Unit 11 XML in VB .NET

Structure:

11.1 Introduction

In the previous unit covers how exception handling works in Visual Basic 2005 and the further improvements over pre-.NET versions of Visual Basic. Common language runtime (CLR) exception handler in detail and the programming methods those are most efficient in catching errors. Previous unit covers the general principles behind exception handling, the exception object's methods and properties. Also discusses the Try...Catch...Finally structure, the Exit Try statement, and nested Try structures. Error and trace logging and how you can use these methods to obtain feedback on how your program is working.

In this unit we are going to discuss the role of XML in VB. NET. The Extensible Markup Language or XML is a technique of using a document, such as a text file, to describe information and making that information available to whatever and whoever can take advantage of it. The description is done so the document can be created by one person or company and used by another person or another company without having to know who first created the document or how it works. This is because the document thus created is not a program, it is not an application: it is just a text-based document. In this unit we are going to discuss how to create an XML in the VB .NET using Data set, methods used to search and filter the XML file.

Creation of excel file from XML file, serialization and deserialization of XML file.

Objectives:

After studying this unit, you will be able to:

- explain the role of XML in VB .NET
- write code to open and read XML file in VB .Net
- create XML file in VB .NET using Dataset
- explain to search and filter data from XML file
- create excel file from XML
- discuss on serialization an deserialization in XML

11.2 Introduction to XML

XML is the Extensible Markup Language. Like its predecessor SGML (Standard Generalized Markup Language), XML is a meta-language used to define other languages. However, XML is much simpler and more straightforward than SGML. XML is a markup language that specifies neither the tag set nor the grammar for that language. The *tag set* for a markup language defines the markup tags that have meaning to a language parser. For example, HTML has a strict set of tags that are allowed. You may use the tag <TABLE> but not the tag <CHAIR>. While the first tag has a specific meaning to an application using the data, and is used to signify the start of a table in HTML, the second tag has no specific meaning, and although most browsers will ignore it, unexpected things can happen when it appears. That is because when HTML was defined, the tag set of the language was defined with it. With each new version of HTML, new tags are defined. However, if a tag is not defined, it may not be used as part of the markup language without generating an error when the document is parsed. The *grammar* of a markup language defines the correct use of the language's tags. Again, let's use HTML as an example. When using the <TABLE> tag, several attributes may be included, such as the width, the background color, and the alignment. However, you cannot define the TYPE of the table because the grammar of HTML does not allow it.

XML is not an extension of HTML (Hyper Text Markup language). XML is used to describe the data or content and to store the content. The difference between XML and HTML is: HTML is a markup language whereas XML is a

markup language that is used to create new markup languages. XML has no predefined tags, so you must define your own elements (tags) in creation of XML documents.

The first line that can be processed in an XML file must specify the version of XML that you are using. When creating an XML file, you should specify what version your file , especially if you are using a version higher than 1.0. For this reason, an XML file should start (again, must, in 1.1), in the top section, with a line known as an XML declaration. It starts with <?xml version=, followed by the version you are using, assigned as a string, and followed by ?>. An example of such a line is:

<?xml version=1.0”?>

By default, an XML file created using Visual Studio .NET 2003 specifies the version as 1.0. Under the XML declaration line, you can then create the necessary tags of the XML file.

As mentioned already, the tags are created using characters of the alphabet and conform to the ISO standard. This is known as the encoding declaration. The two essentials required to indicate encoding is that Encoding type followed by the encoding scheme, which must be supplied as a string, to specify the encoding.

For example, most of the characters used in the English are known as ASCII. These characters use a combination of 7 bits to create a character (because the computer can only recognize 8 bits, the last bit is left for other uses). Such an encoding is specified as UTF-8. There are other standards such as UTF-16 (for wide, 2-Byte, characters).

To specify the encoding you are using, type encoding followed by the encoding scheme you are using, which must be assigned as a string. The encoding is specified in the the first line. Here is an example:

<?xml version="1.0" encoding="utf-8"?>

XML is mainly used to **store, carry and exchange** the data. But XML is not for displaying data.

Store data: (How to store the data in XML file manually)

Plain text files can be used to store the data. XML can also be used to store data into files (. txt, .html) and databases. But some applications

(XML parsers, XML beans) are written to store and retrieve data from database.

