Agenda

- What is Serialization?
- Serialization in .NET Framework
- Implementation

What is Serialization?

- Serialization is the process of converting the state of an object into a form that can be persisted or transported.
- The complement of serialization is deserialization, which converts a stream into an object.
- Together, these processes allow data to be easily stored and transferred.

.NET Framework Serialization

- The .NET Framework features two serializing technologies:
  - Binary Serialization
  - XML Serialization

Binary Serialization

- Binary Serialization
  - During this process, the public and private fields of the object and the name of the class, including the assembly containing the class, are converted to a stream of bytes.
  - The bytes are then written to a data stream.
  - When the object is subsequently deserialized, an exact clone of the original object is created.
  - All items can be serialized in binary serialization.

XML Serialization

- XML Serialization
  - XML serialization converts (serializes) only the public fields and properties of an object, or the parameters and return values of methods, into an XML stream that conforms to a specific XML Schema definition language (XSD) document.
  - XML serialization results in strongly typed classes with public properties and fields that are converted to a serial format (in this case, XML) for storage or transport.
- The following items can be serialized using the XMLSerializable class:
  - Public read/write properties and fields of public classes.
  - Classes that implement ICollection or IEnumerable (Note that only collections are serialized, not public properties).
  - XmlElement objects.
  - XmlNode objects.
  - DataSet objects.

Implementation

```
using System;
using System.Xml.Serialization;
using System.IO;
namespace TestXMLSerialization
{
  [XmlRoot(ElementName="Employee")]
  public class Employee
  {
                //Declare the member variables
                private string empID;
                private string empName;
                private int empAge;
                private string[] empSkills;
                [XmlAttribute(AttributeName="EmpID")]
                public string EmpID
                {
                        get
                        {
                                return empID;
                        }
                        set
                        {
                                empID = value;
                        }
                }
                [XmlElement(ElementName="Name")]
                public string Name
                 {
                        get  {  return empName;}
                        set
                        {
                                empName = value;
                        }
                  }
                [XmlElement(ElementName="Age")]
                public int Age
                {
                        get
                        {
                                return empAge;
                        }
                        set
                        {
                                empAge = value;
                        }
                }
                [XmlElement(ElementName="Skills")]
                public string[] Skills
                {
                        get
                        {
                                return empSkills;
                        }
                        set
                        {
                                empSkills = value;
                        }
                }
```

Implementation (cont.)

```
public static void Main()
{
                    //Create an array of skills
                    string[] empSkills = {"C#", "SQL Server 2000"};
                    //Create a new employee object and set the values
                    Employee serializeEmp = new Employee();
                    serializeEmp.EmpID = "EMP1001";
                    serializeEmp.Name = "George";
                    serializeEmp.Age    = 26;
                    serializeEmp.Skills  = empSkills;

                    //Open a new streamwriter
                    StreamWriter newStream =
File.CreateText(@"C:\SerializeSample.xml");
                    XmlSerializer xmlSerialize = new XmlSerializer(typeof(Employee));
                    //Serialize the employee object
                    xmlSerialize.Serialize(newStream, serializeEmp);
                    newStream.Close();
```

```
  Employee deserializeEmp = new Employee();
  //Create an instance of StreamReader
  StreamReader readStream = File.OpenText(@"C:\SerializeSample.xml");
  xmlSerialize = new XmlSerializer(typeof(Employee));
  //Deserialize the employee object from the file
  deserializeEmp = (Employee) xmlSerialize.Deserialize(readStream);
  readStream.Close();
  //Print the employee name from the deserialized employee object
  Console.WriteLine("Deserialized Object Employee Name : " + deserializeEmp.Name);
  }
 }
}
```

Deserializing back to employee object

Snapshot of the file

```
<?xml version="1.0" encoding="utf-8"?>
<Employee xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" EmpID="EMP1001">
  <Name>George</Name>
  <Age>26</Age>
  <Skills>C#</Skills>
  <Skills>SQL Server 2000</Skills>
</Employee>
```

```
Deserialized object Employee Name : George
```