



Denne guide er oprindeligt udgivet på Eksperten.dk

## Brug af XML i VB.NET

**Denne artikel vil vise lidt om hvordan man kan bruge XML i VB.NET.**

**Den forudsætter kendskab til XML og VB.NET men ikke til brug af XML i VB.NET.**

**Der er en anden artikel med præcis samme indhold bare i C# !**

Skrevet den **15. Feb 2010** af **arne\_v** | kategorien **Programmering / Visual Basic .NET** | ★★★★★

Historie:

V1.0 - 05/11/2004 - original

V1.1 - 25/12/2008 - lidt småjusteringer og link til efterfølgeren

V1.2 - 14/02/2010 - smårettelser

### Indledning

.NET har en glimrende support for XML i namespace System.Xml og under namespaces.

Denne artikel vil vise nogle af mulighederne indenfor basal XML brug.

Jeg vil ikke forklare baggrund bag XML, XPath og XSLT. Det må man finde et andet sted.

Artiklen <http://www.eksperten.dk/artikler/627> "XML hvad, hvorfor og hvornår ?" giver en oversigt over nogle af begreberne.

Eksemplerne vil være console applikationer, men selve XML klasserne og metoderne kan lige så godt bruges i GUI applikationer eller web applikationer.

De fleste af kode eksemplerne vil bruge denne simple XML fil.

test.xml

```
<?xml version='1.0' standalone='yes'?>
<biler>
  <fabrikat>
    <navn>Volvo</navn>
    <land>Sverige</land>
  </fabrikat>
  <fabrikat>
    <navn>Opel</navn>
    <land>Tyskland</land>
  </fabrikat>
  <fabrikat>
    <navn>Toyota</navn>
  </fabrikat>
</biler>
```

```
        <land>Japan</land>
    </fabrikat>
    <fabrikat>
        <navn>Fiat</navn>
        <land>Italien</land>
    </fabrikat>
</biler>
```

## Læse XML fra fil

Det er meget nemt at læse et XML dokument ind fra en fil.

Man laver et nyt tomt XmlDocument og bruger Load metoden.

```
Dim doc As XmlDocument = New XmlDocument
doc.Load("C:\test.xml")
```

## Løbe gennem XML

Man kan finde alle tags med et bestemt navn med GetElementsByTagName metoden som returnerer en collection af noder.

```
Dim fabrikater As XmlNodeList = doc.GetElementsByTagName("fabrikat")
For Each fabrikat As XmlNode In fabrikater
    Dim navn As String = fabrikat.ChildNodes(0).FirstChild.Value
    Dim land As String = fabrikat.ChildNodes(1).FirstChild.Value
    Console.WriteLine(navn & " " & land)
Next
```

## Manipulere XML

Man kan også udvælge noder med SelectNode metoden og et XPath udtryk.

XPath udtryk udvælger noder ved at angive stien til dem i en speciel syntax:

```
xxxx - finder elementer med navn xxxx
//xxxx/yyyy - finder elementer med navn yyyy under elementer med navn xxxx
xxxx[yyyy='abc'] - finder elementer med navn xxxx som har et under element
                med navn yyyy og en tekstværdi 'abc'
xxxx[@yyyy=123] - finder elementer med navn xxxx som har en attribut med navn
                yyyy og en talværdi 123
```

```
Dim volvo As XmlNodeList =
doc.DocumentElement.SelectNodes("fabrikat[navn='Volvo']")
```

```
volvo(0).ChildNodes(1).FirstChild.Value = "USA"  
Dim opel As XmlNodeList =  
doc.DocumentElement.SelectNodes("fabrikat[navn='Opel']")  
opel(0).ChildNodes(1).FirstChild.Value = "USA"
```

Man kan tilføje noder til træet.

```
Dim hyundainavn As XmlNode = doc.CreateElement("navn")  
hyundainavn.AppendChild(doc.CreateTextNode("Hyundai"))  
Dim hyundailand As XmlNode = doc.CreateElement("land")  
hyundailand.AppendChild(doc.CreateTextNode("Syd Korea"))  
Dim hyundai As XmlNode = doc.CreateElement("fabrikat")  
hyundai.AppendChild(hyundainavn)  
hyundai.AppendChild(hyundailand)  
doc.DocumentElement.AppendChild(hyundai)
```

Man kan også fjerne noder fra træet med `element.RemoveChild(node)` !

