that an element is not allowed at this point, it could mean that the element should be nested within another element and that you've either moved the nested element or accidentally deleted the element that should surround it.

Figure W3.7

Although you have many validation options, generally you stick to the defaults when validating an XML document.

**Figure W3.8**

Error messages can help you determine and fix XML coding problems.



For example, in Figure W3.8, you are told that element `h1` is not allowed at this point in `body.head`. If you position the cursor before or after the `h1` tags, you notice in the valid elements list that you could have a headline or byline at this point. You then realize that somehow the headline codes are missing.

If you simply insert the headline tags, the `h1` tags and accompanying text are still not where they should be, nested within the headline tags, and thus the code is still invalid. If you select the `h1` tags and text and move them to within the headline tags, you've fixed the problem. Better yet, before inserting the headline tags, select the `h1` tags and text, and *then* insert the headline tags that correctly wrap around the selected elements.

FORMATTING AN XML DOCUMENT

The most important thing to remember about an XML document is that *data is data!* That is, the content of your document is of primary concern. If the structure and elements are well formed and valid, you or anyone can use that data and display or present it in a variety of formats, or layouts.

UNDERSTANDING XML LAYOUTS

In a typical XML process, the document itself and the DTD don't concern themselves at all with how the document is to be formatted. All they care about is making the document content usable.

For example, you can create an element such as `<model></model>` and valid content might include the various automobile models in your inventory. Such an element is concerned only

with the content, and thus it is very different from typical HTML codes (or even WordPerfect formatting codes) that define *how* something is to be displayed, such as bold, italic, large font, and so on.

Formatting of XML data can be accomplished in a variety of ways, using whatever tools are available. For example, you can create Cascading Style Sheets (CSS) or use other separate programs to describe how various XML entities should be displayed.

Fortunately, WordPerfect includes its own XML Project Designer that enables you to define how various elements display. These definitions are collected in *layouts* and are stored in the same XML template that contains the DTD.

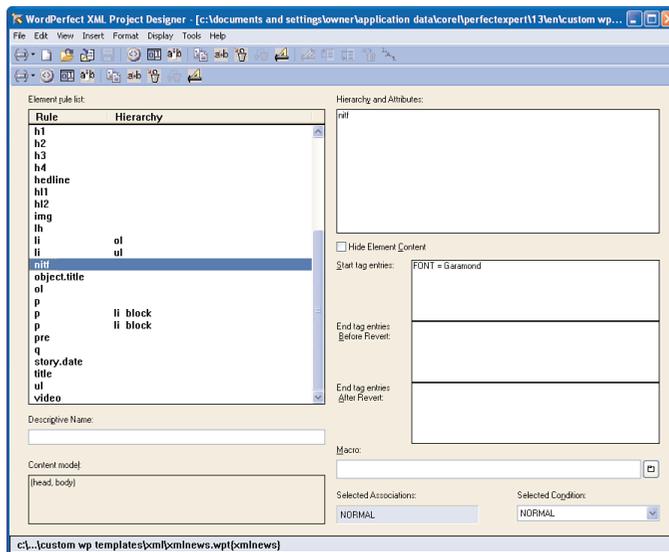
CREATING AN XML LAYOUT

Some templates that ship with WordPerfect contain basic layout information. For example, the news article template displays normal text in a certain font (Garamond), while displaying the headline in a different, larger font (Arial Black, 32 pt).

At some point, however, you or someone else must determine exactly how you want to display the content of an XML document. You do this by using WordPerfect's XML Project Designer, where you can create entirely new layouts, or modify existing ones. Further, you can create up to 10 different layouts and store them in the same XML template.

Suppose, for example, you have created the content for your XML newsletter (refer to Figure W3.4). Click the Edit Layout button on the toolbar. WordPerfect displays the XML Project Designer (see Figure W3.9).

Figure W3.9
The WordPerfect XML Project Designer is a separate program that helps you modify rules for displaying XML elements.



W3

NOTE

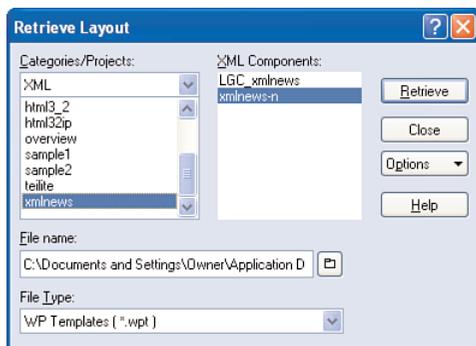
Although the XML Project Designer appears to be just another dialog box, it's actually a separate program. The XML Editor's Edit Layout button makes it easier to access the program and to automatically edit the current layout.

