



Denne guide er oprindeligt
udgivet på Eksperten.dk

XML i ASP

Denne artikel gennemgår kort nogle mulighederne for brug af XML i ASP. Det sker ved brug af eksempler. Eksemplerne vil være i VBS.

Den forudsætter et vist kendskab til både XML og ASP.

Skrevet den **14. Feb 2010** af **arne_v** | kategorien **Programmering / ASP** | ★★☆☆☆☆

Historie:

V1.0 - 29/12/2008 - original

V1.1 - 14/02/2010 - smårettelser

Indledning

Microsoft har udgivet en ActiveX/COM komponent som implementerer W3C XML DOM.

Med W3C XML DOM læses hele filen ind i et hierakisk træ, hvor elementer/attributter/text er children til et element.

Vær opmærksom på at et DOM træ fylder mere i memory end XML filen. DOM er derfor ikke en god løsning til meget store XML filer (3 cifrede MB eller GB størrelser).

Metoderne til at tilgå dette træ er standardiserede, så man kan genbruge sin viden fra andre sprog såsom Java, .NET, PHP etc..

For dokumentation se:

<http://msdn.microsoft.com/en-us/library/ms764730.aspx>

For hurtig intro til XML terminologi se:

<http://www.eksperten.dk/guide/627> "XML hvad, hvorfor og hvornår ?

Eksemplerne vil bruge følgende simple XML fil som input:

test.xml

```
<?xml version='1.0' standalone='yes'?>
<medlemmer>
  <medlem no="1">
    <navn>Niels Nielsen</navn>
    <adresse>Nellikevej 19</adresse>
  </medlem>
  <medlem no="2">
    <navn>Jens Jensen</navn>
    <adresse>Jagtvej 17</adresse>
  </medlem>
  <medlem no="3">
    <navn>Ole Olsen</navn>
    <adresse>Omfartsvejen 13</adresse>
  </medlem>
</medlemmer>
```

```
</medlem>
</medlemmer>
```

Den genererede HTML i eksemplerne er ikke specielt avanceret, men formålet er at vise XML processing ikke HTML generering (og så er jeg iøvrigt ikke specielt HTML kyndig).

Læse XML dokument

Først et eksempel der viser hvordan man kan løbe igennem et XML dokument.

showvbs.asp

```
<%@ Language="VBScript" %>
<html>
<head>
<title>Show i VBS demo</title>
</head>
<body>
<table border="1">
<tr>
<th>No</th>
<th>Navn</th>
<th>Adresse</th>
</tr>
<%
' opret DOM træ
Set doc = CreateObject("MSXML.DOMDocument")
' indlæs XML fil i DOM træ
doc.Async = False
doc.Load(Server.MapPath("test.xml"))
' find alle medlem elementer
Set medlemmer = doc.GetElementsByTagName("medlem")
For Each medlem in medlemmer
    Response.Write "<tr>" & vbCrLf
    ' find og udskriv attribut no
    Response.Write "<td>" & medlem.Attributes.GetNamedItem("no").Value &
"</td>" & vbCrLf
    ' find alle under elementer
    For Each elm in medlem.ChildNodes
        ' find under element navn
        If(elm.NodeName = "navn") Then navn = elm.FirstChild.NodeValue
        ' find under element adresse
        If(elm.NodeName = "adresse") Then adresse = elm.FirstChild.NodeValue
    Next
    ' udskriv de fundne under elementer
    Response.Write "<td>" & navn & "</td>" & vbCrLf
    Response.Write "<td>" & adresse & "</td>" & vbCrLf
    Response.Write "</tr>" & vbCrLf
Next
' cleanup
```

```
Set medlemmer = Nothing
Set doc = Nothing
%>
</table>
</body>
</html>
```

Søge i XML dokument med XPath

Så et eksempel der viser, hvordan man kan søge i en XML fil med XPath.

Jeg vil ikke gå i detaljer med hensyn til XPath syntax det kan man og det er der skrevet bøger om.

Den ultra korte version er:

- 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

search.asp

```
<%@ Language="VBScript" %>
<html>
<head>
<title>Search via XPath i VBS demo</title>
</head>
<body>
<%
If Request.Form("no") <> "" Then
    no = CInt(Request.Form("no"))
    ' opret DOM træ
    Set doc = CreateObject("MSXML.DOMDocument")
    ' indlæs XML fil i DOM træ
    doc.Async = False
    doc.Load(Server.MapPath("test.xml"))
    Set medlemmer = doc.SelectNodes("//medlemmer/medlem[@no=" & no & "]")
    ' test om der var nogen matchende medlemmer
    If medlemmer.Length > 0 Then
        ' tag det første matchende medlem
        Set medlem = medlemmer(0)
        ' udskriv attribut no
        Response.Write "no=" & medlem.Attributes.GetNamedItem("no").NodeValue
        & "<br/>" & vbCrLf
        ' søg efter sub element navn og udskriv det
        Response.Write "navn=" &
        medlem.SelectSingleNode("navn/text()").NodeValue & "<br/>" & vbCrLf
        ' søg efter sub element adresse og udskriv det
        Response.Write "adresse=" &
```

```

medlem.SelectSingleNode("adresse/text()").NodeValue & "<br/>" & vbCrLf
    ' cleanup
    Set medlem = Nothing
Else
    Response.Write "Ingen fundet<br/>" & vbCrLf
End If
' cleanup
Set medlemmer = Nothing
Set doc = Nothing
End If
%>
<form method="POST">
No: <input type="TEXT" name="no"/>
<br/>
<input type="SUBMIT" value="Søg"/>
</form>
</body>
</html>

