Escribiendo con XMLTextWriter en el aula

Nieves Carralero Colmenar.
IES Pedro Mercedes. Junta de Comunidades de Castilla-La Mancha. España.
ncarralero@edu.jccm.es

Resumen

Microsoft .NET usa el objeto XMLTextWriter para poder leer y escribir documentos. Esta alteranativa 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". En él se pretende hacer una aproximación al trabajo con XML como pieza clave en el desarrollo de aplicaciones Web.

1. XMLTextWriter

Escribir datos XML en un archivo

Vamos a leer código XML y escribirlo en un archivo mediante la clase XmlTextWriter. El sistema de escritura proporciona una forma rápida y de desplazamiento sólo hacia delante al generar datos XML, a la vez que ayuda a generar documentos XML que se ajusten a las especificaciones W3C Extensible Markup Language (XML) 1.0 y Namespaces in XML. El objeto XmlTextWriter escribe en una secuencia en lugar de utilizar un modelo de objetos como XML DOM y, por tanto, proporciona mejor rendimiento.

Normalmente, se utiliza un objeto XmlTextReader cuando es necesario escribir datos XML en forma de datos sin formato, sin sufrir la sobrecarga de un modelo DOM. El objeto XmlTextWriter es una implementación de la clase XmlWriter que proporciona la API y que escribe datos XML en un archivo, una secuencia o un objeto TextWriter. Esta clase proporciona muchas reglas de validación y comprobación para garantizar que la sintaxis de los datos XML escritos es correcta. Cuando se producen determinadas

infracciones, se provocan excepciones que deberían detectarse. El objeto XmlTextWriter tiene distintos constructores, cada uno de los cuales especifica un tipo de ubicación diferente en la que se van a escribir los datos XML.

En nuestro ejemplo se utiliza el constructor que escribe datos XML en un archivo. En concreto, el fragmento de código que se verá en el siguiente punto crea un objeto XmlTextWriter con una cadena que representa la ubicación del archivo newbooks.xml.

2. Escribiendo con XMLTextWriter.

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

```
<?xml version="1.0" encoding="UTF-8" ?>
<Books>
    <Book Pages="1088">
        <Author>Delaney, Kalen</Author>
        <Title>Inside Microsoft SQL Server 2000</Title>
        <Publisher>Microsoft Press</Publisher>
    </Book>-
    <Book Pages="997">
        <Author>Burton, Kevin</Author>
        <Title>.NET Common Language Runtime</Title>
        <Publisher>Sams</Publisher>
    </Book>
    <Book Pages="392">
        <Author>Cooper, James W.</Author>
        <Title>C# Design Patterns</Title>
        <Publisher>Addison Wesley</Publisher>
    </Book>
</Books>
```

Práctica 1:

Práctica 1: Realizar la escritura de un documento XML usando XmlTextWrite.

La salida de la práctica 1 un documento XML con una estructura parecida a del XML de arriba..

Solución:

Las funciones de la clase XMLTextWriter son:

Close	Cierra esta secuencia y la secuencia subyacente.
Flush	Vuelca el contenido del búfer en las secuencias subyacentes y también vuelca la secuencia subyacente.
WriteAttributes	Cuando se reemplaza en una clase derivada, escribe todos los atributos que se encuentran en la posición actual en XmlReader.
WriteAttributeString	Cuando se reemplaza en una clase derivada, escribe un atributo con el valor especificado.
WriteComment	Escribe un comentario <!--...--> que contiene el texto especificado.
WriteDocType	Escribe la declaración DOCTYPE con el nombre y los atributos opcionales especificados.
WriteElementString	Cuando se reemplaza en una clase derivada, escribe un elemento que contiene un valor de cadena.
WriteEndAttribute	Cierra la anterior llamada al método WriteStartAttribute.
WriteEndDocument	Cierra todos los elementos o atributos abiertos y coloca de nuevo el sistema de escritura en el estado de inicio.
WriteEndElement	Cierra un elemento y extrae el correspondiente ámbito de espacio de nombres.
WriteStartAttribute	Escribe el inicio de un atributo.
WriteStartDocument	Escribe la declaración XML con la versión "1.0".
.	Escribe la etiqueta de apertura especificada.
WriteString	Escribe el contenido de texto especificado.
WriteWhitespace	Escribe el espacio en blanco especificado.

Al crear este elemento, el ejemplo anterior también muestra que, para cada tipo de nodo XML, hay un método de escritura de datos XML. Por ejemplo, cuando se escribe un elemento se llama al método WriteElementString y cuando se escribe un atributo se llama al método WriteAttributeString.