To display XML data: By using of style sheets XSL sheet (XML style sheet language sheet).

XSL sheet = XML doc + CSS commands

Some generic application (CSS commands) is written for displaying XML data (what is stored as part of XML).

Carry data: XML data send from one place to another place.

Note: XML is a cross platform, software and hard ware independent tool for transmitting the data. So that it is easy for a developer to carry data.

Exchange data: with the XML, data can be exchanged between in compatible systems. When you store data in personal computer and databases, that data is being stored in incompatible format. This process uses a variety of techniques to link the systems or programmes together. Because of the lack of tight coupling between the data source and the goal, we call this kind of integration "loosely coupled."

Converting data to XML can greatly reduce time complexity and create data that can be read by different types of applications. XML is used to store a large amount of data. XML is a self describing language and it gives the data as well as the rules to identify what information it contains.

Self Assessment Questions

1. SGML stands for_____.
2. XML is a markup language that is helps to generate new markup language. State [True/False].
3. XSL sheet is the combination of_____and_____.

11.3 XML in VB.NET

By this time you must have understood that the XML is basically general purpose tag based language. It supports in easy storage of data and transfer of data between the applications. Since the XML is a platform independent language the data that are formatted using XML can be used elsewhere. XML is considered as the self-explanatory language since it provides the data as well as the rules to find what information it contains. Parsing XML files has always been time consuming and sometimes trick.

.NET framework provides powerful new ways of parsing XML. Various techniques like xml files with .NET framework are using XmlReader, XmlDocument, XmlSerializer, Dataset and XpathDocument. You can read the contents of a node in an XML document by using the XmlReader's navigation and reading methods. The reader's position informs the class's attributes, which in turn reflect the value of the node at which they are positioned.

The .Net technology is widely supported XML file format. The .Net Framework provides the Classes for read, write, and other operations in XML formatted files. These classes are stored in the namespaces like System.Xml, System.Xml.Schema, System.Xml.Serialization, System.Xml. XPath, System.Xml.Xsl etc. The Dataset in ADO.NET uses XML as its internal storage format.

You can use any text editor to create an XML file. More over XML files are readable by humans as well as computers. For creating a new XML file in VB.NET, we are using XmlTextWriter class. The class takes FileName and Encoding as argument. Also we are here passing formating details. The following source codes creating an XML file product.xml and add four rows in the file.

As per the given example there are four different product details are given that is product1, product2, product3, and product 4 these four records will be read and formatted and generated as a XML file.

```
Imports System.Xml
Public Class Form1
 Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
    Dim writer As New XmlTextWriter("product.xml", System.Text.
Encoding.UTF8)
    writer.WriteStartDocument(True)
    writer.Formatting = Formatting.Indented
    writer.Indentation = 2
    writer.WriteStartElement("Table")
    createNode(1, "Product 1", "1000", writer)
    createNode(2, "Product 2", "2000", writer)
    createNode(3, "Product 3", "3000", writer)
    createNode(4, "Product 4", "4000", writer)
    writer.WriteEndElement()
    writer.WriteEndDocument()
    writer.Close()
  End Sub
  Private Sub createNode(ByVal pID As String, ByVal pName As
String, ByVal pPrice As String, ByVal writer As XmlTextWriter)
    writer.WriteStartElement ("Product")
```

```
            writer.WriteStartElement ("Product_id")
            writer.WriteString (pID)
            writer.WriteEndElement ()
            writer.WriteStartElement ("Product_name")
            writer.WriteString (pName)
            writer.WriteEndElement ()
            writer.WriteStartElement ("Product_price")
            writer.WriteString (pPrice)
            writer.WriteEndElement()
            writer.WriteEndElement()
        End Sub
    End Class
```

If you execute the above code following output will be generated.

