.NET Serialization

Objectives

- Discuss Object serialization
- Discuss serialization of objects to streams
- Discuss Binary formatters
- Discuss SOAP formatters

Introduction

- Converting an object instance into a format that can either be stored to the disk or transported over the network
- Object can be recreated with its current state at a different location

Object Serialization (1)

- Serializing objects to a stream using binary formatters
- Serializing objects to a stream using SOAP formatters and saving them as XML files

Object Serialization (2)

- Objects may be remoted by serializing an object to a stream of bytes
- This stream is then transmitted to another machine that understands the serialization format

How .NET serialization works?

- A .NET formatter class must be used to control the serialization of the object to and from the stream
- The serialized stream carries information about the objects type, including its assembly name, culture & version

Role of formatters

- Determines the serialization format for objects
- All formatters expose an interface called the IFormatter interface

IFormatter Interface

- The 2 formatters that inherit from the IFormatter Interface are –
  - BinaryFormatter
  - SOAPFormatter

Binary Formatter

- To serialize an object we need –
  - A formatter which is used to serialize objects
  - The object that is to be serialized
  - A stream to hold the serialized object

Serializing an object Binary formatter

- Example 1 -

```
using System;
using System.Runtime.Serialization.Formatters.Binary;
using System.IO;

namespace XML2Ex1
{
      class Class1
      {
      static void Main(string[] args)
        {
        Test MyObj = new Test();

        MyObj.Name = "Garfield";
        MyObj.phoneNumber= 5555555;

        Stream MyStream =
File.OpenWrite("C:\\BinSerialization.ex");
        BinaryFormatter formatter = new BinaryFormatter();
        formatter.Serialize(MyStream, MyObj);
        MyStream.Close();
        }
      }
}
```

Serializing an object Binary formatter

- Example 2 -

```
using System;

namespace Serialization
{
      [Serializable]
      public class Test
      {
            public string Name;
            public int phoneNumber;
      }

}
```

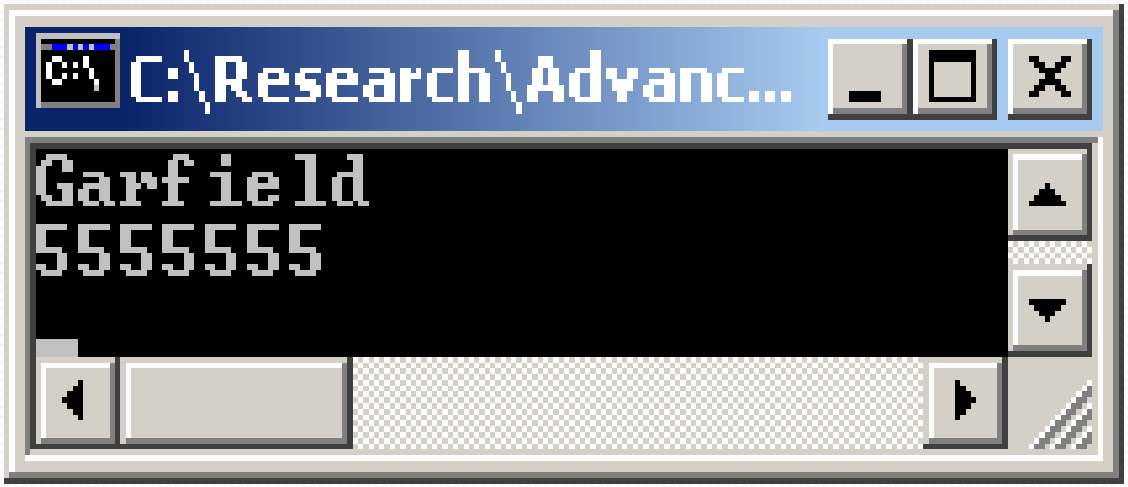
Deserializing an object using Binary formatter (1)

- Example 3 -

```
using System;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;
using System.Runtime.Serialization;

namespace Serialization
{
      class Class2
      {
            static void Main(string[] args)
            {
            FileStream file = new
FileStream("C:\\BinSerialization.ex", FileMode.Open);

            BinaryFormatter formatter = new BinaryFormatter();

            Test MyObj = formatter.Deserialize(file) as Test;

            Console.WriteLine(MyObj.Name);
            Console.WriteLine(MyObj.phoneNumber);
            Console.ReadLine();
            }
      }
}
```

Deserializing an object using Binary formatter (2)