Alternatively, you can open the XML Project Designer by going to the Windows Start menu and choosing All Programs, WordPerfect Office X3, Utilities, WordPerfect XML Project Designer. Then within the program, you choose File, Open and select the XML Project (template) and Component (layout) you want to edit.

You can edit the default xmlnews layout, but you're better off creating a new layout so that you don't lose the original. To create a new layout, based on the original layout, follow these steps:

1. Choose File, New, Layout. WordPerfect clears the XML Project Designer lists.
2. Choose File, Retrieve. WordPerfect displays the Retrieve Layout dialog box (see Figure W3.10).

Figure W3.10
You retrieve an XML layout (component) from an existing XML template (project).

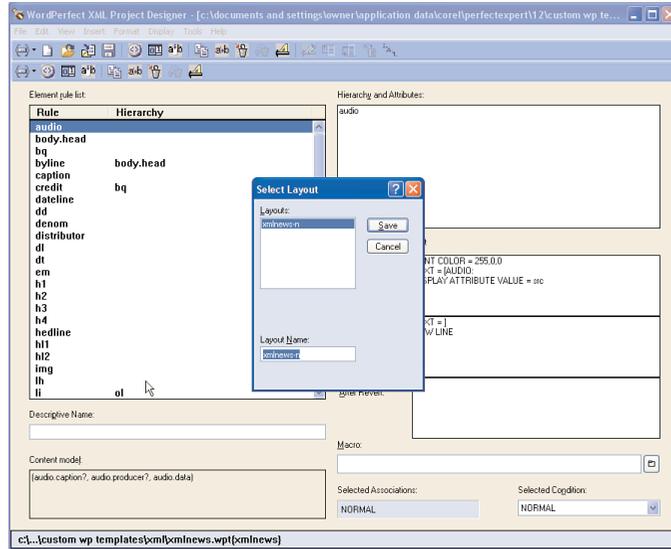


3. Choose the layout you want to insert by choosing a project (template)—for example, choose the XML category, then xmlnews project from the Categories/Projects list, and then choose the layout from the Component list—for example, xmlnews-n.
4. Click retrieve and WordPerfect inserts the selected collection of elements into your new, about to be created, layout.
5. Before you proceed, Choose File, Save to save your new layout. WordPerfect displays the Select Layout dialog box (see Figure W3.11). You can overwrite an existing layout (be careful!) or supply a new name in the Layout Name text box. Click Save.

NOTE

When you save a layout, you save it in an XML template. Thus, it becomes available not only to the current XML document, but also for all existing and new XML documents based on or created using that XML template.

Figure W3.11
Use a unique name to identify the type of layout you're creating for your XML template.



You now can begin to modify the rules for any of the document's elements, which include overall attributes, and what attributes or text to insert before or after the element appears. These rules are what determine how the document's content displays. For example, suppose you don't like the default Garamond font used in the original layout. To change an element, follow these steps:

1. Select the element you want to change in the Element rule list. For example, the default font is set in the nitf element, which is the root, or first element in the document.
2. Choose Edit, Edit Rule, or simply double-click the rule in the Element rule list, directly under the Rule column, not the Hierarchy column. WordPerfect displays the Element Rule dialog box (see Figure W3.12).

Figure W3.12
Elements define content. You create your own rules for displaying that content.

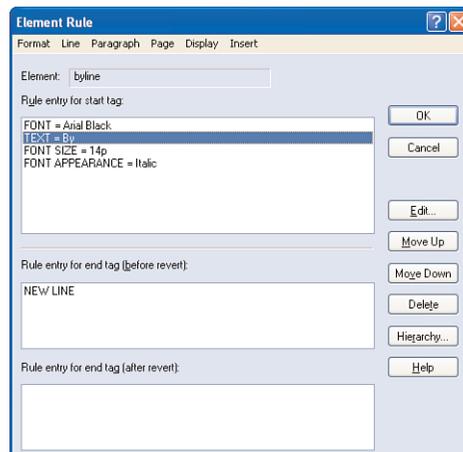


3. Note that the Start Tag rule specifies the Garamond font. Select that rule and click Delete to remove it. If you only want to change it, you could double-click the rule entry to go directly to the Font dialog box.
4. Click Format, Font, and from the Font dialog box, choose the font, size, and other attributes you want to serve as your default document font. Click OK.
5. Click OK again to return to the XML Project Designer.