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
-<Table>
 -<Product>
        <Product_id>1</Product_id>
        <Product_name>Product 1</Product_name>
        <Product_price>1000</Product_price> </Product>
 -<Product>
        <Product_id>2</Product_id>
        <Product_name>Product 2</Product_name>
        <Product_price>2000</Product_price> </Product>
-<Product>
        <Product_id>3</Product_id>
        <Product_name>Product 3</Product_name>
        <Product_price>3000</Product_price> </Product>
         -<Product>
                <Product_id>4</Product_id>
                <Product_name>Product 4</Product_name>
                <Product_price>4000</Product_price> </Product>
        </Table>
```

Using XmlDocument in VB .NET

From this below coding we will understand to parse the data through xmlDocument. XmlDocument is a class supports to read all the XML documents. XML is a standard that can be used in various applications like

windows, web, web services etc. Following is the sample XML file called “XMLFile.xml” will be read through in this example.

```
<?xml version="1.0" encoding="utf-8"?>
<data>
    <Microsoft>
        <product>Visual Studio 2005</product>
        <product>MS SQL SERVER 2005</product>
        <product>Office 2003</product>
     </Microsoft>
    <Adobe>
        <product>Adobe Reader</product>
        <product>PDF Writer</product>
        <product>Distiller</product>
    </Adobe>
    <Dell>
        <product>Computers</product>
        <product>Printers</product>
        <product>Accessories</product>
    </Dell>
 </data>
```

Below code display the contents of the xml file. The main aim of this program is create an xmlDocument object that parse through the various nodes and exhibit the data finally. Double quoted are the command line helps you understand the logic behind the particular executable statement.

```
' Create xml object
Dim xmldoc As New System.Xml.XmlDocument()
 ' Load from file
xmldoc.Load(Server.MapPath(".") + "\XMLFile.xml")
' Get a list of all the child elements
Dim nodelist As XmlNodeList =
xmldoc.DocumentElement.ChildNodes
Response.Write("Number of items in data node: " + nodelist.Count +
"<br />")
Response.Write("Document root node: " +
xmldoc.DocumentElement.Name + "<br />")
' Parse through all nodes
```

