任何XML文档对任何类型的应用以及正确的解析都必须是良构的（well-formed），即每一个打开的标签都必须有匹配的结束标签，不得含有次序颠倒的标签，并且在语句构成上应符合技术规范的要求。

当前我们需要用的：

1:get\_xml();

mathMLControl1.MC\_getXML();（把公式编辑器转化为xml格式）

String k=mathMLControl1.MC\_getXML();(xml转化为字符串格式)

2:save\_xml();

（）中填要储存的位置。

3：load\_xml();

导出xml。（）中可以直接填上述k，可以直接在公式编辑器中同类型转化过来，也可以填储存过程。

主要内容：

创建xml文件，增、删、改、查xml的节点信息。

① ：生成xml文件  
② ：遍历xml文件的节点信息  
③ ：修改xml文件的节点信息  
④ ：向xml文件添加节点信息  
⑤ ：删除指定xml文件的节点信息

创建有两种方法

定义公共对象

XmlDocument xmldoc ;//表示xml文档

XmlNode xmlnode ;//表示xml单个节点

XmlElement xmlelem ;//表示xml元素

方法一：

xmldoc = new XmlDocument ( ) ;

//加入XML的声明段落,<?xml version="1.0" encoding="gb2312"?>

XmlDeclaration xmldecl;

xmldecl = xmldoc.CreateXmlDeclaration("1.0","gb2312",null);

xmldoc.AppendChild (xmldecl);//appendChild() 方法可向节点的子节点列表的末尾添加 新的子节点。此方法可返回这个新的子节点。

//加入一个根元素

xmlelem = xmldoc.CreateElement ( "" , "Employees" , "" ) ;

xmldoc.AppendChild (xmlelem) ;

//加入另外一个元素

for(int i=1;i<3;i++)

{

XmlNode root=xmldoc.SelectSingleNode("Employees");//查找<Employees>

XmlElement xe1=xmldoc.CreateElement("Node");//创建一个<Node>节点

xe1.SetAttribute("genre","DouCube");//设置该节点genre属性

xe1.SetAttribute("ISBN","2-3631-4");//设置该节点ISBN属性

XmlElement xesub1=xmldoc.CreateElement("title");

xesub1.InnerText="CS从入门到精通";//设置文本节点

xe1.AppendChild(xesub1);//添加到<Node>节点中

XmlElement xesub2=xmldoc.CreateElement("author");

xesub2.InnerText="候捷";

xe1.AppendChild(xesub2);

XmlElement xesub3=xmldoc.CreateElement("price");

xesub3.InnerText="58.3";

xe1.AppendChild(xesub3);

root.AppendChild(xe1);//添加到<Employees>节点中

}

//保存创建好的XML文档

xmldoc.Save ( Server.MapPath("data.xml") ) ;

代码结果如下：

<?xml version="1.0" encoding="gb2312"?>

<Employees>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

</Employees>

方法二：

XmlTextWriter xmlWriter;

string strFilename = Server.MapPath("data1.xml") ;

xmlWriter = new XmlTextWriter(strFilename,Encoding.Default);//创建一个xml文档

xmlWriter.Formatting = Formatting.Indented;

xmlWriter.WriteStartDocument();

xmlWriter.WriteStartElement("Employees");

xmlWriter.WriteStartElement("Node");

xmlWriter.WriteAttributeString("genre","DouCube");

xmlWriter.WriteAttributeString("ISBN","2-3631-4");

xmlWriter.WriteStartElement("title");

xmlWriter.WriteString("CS从入门到精通");

xmlWriter.WriteEndElement();

xmlWriter.WriteStartElement("author");

xmlWriter.WriteString("候捷");

xmlWriter.WriteEndElement();

xmlWriter.WriteStartElement("price");

xmlWriter.WriteString("58.3");

xmlWriter.WriteEndElement();

xmlWriter.WriteEndElement();

xmlWriter.Close();

结果如下：

<?xml version="1.0" encoding="gb2312"?>

<Employees>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

</Employees>

1. 遍历xml文件的节点信息

例子：

private static void GetXMLInformation(string xmlFilePath)

{

try

{

//初始化一个xml实例

XmlDocument myXmlDoc = new XmlDocument();

//加载xml文件（参数为xml文件的路径）

myXmlDoc.Load(xmlFilePath);

//获得第一个姓名匹配的节点（SelectSingleNode）：此xml文件的根节点

XmlNode rootNode = myXmlDoc.SelectSingleNode("Computers");

//分别获得该节点的InnerXml和OuterXml信息

string innerXmlInfo = rootNode.InnerXml.ToString();

string outerXmlInfo = rootNode.OuterXml.ToString();

