

# XML Serialization

Here is an example based on the [Advanced Tutorial, Learning Boo Kung-Fu](#).

Note, you have to compile this to an exe and then run it. It won't work in booi, because types in dynamically generated assemblies can't be serialized apparently.

```
import System.Xml from System.Xml
import System.Drawing from System.Drawing
import System.ComponentModel
import System.Xml.Serialization
import System.IO
import System

class Ninja:
    [XmlAttribute("name")]
    public Name as string
    [XmlAttribute("style")]
    public Style as string
    [XmlAttribute("speed")]
    public Speed as int
    [XmlAttribute("strength")]
    public Strength as int
    [XmlAttribute("stamina")]
    public Stamina as int

    //problem serializing Colors:
    http://dotnetified.com/PermaLink.aspx?guid=E86B447E-AA95-49B6-909F-CDA36ACF481F
    _color as Color
    [XmlIgnore] //don't serialize this, we'll using StringColor instead
    Color as Color:
        get:
            return _color
        set:
            _color = value

    [XmlAttribute("color")]
    [Browsable(false)] //so this property isn't used in an IDE
    public StringColor as string:
        get:
            return ColorTranslator.ToHtml(_color)
        set:
            _color = ColorTranslator.FromHtml(value)

    def ToString():
        return "You see $Name, a ninja of the $Style with the stats
        $Strength/$Stamina/$Speed. He is $Color."

//SERIALIZING FROM AN OBJECT TO XML:

ninja = Ninja(Name:"Hiyo", Style:"Drunken Monkey",
              Speed:77, Strength:88, Stamina:99,
              StringColor:"#880000")

print ninja
```

```
s = XmlSerializer(typeof(Ninja))
//convert to xml, print it out:
print "\nSERIALIZED EXAMPLE:"
s.Serialize(System.Console.Out, ninja)
print

//to serialize to a file:
//using out = StreamWriter(filename):
// s.Serialize(out, ninja)

//to serialize to a string variable:
//s as string
//using out = StringWriter(s):
// s.Serialize(out, ninja)

//DESERIALIZING FROM XML (with earlier example xml)
xml = """
<Ninja color="#FF00FF" name="John Kho Kawn" style="Crazy Martial Arts" strength="71"
speed = "74" stamina = "65" />
"""

s = XmlSerializer(typeof(Ninja))
reader = StringReader( xml )
newninja as Ninja = s.Deserialize( reader )
reader.Close()
```

```
print "\nDESERIALIZED EXAMPLE:"
print newninja
```

For more info on XML Serialization in .NET and Mono, see these tutorials:

- [Introducing XML Serialization](#)
- [XML Serialization in C#](#)
- [XML Serialization in the .NET Framework](#)
- [How to Serialize a Hashtable \(and a Date\)](#).  
(you can serialize a hashtable using binary instead of xml, see below)

## Non-XML (Binary) Serialization

This is useful when you really just want to store and retrieve an object like a dictionary/hashtable, and you don't need to use XML.

```
import System.IO
import System.Runtime.Serialization.Formatters.Binary

def savetofile(obj, filename as string):
    using stream = FileStream(filename, FileMode.OpenOrCreate ,FileAccess.Write):
        savetostream(obj, stream)

def savetomemory(obj) as (byte):
    using stream = MemoryStream():
        savetostream(obj, stream)
    return stream.GetBuffer()

def savetostream(obj, str as Stream): //see also CryptoStream
    BinaryFormatter().Serialize(str, obj)

def loadfromfile(filename as string) as object:
    using stream = FileStream(filename, FileMode.Open):
        return loadfromstream(stream)

def loadfrommemory(buffer as (byte)) as object:
    using stream = MemoryStream(buffer):
        return loadfromstream(stream)

def loadfromstream(str as Stream) as object:
    return BinaryFormatter().Deserialize(str)

d = {"one": "item", "two": "another item"}

print "memory example..."
saved = savetomemory(d)
d2 as Hash = loadfrommemory(saved)
for key in d2.Keys:
    print key, ":", d2[key]

print "\nfile example..."
filename = "saved-dict.dat"
savetofile(d, filename)
d3 as Hash = loadfromfile(filename)
for key in d3.Keys:
    print key, ":", d3[key]
```

See also:

- [Encrypt a String](#) for ideas about how you might serialize to a CryptoStream
- [Encrypting data in network connections](#)