You can now select other items you want to change. To see some of the possibilities, try changing two of the headline elements by following these steps:

1. Click the headline element. Notice that it doesn't include font information, although it does indicate that two lines will separate the headline from the next element in the document.
2. Click the h1 element. This is the element that actually contains the headline content. Double-click it to open the Element Rule dialog box.
3. Double-click the Font rule, and change it from Arial Black to something more interesting. Click OK.
4. Choose Paragraph, Justification, Center to center the title, and then click OK.
5. In the Element rule list, double-click the byline element. WordPerfect displays the Element Rule dialog box.
6. Choose Display, Text. Because the content of the byline element contains only a person's name (for example, yours), you need to add text that precedes that content. In the Text dialog box, type **By** followed by a space, and click OK. WordPerfect inserts the new rule following the Font rule (see Figure W3.13).

Figure W3.13
Not only can you add rules, but you can also change the order in which they're applied to an element.



7. Unfortunately, in its current position, “By” will be in the right font but won't be the right size or in italics. Change the order of the text rule by clicking Move Down until it follows the other rules, and then click OK.

You have a great deal of flexibility in defining the rules by which XML elements are to be displayed. You can even choose not to display elements by checking the Hide Element Content box. For example, select the Title element, and check the Hide Element Content box so that the title element doesn't display in the document.

When you finish editing the layout, choose File, Save. If you haven't named your new layout, WordPerfect displays the Select Layout dialog box (refer to Figure W3.11). You can overwrite an existing layout or supply a new name in the Layout Name text box. Click Save. Having saved the layout in its associated WordPerfect XML template, you then close the Project Designer to return to WordPerfect's XML Editor.

You now can apply your new layout, or any other defined layout, by clicking the Layouts button on the toolbar and selecting the layout you want. If you select None, you see what your document content looks like without any formatting.

In the future, if you create a new XML document based on this modified XML template, you can still use the same saved layouts. This is a great tool for standardizing the look of your documents, which can be a real timesaver in a corporate environment. Most individuals, however, will find this process to be more work than is necessary for their personal documents.

PUBLISHING AN XML DOCUMENT

Sharing your XML documents with other people can be as simple as providing them with your basic XML data, or providing them also with DTD and layout information. You can also print the document.

NOTE

WordPerfect offers a Publish to XML option (found under File, Publish To, XML) that is different from the process described in this section.

SAVING AN XML DOCUMENT

By default, when you save a new XML document, WordPerfect assumes you want to save it in XML format. Choose File, Save (or Save As), and WordPerfect displays the Save File dialog box. Note that the file type is XML UTF-8. Supply a filename and click Save. WordPerfect saves the file with an .XML filename extension.

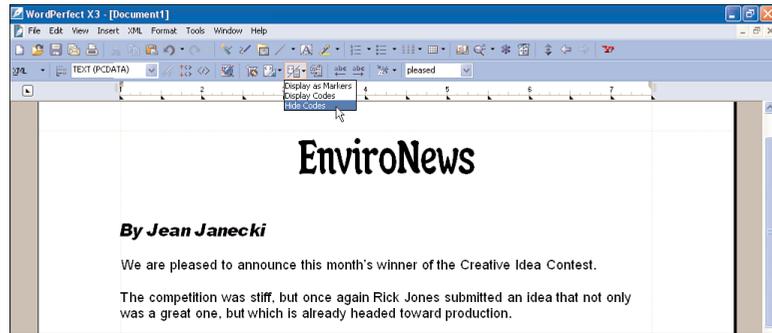
PRINTING AN XML DOCUMENT

You print an XML document just as you do any WordPerfect document. However, consider the following:

- Apply a layout that works well for a printed document. For example, no layout, or one designed for a Web page, might not look good on the printed page.

- What you see in the XML editing screen is what you print. Unless you want them there, first turn off the Display codes or markers. Click the Change XML Codes button on the toolbar, and select Hide Codes (see Figure W3.14). You can turn them back on again if you want them for further editing.

Figure W3.14
If you hide the XML codes, you get a better idea as to how your layout rules affect your XML document.