```

Modificere XML dokument

Eksempel på at rette i XML dokument (og returnere XML fremfor HTML).

change.asp

```

<%@ Language="VBScript" %>
<%
Response.ContentType = "text/xml"
' opret DOM træ
Set doc = CreateObject("MSXML.DOMDocument")
' indlæs XML fil i DOM træ
doc.Async = False
doc.Load(Server.MapPath("test.xml"))
' søg efter medlem med no 2
Set medlem = doc.SelectSingleNode("//medlemmer/medlem[@no=2]")
' fjern medlem
medlem.ParentNode.RemoveChild(medlem)
' tilføj medlem
Set navn = doc.CreateElement("navn")
navn.AppendChild(doc.CreateTextNode("Lars Larsen"))
Set adresse = doc.CreateElement("adresse")
adresse.AppendChild(doc.CreateTextNode("Ledvej 14"))
Set nytmedlem = doc.CreateElement("medlem")
call nytmedlem.SetAttribute("no", "4")
nytmedlem.AppendChild(navn)
nytmedlem.AppendChild(adresse)
doc.DocumentElement.AppendChild(nytmedlem)
' udskriv DOM træ som XML
doc.Save(Response)
' cleanup
Set navn = Nothing

```

```
Set adresse = Nothing
Set nytmedlem = Nothing
Set medlem = Nothing
Set doc = Nothing
%>
```

Oprette XML dokument

Eksempel på at oprette et nyt XML dokument (og returnere XML fremfor HTML).

create.asp

```
<%@ Language="VBScript" %>
<%
Response.ContentType = "text/xml"
' opret DOM træ
Set doc = CreateObject("MSXML.DOMDocument")
' opret document element og 3 sub elementer
Set one1 = doc.CreateElement("one")
one1.AppendChild(doc.CreateTextNode("A"))
Set one2 = doc.CreateElement("one")
one2.AppendChild(doc.CreateTextNode("BB"))
Set one3 = doc.CreateElement("one")
one3.AppendChild(doc.CreateTextNode("CCC"))
Set root = doc.CreateElement("all")
root.AppendChild(one1)
root.AppendChild(one2)
root.AppendChild(one3)
doc.AppendChild(root)
' udskriv DOM træ som XML
doc.Save(Response)
' cleanup
Set one1 = Nothing
Set one2 = Nothing
Set one3 = Nothing
Set root = Nothing
Set doc = Nothing
%>
```

DOM er ikke særligt velegnet til at create XML dokumenter med, så man kunne overveje et simplere alternativ.

createalt.asp

```
<%@ Language="VBScript" %>
<%
Response.ContentType = "text/xml"
' udskriv XML
```

```

Response.Write "<all>"
Response.Write "    <one>A</one>"
Response.Write "    <one>BB</one>"
Response.Write "    <one>CCC</one>"
Response.Write "</all>"
%>

```

XSLT

XSLT er en måde at transformere XML på.

Det kan laves client side. Men det kan også laves server side.

Laver man det server side, så kan man outputte HTML ud fra XML uden at kode presentationen i ASP.

Eksempel.

xslt.asp

```

<html>
<head>
<title>XSL i VBS demo</title>
</head>
<body>
<%
' opret XML og indlæs
Set xml = CreateObject("MSXML.DOMDocument")
xml.Async = False
xml.Load(Server.MapPath("test.xml"))
' opret XSL og indlæs
Set xsl = CreateObject("MSXML.DOMDocument")
xsl.Async = False
xsl.Load(Server.MapPath("test.xsl"))
' udskriv XML transformeret til HTML
Response.Write xml.TransformNode(xsl)
%>
</body>
</html>

```

test.xsl

```

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

<xsl:template match="/">
<table border="1">

```

```

<tr>
<th>No</th>
<th>Navn</th>
<th>Adresse</th>
</tr>
<xsl:apply-templates/>
</table>
</xsl:template>

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

<xsl:template match="medlem">
<tr>
<td><xsl:value-of select="@no"/></td>
<td><xsl:value-of select="navn"/></td>
<td><xsl:value-of select="adresse"/></td>
</tr>
</xsl:template>

</xsl:stylesheet>

```

XSLT er en særdels kraftfuld teknologi. Men vær også opmærksom på, at det er et helt nyt sprog som man skal sætte sig ind i. Og det er et svært sprog - sværere end ASP & VBS !

Kommentar af xcover d. 15. Jan 2009 | 1

Super artikel, men syntes jeg mangler lidt info omkring hvordan jeg henter et XML dokument på fx. dr.dk og udskriver via min egen ASP - så vidt jeg ved kan jeg ikke det med doc.Load(Server.MapPath("test.xml"))

Eller er jeg forkert på den ?