- Output -



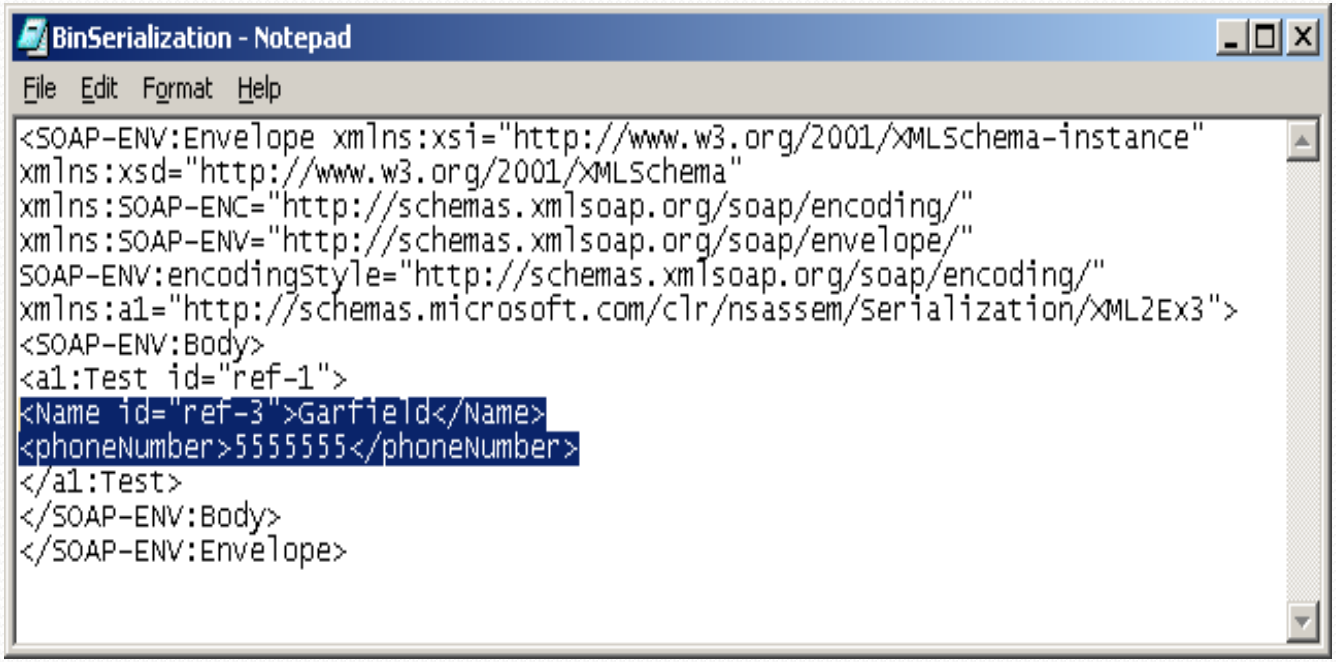
Serializing an object using SOAP formatter (1)

- Example 4 -

```
using System;
using System.Runtime.Serialization.Formatters.Soap;
using System.IO;

namespace Serialization
{
      class Class1
      {
            static void Main(string[] args)
            {
                  Test MyObj = new Test();

                  MyObj.Name = "Garfield";
                  MyObj.phoneNumber= 5555555;

                  Stream MyStream =
File.OpenWrite("C:\\BinSerialization.ex");
                  SoapFormatter formatter = new SoapFormatter();
                  formatter.Serialize(MyStream, MyObj);
                  MyStream.Close();
            }
      }
}
```

Serializing an object using SOAP formatter (2)

- Output -



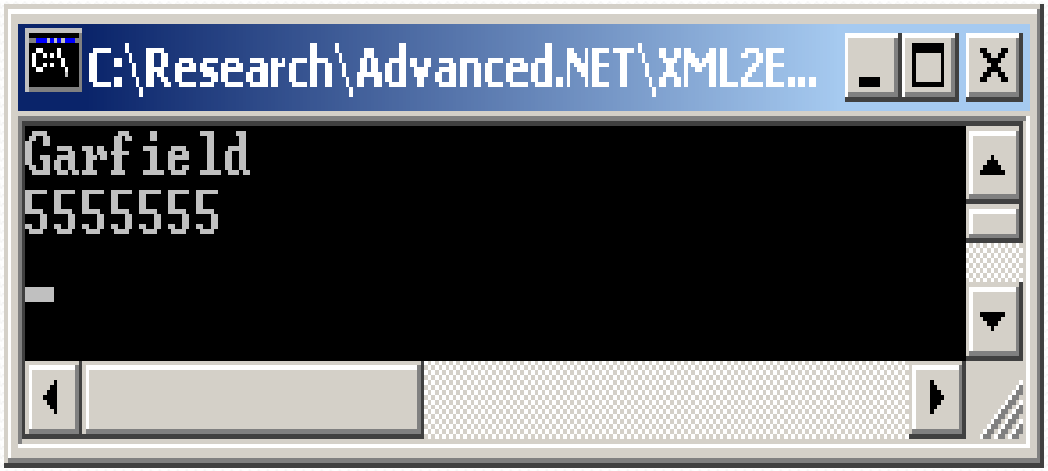
Deserializing an object using SOAP formatter (1)

- Example 5 -

```
using System;
using System.IO;
using System.Runtime.Serialization.Formatters.Soap;
using System.Runtime.Serialization;

namespace Serialization
{
      class Class2
      {
            static void Main(string[] args)
            {
            FileStream file = new
FileStream("C:\\BinSerialization.ex", FileMode.Open);
            SoapFormatter formatter = new SoapFormatter();

            Test MyObj = (Test) formatter.Deserialize(file);

            Console.WriteLine(MyObj.Name);
            Console.WriteLine(MyObj.phoneNumber);
            Console.ReadLine();
            }
      }
}
```

Deserializing an object using SOAP formatter (2)

- Output -



Selectively serializing the members of an object

- Example 6 -

```
using System;

namespace Serialization
{
      [Serializable]
      public class Test
      {
        public string Name;
        public int phoneNumber;

        [NonSerialized]
        public bool CalledToday;
      }

}
```