DISTRIBUTING AN XML DOCUMENT

If you decide to share your XML document with someone else electronically, you also need to provide the XML template from which the document derived. Because the template contains the DTD and layout information, the XML document is nothing more than a collection of data and undefined (invalid) tags without the template.

In a corporate environment, typically, such templates are accessible by everyone in a central network drive and folder. When you open an XML document from someone else, your networked version of WordPerfect automatically checks in the shared template folder and matches the DTD and layout information for that document.

If you don't all share the same template folder, you need to provide *both* the XML document and its template. The recipient then must save the template in his or her default template folder (for example, C:\Documents and Settings\Owner\Application Data\Corel\PerfectExpert\13\EN\Custom WP Templates\XML). Then, when the recipient opens the XML document in WordPerfect, WordPerfect finds the associated template and knows just what to do with the XML document.

UNDERSTANDING ADVANCED XML TASKS

As noted earlier, if you're an individual working alone, using XML is going to be of little importance to you. On the other hand, if you're working in a shared environment where standards are important, XML can be of great value.

Nevertheless, most individuals don't want or need to know about the inner workings of XML. Creating DTD files, compiling DTDs, or creating XML templates and corporate XML layouts likely will be the responsibility of your company's Information Technology

personnel. Such individuals are better equipped to deal with the somewhat arcane intricacies of the XML process.

Nevertheless, the following can give you a sense of what these advanced tasks are all about, in case you want to know, or if you need to discuss these matters with an expert.

Help for Advanced Uses of XML

WordPerfect provides a couple of fairly useful aids to learning about its approach to creating XML documents. One is a tutorial that can be found in the WordPerfect XML Project Designer. In Windows, go to All Programs, WordPerfect Office X3, Utilities, XML Project Designer, and then within that program, choose Help, XML Tutorial. The tutorial doesn't begin with document creation, but rather with creating and compiling DTDs. Thus, it might be useful to survey this entire chapter first to get an overall sense of the process before tackling the tutorial.

The other aid is a comprehensive, 138-page XML Manual that WordPerfect published in PDF format and shipped with WordPerfect Office 2000 and WordPerfect 9. WordPerfect revamped its XML procedures with WordPerfect 9, but little has changed since then except minor modifications and bug fixes, and thus the manual is still applicable to XML and WordPerfect X3. If you can find a copy of WordPerfect Office 2000, you'll find the xml9en.pdf file on the distribution CD. You might also find the file by searching for it on the Internet.

In addition, you can find other helpful resources on the Web. For example, I searched for WordPerfect XML and found an article that talked about using XML for working in an academic law instruction environment (<http://www.cali.org/conference/2001/postconf/2001C23d1>).

Many articles and books are written about XML, but you also need to find those that show how it is used in WordPerfect.

UNDERSTANDING DTD FILES

A Document Type Definition (DTD) file provides definitions for tags you use in an XML document. XML is much more flexible than HTML because you can use literally any tag to define XML content, if you define it so others know what to do with it.

DTD files are nothing more than ASCII files. If you open the xmlnews.dtd file in Notepad, you find definitions for every element in the News Industry Text Format (nif) document type. The following snippets are from that DTD:

```
<!-- Document header -->
<!ELEMENT head (title, base?)>
```

Note that the "head" tag refers to the document heading—a nonprinting part of the document—and includes the required title and optional base tags (indicated by the question mark).

```
<!-- Story headline group -->
<!ELEMENT headline (h11, h12*)>
```

In the headline element of the document, authors can define one main headline (h11) and one or more (*) subheadlines (h12).

```
<!-- Story headline -->
<!ELEMENT h11 (#PCDATA | chron | copyrite | event | function | location | money
| num | object.title | org | person | virtloc | a | br | em | lang |
pronounce | q)*>
```

Finally the headline itself (h1) can include optionally (indicated by |) one or more (indicated by *) elements, and include text (#PCDATA) and other identifiable pieces of information (such as org, person, pronounce).

The language required to create a valid DTD is not terribly difficult. However, because they try to establish standards for certain types of documents, and because such standards usually come from committees, the actual creation of new DTDs usually is left up to the experts.

