

Structure d'un fichier XML

Avant de voir comment manipuler ce format avec C# et le framework .NET, nous allons détailler les différents éléments de la structure d'un fichier XML.

1. Constituants d'un document XML

Un document XML peut être composé des types de blocs suivants :

Instructions de traitement

Les instructions de traitement permettent d'ajouter des informations destinées au processeur XML ou à tout autre programme dans le corps du document. Ces instructions doivent respecter une syntaxe particulière :

```
<?programme instruction ?>
```

La première partie de cette instruction de traitement indique le nom du programme destiné à la traiter. La seconde partie fournit l'instruction à proprement parler.

Un document XML contient la plupart du temps une instruction de traitement spéciale indiquant la version du format XML qu'il utilise ainsi que le jeu de caractères utilisé pour l'encodage du document.

```
<?xml version="1.0" encoding="utf-8" ?>
```

Commentaires

Comme avec C#, il est possible d'inclure des commentaires dans le code XML. Ceux-ci sont destinés aux utilisateurs, le plus souvent pour les aider à comprendre la structure ou le contenu du document. Ils sont ignorés par tout programme manipulant le code.

La syntaxe de définition de commentaires XML est identique à la syntaxe utilisée pour les commentaires HTML :

```
<!-- Ce texte est un commentaire ignoré par le processeur XML -->
```

Éléments

Les éléments sont constitués d'une balise ouvrante et d'une balise fermante. Ils peuvent contenir des données sous la forme d'une chaîne de caractères, d'un ou plusieurs autres éléments ou bien d'attributs. La syntaxe permettant de créer un élément XML est la suivante :

```
<nomElement>contenu</nomElement>
```

Certaines règles doivent être respectées concernant les éléments :

- Les noms d'éléments ne doivent ni contenir d'espace, ni débiter par xml, par un chiffre ou par un signe de ponctuation.
- Les balises de début et de fin doivent avoir la même casse.

- Les noms d'éléments doivent débiter après le symbole <, sans espace.
- Un document XML doit contenir au minimum un élément : l'élément racine du document.
- Si un élément n'a pas de contenu, il peut avoir la forme <nomElement />, sauf si cet élément est l'élément racine du document.
- Aucun élément ne peut être situé au même niveau de l'arborescence que l'élément racine. Les éléments doivent donc être imbriqués dans cet élément racine.

```
<?xml version="1.0" encoding="utf-8"?>
<Clients>
  <Client>
    <IdClient>ALFKI</IdClient>
    <Societe>Alfreds Futterkiste</Societe>
    <NomContact>Alfreds Maria Anders</NomContact>
    <Adresse>Obere Str. 57</Adresse>
    <CodePostal>12209</CodePostal>
    <Ville>Berlin</Ville>
    <DerniereCommande>
      <Produits>
        <Produit>
          <IdProduit>58</IdProduit>
          <Nom>Escargots de Bourgogne</Nom>
          <PrixUnitaire>13,25</PrixUnitaire>
          <Quantite>40</Quantite>
        </Produit>
        <Produit>
          <IdProduit>71</IdProduit>
          <Nom>Floremysost</Nom>
          <PrixUnitaire>21,50</PrixUnitaire>
          <Quantite>20</Quantite>
        </Produit>
      </Produits>
    </DerniereCommande>
  </Client>
</Clients>
```

Attributs d'éléments

Les attributs représentent des données stockées au niveau d'un élément. Ils sont placés dans la balise d'ouverture de l'élément. Les attributs sont définis en utilisant la syntaxe suivante :

```
<nomElement attribut1="valeur" attribut2="valeur"></nomElement>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Clients>
  <Client IdClient="ALFKI">
    <Societe>Alfreds Futterkiste</Societe>
    <NomContact>Alfreds Maria Anders</NomContact>
    <Adresse>Obere Str. 57</Adresse>
    <CodePostal>12209</CodePostal>
    <Ville>Berlin</Ville>
```

```

<DerniereCommande Date="09/04/1998">
  <Produits>
    <Produit IdProduit="58">
      <Nom>Escargots de Bourgogne</Nom>
      <PrixUnitaire>13,25</PrixUnitaire>
      <Quantite>40</Quantite>
    </Produit>
    <Produit IdProduit="71">
      <Nom>Floremysost</Nom>
      <PrixUnitaire>21,50</PrixUnitaire>
      <Quantite>20</Quantite>
    </Produit>
  </Produits>
</DerniereCommande>
</Client>
</Clients>