```
For Each outerNode As XmlNode In nodelist
   ' Write the outerNode name such as Microsoft, Adobe, Dell
    Response.Write("--> Outer node name: " + outerNode.Name +
"<br />")
     ' Check if this matches with our selected item
    For Each InnerNode As XmlNode In outerNode.ChildNodes
       ' Write the InnerNode name and value such as Visual Studio
2005, Adobe Reader, Computers, etc
        Response.Write(" -----> Inner node Name: " +
InnerNode.Name + "<br />")
        Response.Write(" -----> Inner node InnerText: " +
InnerNode.InnerText + "<br />")
     Next
 Next
```

Self Assessment Questions

4. Storage is complicated in XML. State[True/False].
5. _____ in ADO.NET uses XML as its internal storage format.

File operations in XML:

Various operations that are performed using file operations are like creating a file, opening a file and reading the contents of file. We can create an XML file in several ways. In the previous section we create an XML file using XmlTextWriter Class. In the .NET framework, the System.IO namespace is the region of the base class libraries devoted to file based input and output services. Like any namespace, the System.IO namespace defines a set of classes, interfaces, enumerations, structures and delegates. Similar to file operation XML also supports various class reader and writer classes

1. Methods used to create a file:

Three methods to create a file are shown here. XML is a tag based language, means the document is made up of XML tags that contain information. We can create an XML file in several ways. In the previous section we create an XML file using XmlTextWriter Class. Before we start with the discussion of program we shall see the terminologies used in this context.

DataTable: Data is often stored in tables. And these tables often reside in databases. In the .NET Framework, the DataTable type stores data in memory. It is used often in VB.NET programs. It has Columns and Rows properties. DataTable is an in-memory representation

of structured data such as that read from a database.

DataRow: A DataRow contains an individual row of data. It narrows the data abstraction to the level of the row. The DataRow type provides ways to add, remove, or read cells from the enclosing data structure.

Here we are creating an XML file Product.XML using an ADO.NET Dataset. For that we have to manually create a Datatable first and add the data of Product.XML in the Datatable . Then add the Datatable in a Dataset .

Creating an XML file using VB.NET Class:

Creation of XML file is illustrated here. The .Net technology is widely supported XML file format. The .Net Framework provides the Classes for read, write, and other operations in XML formatted files. These classes are stored in the namespaces like System.Xml, System.Xml.Schema, System.Xml.Serialization, System.Xml. XPath, System.Xml.Xsl etc. The Dataset in ADO.NET uses XML as its internal storage format.

You can use any text editor to create an XML file. More over XML files are readable by humans as well as computers. For creating a new XML file in VB.NET, we are using XmlTextWriter class. The class takes FileName and Encoding as argument. Also we are here passing formating details.

The following source codes creating an XML file product.xml and add four rows in the file. XmlTextWriter class is used for creating a new file in XML.

Creating an XML file using VB.NET Dataset:

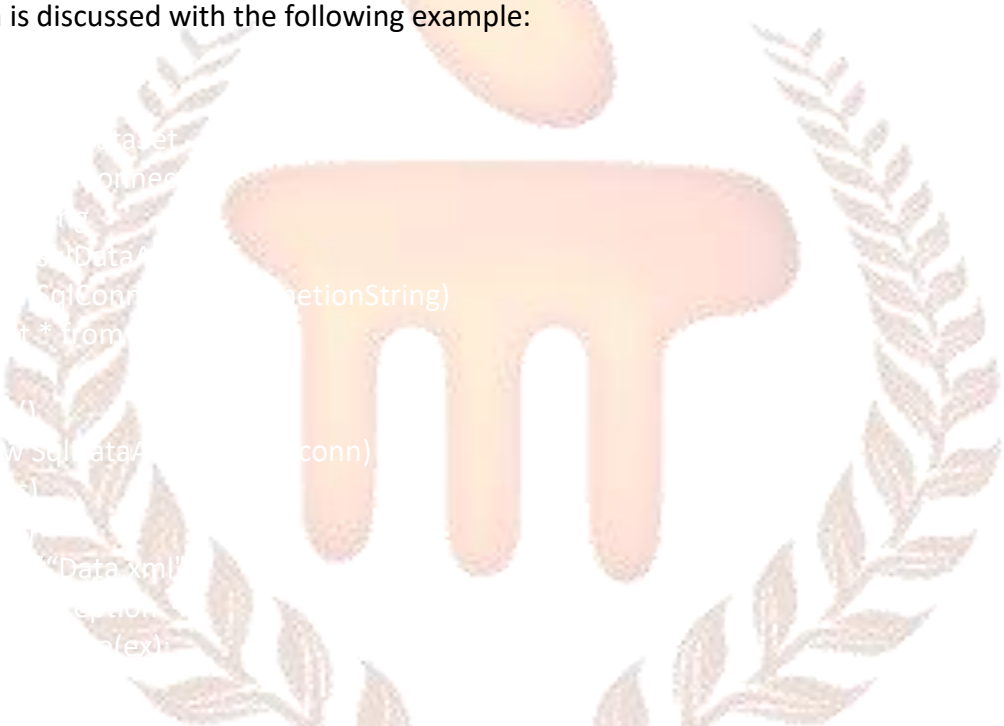
In the previous section we create an XML file using XmlTextWriter Class. Here we are creating an XML file Product.XML using an ADO.NET Dataset. For that we have to manually create a Datatable first and add the data of Product.XML in the Datatable . Then add the Datatable in a Dataset . Call the method WriteXml of Dataset and pass the file name Product.XML as argument. Here, we're going to use an ADO.NET Dataset to build the XML file Product.XML. To do this, we must manually build a Datatable and then populate it with the data from Product.XML. The Datatable is then included in a Dataset. Call the Dataset's WriteXML method and supply the parameter of the file Product.XML.

First, a Datatable is created. Second, the data of XML file is transferred to Datatable, Third, the Datatable is added to the Dataset and finally the forth method the WriteXML is

called and an argument file name is passed.

Creating an XML file using SQL in VB.NET:

Creation of file with SQL is shown here. XML file is created with the help of database. Then, Write.XML method of Dataset is called. At first, the SQLconnection is created to the database. After that the execution of sql occurs and data is stored in the dataset. The execution is discussed with the following example:



Opening an XML file:

Open operation of file is explained here. The XML file is opened to read the data from it. To read a file, it is necessary to open it first. So, we start with the using( ) statement. Inside that, we created a reference of **XmlReader**. Then assigned a reader stream of a XML file using the **Create**() method.

Now, we start reading the XML file and **reader.Read()** returns the Boolean value indicating whether there is a XML statement or not.

If Yes, then we try to check if the current statement contains a starting element or not using **reader.IsStartElement()**.

Since we have a number of different element fields in the Student elements, we are using

a switch block. All the cases are names of Element. Since we want an actual text string of the element, for that we used the **ReadString()**. Since it returns a string type.

11.4 Read XML file in VB.NET

XML is a self describing language and it gives the data as well as the rules to extract what the data it contains. Reading an XML file means that we are reading the information embedded in XML tags in an XML file. In the previous section we have created an XML file and named it as products.xml. The following program read that file and extracts the contents inside the XML tag. We can read an XML file in several ways depending on our requirement. This program read the content in Node wise. Here we are using XmlDataDocument class to read the XML file. In this program it searches the Node < Product > and its child Nodes and extracts the data in child nodes.

