

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ≡ ≡ ↺ 🔍 ↻

## 5 XML-Schema et les espaces de noms

- Déclaration d'un espace de nom
- Espace de noms par défaut
- Espaces de noms et schéma
- Inclusions de schémas
- Importation de schéma

# Motivations

- modularité, réutilisation
- exporter les définitions d'un schéma
- éviter les conflits de nom.
- utilisation de noms *qualifiés* pour les éléments et les attributs.

# Définition

## Définition

Un espace de noms est *identifié* par une adresse *IRI*,  
(Internationalized Resource Identifier)

- IRI = extension des URI permettant l'utilisation de caractères internationaux (par ex. UTF-8) dans l'adresse elle-même.  
`http://www.exemple.org/clés/`
- URI = URN ou URL
  - URN : `urn:isbn-0-395-36341-1`
  - URL : `ftp://ftpperso.free.fr/pxml`

# Remarques

- Le W3C déconseille l'usage d'une IRI relative comme identifiant d'espace de nom
- L'analyse d'un identifiant d'espace de nom tient compte de la casse et ne prend pas en compte la *résolution* de l'IRI. Tous les exemples suivants représentent des identifiants différents :

`http://www.Example.org/wine`

`http://www.example.org/Wine`

`http://www.example.org/rosé`

`http://www.example.org/ros%c3%a9`

`http://www.example.org/ros%c3%A9`

# Finale

## En pratique

identifiant d'espace de nom = URL absolue avec des caractères ASCII.

`http://www.w3.org/XML/1998/namespace`

`http://www.w3.org/1999/xhtml`

`http://www.w3.org/2001/XMLSchema`

`http://www.w3.org/2001/XMLSchema-instance`

`http://www.w3.org/1999/XSL/Transform`

`http://xml.insee.fr/schema/`

`http://fil.univ-lille1.fr/miage-fa-fc`

# Utiliser les espaces de noms

# Déclaration d'un espace de noms

- Le préfixe qui désigne un espace de noms doit avoir été déclaré, grâce à l'attribut `xmlns` ou par un attribut qui commence par `xmlns`:
- Les préfixes `xml` et `xmlns` sont réservés.
- La déclaration se fait dans la balise ouvrante d'un élément
- Lorsqu'on déclare un espace de noms, le préfixe est applicable dès la balise ouvrante où se fait la déclaration, et pour tout le contenu de cet élément, sauf si le même préfixe est utilisé plus bas pour un autre espace de noms.
- L'utilisation du préfixe pour un élément (ou attribut) indique que cet élément (ou attribut) appartient à l'espace de noms associé au préfixe. (nom qualifié)
- On peut déclarer plusieurs espaces de noms dans une même balise ouvrante.



# Déclaration d'un espace de noms

- La déclaration se fait dans la balise ouvrante d'un élément
- Lorsqu'on déclare un espace de noms, le préfixe est applicable dès la balise ouvrante où se fait la déclaration, et pour tout le contenu de cet élément.

```
<x xmlns:edi="http://ecom.exple.org/schema">  
  <!-- the "edi" prefix is bound to  
        http://ecom.exple.org/schema  
        for the "x" element and contents -->  
</x>
```

# Déclaration d'un espace de noms

- La déclaration se fait dans la balise ouvrante d'un élément
- Lorsqu'on déclare un espace de noms, le préfixe est applicable dès la balise ouvrante où se fait la déclaration, et pour tout le contenu de cet élément.

```
<!-- the 'price' element's namespace  
       is http://ecom.exple.org/schema -->  
<edi:price xmlns:edi="http://ecom.exple.org/sch" units="Euro">  
  32.18  
</edi:price>
```

# Déclaration d'un espace de noms

- La déclaration se fait dans la balise ouvrante d'un élément
- Lorsqu'on déclare un espace de noms, le préfixe est applicable dès la balise ouvrante où se fait la déclaration, et pour tout le contenu de cet élément.

```
<x xmlns:edi="http://ecom.exple.org/schema">  
  <!-- the "taxClass" attribute's namespace is  
        http://ecom.exple.org/schema -->  
  <lineItem edi:taxClass="exempt">Baby food</lineItem>  
</x>
```

