

Manipuler des fichiers XML en VBScript avec XPath

par Baptiste Wicht ([home](#))

Date de publication : Le 19 Décembre 2007

Cet article va vous apprendre à manipuler des fichiers XML en Visual Basic Script avec XPath.

- 1 - Introduction
- 2 - Lecture de fichiers XML
- 3 - Ecriture de fichier XML
 - 3.1 - Indenter le fichier XML
- 4 - Modification
- 5 - Conclusion
 - 5.1 - Remerciements
 - 5.2 - Code complet
 - 5.2.1 - Lecture
 - 5.2.2 - Ecriture
 - 5.2.3 - Modification

1 - Introduction

Dans cet article, nous allons commencer par apprendre à lire un fichier XML existant et à en extraire les informations dont on a besoin. Ensuite, nous allons nous pencher sur la création de documents XML. Et enfin, nous allons voir comment faire pour modifier un fichier XML existant.

Pour cela nous allons utiliser l'api DOM de Microsoft (**Microsoft DOM**). Il s'agit d'une implémentation de la recommandation définie par le W3C qui permet d'accéder au contenu d'un document et de le modifier. Le DOM est surtout utilisé pour modifier des documents XML ou pour accéder au contenu de pages web.

Pour le parcours du document DOM, nous allons utiliser XPath. XPath est une syntaxe permettant de désigner une portion précise d'un fichier XML.

2 - Lecture de fichiers XML

Voici le document XML sur lequel nous allons travailler :

```
personnes.xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<personnes>
  <personne age="49">
    <nom>Baud</nom>
    <prenom>Georges</prenom>
    <etat>Marié</etat>
    <enfants>
      <enfant>
        <nom>Tiop</nom>
        <prenom>Elisabeth</prenom>
      </enfant>
    </enfants>
  </personne>
  <personne age="22">
    <nom>Trinzka</nom>
    <prenom>Judith</prenom>
    <etat>Célibataire</etat>
  </personne>
  <personne age="88">
    <nom>Godoh</nom>
    <prenom>Madeleine</prenom>
    <etat>Veuve</etat>
    <enfants>
      <enfant>
        <nom>Godoh</nom>
        <prenom>Jean-Marie</prenom>
      </enfant>
      <enfant>
        <nom>Godoh</nom>
        <prenom>Etienne</prenom>
      </enfant>
      <enfant>
        <nom>Swoti</nom>
        <prenom>Julienne</prenom>
      </enfant>
    </enfants>
  </personne>
</personnes>
```

C'est donc simplement une liste de personnes. Chaque personne a un âge, un état, un nom et un prénom et peut éventuellement avoir des enfants.

On va maintenant passer à la lecture du fichier XML. Pour commencer, il nous faut utiliser un parseur. Pour cela, nous allons utiliser le parseur XMLDOM de Microsoft.

La première chose à faire dans notre code est donc d'initialiser un parseur :

```
Set xmlDoc = CreateObject("Microsoft.XMLDOM")
```

Et tout à la fin de notre script, il ne faut pas oublier de détruire notre objet :

```
Set xmlDoc = Nothing
```

Il nous faut maintenant ouvrir notre fichier. On va également indiquer au parseur de charger tout le fichier en mémoire avant de commencer à le parser :

```
Set xmlDoc = CreateObject("Microsoft.XMLDOM")
xmlDoc.Async = "false"
xmlDoc.Load("personnes.xml")
```

On a donc juste passé le parseur en mode synchrone pour qu'il charge tout le fichier en mémoire avant de le traiter et ensuite on a ouvert le fichier avec la méthode Load.

Pour l'exemple, on va juste créer un petit programme qui va lire le document XML et afficher les personnes dans une boîte de dialogue.

On va maintenant récupérer toutes les personnes contenues dans le fichier XML. Pour cela, on va utiliser la méthode selectNodes à laquelle on passe une requête XPath. La requête XPath est simple, on va rechercher tous les éléments personne dans la balises personnes :

```
'On récupère tous les noeuds personnes
'à l'intérieur d'un noeud personnes
For Each personneElement In xmlDoc.selectNodes("/personnes/personne")
Next
```

On a donc bouclé sur tous les éléments personnes que l'on a trouvés dans l'élément racine.