```
Imports System.Xml
Imports System.IO
Public Class Form1
 Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
    Dim xmldoc As New XmlDataDocument()
```

```
        Dim xmlnode As XmlNodeList
        Dim i As Integer
        Dim str As String
        Dim fs As New FileStream("products.xml", FileMode.Open,
FileAccess.Read)
        xmldoc.Load(fs)
        xmlnode = xmldoc.GetElementsByTagName("Product")
        For i = 0 To xmlnode.Count - 1
            xmlnode(i).ChildNodes.Item(0).InnerText.Trim()
            str = xmlnode(i).ChildNodes.Item(0).InnerText.Trim() & " | " &
xmlnode(i).ChildNodes.Item(1).InnerText.Trim() & " | " &
xmlnode(i).ChildNodes.Item(2).InnerText.Trim()
            MsgBox(str)
        Next
    End Sub
End Class
```

11. 5 Create XML file in VB .NET using Dataset

XML is a tag based language, means the document is made up of XML tags that contain information. We can create an XML file in several ways. In the previous section we create an XML file using XmlTextWriter Class. To prevent the namespace declaration from being repeated on the two child items, XmlTextWriter moves it up to the root element. The namespace prefix is inherited by the child components. You can also alter the existing namespace declaration with XmlTextWriter. Before we start with the discussion of program we shall see the terminologies used in this context.

DataTable: Data is often stored in tables. And these tables often reside in databases. In the .NET Framework, the DataTable type stores data in memory. It is used often in VB.NET programs. It has Columns and Rows properties. DataTable is an in-memory representation of structured data such as that read from a database.

DataRow: A DataRow contains an individual row of data. It narrows the data abstraction to the level of the row. The DataRow type provides ways to add, remove, or read cells from the enclosing data structure.

Here we are creating an XML file Product.XML using an ADO.NET Dataset.

For that we have to manually create a Datatable first and add the data of Product.XML in the Datatable . Then add the Datatable in a Dataset . Call

the method WriteXml of Dataset and pass the file name Product.XML as argument.

```
Imports System.Xml
Imports System.Data
Public Class Form1
   Dim dt As DataTable
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
   System.EventArgs) Handles Button1.Click
      Dim ds As New DataSet
      dt = New DataTable()
      dt.Columns.Add(New DataColumn("Product_ID", Type.GetType
      ("  System.Int32")))
      dt.Columns.Add(New DataColumn("Product_Name",
Type.GetType(" System.String")))
      dt.Columns.Add(New DataColumn("product_Price",
Type.GetType("
      System.Int32")))
      fillRows(1, "product1", 1111)
      fillRows(2, "product2", 2222)
      fillRows(3, "product3", 3333)
      fillRows(4, "product4", 4444)
      ds.Tables.Add(dt)
      ds.Tables(0).TableName = "product"
      ds.WriteXml("Product.xml")
      MsgBox("Done")
   End Sub
```