# Déclaration d'un espace de noms

- Lorsqu'on déclare un espace de noms, le préfixe est applicable dès la balise ouvrante où se fait la déclaration, et pour tout le contenu de cet élément, sauf si le même préfixe est utilisé plus bas pour un autre espace de noms.

```
<x xmlns:edi="http://ecom.exple.org/schema-1">
  <edi:a>
    <edi:b xmlns:edi="http://ecom.exple.org/schema-2">
      <edi:a><edi:c/></edi:a>
    </edi:b>
  <edi:b/>
</edi:a>
</x>
```

# Déclaration d'un espace de noms

- On peut déclarer plusieurs espaces de noms dans une même balise ouvrante.

```
<!-- both namespace prefixes are available throughout -->  
<bk:book  
  xmlns:bk="urn:loc.gov:books"  
  xmlns:isbn="urn:ISBN:0-395-36341-6">  
  <bk:title>  
    Cheaper by the Dozen  
  </bk:title>  
  <isbn:number>  
    1568491379  
  </isbn:number>  
</bk:book>
```

# Espace de noms par défaut

- Si on utilise l'attribut `xmlns` (sans `:`), on définit alors un espace de noms par défaut, pour lequel il n'existe pas de préfixe associé.
- L'espace de nom par défaut ne s'applique pas aux attributs.

```
<?xml version="1.1"?>
<!-- elements are in the HTML namespace, by default -->
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>Frobnostication</title></head>
  <body>
    <p>
      Moved to <a href="http://frob.example.com">here</a>.
    </p>
  </body>
</html>
```

# Espace de noms par défaut

- Si on utilise l'attribut `xmlns` (sans `:`), on définit alors un espace de noms par défaut, pour lequel il n'existe pas de préfixe associé.

```
<?xml version="1.1"?>
<!-- initially, the default namespace is "books" -->
<book xmlns="urn:loc.gov:books"
      xmlns:isbn="urn:ISBN:0-395-36341-6">
  <title>Cheaper by the Dozen </title>
  <isbn:number>1568491379</isbn:number>
  <notes>
    <!-- make HTML the default namespace for some commentary -->
    <p xmlns="http://www.w3.org/1999/xhtml">
      This is a <i>funny</i> book!
    </p>
  </notes>
</book>
```

# Créer des espaces de noms



# Exporter un espace de noms

## attribut targetNamespace

Pour exporter (créer) un espace de nom dans un schéma, on utilise l'attribut `targetNamespace` de la balise `xsd:schema`

```
<?xml version="1.0"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://www.w3.org/2001/XMLSchema
     http://www.w3.org/2001/XMLSchema.xsd"
  targetNamespace="http://fil.univ-lille1.fr/miage-fa-fc">

<!-- description de la miage fa fc -->
...
```

# Exporter un espace de noms

```
<?xml version="1.0" encoding="utf-8"?>
<miage-fa-fc
  xmlns="http://fil.univ-lille1.fr/miage-fa-fc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://fil.univ-lille1.fr/miage-fa-fc
    http://www.fil.univ-lille1.fr/FORMATIONS/MIAGE-FC-FA/
      schemas/miage-fa-fc.xsd"
  annee="2011">
  <creneaux>
    <creneau>
      <trimestre>T1</trimestre>
      <jour>lundi</jour>
      <de>09:00:00</de>
      <a>12:00:00</a>
      <salle>M5-A2</salle>
```

...

# Espaces de noms et schémas

## attribut elementFormDefault

Quand on exporte un espace de noms,

- Les déclarations globales appartiennent à l'espace de nom
- Les déclarations locales n'appartiennent pas à l'espace de nom, sauf si on ajoute l'attribut `elementFormDefault="qualified"`

```
<?xml version="1.0"?>
```

```
<xsd:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xsi:schemaLocation="http://www.w3.org/2001/XMLSchema
                               http://www.w3.org/2001/XMLSchema.xsd"
            xmlns="http://fil.univ-lille1.fr/miage-fa-fc"
            targetNamespace="http://fil.univ-lille1.fr/miage-fa-fc"
            elementFormDefault="qualified">
```

# Inclusion de schémas

On peut assembler plusieurs composants de schemas (définitions de types, déclarations d'éléments, ...), provenant de plusieurs documents.