On va maintenant récupérer les informations principales (nom, prénom, état) et les afficher avec la méthode MsgBox :

```
'On récupère les informations sur la personne
nom = personneElement.selectSingleNode("nom").text
prenom = personneElement.selectSingleNode("prenom").text
etat = personneElement.selectSingleNode("etat").text

MsgBox "Nom : " & nom & vbCrLf & _
      "Prénom : " & prenom & vbCrLf & _
      "Etat civil : " & etat
```

Ce qui devrait vous afficher toutes les personnes dans des boites de dialogue. Pour récupérer le contenu d'un élément, on a utilisé la méthode selectSingleNode sur le noeud XML qui permet de récupérer un noeud unique depuis le noeud courant et on a ensuite récupérer la valeur du noeud avec l'attribut text.

On va maintenant récupérer l'âge de la personne et ajouter ça dans notre fenêtre :

```
'On récupère les informations sur la personne
nom = personneElement.selectSingleNode("nom").text
prenom = personneElement.selectSingleNode("prenom").text
etat = personneElement.selectSingleNode("etat").text
age = personneElement.getAttribute("age")

MsgBox "Nom : " & nom & vbCrLf & _
      "Prénom : " & prenom & vbCrLf & _
      "Etat civil : " & etat & vbCrLf & _
      "Age : " & age
```

Cette fois, c'est donc la méthode getAttribute que l'on a utilisé sur notre noeud. Enfin, on va maintenant voir comment faire pour récupérer les enfants d'une personne :

```
'On récupère tous les noeuds enfant à l'intérieur d'une balise enfants dans le noeud personne
courant
Set enfants = personneElement.selectNodes("enfants/enfant")

'S'il y a des enfants
```

```
If enfants.length > 0 Then
text = prenom & " " & nom & " a des enfants : "

'On boucle sur tous les enfants
For Each enfantElement In enfants
nomEnfant = enfantElement.selectSingleNode("nom").text
prenomEnfant = enfantElement.selectSingleNode("prenom").text

text = text & vbcrLf & " - " & prenomEnfant & " " & nomEnfant
Next

MsgBox text
Else
MsgBox prenom & " " & nom & " n'a pas d'enfants"
End If
```

On a donc commencé par récupérer tous les éléments enfant de la personne et s'il y en a, on les a parcourus pour les ajouter et pour enfin, les afficher. S'il n'y en a pas, on affiche tout simplement un message d'erreur.

Voilà. On a donc réussi à parser notre fichier XML et à en extraire les informations nécessaires. La lecture de fichier repose toujours sur le même principe. On récupère des noeuds, leurs enfants, leur contenu et leurs attributs.

Maintenant que l'on a vu comment lire un fichier XML, on va voir comment est-ce qu'on peut en écrire un.

3 - Ecriture de fichier XML

On va maintenant faire le travail inverse de l'exercice précédent, on va partir des données contenues dans notre programme VBS pour écrire un fichier personnes2.xml.

Voilà la forme sous laquelle on a nos données :

```
personnes = array()  
Redim personnes(1)  
  
georges = array()  
Redim georges(4)  
  
georges(0) = "Baud"  
georges(1) = "Georges"  
georges(2) = "Marié"  
georges(3) = 49  
  
enfants = array()  
Redim enfants(0)  
  
elisabeth = array()  
Redim elisabeth(1)  
  
elisabeth(0) = "Tiop"  
elisabeth(1) = "Elisabeth"  
  
enfants(0) = elisabeth  
  
georges(4) = enfants  
  
personnes(0) = georges  
  
judith = array()  
Redim judith(4)  
  
judith(0) = "Trinzka"  
judith(1) = "Judith"  
judith(2) = "Célibataire"  
judith(3) = 22  
judith(4) = array()  
  
personnes(1) = judith
```

Et on va transformer en un XML de la même forme que celui qu'on lisait au chapitre précédent

Pour l'écriture, on va également devoir initialiser et configurer notre parseur :

```
Set xmlDoc = CreateObject("Microsoft.XMLDOM")  
  
Set oCreation = xmlDoc.createProcessingInstruction("xml", "version='1.0' encoding='ISO-8859-1'")  
xmlDoc.insertBefore oCreation, xmlDoc.childNodes.Item(0)
```

Les 2 lignes qui suivent l'initialisation permettent de générer l'entête XML.

