//пример 1

[Serializable]

class FullNameClass {

public string Name { get; set; }

public string Surname { get; set; }

public string MiddleName { get; set; }

public FullNameClass(string name, string surname, string middle){

Name = name; Surname = surname; MiddleName = middle;

}

public void Serialize(FileStream fs){

BinaryFormatter bf = new BinaryFormatter();

bf.Serialize(fs, this);

fs.Flush();

fs.Close();

}

public void Deserialize(FileStream fs) {

BinaryFormatter bf = new BinaryFormatter();

FullNameClass deserialized = (FullNameClass)bf.Deserialize(fs);

Name = deserialized.Name;

Surname = deserialized.Surname;

MiddleName = deserialized.MiddleName;

fs.Close();

}

public void Print() {

Console.WriteLine("Name={0} Surname={1} Middle={2}", Name, Surname, MiddleName);

}

}

class Program

{

static void Main(string[] args)

{

FileStream fs = new

FileStream("c:\\FullNameSerialize.bin",

FileMode.OpenOrCreate, FileAccess.Write);

FullNameClass fnc = new FullNameClass("Ivan", "Ivanov", "Ivanovich");

fnc.Print();

fnc.Serialize(fs);

fnc = new FullNameClass("Petr", "Petrov", "Petrovich");

fnc.Print();

fs = new

FileStream("c:\\FullNameSerialize.bin",

FileMode.OpenOrCreate, FileAccess.Read);

fnc.Deserialize(fs);

fnc.Print();

}

}

// пример 2

using System;

using System.Text;

using System.IO;

// Add references to Soap and Binary formatters.

using System.Runtime.Serialization.Formatters.Binary;

using System.Runtime.Serialization.Formatters.Soap ;

using System.Runtime.Serialization;

[Serializable]

public class MyItemType : ISerializable

{

public MyItemType()

{

// Empty constructor required to compile.

}

// The value to serialize.

private string myProperty\_value;

public string MyProperty

{

get { return myProperty\_value; }

set { myProperty\_value = value; }

}

// Implement this method to serialize data. The method is called

// on serialization.

public void GetObjectData(SerializationInfo info, StreamingContext context)

{

// Use the AddValue method to specify serialized values.

info.AddValue("props", myProperty\_value, typeof(string));

}

// The special constructor is used to deserialize values.

public MyItemType(SerializationInfo info, StreamingContext context)

{

// Reset the property value using the GetValue method.

myProperty\_value = (string) info.GetValue("props", typeof(string));

}

}

// This is a console application.

public static class Test

{

static void Main()

{

// This is the name of the file holding the data. You can use any file extension you like.

string fileName = "dataStuff.myData";

// Use a BinaryFormatter or SoapFormatter.

IFormatter formatter = new BinaryFormatter();

//IFormatter formatter = new SoapFormatter();

Test.SerializeItem(fileName, formatter); // Serialize an instance of the class.

Test.DeserializeItem(fileName, formatter); // Deserialize the instance.

Console.WriteLine("Done");

Console.ReadLine();

}

public static void SerializeItem(string fileName, IFormatter formatter)

{

// Create an instance of the type and serialize it.

MyItemType t = new MyItemType();

t.MyProperty = "Hello World";

FileStream s = new FileStream(fileName , FileMode.Create);

formatter.Serialize(s, t);

s.Close();

}

public static void DeserializeItem(string fileName, IFormatter formatter)

{

FileStream s = new FileStream(fileName, FileMode.Open);

MyItemType t = (MyItemType)formatter.Deserialize(s);

Console.WriteLine(t.MyProperty);

}

}