XML Version 2

In this version of the xml reader and writer we will be using the XDocument Class from the System.Xml.Linq Namespace in addition to System.Xml.

using System.Xml.Linq;

using System.Xml;

The xml File “test.xml” will contain the following

<?xml version="1.0" encoding="utf-8"?>

<Bookshelf>

 <Book>

 <Name>C# Programming Basics</Name>

 <Author>Lony Hem</Author>

 </Book>

 <Book>

 <Name>Peter White the Bear</Name>

 <Author>Gary Fischer</Author>

 </Book>

</Bookshelf>

Let us build a class to store book information (Name and Author) from an xml file.

We will also include a constructor to help make our class easier to use

 public class Book

 {

 public string Name;

 public string Author;

 //Constructor for book class takes two parameters name and author

 public Book(string name, string author)

 {

 this.Name = name;

 this.Author = author;

 }

 }

**Xml Reader**

//loads xml file into the XDocument object

XDocument doc = XDocument.Load("test.xml");

var books = doc.Descendants("book");

List<Book> bookList = new List<Book>();

//copies the contains of the XDocument into the bookList

foreach (var book in books)

 {

 var bookName = book.Descendants("name").First<XElement>().Value;

 var bookAuthor = book.Descendants("author").First<XElement>().Value;

 bookList.Add(new Book(bookName, bookAuthor));

 }

<object>.Descendants(elementName) is a method that generates a collection of XElements.

foreach is used to iterate through each item in the collection.

We intend to have only unique sub-childs “name” and “author” in each child “book”.

Since book.Descendants("name") will get a collection with only 1 unique element. We need to take the first (and only) XElement value hence

book.Descendants("name").First<XElement>().Value;

**Xml Writer**

var bookElements = new List<XElement>();

foreach (var book in bookList)

 {

 //build a list of sub-child elements "Name" and "Author"

 var elements = new List<XElement>();

 elements.Add(new XElement("Name", book.Name));

 elements.Add(new XElement("Author", book.Author));

 //add list of sub-child to child "Book"

 bookElements.Add(new XElement("Book", elements));

 }

//create new xdoc and add list of child "Book" to root "Bookshelf"

XDocument newdoc = new XDocument();

newdoc.Add(new XElement("Bookshelf", bookElements));

//write new xdoc to new xml file with xml indentation

XmlWriterSettings xws = new XmlWriterSettings();

xws.Indent = true;

xws.IndentChars = "\t";

var xr = XmlWriter.Create(new StreamWriter("new.xml"), xws);

newdoc.WriteTo(xr);

xr.Close();