Introduction

Avant d'aborder la moindre explication sur cette technologie, pourquoi peut-on avoir besoin du XML en ActionScript, dans Flash ?

Nous ne commencerons pas à répondre à cette question par une énumération d'exemples, car nous n'en finirions pas, répondons simplement par cette affirmation :

"Le XML permet de constituer une 'base de données' acessible off-line et/ou on-line".

Vous aurez parfois besoin d'acceder à une simple liste d'informations dans un programme (ex. : Le nom des jours de la semaine, une liste de prénoms ou de noms, une liste de nombres), vous pourrez dans ce cas créer un tableau, mais vous devrez faire appel de préférence à un fichier XML si les données à stocker/manipuler sont trop nombreuses pour être gérées dans un tableau.

Imaginons que nous ayons besoin de gérer un carnet d'adresses qui contient des fiches avec les informations suivantes :

- nom
- prenom
- adresse mail

Voici comment nous procéderions pour créer un fichier XML à manipuler à partir d'un programme.

Créer un fichier XML

1. A partir d'un éditeur de texte, créez un nouveau document que vous enregistrez au format UTF-8 ou en UNICODE*.
2. Dans ce nouveau document, commencez par saisir sur 2 lignes, les balises <carnet> et </carnet>. Il s'agit de la balise d'ouverture et de fermeture du nœud racine.

```
<carnet>

</carnet>
```

3. Créez ensuite à l'intérieur de ce NŒUD RACINE (ci-dessus), les différents NŒUDS ENFANTS. Voici un exemple de nœud enfant (ci-dessous) :

```
<personne>
```

```
    <nom>Durand</nom>

    <prenom>Eric</prenom>

    <adressemail>adresse@domaine.com</adressemail>

  </personne>
```

Ce premier noeud enfant (ci-dessus), contient lui-même trois noeuds enfants (dont les noms de balises sont nom, prenom et adressemail) qui pourraient à leurs tours en contenir d'autres. Pour ne pas complexifier notre exemple, nous nous arrêterons à ce niveau de complexité.
Reprenons le déroulement de la phase de création de notre document XML en constituant le contenu définitif de notre arborescence XML.

```
<carnet>
<personne>
  <nom>Durand</nom>
  <prenom>Eric</prenom>
  <adressemail>adresse@domaine.com</adressemail>
</personne>
<personne>
  <nom>Martin</nom>
  <prenom>Eva</prenom>
  <adressemail>adresse@domaine.com </adressemail>
</personne>
<personne>
  <nom>Dupont</nom>
  <prenom>Luc</prenom>
  <adressemail>adresse@domaine.com</adressemail>
</personne>
</carnet>
```

4. Effectuez un dernier enregistrement (CTR-S (Windows) ou Commande-S (Mac)) de votre document.
5. Validez votre document en le prévisualisant à travers un navigateur (Glissez-déplacez l'icône du fichier de votre document XML dans la fenêtre d'un navigateur). Si un message d'erreur apparaît, corrigez la (suivez le message d'erreur qu'il vous indique à la ligne spécifiée). Il arrive très souvent qu'une majuscule soit utilisée en début de nom d'une balise d'ouverture et pas dans la balise de fermeture, ou inversement. (ex. : <nom> </Nom>).

* Dans la fenêtre d'enregistrement de votre logiciel, au moment où vous allez indiquer le nom du fichier, il y aura un menu déroulant ou un bouton d'accès à une fenêtre dans laquelle vous pourrez alors sélectionner UTF-8 ou UNICODE.

Voilà, notre document XML est à présent terminé, il ne nous reste plus qu'à essayer d'exploiter ses données à partir de Flash et plus précisement à partir de l'ActionScript 3. Avant de découvrir le code AS3 nécessaire pour l'exploitation d'un fichier XML ajoutons qu'il est possible de stocker les informations différement dans une arborescence XML, en faisant appel aux attributs d'une balise.

En reprenant l'exemple de nos noeuds enfants, voici ce que nous aurions pu faire :

```
<personne nom="Martin">
  <prenom>Eva</prenom>
  <adressemail>adresse@domaine.com</adressemail>
</personne>
```

Comme vous pouvez le remarquer, nous avons ajouté la chaine de caractères nom="Martin" à l'intérieur de la première balise. Le mot "nom" est qualifié d'attribut et le mot "Martin" représente la valeur de cet attribut. Il est très important de laissé une espace entre le nom de la balise et celui de l'attribut : <personne nom="Martin"> et non <personnenom="Martin">. De plus, la valeur de l'attribut doit toujours être contenue entre guillemets. Dans certains cas, une balise pourra même ne contenir qu'un seul ou plusieurs attributs :

```
<personne nom="Martin" prenom="Eva" adressemail=""/>
```