Para niveles anidados se utiliza el par WriteStartElement/WriteEndElement y para crear atributos más complejos se puede utilizar el par WriteStartAttribute/WriteEndAttribute.

Al escribir datos XML, hay que tener en cuenta que el código de ejemplo escribe la declaración XML con la versión "1.0" mediante el método WriteStartDocument. Si se desea que el sistema de escritura compruebe

que la sintaxis del documento es correcta (primero la declaración XML, DOCTYPE en el prólogo, sólo un elemento del nivel raíz, etc.), hay que llamar a este método WriteStartDocument opcional antes de llamar a cualquier otro método. A continuación, el código llama al método WriteDocType para escribir el tipo de documento con el nombre "bookstore". El tercer parámetro de la llamada a WriteDocType especifica que el sistema de escritura debe escribir SYSTEM "books.dtd". Al escribir esto, el archivo XML indica que hay una definición DTD externa con la que se puede validar.

Por último, el código de ejemplo llama al método Flush para almacenar los datos XML en un archivo antes de llamar al método Close. Aunque este ejemplo sólo necesita el método Close, hay situaciones en las que hay que almacenar los datos XML generados y reutilizar el sistema de escritura.

El código comentado es el siguiente:

```
public void Run(String args)
{
      XmlTextWriter myXmlTextWriter = null;
      XmlTextReader myXmlTextReader = null;

      try
      {
            myXmlTextWriter = new XmlTextWriter (args, null);

            myXmlTextWriter.Formatting = Formatting.Indented;
            //Hace que los elementos secundarios tengan la
sangría que indican los valores de las propiedades Indentation y
IndentChar.
            myXmlTextWriter.WriteStartDocument();
            myXmlTextWriter.WriteDocType("bookstore", null,
"books.dtd", null);
```

```
            myXmlTextWriter.WriteComment("Este archivo representa
otro fragmento de una base de datos de inventario de una
librera.");

            myXmlTextWriter.WriteStartElement("bookstore");

            myXmlTextWriter.WriteStartElement("book", null);

      myXmlTextWriter.WriteAttributeString("genre","autobiography
");


      myXmlTextWriter.WriteAttributeString("publicationdate","197
9");

            myXmlTextWriter.WriteAttributeString("ISBN","0-7356-
0562-9");

            myXmlTextWriter.WriteElementString("title", null,
"The Autobiography of Mark Twain");

            myXmlTextWriter.WriteStartElement("Author", null);

            myXmlTextWriter.WriteElementString("first-name",
"Mark");

            myXmlTextWriter.WriteElementString("last-name",
"Twain");

            myXmlTextWriter.WriteEndElement();

            myXmlTextWriter.WriteElementString("price", "7.99");

            myXmlTextWriter.WriteEndElement();

            myXmlTextWriter.WriteEndElement();


            //Escrbie el XML a fichero y lo cierra

            myXmlTextWriter.Flush();

            myXmlTextWriter.Close();



      }
```

En nuestro ejemplo se utiliza el constructor que escribe datos XML en un archivo. En concreto, el fragmento de código que se verá en el siguiente punto crea un objeto XmlTextWriter con una cadena que representa la ubicación del archivo newbooks.xml.

Además del nombre del archivo, este constructor también utiliza la codificación que se desea generar. Si la codificación es "null", el sistema de escritura elige UTF-8. Para obtener más información sobre cómo se utiliza codificación en documentos XML, vea la especificación W3C XML 1.0.

El siguiente fragmento de código crea un archivo XML con un único elemento libro. Este código empieza por utilizar la propiedad Formatting para especificar el formato de los datos XML que se están escribiendo. Al establecer el valor de esta propiedad en Indented, el sistema de escritura aplica una sangría a los elementos secundarios mediante las propiedades Indentation y IndentChar.

3. **Aspectos a tener en cuenta en el aula**.

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

El objeto XmlTextWriter ayuda a escribir documentos XML que se ajusten a las especificaciones W3C Extensible Markup Language (XML) 1.0 y Namespaces in XML.

El objeto XmlTextWriter proporciona constructores para leer datos XML de un archivo, una secuencia o un objeto TextWriter.

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

5. Bibliografía

MSDN. http://msdn.microsoft.com/es-es/library/bb972196.aspx
XML desde .NET: http://zarza.usal.es/~fgarcia/docencia/poo/04-05/Trabajos/XML.NET.pdf

.NET Framework: http://msdn.microsoft.com/en-us/magazine/cc302158.aspx

XML tutorial: http://www.programacion.net/html/xml/

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