### Skrive XML til fil

Det er meget nemt at skrive et XML dokument til en fil.

Man kalder bare `Save` metoden.

```
doc.Save("C:\test1.xml")
```

### Opsamling

Vi kombinerer nu alle kode fragmenterne ovenfor til et helt program.

```
Imports System  
Imports System.Xml  
  
Class MainClass  
    Public Shared Sub Main(ByVal args As String())  
        ' Læse XML fra fil  
        Dim doc As XmlDocument = New XmlDocument  
        doc.Load("C:\test.xml")  
        ' Løbe gennem XML  
        Dim fabrikater As XmlNodeList = doc.GetElementsByTagName("fabrikat")  
        For Each fabrikat As XmlNode In fabrikater  
            Dim navn As String = fabrikat.ChildNodes(0).FirstChild.Value  
            Dim land As String = fabrikat.ChildNodes(1).FirstChild.Value
```

```

        Console.WriteLine(navn & " " & land)
    Next
    ' Manipulere XML
    ' 1. sæt Volvo og Opel til USA
    Dim volvo As XmlNodeList =
doc.DocumentElement.SelectNodes("fabrikat[navn='Volvo']")
    volvo(0).ChildNodes(1).FirstChild.Value = "USA"
    Dim opel As XmlNodeList =
doc.DocumentElement.SelectNodes("fabrikat[navn='Opel']")
    opel(0).ChildNodes(1).FirstChild.Value = "USA"
    ' 2. tilføj Hyundai
    Dim hyundainavn As XmlNode = doc.CreateElement("navn")
    hyundainavn.AppendChild(doc.CreateTextNode("Hyundai"))
    Dim hyundailand As XmlNode = doc.CreateElement("land")
    hyundailand.AppendChild(doc.CreateTextNode("Syd Korea"))
    Dim hyundai As XmlNode = doc.CreateElement("fabrikat")
    hyundai.AppendChild(hyundainavn)
    hyundai.AppendChild(hyundailand)
    doc.DocumentElement.AppendChild(hyundai)
    ' Skrive XML til fil
    doc.Save("C:\test1.xml")
End Sub
End Class

```

test1.xml kommer til at se ud som:

```

<?xml version="1.0" standalone="yes"?>
<biler>
  <fabrikat>
    <navn>Volvo</navn>
    <land>USA</land>
  </fabrikat>
  <fabrikat>
    <navn>Opel</navn>
    <land>USA</land>
  </fabrikat>
  <fabrikat>
    <navn>Toyota</navn>
    <land>Japan</land>
  </fabrikat>
  <fabrikat>
    <navn>Fiat</navn>
    <land>Italien</land>
  </fabrikat>
  <fabrikat>
    <navn>Hyundai</navn>
    <land>Syd Korea</land>
  </fabrikat>
</biler>

```

## Skrive XML til fil på en anden måde

Man kan naturligvis lave et helt nyt XML dokument på samme måde som der ovenfor tilføjes elementer.

```
Imports System
Imports System.Xml

Class MainClass
    Public Shared Sub Main(ByVal args As String())
        Dim doc As XmlDocument = New XmlDocument
        Dim root As XmlNode = doc.CreateElement("all")
        doc.AppendChild(root)
        Dim i As Integer
        For i = 1 To 5
            Dim elm As XmlNode = doc.CreateElement("one")
            elm.AppendChild(doc.CreateTextNode("Element #" & i))
            root.AppendChild(elm)
        Next
        doc.InsertBefore(doc.CreateXmlDeclaration("1.0", "UTF-8", "yes"),
doc.DocumentElement)
        doc.Save("C:\test2.xml")
    End Sub
End Class
```

test2.xml kommer til at se ud som:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<all>
  <one>Element #1</one>
  <one>Element #2</one>
  <one>Element #3</one>
  <one>Element #4</one>
  <one>Element #5</one>
</all>
```

Men der findes en alternativ måde som nogen gange er mere hensigtsmæssig.

