

LESSON 4.3

10754 Microsoft .NET Fundamentals

Understand XML Classes in the .NET Framework

Lesson Overview

Understand XML classes in the .NET Framework.

In this lesson, you will:

- See samples of Extensible Markup Language (XML) code.
- Examine the classes and methods for working with XML in the Microsoft® .NET Framework.
- Learn the purpose of XML Schemas.

Guiding Questions

1. What is XML?
2. What classes can you use to read and write XML?
3. What is an XML Schema Definition (XSD) file?

Anticipatory Set

- Describe the output produced by the following Hypertext Markup Language (HTML) code:

```
<p>The quick brown <strong>fox</strong> <br/>  
jumps over the lazy dog<p>
```

- What are some other simple HTML tags you could add to the text?

XML

- XML is a markup language for formatting data in a document.
 - Like HTML for web pages, XML uses tags identified by angle brackets:

`<` `>` `</` `>`
- Unlike HTML, XML does not specify how data should look; instead, it identifies how the data is structured.
 - An HTML document may specify that “Microsoft Word” will be displayed in a large, bold font; an XML document might simply identify “Microsoft Word,” with no information about how to display it.
- XML provides a standardized format for storing data or transferring it between different platforms or applications.
 - XML is an “open standard”—that is, it is not specific to any operating system or software application. This enables easier transfer of data between software developed by different vendors.

XML Files in Microsoft Visual Studio

- Visual Studio® provides an editor that assists with XML files.
- To create an XML file that is not part of a project, perform the following steps:
 1. From the File menu, go to New and select File.
 2. Choose XML File from the Add Existing Item dialog box.
- To add a new XML file to a project, perform the following steps:
 1. From the Project menu, select Add New Item.
 2. Choose XML File from the Add New Item dialog box.

Create a Windows Forms application called XMLPractice; then add a new XML file named Computers.xml, with the code shown on the next slide.

XML Example

```
<?xml version="1.0" encoding="utf-8"?>
<ComputerSales>
  <Computer name="Desktop">
    <ProcessorSpeed>2.0 GHz</ProcessorSpeed>
    <RAM>4 Gb</RAM>
    <HDD>500 Mb</HDD>
  </Computer>
  <Computer name="Laptop">
    <ProcessorSpeed>1.8 GHz</ProcessorSpeed>
    <RAM>2 Gb</RAM>
    <HDD>250 Mb</HDD>
  </Computer>
</ComputerSales>
```

XmlReader and XmlWriter Classes

- The .NET Framework includes these two classes (in the `System.Xml` namespace) for reading and writing XML.
 - Both classes are abstract, so you cannot create objects from them.
- The `System.Xml` namespace also includes `XmlTextReader` and `XmlTextWriter`, which can be instantiated and used to read and write XML data.

Reading XML

- To read XML, first create an instance of the `XmlTextReader` class. It has constructors that will accept a `Stream`, a file name, or a Uniform Resource Locator (URL).
- The `XmlTextReader.Read` method reads the next node of the XML. It returns a `boolean` indicating the success of the operation.
- Use properties such as `Name` and `Value` to access information about the current node.
- The `Read` method can be used in a `while` loop to iterate through all XML data. For example, given an instance named `reader`:

```
while (reader.Read())  
{  
    // code for processing the nodes here  
}
```

Writing XML

- Instantiating an `XmlTextWriter` object exposes methods that allow the developer to write well-formed XML.
- Much of the process for creating the XML involves using method pairs that write corresponding tag pairs:
 - `WriteStartDocument` and `WriteEndDocument` opens and closes an XML document.
 - `WriteStartElement` and `WriteEndElement` delimit an element, with the first method accepting strings to indicate the name and value of the element.
 - `WriteStartAttribute` and `WriteEndAttribute` start and end attributes for the current element.
- After writing all the elements, use the `Close` method; this verifies that all elements and attributes have been ended and closes the `Stream`.

XML Schema

- XML Schema is a technology that allows developers to specify the structure of an XML document, including data type information.
- XML Schema can be defined in an XSD file.
 - XML Schemas are actually defined using XML, so you don't have to learn a new format or language just for that purpose.
- Developers can use `XmlValidatingReader` (which implements the `XmlReader` class) to validate XML against an XSD file.

Assignment

Create the application described in the Student Activity document.

Use the Visual Studio project “XMLPractice,” created with the instructions beginning on slide 6 of this presentation, to complete the activity.