Serialization is the process of converting an object into a stream of bytes in order to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate it when needed. The reverse process is called deserialization.

For examples of serialization, see [Related Topics and Examples](http://msdn.microsoft.com/en-IN/library/ms233843.aspx#bkmk_relatedtopics) later in this topic.

[How Serialization Works](javascript:void(0))

This illustration shows the overall process of serialization.



The object is serialized to a stream, which carries not just the data, but information about the object's type, such as its version, culture, and assembly name. From that stream, it can be stored in a database, a file, or memory.

Serialization allows the developer to save the state of an object and recreate it as needed, providing storage of objects as well as data exchange. Through serialization, a developer can perform actions like sending the object to a remote application by means of a Web Service, passing an object from one domain to another, passing an object through a firewall as an XML string, or maintaining security or user-specific information across applications.

### [Making an Object Serializable](javascript:void(0))

To serialize an object, you need the object to be serialized, a stream to contain the serialized object, and a [Formatter](http://msdn.microsoft.com/en-IN/library/system.runtime.serialization.formatter.aspx). [System.Runtime.Serialization](http://msdn.microsoft.com/en-IN/library/system.runtime.serialization.aspx) contains the classes necessary for serializing and deserializing objects.

Apply the [SerializableAttribute](http://msdn.microsoft.com/en-IN/library/system.serializableattribute.aspx) attribute to a type to indicate that instances of this type can be serialized. A [SerializationException](http://msdn.microsoft.com/en-IN/library/system.runtime.serialization.serializationexception.aspx) exception is thrown if you attempt to serialize but the type does not have the [SerializableAttribute](http://msdn.microsoft.com/en-IN/library/system.serializableattribute.aspx) attribute.

If you do not want a field within your class to be serializable, apply the [NonSerializedAttribute](http://msdn.microsoft.com/en-IN/library/system.nonserializedattribute.aspx) attribute. If a field of a serializable type contains a pointer, a handle, or some other data structure that is specific to a particular environment, and the field cannot be meaningfully reconstituted in a different environment, then you may want to make it nonserializable.

If a serialized class contains references to objects of other classes that are marked [SerializableAttribute](http://msdn.microsoft.com/en-IN/library/system.serializableattribute.aspx), those objects will also be serialized.

* SOAP stands for Simple Object Access Protocol
* SOAP is a communication protocol
* SOAP is for communication between applications
* SOAP is a format for sending messages
* SOAP communicates via Internet
* SOAP is platform independent
* SOAP is language independent
* SOAP is based on XML
* SOAP is simple and extensible
* SOAP allows you to get around firewalls
* SOAP is a W3C recommendation