

XMLDocument en el aula

Nieves Carralero Colmenar.

IES Pedro Mercedes. Junta de Comunidades de Castilla-La Mancha. España.

ncarralero@edu.jccm.es

Resumen

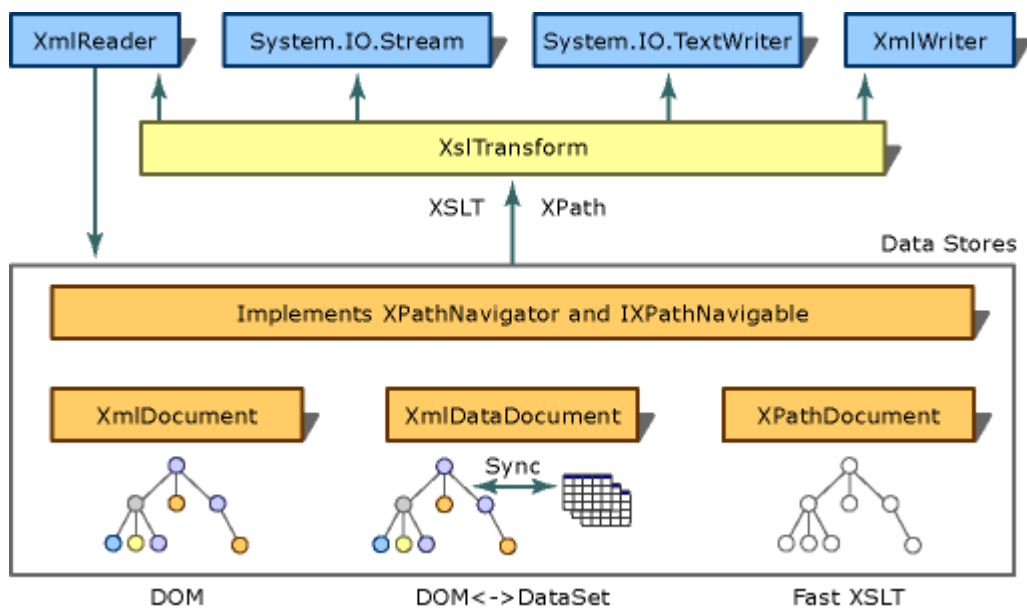
XMLDocument es la manera más sencilla y potente de poder acceder a XML con .NET. En otros artículos de esta serie [XML1] [XML2] [XML3] [XML4] se han mostrado otras alternativas que se ajustan a diferentes tipos de tratamiento. Sin embargo, esta solución es la más versátil ya que permite recorrer y modificar documentos al mismo tiempo. Esta alternativa se ubica dentro del ciclo formativo de Grado Superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web, se define un módulo con título: "Módulo Profesional: Lenguajes de marcas y sistemas de gestión de información".

Introducción

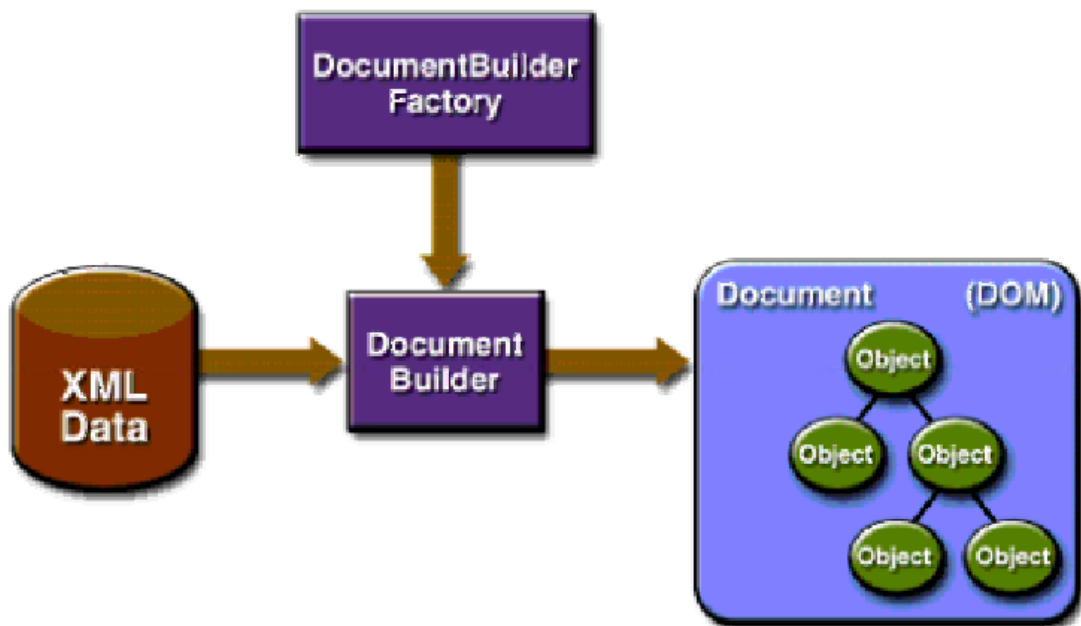
En publicaciones anteriores [XML1] [XML2] [XML3] [XML4] se han mostrado diferentes soluciones, más o menos flexibles para poder acceder a documentos XML.

