

Accueil | ALM | Java | **.NET** | Dév. Web | EDI | Programmation | SGBD | Office | Solutions d'entreprise

Applications | Mobiles | Systèmes

**.NET**

Visual Studio

ASP.NET

C#

Visual Basic.NET

Windows Phone

Microsoft Azure

FORUMS .NET

FAQs .NET

TUTORIELS .NET

VIDÉOS .NET

SOURCES .NET

LIVRES .NET

OUTILS .NET

AZURE DEV CAMP

## Les fichiers XML et VB.Net

Date de publication : 08/03/2005 , Date de mise a jour : 08/03/2005

Par Selkis

Création et Lecture un Fichier Xml à schéma connu en utilisant la classe XmlDocument

### I. Création d'un fichier XML

#### I.A. Le principe en 6 points

#### II.B. Code source

### II. Lecture d'un fichier XML

#### II.A. Extraction des données

#### II.B. Le principe en 5 points

#### II.C. Code source



Télécharger le code source commenté  
 exemple vb.net Lecteur de flux rss

## I. Création d'un fichier XML

Un fichier Xml doit être écrit selon un schéma bien défini ,nous allons donc définir la structure de notre fichier suivant ce schéma : pour créer un fichier xml standard à la syndication de contenu au format rss, voir les référence en fin d'article

```
<lstSite>
  <SITE>
    <URL>http://www.peuw.net/index.xml</URL>
    <NOM>peuw.net</NOM>
  </SITE>
  <SITE>
    <URL>http://www.poxx.net/index.xml</URL>
    <NOM>poxx - Science Fair</NOM>
  </SITE>
</lstSite>
```



Imports System.Xml

Nous allons donc déclarer l'objet XmlDocument représentant notre document avec la classe XmlDocument.

```
'création d'une nouvelle instance du membre xmldocument
Dim XmlDoc As XmlDocument = New XmlDocument()
```

Puis nous allons tout créer et baliser un document vierge de part la propriété **LoadXml** (Cree un document suivant le code passé en paramètre) du **XmlDocument** .

```
'création du document
XmlDoc.LoadXml("[1stSite][1stSite]")
```

Ensuite pour chacun des flux nous allons créer une balise [SITE][SITE] qui contiendra , une balise [URL][URL] et une balise [NOM][NOM]

Chaque flux est représenté par un Élément (classe XmlElement), nous devons donc les déclarer

```
'a savoir :
Dim elemSite As XmlElement 'ElemSite pour le nœud [SITE][SITE]
Dim elemUrl As XmlElement 'ElemSite pour le nœud [URL][URL]
Dim elemNom As XmlElement 'ElemSite pour le nœud [NOM][NOM]
```

Pour créer nos balises, regardons de plus près les propriétés de la classe XmlElement qui nous seront utiles à savoir :

- .CreateElement : Pour crée un nouveau nœud (une nouvelle balise)
- .InnerText : Pour définir la valeur de la balise ( noeud)
- .AppendChild : pour ajouter le oeud à notre document (à la fin de la liste des enfants de ce noeud).

## I.A. Le principe en 6 points

1) Créons d'abord notre balise [SITE][SITE] qui est en fait le nœud parent des balises [URL][URL] et [NOM][NOM] et le nœud enfant de la balise [1stSite][1stSite]).

```
'creation de la balise [SITE][SITE]
elemSite = XmlDoc.CreateElement("SITE")
```

2) Créons ensuite nos balises [URL][URL] et [NOM][NOM]

```
elemUrl = XmlDoc.CreateElement("URL")
elemNom = XmlDoc.CreateElement("NOM")
```

3) Définissons les valeur de nos balises [URL][URL] et [NOM][NOM]

```
elemUrl.InnerText = " http://www.peuw.net/index.xml "
elemNom.InnerText = " peuw.net "
```

4) ajouter les balises enfants à la balise parent,

```
elemSite.AppendChild(elemUrl)
elemSite.AppendChild(elemNom)
```

5) ajouter la balise parent au document Xml

```
'on ajoute la balise parent au document
XmlDoc.DocumentElement.AppendChild(elemSite)
```

6) Ecrire le schéma Xml

## II.B. Code source

```
XmlDoc.Save(Application.StartupPath & "Newsite.XML")
```

Résumé du code

```
Dim elemSite As XmlElement
elemSite = XmlDoc.CreateElement("SITE")

'creation et definition de la balise url (enfant)
Dim elemUrl As XmlElement
elemUrl = XmlDoc.CreateElement("URL")
elemUrl.InnerText = " http://www.peuw.net/index.xml "

'creation et definition de la balise Nom (enfant)
Dim elemNom As XmlElement
elemNom = XmlDoc.CreateElement("NOM")
elemNom.InnerText = peuw.net

'ajout des 2 balise enfants a la balise parent
elemSite.AppendChild(elemUrl)
elemSite.AppendChild(elemNom)

'on ajoute la balise parent au document
XmlDoc.DocumentElement.AppendChild(elemSite)

'Ecriture du Xml
XmlDoc.Save(Application.StartupPath & "Newsite.XML")
MsgBox("Enregistrement réussi")
```

## II. Lecture d'un fichier XML

Il est très aisé de lire un fichier Xml dont nous connaissons la structure.