Comme vous pouvez le remarquer nous n'avons plus qu'une seule balise autofermante (et non deux), voilà ce que nous aurions donc pu obtenir en reprenant l'exemple ci-dessus.

```
<carnet>
  <personne nom="Durand" prenom="Eric" adressemail="eric@durand.com"/>
  <personne nom="Martin" prenom="Eva" adressemail="eva@martin.com"/>
  <personne nom="Dupont" prenom="Luc" adressemail="luc@dupont.com"/>
</carnet>
```

Attributs ou noeuds enfants ? Vous vous posez ou vous poserez peut-être cette question un jour, mais la réponse est simple et évidente : Tout dépend de la quantité d'informations à stocker et de la structure que vous voulez donner à votre arborescence XML. Dans le cas d'un nom, un prénom ou une adresse mail, il s'agit d'informations assez courtes, vous pouvez donc les stocker en tant que valeurs d'attributs, en revanche, si vous aviez eu à stocker un texte bien plus long, vous auriez eu tout intérêt à utiliser une balise ouvrante et une autre fermante.

Remarque : Vous noterez que dans certains documents XML, la première ligne ne correspond pas à celle que vous voyez dans notre exemple, mais elle correspond à une autre qui décrit le type de document. En AS3, nous n'avons pas initialement besoin de manipuler cette information, nous l'omettons donc volontairement.

Charger les données XML dans un document Flash

Avant de parcourir les données contenues dans notre document XML, nous devons chercher à charger l'arborescence qu'il contient. Utilisez alors le script ci-dessous :

```
var adresseFichierXML:URLRequest = new URLRequest("nomdufichier.xml");

var conteneurXML:URLLoader = new URLLoader();

conteneurXML.load(adresseFichierXML);
```

Si ce code vous parait trop complexe, vous pouvez alors simplifier le script de la façon suivante (nous ne typons pas les instances et optons pour des noms plus courts mais moins représentatifs) :

```
var adresse = new URLRequest("nomdufichier.xml");

var conteneur = new URLLoader();

conteneur.load(adresse);
```

Pour l'instant, nous indiquons simplement au compilateur de Flash de télécharger les données du fichier, mais nous n'avons pas encore vérifié la fin du chargement, ni même instancié la classe XML pour pouvoir manipuler les données, nous devons donc complété notre script initial :

```
var arboXML:XML;

conteneurXML.addEventListener(Event.COMPLETE,chargementXMLTermine);

function chargementXMLTermine(evt:Event) {
```

```
    arboXML = new XML(conteneurXML.data);



}
```

Note : Si vous avez opté pour une simplification du code, pensez à remplacer conteneurXML par conteneur.
Voilà, vous venez d'apprendre à charger des données XML et à les placer dans une instance de type XML. Toute la difficulté réside à présent dans la manipulation de l'arborescence XML afin de lire les noeuds et attributs contenus dans le fichier.

Parcourir les données d'une arborescence XML

Nous avons le fichier XML suivant (attention, nous avons ajouté un attribut intitulé "numeroSecu" aux balises <personne> :

```
<carnet>
<personne numeroSecu="123">
  <nom>Durand</nom>
  <prenom>Eric</prenom>
  <adressemail>eric@durand.com</adressemail>
</personne>
<personne numeroSecu="456">
  <nom>Martin</nom>
  <prenom>Eva</prenom>
  <adressemail>eva@martin.com</adressemail>
</personne>
<personne numeroSecu="789">
  <nom>Dupont</nom>
  <prenom>Luc</prenom>
  <adressemail>luc@dupont.com</adressemail>
</personne>
</carnet>
```

…et nous avons le script suivant :

```
var adresseFichierXML:URLRequest = new URLRequest("nomdufichier.xml");



var conteneurXML:URLLoader = new URLLoader();
```

```
conteneurXML.load(adresseFichierXML);

var arboXML:XML;

conteneurXML.addEventListener(Event.COMPLETE,chargementXMLTermine);

function chargementXMLTermine(evt:Event) {

  arboXML = new XML(conteneurXML.data);

}
```

Dans la fonction chargementXMLTermine, nous avons instancié la classe XML(), mais nous n'avons pas cherché à lire un noeud, voici donc comment vous devez procéder :

(Les lignes d'instructions qui figurent ci-dessous sont à ajouter dans la fonction chargementXMLTermine).

```
Lire le premier noeud : trace(arboXML.personne[0])
Lire le contenu du noeud dont la balise est nom dans le noeud à l'index 2 :
trace(arboXML.personne[2].nom)
Lire la valeur de l'atribut numeroSecu du noeud numéro 1 :
trace(arboXML.personne[1].@numeroSecu)
```

Note : Le premier noeud d'une arborescence XML porte l'index 0.

Résultats de l'exécution des 3 lignes d'instructions ci-dessus :

```
<personne numeroSecu="123">
  <nom>Durand</nom>
  <prenom>Eric</prenom>
  <adressemail>eric@durand.com</adressemail>
</personne>

Dupont
```

456

Voilà, faites à présent différents essais pour essayer de maitriser la gestion du XML en ActionScript. Courage, cela n'est pas si difficile !!!

