# How to Store Data of DataSet into XML File

## Introduction

In this sample, we will demonstrate how to write data into XML file from DataSet and read data into DataSet from XML.

1. We will create one dataset with two tables.

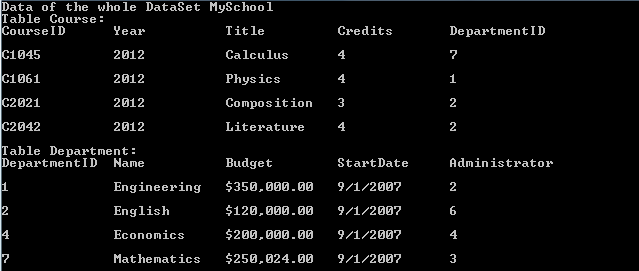
2. We will use two ways to export dataset into the XML files:WriteXml method and GetXml method.

3. We will use two ways to import dataset from the XML files:ReadXml method and InferXmlSchema method.

## Running the Sample

Press F5 to run the sample, the following is the result.

First, we create one dataset that contains two tables.



Second, we use two ways to export the dataset into XML files.

a. Use WriteXml method to export the dataset.



b. Use GetXml method to export the dataset.



Third, we import the dataset from the XML files.

a. Use ReadXml to import the dataset.

We use ReadXml method to read the XML file that WriteXml method created early. Because we also read the schema from the XML file, we can find that the schema is as same as the schema of the original dataset.



b. Use InferXmlSchema method to infer the schemas.

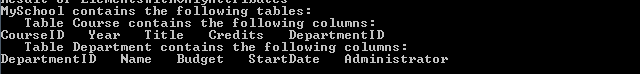
We display four types of XML structures.

1. Elements that only have attributes

Following is the Xml document:

|  |
| --- |
| -Code block start-  --XML code snippet start--  <MySchool>  <Course CourseID="C1045" Year="2012" Title="Calculus" Credits="4" DepartmentID="7" />  <Course CourseID="C1061" Year="2012" Title="Physics" Credits="4" DepartmentID="1" />  <Department DepartmentID="1" Name="Engineering" Budget="350000" StartDate="2007-09-01T00:00:00+08:00" Administrator="2" />  <Department DepartmentID="7" Name="Mathematics" Budget="250024" StartDate="2007-09-01T00:00:00+08:00" Administrator="3" />  </MySchool>  --XML code snippet end--  -Code block end- |

Following is the result of inferring from the above Xml document:



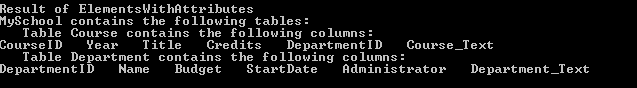
Now we can find that the name of root element becomes the name of dataset, the names of elements become the name of table and the names of attributes become the name of columns.

1. Elements that have attributes and the element text

Following is the Xml document:

|  |
| --- |
| -Code block start-  --XML code snippet start--  <MySchool>  <Course CourseID="C1045" Year="2012" Title="Calculus" Credits="4" DepartmentID="7">New</Course>  <Course CourseID="C1061" Year="2012" Title="Physics" Credits="4" DepartmentID="1" />  <Department DepartmentID="1" Name="Engineering" Budget="350000" StartDate="2007-09-01T00:00:00+08:00" Administrator="2" />  <Department DepartmentID="7" Name="Mathematics" Budget="250024" StartDate="2007-09-01T00:00:00+08:00" Administrator="3">Cancelled</Department>  </MySchool>  --XML code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

Following is the result of inferring from the above Xml document:



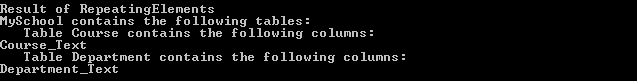
The only difference between type 1 and 2 is that type 2 have the element text in Xml document. So we can find the only difference of the results is that the new columns, “Course\_Text”, “Department\_Text”.

1. Repeating Elements

Following is the Xml document:

|  |
| --- |
| -Code block start-  --XML code snippet start--  <MySchool>  <Course>C1045</Course>  <Course>C1061</Course>  <Department>Engineering</Department>  <Department>Mathematics</Department>  </MySchool>  --XML code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

Following is the result of inferring from the above Xml document:



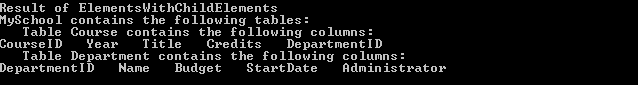
We can find that the repeat elements result in a single inferred table.

1. Elements With Child Elements

Following is the Xml document:

|  |
| --- |
| -Code block start-  --XML code snippet start--  <MySchool>  <Course>  <CourseID>C1045</CourseID>  <Year>2012</Year>  <Title>Calculus</Title>  <Credits>4</Credits>  <DepartmentID>7</DepartmentID>  </Course>  <Course>  <CourseID>C1061</CourseID>  <Year>2012</Year>  <Title>Physics</Title>  <Credits>4</Credits>  <DepartmentID>1</DepartmentID>  </Course>  .................................  <Department>  <DepartmentID>1</DepartmentID>  <Name>Engineering</Name>  <Budget>350000</Budget>  <StartDate>2007-09-01T00:00:00+08:00</StartDate>  <Administrator>2</Administrator>  </Department>  <Department>  <DepartmentID>2</DepartmentID>  <Name>English</Name>  <Budget>120000</Budget>  <StartDate>2007-09-01T00:00:00+08:00</StartDate>  <Administrator>6</Administrator>  </Department>  .................................  </MySchool>  --XML code snippet end--  -Code block end- |

Following is the result of inferring from the above Xml document:



The above Xml document is as same as the document that GetXml method created early and we get the same structure as the original dataset. For this type structure, we can find that the name of root becomes the name of dataset, the names of second level elements become the names of tables and the names of the third level elements become the names of attributes.