Ensuite, on va créer la racine et l'ajouter au document :

```
Set root = xmlDoc.createElement("personnes")  
  
xmlDoc.appendChild(root)
```

On a donc utilisé la méthode `createElement` pour créer un nouveau noeud et la méthode `appendChild` pour ajouter ce noeud à la racine du document.

On va maintenant parcourir notre tableau `personnes` pour créer un élément `personne` pour chacun des éléments du tableau :

```
For Each personne In personnes
    Set personneElement = xmlDoc.createElement("personne")
    root.appendChild(personneElement)
Next
```

Cette fois, on ajouté les éléments dans le noeud racine.

On va maintenant ajouter les informations principales sur la personne :

```
Set personneElement = xmlDoc.createElement("personne")

Set nomElement = xmlDoc.createElement("nom")
nomElement.Text = personne(0)
personneElement.appendChild(nomElement)

Set prenomElement = xmlDoc.createElement("prenom")
prenomElement.Text = personne(1)
personneElement.appendChild(prenomElement)

Set etatElement = xmlDoc.createElement("etat")
etatElement.Text = personne(2)
personneElement.appendChild(etatElement)

root.appendChild(personneElement)
```

Le principe reste toujours le même, on crée un élément et on l'ajoute à l'élément parent.

Passons maintenant à l'ajout de l'attribut `age` :

```
personneElement.setAttribute "age", personne(3)
```

Rien de bien méchant de ce côté-là. Pour les enfants, non plus, rien de difficile :

```
If UBound(personne(4)) > -1 Then
    Set enfantsElement = xmlDoc.createElement("enfants")

    For Each enfant In personne(4)
        Set enfantElement = xmlDoc.createElement("enfant")

        Set nomElement = xmlDoc.createElement("nom")
        nomElement.Text = enfant(0)
        enfantElement.appendChild(nomElement)

        Set prenomElement = xmlDoc.createElement("prenom")
        prenomElement.Text = enfant(1)
        enfantElement.appendChild(prenomElement)

        enfantsElement.appendChild(enfantElement)
    Next

    personneElement.appendChild(enfantsElement)
End If
```

S'il y a des enfants, on crée donc un élément enfants auquel on ajoute des éléments enfant pour chaque enfant de la personne et enfin on ajoute un nom et un prénom à chaque enfant.

On va maintenant sauvegarder notre fichier.

```
xmlDoc.save ("personnes2.xml")
```

Et voilà le résultat :

```
personnes2.xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<personnes><personne
  age="49"><nom>Baud</nom><prenom>Georges</prenom><etat>Marié</etat><enfants><enfant><nom>Tiop</
nom><prenom>Elisabeth</prenom></enfant></enfants></personne><personne
  age="22"><nom>Trinzka</nom><prenom>Judith</prenom><etat>Célibataire</etat></personne></personnes>
```

Ce fichier XML bien que tout à fait valide n'est pas très lisible. On va voir dans le chapitre suivant comment faire pour le rendre plus lisible.

3.1 - Indenter le fichier XML

La méthode à utiliser est malheureusement bien plus compliquée cette fois. Il faut remplacer xmlDoc.Save par :

```
set rdr = CreateObject("MSXML2.SAXXMLReader")
set wrt = CreateObject("MSXML2.MXXMLWriter")
Set oStream = CreateObject("ADODB.STREAM")
oStream.Open
oStream.Charset = "ISO-8859-1"

wrt.indent = True
wrt.encoding = "ISO-8859-1"
wrt.output = oStream
Set rdr.contentHandler = wrt
Set rdr.errorHandler = wrt
rdr.Parse xmlDoc
wrt.flush

oStream.SaveToFile "personnes2.xml", 2

Set rdr = Nothing
Set wrt = Nothing
```

Ce qu'on fait dans ce code, c'est qu'on ouvre un reader SAX qui va parser notre fichier XML et on donne le contenu de la lecture à un MXXMLWriter qui va nous permettre d'écrire notre XML avec les balises indentées et enfin, on configure le stream de sortir du writer on lui dit d'écrire dans le fichier de notre choix.