```
Imports System
Imports System.IO
Imports System.Text
Imports System.Xml

Class MainClass
    Public Shared Sub Main(ByVal args As String())
```

```

    Dim xtw As XmlTextWriter = New XmlTextWriter("C:\test3.xml",
Encoding.UTF8)
    xtw.Formatting = Formatting.Indented
    xtw.WriteStartDocument
    xtw.WriteStartElement("all")
    Dim i As Integer
    For i = 1 To 5
        xtw.WriteStartElement("one")
        xtw.WriteString("Element #" & i)
        xtw.WriteEndElement
    Next
    xtw.WriteEndElement
    xtw.WriteEndDocument
    xtw.Close
End Sub
End Class

```

test3.xml kommer til at se ud som:

```

<?xml version="1.0" encoding="utf-8"?>
<all>
  <one>Element #1</one>
  <one>Element #2</one>
  <one>Element #3</one>
  <one>Element #4</one>
  <one>Element #5</one>
</all>

```

## Bruge XSLT

Hvis man gerne vil manipulere sit XML dokument er det også en mulighed at bruge XSLT.

```

Imports System
Imports System.IO
Imports System.Xml
Imports System.Xml.Xsl

Class MainClass
    Public Shared Sub Main(ByVal args As String())
        Dim doc As XmlDocument = New XmlDocument
        doc.Load("C:\test.xml")
        Dim xslt As XslTransform = New XslTransform
        xslt.Load("C:\test.xsl")
        Dim wrt As XmlTextWriter = New XmlTextWriter(New
StreamWriter("C:\test4.xml"))
        wrt.Formatting = Formatting.Indented
        xslt.Transform(doc, Nothing, wrt, Nothing)
    End Sub
End Class

```

```
End Sub
End Class
```

test.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

<xsl:template match="biler">
<biler>
<xsl:apply-templates/>
</biler>
</xsl:template>

<xsl:template match="fabrikat">
<fabrikat>
<xsl:attribute name="land"><xsl:value-of select="land"/></xsl:attribute>
<xsl:value-of select="navn"/>
</fabrikat>
</xsl:template>

</xsl:stylesheet>
```

test4.xml kommer til at se ud som:

```
<biler>
  <fabrikat land="Sverige">Volvo</fabrikat>
  <fabrikat land="Tyskland">Opel</fabrikat>
  <fabrikat land="Japan">Toyota</fabrikat>
  <fabrikat land="Italien">Fiat</fabrikat>
</biler>
```

## Nyt i .NET 2.0 og 3.5

<http://www.eksperten.dk/artikler/1260> beskriver:

- hurtige readonly XPathDocument
- XsltTransform -> XsltCompiledTransform i .NET 2.0
- LINQ for XML i .NET 3.5

**Kommentar af martinm d. 11. Nov 2004 | 1**

**Kommentar af adhn d. 24. Oct 2008 | 2**

### **Kommentar af basementjack d. 04. Dec 2004 | 3**

God for nybegyndere.. :)

### **Kommentar af simonth d. 05. Jul 2007 | 4**

### **Kommentar af groyk d. 05. Dec 2008 | 5**

God artikel, meget brugbar.

Men hvordan skriver jeg et tag således?

```
<tag id="xx">  
Andet indhold  
</tag>
```

### **Kommentar af jesperwerge d. 17. Nov 2006 | 6**

Alltidens artikel, tak - Håber på en opfølger med redigering af elementer i en XML fil.

### **Kommentar af baxos d. 27. Nov 2006 | 7**

Fed Artikel!

### **Kommentar af jaffafo d. 16. Oct 2005 | 8**

Udmærket artikel

### **Kommentar af nicklasw d. 05. May 2006 | 9**

rigtig god artikel for nybegyndere

### **Kommentar af horsdal d. 09. Nov 2004 | 10**

var lige noget jeg kunne bruge... tak :)