```
Private Sub fillRows(ByVal pID As Integer, ByVal pName As String, ByVal
pPrice As Integer)
   Dim dr As DataRow
   dr = dt.NewRow()
   dr("Product_ID") = pID
   dr("Product_Name") = pName
   dr("product_Price") = pPrice
   dt.Rows.Add(dr)
  End Sub
End Class
```

In the above example dt is the object created for the data table. Using (columns.Add method) three columns are added in the data table. ds is the data row created to update the data in each column using fillrows method. Four records are stored in the table called product.

Self Assessment Questions

6. In XmlDataDocument_____method is used to load the XML file.
7. _____is available in datatable to hold data.
8. DataRow consists of row of a particular data. State [True/False].

11.6 Search and Filter in XML File

Search and filter are the options used to search locate and list the searching items from the table.

Search in XML file

XML files are made up of tags that contain information. The .NET technology is widely supported XML file format. Also the Dataset in ADO.NET uses XML format as its internal storage format.

The following source code shows how to search an item in an XML file using Dataset. Here Dataset using an XmlReader for read the content of the file. Locate the XML file using XmlReader and pass the XmlReader as argument of Dataset. By using the Dataset, search the product Product2 in the file Product.XML with the help of DataView.

Data View: Data View offers multiple views of the data stored in DataTable, helps to customize the view of data. It can be used to filter, sort and search the data from the datatable. Also allow the user to update the existing content and add the new data into the table.

```
Imports System.Xml
Public Class Form1
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button1.Click
        Dim xmlFile As XmlReader
        xmlFile = XmlReader.Create("Product.xml", New XmlReader
        Settings())
        Dim ds As New DataSet
        Dim dv As DataView
        ds.ReadXml(xmlFile)
```

```
    dv = New DataView(ds.Tables(0))
    dv.Sort = "Product_Name"
    Dim index As Integer = dv.Find("Product2")
    If index = -1 Then
       MsgBox("Item Not Found")
    Else
   MsgBox(dv(index)("Product_Name").ToString() & "  " & dv(index)
    ("Product_Price").ToString())
    End If
  End Sub
```

Using ReadXml the XML file is read and the column product name is been sorted with the sort method of dataview. Then using the find method the data is been searched during this process of the index value =-1 means you reched the end of the record stop the searching process and display Item not found. Else if the item found during the process it displays the product name along with the index value.

Filter in XML file

As we discussed earlier XML is a platform independent language, so the information formatted in XML can be used in any other platforms (Operating Systems). If we create an XML file in one platform it can be used in other platforms also. The .Net technology is widely supported XML file format. Also the Dataset in ADO.NET uses XML format as its internal storage format.

Here we are going to filter XML file content and store the result in a newly created XML file. We have an XML file Product.xml, and it has a field Product_Price. We are giving a search criteria like the Product_Price >= 3000 and store the result in a newly created XML file Result.xml.

```
Imports System.Xml
Public Class Form1
 Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
    Dim xmlFile As XmlReader
    xmlFile = XmlReader.Create("Product.xml", New
XmlReaderSettings())
```

```
        Dim ds As New DataSet
        Dim dv As DataView
        ds.ReadXml(xmlFile)
        dv = New DataView(ds.Tables(0), "Product_price > = 3000",
"Product_Name", DataViewRowState.CurrentRows)
        dv.ToTable().WriteXml("Result.xml")
        MsgBox("Done")
    End Sub
End Class
```

11.7 Create Excel File from XML

Now we are going to discuss to read a content from the XML file and the same will be written to the excel file. In this program we are going use the XmlReader to read a data from the XML file. The read information will be holed in the dataset. Then write the code to move the content from dataset to the excel file.

```
Imports System.Xml
Imports System.Data
Imports Excel = Microsoft.Office.Interop.Excel
Public Class Form1
 Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        Dim xlApp As Excel.Application
        Dim xlWorkBook As Excel.Workbook
        Dim xlWorkSheet As Excel.Worksheet
        Dim misValue As Object = System.Reflection.Missing.Value

        Dim ds As New DataSet
        Dim xmlFile As XmlReader
        Dim i, j As Integer

        xlApp = New Excel.ApplicationClass
        xlWorkBook = xlApp.Workbooks.Add(misValue)
        xlWorkSheet = xlWorkBook.Sheets("sheet1")

        xmlFile = XmlReader.Create("Product.xml", New
XmlReaderSettings())
        ds.ReadXml(xmlFile)
```