Les XMLList

Lorsque vous effectuerez un filtre, c'est-à-dire une requête, une recherche, vous aurez peut-être besoin de stocker l'ensemble des données qui constituent votre résultat dans une liste XML. Instanciez alors la classe XMLList() et affectez lui comme valeur, le résultat de votre recherche. Dans l'exemple ci-dessous, nous stockons dans une instance intitulée requête, l'ensemble des valeurs contenues dans les noeuds nom des noeuds personne.

```
var adresseFichierXML:URLRequest = new URLRequest("nomdufichier.xml");

var conteneurXML:URLLoader = new URLLoader();
var arboXML:XML;

var requete:XMLList;
conteneurXML.load(adresseFichierXML);
conteneurXML.addEventListener(Event.COMPLETE,chargementXMLTermine);

function chargementXMLTermine(evt:Event) {

    arboXML = new XML(conteneurXML.data);

    requete = new XMLList(arboXML.personne.nom);

    for each (var individu:String in requete) {
```

```
        trace(individu);


    }


}
```

Autre exemple

Fichier XML de départ :

```
<galerie>
<artiste nom="TARDIVO">
  <toile numero="12">
    <nom technique="acrylique">Les belles de mai</nom>
    <taille>120x80</taille>
  </toile>
  <toile numero="13">
    <nom technique="acrylique">Gudule</nom>
    <taille>120x120</taille>
  </toile>
  <toile numero="14">
    <nom technique="huille">Queue ménie</nom>
    <taille>80x80</taille>
  </toile>
</artiste>
<artiste nom="LEMARCHAND">
  <toile numero="28">
    <nom technique="huile">Le jaune d'or</nom>
    <taille>140x80</taille>
  </toile>
  <toile numero="31">
    <nom technique="aquarelle">Océan y levène</nom>
    <taille>90x90</taille>
  </toile>
  <toile numero="63">
    <nom technique="acrylique">Vertige de la vague</nom>
    <taille>60x60</taille>
  </toile>
</artiste>
</galerie>
```

Code :

```
var arboXML:XML;

var adresseFichierXML:URLRequest = new URLRequest("galerie.xml");

var conteneurXML:URLLoader = new URLLoader();

function chargementXMLTermine(evt:Event) {

        arboXML = new XML(conteneurXML.data);

        trace(arboXML.artiste.(@nom=="TARDIVO"));

        trace(arboXML.children().length());

        trace(arboXML.artiste[1].children().length());

        trace(arboXML.artiste[1]);

        trace(arboXML.artiste.toile.(@numero==28));

        trace(arboXML.artiste.toile.(nom=="Vertige de la vague").taille);
```

```
        trace(arboXML.artiste.toile.(nom=="Vertige de la vague"));

        trace(arboXML.artiste[1].toile.(@numero=="28").nom);

}

conteneurXML.addEventListener(Event.COMPLETE,chargementXMLTermine);

conteneurXML.load(adresseFichierXML);
```

Voici le résultat de l'exécution des lignes d'instructions :

```
trace(arboXML.artiste.(@nom=="TARDIVO"));

<artiste nom="TARDIVO">
  <toile numero="12">
    <nom technique="acrylique">Les belles de mai</nom>
    <taille>120x80</taille>
  </toile>
  <toile numero="13">
    <nom technique="acrylique">Gudule</nom>
    <taille>120x120</taille>
  </toile>
  <toile numero="14">
    <nom technique="huille">Queue nénie</nom>
    <taille>80x80</taille>
  </toile>
</artiste>

trace(arboXML.children().length());

2
```

```
trace(arboXML.artiste[1].children().length());
```

3

```
trace(arboXML.artiste[1]);
```

```
<artiste nom="LEMARCHAND">
  <toile numero="28">
    <nom technique="huile">Le jaune d'or</nom>
    <taille>140x80</taille>
  </toile>
  <toile numero="31">
    <nom technique="aquarelle">Océan y levène</nom>
    <taille>90x90</taille>
  </toile>
  <toile numero="63">
    <nom technique="acrylique">Vertige de la vague</nom>
    <taille>60x60</taille>
  </toile>
</artiste>
```

```
trace(arboXML.artiste.toile.(@numero==28));
```

```
<toile numero="28">
  <nom technique="huile">Le jaune d'or</nom>
  <taille>140x80</taille>
</toile>
```

```
trace(arboXML.artiste.toile.(nom=="Vertige de la vague").taille);
```

60x60

```
trace(arboXML.artiste.toile.(nom=="Vertige de la vague"));
```

```
<toile numero="63">
  <nom technique="acrylique">Vertige de la vague</nom>
  <taille>60x60</taille>
</toile>
```

```
trace(arboXML.artiste[1].toile.(@numero=="28").nom);
```

```
Le jaune d'or
```

Remarque : Vous noterez que dans certains documents XML, la première ligne ne correspond pas à celle que vous voyez dans notre exemple, mais elle correspond à une autre qui décrit le type de document. En AS3, nous n'avons pas initialement besoin de manipuler cette information, nous l'omettons donc volontairement.