Ce qui nous donne comme fichier en sortie :

```
personnes2.xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<personnes>
  <personne age="49">
    <nom>Baud</nom>
    <prenom>Georges</prenom>
    <etat>Marié</etat>
    <enfants>
      <enfant>
        <nom>Tiop</nom>
```

```
personnes2.xml
  <prenom>Elisabeth</prenom>
</enfant>
</enfants>
</personne>
<personne age="22">
  <nom>Trinzka</nom>
  <prenom>Judith</prenom>
  <etat>Célibataire</etat>
</personne>
</personnes>
```

Il est donc directement beaucoup plus lisible.

4 - Modification

On va maintenant voir comment faire pour modifier un fichier XML existant. En fait, ce n'est pas bien compliqué et on a déjà tout vu. Il suffit en fait de lire le fichier XML, de le modifier avec les commandes vues dans la partie d'écriture et de le sauvegarder ensuite.

Pour l'exemple, on va modifier les âges des personnes. On va rajouter une année à chacune des personnes dans le XML.

On va l'ouvrir de la même manière que pour une lecture normale :

```
Set xmlDoc = CreateObject("Microsoft.XMLDOM")
xmlDoc.Async = "false"
xmlDoc.Load("personnes.xml")
```

Ensuite, on va boucler sur toutes les personnes et leur rajouter une année :

```
'On récupère tous les noeuds personnes à l'intérieur d'un noeud personnes
For Each personneElement In xmlDoc.selectNodes("/personnes/personne")
'On récupère l'âge de la personne
age = personneElement.getAttribute("age")

'On lui rajoute un an
personneElement.setAttribute "age", age + 1
Next
```

Et finalement, on va sauvegarder notre fichier. Vous pouvez choisir la méthode que vous préférez pour le sauvegarder. Pour l'exemple, on va prendre la méthode avec indentation :

```
set rdr = CreateObject("MSXML2.SAXXMLReader")
set wrt = CreateObject("MSXML2.MXXMLWriter")
Set oStream = CreateObject("ADODB.STREAM")
oStream.Open
oStream.Charset = "ISO-8859-1"

wrt.indent = True
wrt.encoding = "ISO-8859-1"
wrt.output = oStream
Set rdr.contentHandler = wrt
Set rdr.errorHandler = wrt
rdr.Parse xmlDoc
wrt.flush

oStream.SaveToFile "personnes.xml", 2

Set rdr = Nothing
Set wrt = Nothing
Set xmlDoc = Nothing
```

Ce qui vous donnera comme fichier XML :

personnes.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<personnes>
  <personne age="50">
    <nom>Baud</nom>
    <prenom>Georges</prenom>
    <etat>Marié</etat>
  <enfants>
```

personnes.xml

```
<enfant>
  <nom>Tiop</nom>
  <prenom>Elisabeth</prenom>
</enfant>
</enfants>
</personne>
<personne age="23">
  <nom>Trinzka</nom>
  <prenom>Judith</prenom>
  <etat>Célibataire</etat>
</personne>
<personne age="89">
  <nom>Godoh</nom>
  <prenom>Madeleine</prenom>
  <etat>Veuve</etat>
  <enfants>
    <enfant>
      <nom>Godoh</nom>
      <prenom>Jean-Marie</prenom>
    </enfant>
    <enfant>
      <nom>Godoh</nom>
      <prenom>Etienne</prenom>
    </enfant>
    <enfant>
      <nom>Swoti</nom>
      <prenom>Julienne</prenom>
    </enfant>
  </enfants>
</personne>
</personnes>
```

Comme vous le voyez, une fois que vous savez lire et écrire dans un fichier XML, la modification n'est plus qu'un jeu d'enfant.

On va prendre un deuxième exemple : Mme Madeleine Godoh a renié ses enfants pour une raison quelconque, on va donc devoir les supprimer du fichier XML.

Je ne vais montrer que la partie de modification le reste étant le même code que pour le premier exemple :