```
        For i = 0 To ds.Tables(0).Rows.Count - 1
            For j = 0 To ds.Tables(0).Columns.Count - 1
                xlWorkSheet.Cells(i + 1, j + 1) = _
                ds.Tables(0).Rows(i).Item(j)
            Next
        Next

        xlWorkSheet.SaveAs("xml2excel.xlsx")
        xlWorkBook.Close()
        xlApp.Quit()

        releaseObject(xlApp)
        releaseObject(xlWorkBook)
        releaseObject(xlWorkSheet)
    End Sub

    Private Sub releaseObject(ByVal obj As Object)
        Try
System.Runtime.InteropServices.Marshal.ReleaseComObject(obj)
            obj = Nothing
        Catch ex As Exception
            obj = Nothing
        Finally
            GC.Collect()
        End Try
    End Sub
End Class
```

Self Assessment Question

9. _____offers several views of the data stored in DataTable.
10. _____method of Dataview helps to search the data from the datatable.
11. _____ object in the function inform explicitly that the optional argument is omitted.

11.8 Serialization in XML

XML Serialization is the process of serializing a .Net Object to the form of XML or from an XML to .Net Object. The primary purpose of XML

serialization in the .NET Framework is to enable the conversion of XML documents and streams to common language runtime objects and vice versa. This is the process of converting an object into a form that can be readily transported.

During XML serialization, only the public properties and fields of an object are serialized. Serialization of XML to common language runtime objects enables one to convert XML documents into a form where they are easier to process using conventional programming languages. The .Net technology is widely supported XML file format. The .Net Framework provides the Classes for read, write, and other operations in XML formatted files.

The following program shows how to serialize a Dataset to an XML disk file. Here we are using XmlSerializer class for serialize the Dataset Object.

```
Public Class Form1
   Dim dt As DataTable
 Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
      Dim ds As New DataSet
      dt = New DataTable()
      dt.Columns.Add(New DataColumn("Product_ID", Type.GetType
     ("  System.Int32" )))
      dt.Columns.Add(New DataColumn("Product_Name",
Type.GetType(" System.String")))
      dt.Columns.Add(New DataColumn("product_Price",
Type.GetType("System.Int32")))
      fillRows(1, "product1", 9999)
      fillRows(2, "product2", 2222)
      fillRows(3, "product3", 3333)
      fillRows(4, "product4", 4444)
      ds.Tables.Add(dt)
      ds.Tables(0).TableName = "product"

      Dim serialWriter As StreamWriter
      serialWriter = New StreamWriter("serialXML.xml")
      Dim xmlWriter As New XmlSerializer(ds.GetType())
      xmlWriter.Serialize(serialWriter, ds)
      serialWriter.Close()
```

```
        ds.Clear()
    End Sub

    Private Sub fillRows(ByVal pID As Integer, ByVal pName As
String, ByVal pPrice As Integer)
        Dim dr As DataRow
        dr = dt.NewRow()
        dr("Product_ID") = pID
        dr("Product_Name") = pName
        dr("product_Price") = pPrice
        dt.Rows.Add(dr)
    End Sub
End Class
```

Things involved in Serialization:

Two things can be serialized in XML serialisation . serialization allows the developer to save the state of an object and recreate it as needed, providing storage of objects as well as data exchange. Through serialization, a developer can perform actions like sending the object to a remote application by means of a Web Service, passing an object from one domain to another, passing an object through a firewall as an XML string, or maintaining security or user-specific information across applications. Public fields of an object are serialised in XML serialisation. Property values of an object are also serialised in XML serialisation

XML serialization does not convert methods, indexers, private fields, or read-only properties (except read-only collections). To serialize all an object's fields and properties, both public and private, use the DataContractSerializer instead of XML serialization. Sometimes XML serialization will not be considered for few things like ‘type information’ is not included in XML serialization because All serialised kinds must be known to it. You can't provide it an interface representing a type that the serializer isn't familiar with and expect it to be processed. Secondly

Things not involved in Serialization:

XML serialization does not convert methods, indexers, private fields, or read-only properties (except read-only collections). To serialize all an object's fields and properties, both public and private, use the DataContractSerializer instead of XML serialization. Sometimes XML serialization will not be considered for few things like 'type information' is not included in XML serialization because All serialised kinds must be known to it. You can't provide it an interface representing a type that the serializer isn't familiar with and expect it to be processed. Secondly

XML Serialisation: Let us know more about serialization types, first one is serialization of Dataset which takes cares of XmlElement and XmlNode, the second type is serialization of class were it involves serialization of collection of interface , and it take scares of serializing the array of objects

Various steps involved in XML serialisation are shown here.

1.A .NET object is serialised to XML form and vice-versa in XML serialization.
2.Only public fields are serialised in XML serialisation.
3.Once the object is serialised, it is also de -serialised.

Advantage and Disadvantage of Serialization:

- Let us see some advantages of XML serialization , Serialization facilitate the transportation of an object through a network and also it Creates a clone of an object, it provides the flexible environment and it confirms the specific schema classes and its associated members.

- There are two major disadvantages of serialization .Very importantly the serialization in XL is not secure and second important disadvantages is XML serialization process takes lot of disk space

Deserialize

XML is a general purpose tag based language and very easy to transfer and store data across applications. XML Serialization is the process of serializing a .Net Object to the form of XML file or from an XML to .Net Object. During XML serialization, only the public properties and fields of an object are serialized. The following source code shows how to de-serialize

the DataSet as it is streamed from an XML file back into memory.