## Using the Code

1. Use two ways to export the dataset into the XML files.

a. Use WriteXml method to export the dataset.

|  |
| --- |
| -Code block start-  --C# code snippet start--  using (FileStream fsWriterStream = new FileStream(xmlFileName, FileMode.Create))  {  using (XmlTextWriter xmlWriter = new XmlTextWriter(fsWriterStream, Encoding.Unicode))  {  dataset.WriteXml(xmlWriter, XmlWriteMode.WriteSchema);  Console.WriteLine("Write {0} to the File {1}.", dataset.DataSetName, xmlFileName);  Console.WriteLine();  }  }  --C# code snippet end--  --VB code snippet start--  Using fsWriterStream As New FileStream(xmlFileName, FileMode.Create)  Using xmlWriter As New XmlTextWriter(fsWriterStream, Encoding.Unicode)  dataset.WriteXml(xmlWriter, XmlWriteMode.WriteSchema)  Console.WriteLine("Write {0} to the File {1}.", dataset.DataSetName, xmlFileName)  Console.WriteLine()  End Using  End Using  --VB code snippet end--  -Code block end- |

b. Use GetXml method to export the dataset.

|  |
| --- |
| -Code block start-  --C# code snippet start--  using (StreamWriter writer = new StreamWriter(xmlFileName))  {  writer.WriteLine(dataset.GetXml());  Console.WriteLine("Get Xml data from {0} and write to the File {1}.", dataset.DataSetName, xmlFileName);  Console.WriteLine();  }  --C# code snippet end--  --VB code snippet start--  Using writer As New StreamWriter(xmlFileName)  writer.WriteLine(dataset.GetXml())  Console.WriteLine("Get Xml data from {0} and write to the File {1}.",  dataset.DataSetName, xmlFileName)  Console.WriteLine()  End Using  --VB code snippet end--  -Code block end- |

2. Use two ways to import the dataset from the XML files.

a. Use ReadXml method to import the dataset.

|  |
| --- |
| -Code block start-  --C# code snippet start--  using (FileStream fsReaderStream = new FileStream(xmlFileName, FileMode.Open))  {  using (XmlTextReader xmlReader = new XmlTextReader(fsReaderStream))  {  newDataSet.ReadXml(xmlReader, XmlReadMode.ReadSchema);  }  }  --C# code snippet end--  --VB code snippet start--  Using fsReaderStream As New FileStream(xmlFileName, FileMode.Open)  Using xmlReader As New XmlTextReader(fsReaderStream)  newDataSet.ReadXml(xmlReader, XmlReadMode.ReadSchema)  End Using  End Using  --VB code snippet end--  -Code block end- |

b. Use InferXmlSchema method to import the datset.

|  |
| --- |
| -Code block start-  --C# code snippet start--  String[] xmlFileNames = {  @"XMLFiles\ElementsWithOnlyAttributes.xml",  @"XMLFiles\ElementsWithAttributes.xml",  @"XMLFiles\RepeatingElements.xml",  @"XMLFiles\ElementsWithChildElements.xml" };  foreach (String xmlFileName in xmlFileNames)  {  Console.WriteLine("Result of {0}", Path.GetFileNameWithoutExtension(xmlFileName));  DataSet newSchool = new DataSet();  newSchool.InferXmlSchema(xmlFileName,null);  DataTableHelper.ShowDataSetSchema(newSchool);  Console.WriteLine();  }  --C# code snippet end--  --VB code snippet start--  Dim xmlFileNames() As String =  {"XMLFiles\ElementsWithOnlyAttributes.xml",  "XMLFiles\ElementsWithAttributes.xml",  "XMLFiles\RepeatingElements.xml",  "XMLFiles\ElementsWithChildElements.xml"}  For Each xmlFileName As String In xmlFileNames  Console.WriteLine("Result of {0}", Path.GetFileNameWithoutExtension(xmlFileName))  Dim newSchool As New DataSet()  newSchool.InferXmlSchema(xmlFileName, Nothing)  DataTableHelper.ShowDataSetSchema(newSchool)  Console.WriteLine()  Next xmlFileName  --VB code snippet end--  -Code block end- |

## More Information

[DataSet.GetXml Method](http://msdn.microsoft.com/en-us/library/system.data.dataset.getxml.aspx)

[DataSet.ReadXml Method (XmlReader)](http://msdn.microsoft.com/en-us/library/d6swf149.aspx)

[Inferring Tables](http://msdn.microsoft.com/en-us/library/cat50f7f.aspx)

[Loading a DataSet from XML](http://msdn.microsoft.com/en-us/library/fx29c3yd(VS.110).aspx)