- [XML1] muestra un ejemplo de lectura de documento XML usando el objeto XMLReader. La salida será los elementos y sus etiquetas mostrados en formato secuencial.
- [XML2] muestra un ejemplo de lectura de documento XML usando el objeto XMLTextReader. La salida será los elementos y sus etiquetas mostrados en formato secuencial.
- [XML3] muestra un ejemplo para el trabajo con XML usando el objeto XmlNode para poder leer y escribir documentos.
- [XML4] muestra un ejemplo para escribir documentos XML usando el objeto XMLTextWriter para poder leer y escribir documentos.

La siguiente imagen muestra un resumen de las alternativas mostradas y su integración en .NET



En esta práctica, al usar XmlDocument se leerá código XML y se escribirá en un archivo, apoyándose en el modelo DOM, que esquematiza la siguiente figura.



XmlDocument implementa el Core Document Object Model (DOM) Level 1 y el Core DOM Level 2 del consorcio W3C. DOM es la representación en forma de árbol en memoria (caché) de un documento XML y permite la exploración y edición del documento. Dado que XmlDocument implementa la interfaz XPathNavigator, también se puede utilizar como documento de origen de la clase XslTransform.

La clase XmlDocument extiende XmlDocument y permite que los datos estructurados se almacenen, recuperen y manipulen mediante un DataSet relacional. Esta clase permite que los componentes combinen vistas XML y relacionales de los datos subyacentes.

1. Escribiendo con XmlDocument.

En la siguiente práctica se muestra un ejemplo de escritura de documento XML usando el objeto XmlDocument. A continuación se muestra una imagen de un documento XML que será el que se usará de ejemplo:

```
<?xml version="1.0" ?>
<!-- This file represents a fragment of a book store inventory database -->
- <bookstore>
- <book genre="autobiography" publicationdate="1981" ISBN="1-861003-11-0">
  <title>The Autobiography of Benjamin Franklin</title>
  - <author>
    <first-name>Benjamin</first-name>
    <last-name>Franklin</last-name>
  </author>
  <price>8.99</price>
</book>
- <book genre="novel" publicationdate="1967" ISBN="0-201-63361-2">
  <title>The Confidence Man</title>
  - <author>
    <first-name>Herman</first-name>
    <last-name>Melville</last-name>
  </author>
  <price>11.99</price>
</book>
- <book genre="philosophy" publicationdate="1991" ISBN="1-861001-57-6">
  <title>The Gorgias</title>
  - <author>
    <name>Plato</name>
  </author>
  <price>9.99</price>
</book>
</bookstore>
```

Práctica 1:

Práctica 1: Esta aplicación de ejemplo recorre el documento mostrado en la imagen anterior para aumentar el precio de todos los libros en un 2%. A continuación, la aplicación guarda los datos XML en un nuevo archivo denominado [updatebooks.xml](#). Este documento es el siguiente:

```

<?xml version="1.0" ?>
<!-- This file represents a fragment of a book store inventory database -->
- <bookstore>
- <book genre="autobiography" publicationdate="1981" ISBN="1-861003-11-0">
  <title>The Autobiography of Benjamin Franklin</title>
  - <author>
    <first-name>Benjamin</first-name>
    <last-name>Franklin</last-name>
  </author>
  <price>917</price>
</book>
- <book genre="novel" publicationdate="1967" ISBN="0-201-63361-2">
  <title>The Confidence Man</title>
  - <author>
    <first-name>Herman</first-name>
    <last-name>Melville</last-name>
  </author>
  <price>1.223</price>
</book>
- <book genre="philosophy" publicationdate="1991" ISBN="1-861001-57-6">
  <title>The Gorgias</title>
  - <author>
    <name>Plato</name>
  </author>
  <price>1.019</price>
</book>
</bookstore>

```

Solución:

No existe un camino directo que cree un XmlNode para representar una entidad de un documento XML. Es necesario recuperar los XmlNode desde objetos XmlDocument. Un objeto XmlDocument representa un documento XML completo.

Los métodos y propiedades más importantes de XmlDocument son mostrados en la siguiente tabla:

<i>Member</i>	<i>Type</i>	<i>Description</i>
CreateAttribute()	Method	Creates an attribute node
CreateElement()	Method	Creates an element node
CreateNode()	Method	Creates an XmlNode object
DocumentElement	Property	Returns the root XmlNode object for this document
DocumentType	Property	Returns the node containing the DTD declaration for this document, if it has one
ImportNode()	Method	Imports a node from another XML document
Load()	Method	Loads an XML document into the XmlDocument object