```
Imports System.Xml.Serialization
Imports System.io
Public Class Form1
   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
      Dim ds As New DataSet
      Dim xmlSerializer As XmlSerializer = New XmlSerializer(ds.GetType)
      Dim readStream As FileStream = New FileStream("serialXML.xml",
FileMode.Open)
      ds = CType(xmlSerializer.Deserialize(readStream), DataSet)
      readStream.Close()
      DataGridView1.DataSource = ds.Tables(0)
   End Sub
End Class
```

Self Assessment Questions

12. Conversion of XML documents and streams to CLR objects is called _____.
13. _____ class is used to serialize the objects of Dataset.

11.9 Summary

- XML (Extensible Markup Language) is useful to project the useful information in a structured type.
- The main role of XML is to carry exchange and store information.
- .NET framework provides strong support to parse the XML documents.
- There are plenty of ways available to read the XML file in this unit we are using node wise content reading.
- DataTable is the type of data storage available in .NET framework to store data in row and column wise.
- DataRow is used to hold the row of data one at a time.
- DataView is helpful in viewing the datarable content in a multiple ways.
- System.Reflection.Missing.Value inform explicitly that the optional argument is omitted that we refer in our program to create excel file.
- XML Serialization is the process of serializing a .Net Object to the form of XML or from an XML to .Net Object

11.10 Terminal Questions

1. Discuss the uses of XML
2. Explain the role of XML in VB .NET
3. Write a code to open and read a XML file in VB .NET
4. Explain the role of DataTable and DataRow in XML with appropriate example.
5. How to search and filter a document from the XML file.
6. What is serialization and deserialization in XML

11.11 Answers

Self Assessment Questions

1. Standard Generalized Markup Language
2. True

3. XML doc & CSS commands
4. False
5. Dataset
6. Load
7. DataTable
8. True.
9. Data View
10. Find
11. System.Reflection.Missing.Value
12. Serialization
13. XmlSerializer

Terminal Questions

1. XML is useful to project the useful information in a structured type. The main role of XML is to carry exchange and store information. For more details refer section 11.2.
2. .NET framework provides strong support to parse the XML documents. For more details refer section 11.3.
3. XML is a self describing language and it gives the data as well as the rules to extract what the data it contains. There are plenty of ways available to read a data from XML file. For more details refer section 11.4.
4. DataRow is used to hold the row of data one at a time. DataView is helpful in viewing the datarable content in a multiple ways. For more details refer section 11.5.
5. Using datatable object you can filter and search a data from the XML file. For more details refer section 11.6.
6. XML Serialization is the process of serializing a .Net Object to the form of XML or from an XML to .Net Object. For more details refer section 11.8.

E-Reference:

- http://www.codeproject.com/Articles/4826/XML-File-Parsing-in-VB-NET
- http://www.progtalk.com/viewarticle.aspx?articleid=86
- http://vb.net-informations.com/xml/open-xml-vb.net.htm
- http://social.msdn.microsoft.com/Forums/en-US