```

Espaces de noms

Les espaces de noms sont des identifiants permettant de dissocier plusieurs éléments dont le nom est identique. Ils permettent d'éviter les confusions lorsque plusieurs éléments ont le même nom, mais une signification différente. Ce type de cas peut arriver lorsque deux fichiers XML sont fusionnés.

Un espace de noms est défini en concaténant l'attribut `xmlns` suivi du symbole `:` et d'un préfixe pour l'espace de noms. La valeur de l'attribut est un identifiant unique, souvent écrit sous la forme d'une URL. Cette valeur nécessite d'être unique pour anticiper tout conflit en cas de transmission du document à une application pour laquelle le même élément est déjà défini.

```
<nomElement xmlns:prefixe="identifiant" />
```

Le document XML suivant fusionne des objets de type `Client` gérés respectivement par une application web et une application WPF en boutique. Ils sont distingués dans le document résultant par leur espace de noms.

```

<?xml version="1.0" encoding="utf-8"?>
<Clients xmlns:web="http://www.masociete.com/clients/web"

xmlns:boutique="http://www.masociete.com/clients/boutique">
  <web:Client Id="ALFKI">
    <Societe>Alfreds Futterkiste</Societe>
    <NomContact>Alfreds Maria Anders</NomContact>
    <Adresse>Obere Str. 57</Adresse>
    <CodePostal>12209</CodePostal>
    <Ville>Berlin</Ville>
    <DerniereCommande Date="09/04/1998">
      <Produits>
        <Produit IdProduit="58">
          <Nom>Escargots de Bourgogne</Nom>
          <PrixUnitaire>13,25</PrixUnitaire>
          <Quantite>40</Quantite>
        </Produit>
        <Produit IdProduit="71">
          <Nom>Floremysost</Nom>

```

```

        <PrixUnitaire>21,50</PrixUnitaire>
        <Quantite>20</Quantite>
    </Produit>
</Produits>
</DerniereCommande>
</web:Client>

<boutique:Client Id="684316" EstAnonyme="True">
    <Achat Date="12/04/1998" ModePaieement="CB">
        <Produits>
            <Produit IdProduit="11">
                <Nom>Queso Cabrales</Nom>
                <PrixUnitaire>21,00</PrixUnitaire>
                <Quantite>2</Quantite>
            </Produit>
            <Produit IdProduit="19">
                <Nom>Teatime Chocolate Biscuits</Nom>
                <PrixUnitaire>9,20</PrixUnitaire>
                <Quantite>7</Quantite>
            </Produit>
        </Produits>
    </Achat>
</boutique:Client>
</Clients>

```

2. Document bien formé et document valide

Le langage XML définit plusieurs types de constituants permettant de décrire une structure de données cohérente. Mais cette notion seule ne suffit pas à définir la qualité d'un document XML. Deux notions complémentaires sont utilisées pour cela : le document bien formé et le document valide.

Un document bien formé est un document XML respectant les différentes règles syntaxiques du langage. Ces règles incluent les différents points évoqués ci-dessus.

Il est nécessaire que le document soit bien formé pour que le processeur XML soit en mesure de le traiter.

La validité du document est un point vérifiable lorsqu'une DTD (*Document Type Definition*) ou un schéma XSD (*Xml Schema Definition*) sont liés au fichier. Ces types de documents fournissent la description de la structure de données utilisées dans le fichier XML. Si cette structure déroge aux règles fixées par la DTD ou le schéma XSD, le fichier n'est pas valide.