**How to: Serialize an Object**

**.NET Framework 4**

To serialize an object, first create the object that is to be serialized and set its public properties and fields. To do this, you must determine the transport format in which the XML stream is to be stored, either as a stream or as a file. For example, if the XML stream must be saved in a permanent form, create a [FileStream](http://msdn.microsoft.com/en-us/library/system.io.filestream.aspx) object.

|  |
| --- |
| **Description: szzyf24s.note(en-us,VS.100).gifNote:** |
| For more examples of XML serialization, see [Examples of XML Serialization](http://msdn.microsoft.com/en-us/library/58a18dwa.aspx). |

**To serialize an object**

1. Create the object and set its public fields and properties.
2. Construct a [XmlSerializer](http://msdn.microsoft.com/en-us/library/system.xml.serialization.xmlserializer.aspx) using the type of the object. For more information, see the **XmlSerializer** class constructors.
3. Call the [Serialize](http://msdn.microsoft.com/en-us/library/system.xml.serialization.xmlserializer.serialize.aspx) method to generate either an XML stream or a file representation of the object's public properties and fields. The following example creates a file.

**VB**

Dim myObject As MySerializableClass = New MySerializableClass()

' Insert code to set properties and fields of the object.

Dim mySerializer As XmlSerializer = New XmlSerializer(GetType(MySerializableClass))

' To write to a file, create a StreamWriter object.

Dim myWriter As StreamWriter = New StreamWriter("myFileName.xml")

mySerializer.Serialize(myWriter, myObject)

myWriter.Close()

**C#**

MySerializableClass myObject = new MySerializableClass();

// Insert code to set properties and fields of the object.

XmlSerializer mySerializer = new

XmlSerializer(typeof(MySerializableClass));

// To write to a file, create a StreamWriter object.

StreamWriter myWriter = new StreamWriter("myFileName.xml");

mySerializer.Serialize(myWriter, myObject);

myWriter.Close();

**How to: Deserialize an Object**

**.NET Framework 4**

When you deserialize an object, the transport format determines whether you will create a stream or file object. After the transport format is determined, you can call the [Serialize](http://msdn.microsoft.com/en-us/library/system.xml.serialization.xmlserializer.serialize.aspx) or [Deserialize](http://msdn.microsoft.com/en-us/library/system.xml.serialization.xmlserializer.deserialize.aspx) methods, as required.

**To deserialize an object**

1. Construct a [XmlSerializer](http://msdn.microsoft.com/en-us/library/system.xml.serialization.xmlserializer.aspx) using the type of the object to deserialize.
2. Call the **Deserialize** method to produce a replica of the object. When deserializing, you must cast the returned object to the type of the original, as shown in the following example, which deserializes the object into a file (although it could also be deserialized into a stream).

**VB**

Dim myObject As MySerializableClass

' Construct an instance of the XmlSerializer with the type

' of object that is being deserialized.

Dim mySerializer As XmlSerializer = New XmlSerializer(GetType(MySerializableClass))

' To read the file, create a FileStream.

Dim myFileStream As FileStream = \_

New FileStream("myFileName.xml", FileMode.Open)

' Call the Deserialize method and cast to the object type.

myObject = CType( \_

mySerializer.Deserialize(myFileStream), MySerializableClass)

**C#**

MySerializableClass myObject;

// Construct an instance of the XmlSerializer with the type

// of object that is being deserialized.

XmlSerializer mySerializer =

new XmlSerializer(typeof(MySerializableClass));

// To read the file, create a FileStream.

FileStream myFileStream =

new FileStream("myFileName.xml", FileMode.Open);

// Call the Deserialize method and cast to the object type.

myObject = (MySerializableClass)

mySerializer.Deserialize(myFileStream)