Several industry standard DTDs have emerged, including those from the government, the technical publishing industry, and the news industry. One of the most common DTDs is the docbook standard, which is currently in revision 4.2. This is the standard DTD by which WordPerfect documents are published to XML, unless otherwise designated. Industry standard DTDs also tend to develop slowly over time, with lots of feedback from users and discussion by committee members before new DTD versions are released. Therefore, some DTDs are more than a couple of years old, but on the other hand, that does provide for a certain degree of stability in an otherwise rapidly changing computing world.

COMPILING DTD FILES

To use a DTD to create a new WordPerfect XML document, you must first compile the DTD and make it part of a WordPerfect template. You do so using WordPerfect's XML Project Designer, and the result is incorporated in a WordPerfect XML template.

The DTD itself must be properly created. WordPerfect provides virtually no help in troubleshooting compiling problems, other than to indicate there are problems. If the compiler encounters errors, it will not create a template. Thus, you have to resolve all errors before you can create a new template based on a DTD.

In addition, if any additional information is needed, such as reference to external files that describe how to deal with graphic images, these must be properly noted in a catalog file.

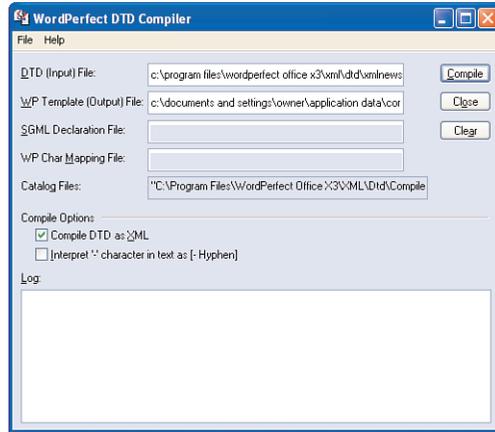
NOTE

The requirements for creating a valid DTD are quite stringent. The possible combinations and problems that can arise are far beyond the scope of this book. This section is included if you are provided with a DTD and need to compile it before using it.

To compile a DTD file and generate an XML template, follow these steps:

1. Open the WordPerfect XML Project Designer by going to the Windows Start menu and choosing All Programs, WordPerfect Office X3, Utilities, WordPerfect XML Project Designer. WordPerfect opens the XML Project Designer.
2. Choose File, DTD Compile. WordPerfect displays the WordPerfect DTD Compiler dialog box (see Figure W3.15).

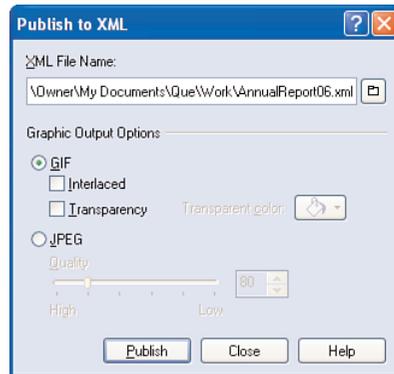
Figure W3.15
When you compile a valid DTD file, it then becomes part of an XML template.



Nevertheless, if you need to convert existing documents to XML so that someone else can work with the resulting raw XML content, these are the steps:

1. Open an existing .wpd file into WordPerfect.
2. Choose File, Publish To, XML. WordPerfect displays the Publish to XML dialog box (see Figure W3.16).

Figure W3.16
The Publish to XML feature enables you to quickly convert existing WordPerfect documents to raw XML, without any layout information.



3. By default, WordPerfect converts the document using the same filename, but with an .XML extension. Change the filename if you want.
4. If the document contains graphic images, you can specify how these are to be converted.
5. Click Publish to convert the document to XML content.

TROUBLESHOOTING

CRASH

While working with the XML Editor, it sometimes crashes for no apparent reason.

Unfortunately, that is the nature of WordPerfect's XML Editor beast. You'll often find that if something you're working with crashes, it will do so over and over again. You can try two things.

First, make sure you save your projects early and often. Press Ctrl+S regularly to save your work, so you won't lose it should the XML Editor crash.

Second, try exiting WordPerfect and Windows altogether and starting them again. Help the editor forget its crashes and perhaps it will work better for you.

CAN'T VALIDATE

When I try to save my XML project, I sometimes come up with validation errors. Should I save anyway?

You can certainly save your documents with its errors. However, before you share the document with others, you should be sure to resolve any validation errors.

Remember that errors can be related to structure or syntax, and thus these are not well formed. Errors can also involve undefined or misused elements or tags, in which case the code is not valid. Although it can be difficult to figure out why the document does not validate, take the time to make sure your document is valid.