//获得该节点的子节点（即：该节点的第一层子节点）

XmlNodeList firstLevelNodeList = rootNode.ChildNodes;

foreach (XmlNode node in firstLevelNodeList)

{

//获得该节点的属性集合

XmlAttributeCollection attributeCol = node.Attributes;

foreach (XmlAttribute attri in attributeCol)

{

//获取属性名称与属性值

string name = attri.Name;

string value = attri.Value;

Console.WriteLine("{0} = {1}", name, value);

}

//判断此节点是否还有子节点

if (node.HasChildNodes)

{

//获取该节点的第一个子节点

XmlNode secondLevelNode1 = node.FirstChild;

//获取该节点的名字

string name = secondLevelNode1.Name;

//获取该节点的值（即：InnerText）

string innerText = secondLevelNode1.InnerText;

Console.WriteLine("{0} = {1}", name, innerText);

//获取该节点的第二个子节点（用数组下标获取）

XmlNode secondLevelNode2 = node.ChildNodes[1];

name = secondLevelNode2.Name;

innerText = secondLevelNode2.InnerText;

Console.WriteLine("{0} = {1}", name, innerText);

}

}

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

}

1. 修改xml文件的节点信息

1、**修改结点的值（属性和子结点）：**

XmlDocument xmlDoc=new XmlDocument();

xmlDoc.Load( Server.MapPath("data.xml") );

XmlNodeList nodeList=xmlDoc.SelectSingleNode("Employees").ChildNodes;//获取Employees节点的所有子节点

foreach(XmlNode xn in nodeList)//遍历所有子节点

{

XmlElement xe=(XmlElement)xn;//将子节点类型转换为XmlElement类型

if(xe.GetAttribute("genre")=="张三")//如果genre属性值为“张三”

{

xe.SetAttribute("genre","update张三");//则修改该属性为“update张三”

XmlNodeList nls=xe.ChildNodes;//继续获取xe子节点的所有子节点

foreach(XmlNode xn1 in nls)//遍历

{

XmlElement xe2=(XmlElement)xn1;//转换类型

if(xe2.Name=="author")//如果找到

{

xe2.InnerText="亚胜";//则修改

}

}

}

}

xmlDoc.Save( Server.MapPath("data.xml") );//保存。

结果如下：

<?xml version="1.0" encoding="gb2312"?>

<Employees>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node genre="update张三" ISBN="1-1111-1">

<title>C#入门帮助</title>

<author>亚胜</author>

<price>158.3</price>

</Node>

</Employees>

1. **修改结点（添加结点的属性和添加结点的自结点）：**

XmlDocument xmlDoc=new XmlDocument();

xmlDoc.Load( Server.MapPath("data.xml") );

XmlNodeList nodeList=xmlDoc.SelectSingleNode("Employees").ChildNodes;//获取Employees节点的所有子节点

foreach(XmlNode xn in nodeList)

{

XmlElement xe=(XmlElement)xn;

xe.SetAttribute("test","111111");

XmlElement xesub=xmlDoc.CreateElement("flag");

xesub.InnerText="1";

xe.AppendChild(xesub);

}

xmlDoc.Save( Server.MapPath("data.xml") );

结果如下：

<?xml version="1.0" encoding="gb2312"?>

<Employees>

<Node genre="DouCube" ISBN="2-3631-4" test="111111">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

<flag>1</flag>

</Node>

<Node genre="DouCube" ISBN="2-3631-4" test="111111">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

<flag>1</flag>

</Node>

<Node genre="update张三" ISBN="1-1111-1" test="111111">

<title>C#入门帮助</title>

<author>亚胜</author>

<price>158.3</price>

<flag>1</flag>

</Node>

</Employees>

例子：

private static void ModifyXmlInformation(string xmlFilePath)

{

try

{

XmlDocument myXmlDoc = new XmlDocument();

myXmlDoc.Load(xmlFilePath);

XmlNode rootNode = myXmlDoc.FirstChild;

XmlNodeList firstLevelNodeList = rootNode.ChildNodes;

foreach (XmlNode node in firstLevelNodeList)

{

//修改此节点的属性值

if (node.Attributes["Description"].Value.Equals("Made in USA"))

{

node.Attributes["Description"].Value = "Made in HongKong";

}

}

//要想使对xml文件所做的修改生效，必须执行以下Save方法

myXmlDoc.Save(xmlFilePath);

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

}

1. 向xml文件添加节点信息

XmlDocument xmlDoc=new XmlDocument();

xmlDoc.Load(Server.MapPath("data.xml"));