LoadXml()	Method	Loads the XmlDocument object from a string of XML data
NodeChanged	Event	Occurs after the value of a node has been changed
NodeChanging	Event	Occurs when the value of a node is about to be changed
NodeInserted	Event	Occurs when a new node has been inserted
NodeInserting	Event	Occurs when a new node is about to be inserted
NodeRemoved	Event	Occurs when a node has been removed
NodeRemoving	Event	Occurs when a node is about to be removed
PreserveWhitespace	Property	Returns true if whitespace in the document should be preserved when loading or saving the XML
Save()	Method	Saves the XmlDocument object as a file or stream
WriteTo()	Method	Saves the XmlDocument object to an XmlWriter object

Las funciones de la clase XmlNode son:

<i>Member</i>	<i>Type</i>	<i>Description</i>
AppendChild()	Method	Adds a new child node to the end of this node's list of children
Attributes	Property	Returns the attributes of the node as an <code>XmlAttributeCollection</code> object
ChildNodes	Property	Returns all child nodes of this node
CloneNode()	Method	Creates a duplicate of this node
FirstChild	Property	Returns the first child node of this node
HasChildNodes	Property	Returns <code>true</code> if this node has any children
InnerText	Property	Specifies the value of the node and all its children
InnerXml	Property	Specifies the markup representing only the children of this node
InsertAfter()	Method	Inserts a new node after this node
InsertBefore()	Method	Inserts a new node before this node
LastChild	Property	Returns the last child node of this node
Name	Property	Specifies the node's name
NextSibling	Property	Returns the next child of this node's parent node
NodeType	Property	Specifies this node's type
OuterXml	Property	Specifies the markup representing this node and its children
OwnerDocument	Property	Specifies the <code>XmlDocument</code> object that contains this node
ParentNode	Property	Returns this node's parent

PrependChild()	Method	Adds a new child node to the beginning of this node's list of children
PreviousSibling	Property	Returns the previous child of this node's parent node
RemoveAll()	Method	Removes all children of this node
RemoveChild()	Method	Removes a specified child of this node
ReplaceChild()	Method	Replaces a child of this node with a new node
SelectNodes()	Method	Selects a group of nodes matching an XPath expression
SelectSingleNode()	Method	Selects the first node matching an XPath expression
WriteContentTo()	Method	Writes all children of this node to an XmlWriter object
WriteTo()	Method	Writes this node to an XmlWriter object

El código comentado es el siguiente:

```
public class Cpractica5
{
    private const String document = "..\\..\\books.xml";
    private const String updatedDocument = "..\\..\\updatebooks.xml";

    public static void Main()
    {
        String[] args = {document, updatedDocument};
        Cpractica5 myCpractica5 = new Cpractica5();
        myCpractica5.Run(args);
    }

    public void Run(String[] args)
    {
```

```

try
{
    // Crea un XmlDocument y lo carga desde fichero
    XmlDocument myXmlDocument = new XmlDocument();
    myXmlDocument.Load (new XmlTextReader (args[0]));
    Console.WriteLine ("Los datos XML se han cargado correctamente en
el objeto XmlDocument...");

    Console.WriteLine();

    IncreasePrice(myXmlDocument.DocumentElement);

    // Escribe el fichero de salida en XML
    myXmlDocument.Save(args[1]);
    Console.WriteLine();
    Console.WriteLine("Los precios actualizados se han guardado en el
archivo {0}", args[1]);
}
catch (Exception e)
{
    Console.WriteLine ("Excepcin: {0}", e.ToString());
}
}

```

```

public void IncreasePrice(XmlNode node)
{
    if (node.Name == "price")
    {
        node = node.FirstChild;
        Decimal price = Decimal.Parse(node.Value);//Convierte string en
Decimal
        // Incrementa el valor de los precios a un 2%
        String newprice = ((Decimal)price*(new
Decimal(1.02))).ToString("#,00");

```



```
Console.WriteLine("Precio anterior = " + node.Value + "\tPrecio actual
= " + newprice);

        node.Value = newprice;
    }

    node = node.FirstChild;
    while (node != null)
    {
        IncreasePrice(node);
        node = node.NextSibling;
    }
}

} // End class
```

2. Aspectos a tener en cuenta en el aula.

En resumen, la clase XmlDocument proporciona una forma rápida y de desplazamiento sólo hacia delante para generar datos XML.

1. La clase XmlDocument permite guardar datos XML en un archivo, una secuencia o un objeto XmlWriter.
2. La clase XmlNode permite recorrer y corregir valores de los nodos en un documento XML

Para cada tipo de nodo XML hay métodos de escritura de datos XML.

4. Bibliografía

[XML1] Leer XML con XmlReader en el aula. Nº 26. MARZO 2011.
www.sociedadelainformacion.com

[XML2] XmlTextReader en el aula. Nº 26. MARZO 2011.
www.sociedadelainformacion.com

[XML3] XMLNode y XMLDocument en el aula. Nº 27. ABRIL 2011.
www.sociedadelainformacion.com

[XML4] Escribiendo con XMLTextWriter en el aula. Nº 27. ABRIL 2011.
www.sociedadelainformacion.com

SOCIEDAD DE LA INFORMACION

www.sociedadelainformacion.com

Edita:



Director: José Ángel Ruiz Felipe

Jefe de publicaciones: Antero Soria Luján

D.L.: AB 293-2001

ISSN: 1578-326x