```
'On récupère tous les noeuds enfant à l'intérieur d'une balise enfants
'se trouvant dans un noeud personne qui a comme nom Godoh et comme prénom Madeleine
xmlDoc.selectNodes("/personnes/personne[nom='Godoh' and
  prenom='Madeleine']/enfants/enfant").removeAll
```

Cette fois, on voit toute la puissance de XPath. Il nous suffit d'une seule ligne pour supprimer les enfants d'une personne. On a donc mis la recherche directement dans la requête XPath et on a ensuite demandé les enfants pour la personne correspondante à la requête et enfin on a supprimé tous ces éléments.

Et voici donc le résultat :

personnes.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<personnes>
  <personne age="50">
    <nom>Baud</nom>
    <prenom>Georges</prenom>
    <etat>Marié</etat>
    <enfants>
      <enfant>
        <nom>Tiop</nom>
```


5 - Conclusion

Voilà, vous savez maintenant comment lire et écrire dans des fichiers XML en Visual Basic Script. J'espère que cet article vous aura été utile.

Si vous voulez en savoir plus sur XPath, je vous conseille de lire [cet article](#) qui montre une grande quantité d'exemples de requêtes XPath.

5.1 - Remerciements

Je tiens à remercier **Erwy** et **GrandFather** pour leurs conseils avisés sur XPath.

Je tiens également à remercier **bbil** pour ses corrections.

5.2 - Code complet

5.2.1 - Lecture

Voici le code complet pour la lecture de notre fichier :

```
Set xmlDoc = CreateObject("Microsoft.XMLDOM")
xmlDoc.Async = "false"
xmlDoc.Load("personnes.xml")

'On récupère tous les noeuds personnes à l'intérieur d'un noeud personnes
For Each personneElement In xmlDoc.selectNodes("/personnes/personne")
    'On récupère les informations sur la personne
    nom = personneElement.selectSingleNode("nom").text
    prenom = personneElement.selectSingleNode("prenom").text
    etat = personneElement.selectSingleNode("etat").text
    age = personneElement.getAttribute("age")

    MsgBox "Nom : " & nom & vbcrLf & _
           "Prénom : " & prenom & vbcrLf & _
           "Etat civil : " & etat & vbcrLf & _
           "Age : " & age

'On récupère tous les noeuds enfants à l'intérieur d'une balise enfants
'dans le noeud personne courant
Set enfants = personneElement.selectNodes("enfants/enfant")

'S'il y a des enfants
If enfants.length > 0 Then
    text = prenom & " " & nom & " a des enfants : "

    'On boucle sur tous les enfants
    For Each enfantElement In enfants
        nomEnfant = enfantElement.selectSingleNode("nom").text
        prenomEnfant = enfantElement.selectSingleNode("prenom").text

        text = text & vbcrLf & " - " & prenomEnfant & " " & nomEnfant
    Next

    MsgBox text
Else
    MsgBox prenom & " " & nom & " n'a pas d'enfants"
End If
Next
```

```
Set xmlDoc = Nothing
```