XmlNode root=xmlDoc.SelectSingleNode("Employees");//查找<Employees>

XmlElement xe1=xmlDoc.CreateElement("Node");//创建一个<Node>节点

xe1.SetAttribute("genre","张三");//设置该节点genre属性

xe1.SetAttribute("ISBN","1-1111-1");//设置该节点ISBN属性

XmlElement xesub1=xmlDoc.CreateElement("title");

xesub1.InnerText="C#入门帮助";//设置文本节点

xe1.AppendChild(xesub1);//添加到<Node>节点中

XmlElement xesub2=xmlDoc.CreateElement("author");

xesub2.InnerText="高手";

xe1.AppendChild(xesub2);

XmlElement xesub3=xmlDoc.CreateElement("price");

xesub3.InnerText="158.3";

xe1.AppendChild(xesub3);

root.AppendChild(xe1);//添加到<Employees>节点中

xmlDoc.Save ( Server.MapPath("data.xml") );

结果：

<?xml version="1.0" encoding="gb2312"?>

<Employees>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node genre="张三" ISBN="1-1111-1">

<title>C#入门帮助</title>

<author>高手</author>

<price>158.3</price>

</Node>

</Employees>

例子：

private static void AddXmlInformation(string xmlFilePath)

{

try

{

XmlDocument myXmlDoc = new XmlDocument();

myXmlDoc.Load(xmlFilePath);

//添加一个带有属性的节点信息

foreach (XmlNode node in myXmlDoc.FirstChild.ChildNodes)

{

XmlElement newElement = myXmlDoc.CreateElement("color");

newElement.InnerText = "black";

newElement.SetAttribute("IsMixed", "Yes");

node.AppendChild(newElement);

}

//保存更改

myXmlDoc.Save(xmlFilePath);

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

}

1. 删除指定xml文件的节点信息
2. **删除结点中的某一个属性：**

XmlDocument xmlDoc=new XmlDocument();

xmlDoc.Load( Server.MapPath("data.xml") );

XmlNodeList xnl=xmlDoc.SelectSingleNode("Employees").ChildNodes;

foreach(XmlNode xn in xnl)

{

XmlElement xe=(XmlElement)xn;

xe.RemoveAttribute("genre");//删除genre属性

XmlNodeList nls=xe.ChildNodes;//继续获取xe子节点的所有子节点

foreach(XmlNode xn1 in nls)//遍历

{

XmlElement xe2=(XmlElement)xn1;//转换类型

if(xe2.Name=="flag")//如果找到

{

xe.RemoveChild(xe2);//则删除

}

}

}

xmlDoc.Save( Server.MapPath("data.xml") );

结果：

<?xml version="1.0" encoding="gb2312"?>

<Employees>

<Node ISBN="2-3631-4" test="111111">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node ISBN="2-3631-4" test="111111">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node ISBN="1-1111-1" test="111111">

<title>C#入门帮助</title>

<author>亚胜</author>

<price>158.3</price>

</Node>

</Employees>

1. **删除结点：**

XmlDocument xmlDoc=new XmlDocument();

xmlDoc.Load( Server.MapPath("data.xml") );

XmlNode root=xmlDoc.SelectSingleNode("Employees");

XmlNodeList xnl=xmlDoc.SelectSingleNode("Employees").ChildNodes;

for(int i=0;i<xnl.Count;i++)

{

XmlElement xe=(XmlElement)xnl.Item(i);

if(xe.GetAttribute("genre")=="张三")

{

root.RemoveChild(xe);

if(i<xnl.Count)i=i-1;

}

}

xmlDoc.Save( Server.MapPath("data.xml") );

结果：

（原来的内容）

<?xml version="1.0" encoding="gb2312"?>

<Employees>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node genre="张三" ISBN="1-1111-1">

<title>C#入门帮助</title>

<author>高手</author>

<price>158.3</price>

</Node>

<Node genre="张三" ISBN="1-1111-1">

<title>C#入门帮助</title>

<author>高手</author>

<price>158.3</price>

</Node>

</Employees>

（删除后的内容）

<?xml version="1.0" encoding="gb2312"?>

<Employees>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

<Node genre="DouCube" ISBN="2-3631-4">

<title>CS从入门到精通</title>

<author>候捷</author>

<price>58.3</price>

</Node>

</Employees>

例子：

private static void DeleteXmlInformation(string xmlFilePath)

{

try

{

XmlDocument myXmlDoc = new XmlDocument();

myXmlDoc.Load(xmlFilePath);

foreach (XmlNode node in myXmlDoc.FirstChild.ChildNodes)

{

//记录该节点下的最后一个子节点（简称：最后子节点）

XmlNode lastNode = node.LastChild;

//删除最后子节点下的左右子节点

lastNode.RemoveAll();

//删除最后子节点

node.RemoveChild(lastNode);

}

//保存对xml文件所做的修改

myXmlDoc.Save(xmlFilePath);

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

}

}