La première chose à faire est de déclarer l'objet qui va contenir les données du fichier.

Les fichiers Xml ont un schéma bien défini et pour pouvoir le lire nous devons nous servir du membre XmlDocument dépendant de la classe System.Xml

```
'creation d'une nouvelle instance du membre xmldocument
Dim XmlDoc As XmlDocument = New XmlDocument()
```

L'instruction **Load** nous permet de charger les données du document Xml dans l'objet XmlDocument déclaré précédemment pour nous permettre de le lire.

```
XmlDoc.Load(Application.StartupPath & "Newsite.XML")
```

## II.A. Extraction des données

Exemple de la structure de notre fichier XML :

```
<lstSite>
  <SITE>
    <URL>http://www.peuw.net/index.xml</URL>
    <NOM>peuw.net</NOM>
  </SITE>
  <SITE>
    <URL>http://www.poxx.net/index.xml</URL>
    <NOM>poxx - Science Fair</NOM>
  </SITE>
</lstSite>
```

Notre but est de récupérer les nœuds enfants [URL] [/URL] et [NOM] [/NOM] des nœuds parents [SITE] [/SITE] pour pouvoir lire leur valeur .

Intéressons nous a la propriété **DocumentElement** qui représente la racine de l'arbre et qui grâce à sa méthode **GetElementsByTagName** va nous permettre de retourner sous forme de liste (**XmlNodeList**) tous les nœuds portant le nom passé en paramètre (dans notre cas " site ").

Exemple

```
Dim element As XmlNodeList
element = XmlDoc.DocumentElement.GetElementsByTagName("SITE")
```

Maintenant que nous avons notre liste (**element (XmlNodeList)**) de nœuds " parents " il nous reste plus qu'a récupérer tous les nœuds enfants.

**element** contient donc la liste de tous les balises **[SITE][/SITE]**, chaque balise étant représenté par un nœud.

chaque nœud parent et enfant représente un **XmlNode** qu'il faut déclarer.

```
Dim noeud, noeudEnf As XmlNode
```

## II.B. Le principe en 5 points

Pour lire les noeud enfants nous utiliserons la propriété **ChildNodes** qui nous retourne pour le nœud sélectionné la liste des ses nœuds enfants sous forme de **XmlNode**.

il ne nous reste plus qu'a vérifier sur quel nœud on se trouve pour récupérer sa valeur et la traiter.

On va donc :

1. Parcourir un a un tous les noeuds " parent " (noeud) compris dans la liste element
2. Pour chaque un de ces noeuds, récupérer la liste des noeuds " enfants " (noeudEnf)
3. Parcourir un a un tous les noeuds enfants (noeudEnf) contenu dans cette liste
4. Récupérer grâce a la propriété LocalName le nom de la balise du noeud " enfant " sélectionné
5. Récupérer le contenu de la balise à l'aide de la propriété InnerText.

## II.C. Code source

```
For Each noeud In element
    For Each noeudEnf In noeud.ChildNodes
        If noeudEnf.LocalName = "URL" Then
            UrlSite = noeudEnf.InnerText
        Else
            If (noeudEnf.LocalName = "NOM") Then
                NomSite = noeudEnf.InnerText
            End If
        End If
        MsgBox(NomSite & " à " & UrlSite)
    Next
Next
```

Liste de mes articles Vb.Net:

Accueil de la section VB.NET.

Utilisation d'une base Access en mode déconnecté avec ADO.NET.

Lecture et écriture d'un fichier Xml à schéma connu.

Création d'une application de type Service Windows.

Création d'un logo xara3d en utilisant un service webdistant

Copyright © 2005 selkis. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

---

**Responsable bénévole de la rubrique Microsoft DotNET : Hinault Romaric - Contacter par email**

---

[Nous contacter](#)   [Participez](#)   [Hébergement](#)   [Informations légales](#)

**Copyright © 2000-2017 - [www.developpez.com](http://www.developpez.com)**