🛷**Serialization**

# **Serialization in C#**

Serialization is the process of bringing an object into a form that it can be written on stream. This converted object into a form that can be stored on a file, database, memory, or transfered accross the network.

There is deserialization, is the reverse purpose of serialization. Its getting back the serialized object so it can be loaded into memory.

## **Types of Serialization**

* Binary Serialization
* XML Serialization
* JSON Serialization

## **Step by Step Process for Serialization in C#**

Create the object you want to serialize:  
MyObject obj = new MyObject();

Create a Stream object that points to a file or memory stream that will be used to store the serialized object:  
Stream stream = new FileStream("MyObject.dat", FileMode.Create);

Create a Formatter object, this will be used to serialize the object:  
BinaryFormatter formatter = new BinaryFormatter();

Call the Formatter's Serialize method, passing in the Stream and the object you want to serialize:  
formatter.Serialize(stream, obj);

Close the Stream:  
stream.Close();

## **Step by Step Process for Deserialization in C#**

1. Create a Stream object that points to a file or memory stream that contains the serialized object:  
   Stream stream = new FileStream("MyObject.dat", FileMode.Open);
2. Create a Formatter object, this will be used to deserialize the object:  
   BinaryFormatter formatter = new BinaryFormatter();
3. Call the Formatter's Deserialize method, passing in the Stream and the type of object you want to deserialize:  
   MyObject obj = (MyObject)formatter.Deserialize(stream);
4. Close the Stream:  
   stream.Close();

Example of Serialization

using System;

using System.IO;

using System.Xml.Serialization;

// A class to serialize and deserialize

[Serializable]

public class Tutorial

{

public int ID { get; set; }

public string Name { get; set; }

}

class Program

{

static void Main(string[] args)

{

// Create an object of Tutorial class

Tutorial t1 = new Tutorial();

t1.ID = 1;

t1.Name = ".Net";

// Create a file stream to write the object

FileStream fs = new FileStream("Example.xml", FileMode.Create);

// Create an XML serializer to serialize the object

XmlSerializer xs = new XmlSerializer(typeof(Tutorial));

// Serialize the object and write it to the file stream

xs.Serialize(fs, t1);

// Close the file stream

fs.Close();

// Create another file stream to read the object

FileStream fs2 = new FileStream("Example.xml", FileMode.Open);

// Create another XML serializer to deserialize the object

XmlSerializer xs2 = new XmlSerializer(typeof(Tutorial));

// Deserialize the object and cast it to Tutorial class

Tutorial t2 = (Tutorial)xs2.Deserialize(fs2);

// Close the file stream

fs2.Close();

// Print the properties of the deserialized object

Console.WriteLine("ID: {0}", t2.ID);

Console.WriteLine("Name: {0}", t2.Name);

}

}