5.2.2 - Ecriture

Voici le code complet de notre mini-programme d'écriture dans un fichier XML :

```
personnes = array()  
Redim personnes(1)  
  
georges = array()  
Redim georges(4)  
  
georges(0) = "Baud"  
georges(1) = "Georges"  
georges(2) = "Marié"  
georges(3) = 49  
  
enfants = array()  
Redim enfants(0)  
  
elisabeth = array()  
Redim elisabeth(1)  
  
elisabeth(0) = "Tiop"  
elisabeth(1) = "Elisabeth"  
  
enfants(0) = elisabeth  
  
georges(4) = enfants  
  
personnes(0) = georges  
  
judith = array()  
Redim judith(4)  
  
judith(0) = "Trinzka"  
judith(1) = "Judith"  
judith(2) = "Célibataire"  
judith(3) = 22  
judith(4) = array()  
  
personnes(1) = judith  
  
Set xmlDoc = CreateObject("Microsoft.XMLDOM")  
  
Set oCreation = xmlDoc.createProcessingInstruction("xml", "version='1.0' encoding='ISO-8859-1'")  
xmlDoc.insertBefore oCreation, xmlDoc.childNodes.Item(0)  
  
Set root = xmlDoc.createElement("personnes")  
  
xmlDoc.appendChild(root)  
  
For Each personne In personnes  
Set personneElement = xmlDoc.createElement("personne")  
  
Set nomElement = xmlDoc.createElement("nom")  
nomElement.Text = personne(0)  
personneElement.appendChild(nomElement)  
  
Set prenomElement = xmlDoc.createElement("prenom")  
prenomElement.Text = personne(1)  
personneElement.appendChild(prenomElement)  
  
Set etatElement = xmlDoc.createElement("etat")
```

```

etatElement.Text = personne(2)
personneElement.appendChild(etatElement)

personneElement.setAttribute "age", personne(3)

If UBound(personne(4)) > -1 Then
  Set enfantsElement = xmlDoc.createElement("enfants")

  For Each enfant In personne(4)
    Set enfantElement = xmlDoc.createElement("enfant")

    Set nomElement = xmlDoc.createElement("nom")
    nomElement.Text = enfant(0)
    enfantElement.appendChild(nomElement)

    Set prenomElement = xmlDoc.createElement("prenom")
    prenomElement.Text = enfant(1)
    enfantElement.appendChild(prenomElement)

    enfantsElement.appendChild(enfantElement)
  Next

  personneElement.appendChild(enfantsElement)
End If

root.appendChild(personneElement)
Next

set rdr = CreateObject("MSXML2.SAXXMLReader")
set wrt = CreateObject("MSXML2.MXXMLWriter")
Set oStream = CreateObject("ADODB.STREAM")
oStream.Open
oStream.Charset = "ISO-8859-1"

wrt.indent = True
wrt.encoding = "ISO-8859-1"
wrt.output = oStream
Set rdr.contentHandler = wrt
Set rdr.errorHandler = wrt
rdr.Parse xmlDoc
wrt.flush

oStream.SaveToFile "personnes2.xml", 2

Set rdr = Nothing
Set wrt = Nothing
Set xmlDoc = Nothing

```

5.2.3 - Modification

Voici le code complet que l'on a utilisé pour notre petit programme de modification :

```

Set xmlDoc = CreateObject("Microsoft.XMLDOM")
xmlDoc.Async = "false"
xmlDoc.Load("personnes.xml")

'On récupère tous les noeuds personnes à l'intérieur d'un noeud personnes
For Each personneElement In xmlDoc.selectNodes("/personnes/personne")
  'On récupère l'âge de la personne
  age = personneElement.getAttribute("age")

  'On lui rajoute un an
  personneElement.setAttribute "age", age + 1
Next

```

```
set rdr = CreateObject("MSXML2.SAXXMLReader")
set wrt = CreateObject("MSXML2.MXXMLWriter")
Set oStream = CreateObject("ADODB.STREAM")
oStream.Open
oStream.Charset = "ISO-8859-1"

wrt.indent = True
wrt.encoding = "ISO-8859-1"
wrt.output = oStream
Set rdr.contentHandler = wrt
Set rdr.errorHandler = wrt
rdr.Parse xmlDoc
wrt.flush

oStream.SaveToFile "personnes.xml", 2

Set rdr = Nothing
Set wrt = Nothing
Set xmlDoc = Nothing
```

Et voici le code complet du deuxième exemple :

```
Set xmlDoc = CreateObject("Microsoft.XMLDOM")
xmlDoc.Async = "false"
xmlDoc.Load("personnes.xml")

'On récupère tous les noeuds enfant à l'intérieur d'une balise enfants
'se trouvant dans un noeud personne qui a comme nom Godoh et comme prénom Madeleine
xmlDoc.selectNodes("/personnes/personne[nom='Godoh' and
prenom='Madeleine']/enfants/enfant").removeAll

set rdr = CreateObject("MSXML2.SAXXMLReader")
set wrt = CreateObject("MSXML2.MXXMLWriter")
Set oStream = CreateObject("ADODB.STREAM")
oStream.Open
oStream.Charset = "ISO-8859-1"

wrt.indent = True
wrt.encoding = "ISO-8859-1"
wrt.output = oStream
Set rdr.contentHandler = wrt
Set rdr.errorHandler = wrt
rdr.Parse xmlDoc
wrt.flush

oStream.SaveToFile "personnes.xml", 2

Set rdr = Nothing
Set wrt = Nothing
Set xmlDoc = Nothing
```

