

XML

XML (произнася се [екс-ем-ел], от англ. eXtensible Markup Language - разширяем маркиращ език) е стандарт (метаезик) дефиниращ правила за създаване на специализирани маркиращи езици както и синтаксисът на който тези езици трябва да се подчиняват. Сам по себе си той е безполезен, защото указва само как да бъде структуриран един документ (чрез маркиране с етикети), но не и какво означават отделните маркери (етикети). Ето защо е по-правилно да се нарече метаезик отколкото език - обикновените езици имат семантика, т.е. предават някаква информация, докато XML указва само граматиката (по-точно синтаксиса) на езиците, базирани на него.

Ползата от XML се състои в това, че синтаксисът (структурирането) на документите се измисля само веднъж, а специализираните маркиращи езици само дефинират семантиката т.е. набора от маркиращи етикети и тяхното значение. Другата изгода е, че XML прави възможно разработването на семантично агностични програмни средства които могат да обработват всички XML базирани езици, дори и такива които не са съществували по време на съставянето на софтуера. Тези програмни средства могат след това да се използват и в изработването на софтуер който е написан за конкретен XML език. Или накратко - разделянето на синтаксиса от семантиката е пореденото въплъщение на принципа "Разделяй и владей".

- Създайте нов проект: File -> New Project -> Windows Form Application.
- Добавете едно TextBox поле (textBox1), задайте му свойство Multiline = true и го разпънете в прозореца на вашето приложение.
- Библиотеката using System.Xml; е необходима за да може да се работи с XML.

Създаване на XML

Добавете бутон и в onClick метода му добавете следния код:

```
// Create the xml document containe
XmlDocument doc = new XmlDocument();
// Create the XML Declaration, and append it to XML
document
XmlDeclaration dec = doc.CreateXmlDeclaration("1.0", null,
null);
doc.AppendChild(dec); // Create the root element
XmlElement root = doc.CreateElement("Library");
doc.AppendChild(root);

// Create Books
// Note that to set the text inside the element,
// you use .InnerText instead of .Value
// (which will throw an exception).
// You use SetAttribute to set attribute
XmlElement book = doc.CreateElement("Book");
```

```

book.SetAttribute("BookType", "Hardcover");
XmlElement title = doc.CreateElement("Title");
title.InnerText = "Door Number Three";
XmlElement author = doc.CreateElement("Author");
author.InnerText = "O'Leary, Patrick";
book.AppendChild(title);
book.AppendChild(author);
root.AppendChild(book);

book = doc.CreateElement("Book");
book.SetAttribute("BookType", "Paperback");
title = doc.CreateElement("Title");
title.InnerText = "Lord of Light";
author = doc.CreateElement("Author");
author.InnerText = "Zelanzy, Roger";
book.AppendChild(title);
book.AppendChild(author);
root.AppendChild(book);

string xmlOutput = doc.OuterXml;

textBox1.Text = xmlOutput;

```

Това е създаване на XML чрез обекта XmlDocument. Наблюдавайте резултатът с CTRL+F5.

Друг начин за създаване на XML е чрез XML Writer. Добавете нов Бутон и в неговия onClick метод добавете следния код:

```

XmlWriterSettings wSettings = new XmlWriterSettings();
wSettings.Indent = true;
MemoryStream ms = new MemoryStream();
XmlWriter xw = XmlWriter.Create(ms, wSettings);
    // Write Declaration
xw.WriteStartDocument();

// Write the root node
xw.WriteStartElement("Library");

// Write the books and the book elements
xw.WriteStartElement("Book");
xw.WriteStartAttribute("BookType");
xw.WriteString("Hardback");
xw.WriteEndAttribute();

xw.WriteStartElement("Title");
xw.WriteString("Door Number Three");

```

```
xw.WriteEndElement();
xw.WriteStartElement("Author");
xw.WriteString("O'Leary, Patrick");
xw.WriteEndElement();

xw.WriteEndElement();

// Write another book
xw.WriteStartElement("Book");
xw.WriteStartAttribute("BookType");
xw.WriteString("Paperback");
xw.WriteEndAttribute();

xw.WriteStartElement("Title");
xw.WriteString("Lord of Light");
xw.WriteEndElement();
xw.WriteStartElement("Author");
xw.WriteString("Zelanzy, Roger");
xw.WriteEndElement();

xw.WriteEndElement();

// Close the document
xw.WriteEndDocument();

// Flush the write
xw.Flush();

Byte[] buffer = new Byte[ms.Length];
buffer = ms.ToArray();
string xmlOutput =
System.Text.Encoding.UTF8.GetString(buffer);

textBox1.Text = xmlOutput;
```

Добавете библиотеката: `using System.IO;`

Стартирайте приложението и сравнете получените XML текстове от различните методи.

За .NET 3.5 съществува още един начин за създаване на XML. Това е `XDocument`. За него е необходима библиотеката `using System.Xml.Linq;`

За да видите разликите с `XmlDocument` добавете още един бутон и следния код за него:

```
//XDocument
MessageBox.Show(new XElement("Foo",
new XAttribute("Bar", "some value"),
```

```
new XElement("Nested1", "data"),
new XElement("Nested2", "other data")
).ToString());

//XmlDocument
XmlDocument doc = new XmlDocument();
XmlElement el =
(XmlElement)doc.AppendChild(doc.CreateElement("Foo"));
el.SetAttribute("Bar", "some value");
el.AppendChild(doc.CreateElement("Nested1")).InnerText =
"data";
el.AppendChild(doc.CreateElement("Nested2")).InnerText =
"other data";
MessageBox.Show(doc.OuterXml.ToString());
```

XDocument е лесен за употреба, но когато става дума за много големи обеми от информация се препоръчва ползването на XML Writer. Той е по-сложен за писане, но по-ефективен.

Четене на XML:

Изтеглете [следния XML файл](#) и го копирайте в Bin/Debug директорията на вашия проект.

Четенето на един XML файл може да става отделно по определен таг.

Добавете бутон „Read XML“ и в неговия метод добавете следния код:

```
XmlDocument xDoc = new XmlDocument();
xDoc.Load("cd_catalog.xml");
XmlNodeList name = xDoc.GetElementsByTagName("TITLE");
XmlNodeList age = xDoc.GetElementsByTagName("ARTIST");

//XmlNodeList represents an ordered collection of nodes.
//Display the results in a message box.
string xmlText = "";
for (int i = 0; i < name.Count; i++)
{
    xmlText += "Title: " + name[i].InnerText + "\t Artist:
" + age[i].InnerText + "\n";
}
MessageBox.Show(xmlText);
```

Забележете, че от посочения файл са извадени само данните за име албум и изпълнител.

За да визуализираме всички данни като таблица ни е необходим DataGridView. Добавете още един бутон „Read All“. За неговия onClick метод добавете кода:

```
XmlDataDocument xmlDatadoc = new
XmlDataDocument();
xmlDatadoc.DataSet.ReadXml("cd_catalog.xml");
DataSet ds = new DataSet("CD DataSet");
ds = xmlDatadoc.DataSet;
dataGridView1.DataSource = ds.DefaultViewManager;
dataGridView1.DataMember = "CD";
```

Стартирайте с CTRL+F5.

Самостоятелна задача

Създайте програма за работа с данни за студенти в XML формат (име, фак. Номер, група и курс). Реализирайте добавяне, триене и търсене на студенти в списък. Добавете функционалност за четене и запис от файл.