- élément `include` qui permet d'inclure les définitions provenant d'autres schémas mais pas de plusieurs espaces de noms.
- Les schémas inclus doivent avoir
  - 1 soit le même espace de noms cible que le document qui les inclut
  - 2 soit pas d'espace de noms, dans ce cas, c'est l'espace de noms du schéma qui inclut tous les autres qui est pris en compte.

# Exemple d'inclusion sans espace de noms cible

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation="dept.xsd"/>
  <xsd:include schemaLocation="emp.xsd"/>
  <xsd:element name="ent">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="employees"/>
        <xsd:element ref="departements"/>
      </xsd:sequence>
    </xsd:complexType>
    ... les clefs étrangères ...
  </xsd:element>
</xsd:schema>
```

- dept.xsd et emp.xsd sont des fichiers dans le même répertoire.
- dept.xsd (resp. emp.xsd) contient les déclarations de l'élément departements (resp. employees) et de tous ses sous-éléments.

# Exemple d'instance du schéma précédent

```
<?xml version="1.0" encoding="utf-8"?>
<ent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="entreprise.xsd">
  <employees>
    <emp num="1">
      <nom>toto</nom><prenom>jules</prenom>
      <salaire>3452</salaire>
      <dept>informatique</dept>
    </emp>
  </employees>
  <departements>
    <dept>
      <nom>informatique</nom>
      <contact>Mme Machin 45-76-77-09-54</contact>
      <chef>1</chef>
    </dept>
  </departements>
</ent>
```

# Exemple d'inclusion avec espace de noms cible

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.lifl.fr/~yroos/schema"
  targetNamespace="http://www.lifl.fr/~yroos/schema"
  elementFormDefault="qualified"
>
  <xsd:include schemaLocation="dept.xsd"/>
  <xsd:include schemaLocation="emp.xsd"/>

  <xsd:element name="ent">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="employes"/>
        <xsd:element ref="departements"/>
      </xsd:sequence>
    </xsd:complexType>
    ... et les clefs étrangères ...
  </xsd:element>
</xsd:schema>
```

# Instance du schéma précédent

```
<?xml version="1.0" encoding="utf-8"?>
<ent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.lifl.fr/~yroos/schema"
    xsi:schemaLocation="http://www.lifl.fr/~yroos/schema
    http://saxo.lifl.fr/~yroos/schema/entreprise.xsd">
  <employes>
    <emp num="1">
      <nom>toto</nom> <prenom>jules</prenom>
      <salaire>3452</salaire> <dept>informatique</dept>
    </emp>
  </employes>
  <departements>
    <dept>
      <nom>informatique</nom>
      <contact>Mme Machin 45-76-77-09-54</contact>
      <chef>1</chef>
    </dept>
  </departements>
</ent>
```



# Importation de schémas

- Un schéma est associé à un espace de noms cible
- L'élément `import` permet de faire référence à des composants d'un schéma qui appartient à un autre espace de noms que le schéma dans lequel on fait référence à ces composants.
- Dans l'exemple qui suit, on utilise un composant du schéma de XHTML pour notre propre schéma.

# Exemple d'importation de schémas

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.lifl.fr/~yroos/schema"
  xmlns:art="http://www.lifl.fr/~yroos/schema"
  xmlns:html="http://www.w3.org/1999/xhtml"
  elementFormDefault="qualified"
>
  <xsd:import namespace="http://www.w3.org/1999/xhtml"
    schemaLocation="http://www.w3.org/2002/08/xhtml/xhtml1-strict.xsd"/>
...
```

# Exemple d'importation de schémas

```
...  
<xsd:element name="dansRevue">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element name="auteur" maxOccurs="unbounded"  
        type="xsd:string"/>  
      <xsd:element name="revue" type="xsd:string"/>  
      <xsd:element name="titre" type="xsd:string"/>  
      <xsd:element minOccurs="0" name="resume" type="html:Block"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>  
</xsd:schema>
```

# Exemple d'instance de ce schéma

```
<?xml version="1.0" encoding="utf-8"?>
<bibliographie xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.lifl.fr/~yroos/schema"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xsi:schemaLocation="http://www.lifl.fr/~yroos/schema
  http://saxo.lifl.fr/~yroos/schema/articles.xsd"
>
  <dansRevue>
    <auteur>Tryphon Tournesol</auteur>
    <revue>Revue de Physique</revue>
    <titre>Ma machine à voyager dans le temps</titre>
    <resume><html:div>et patati <html:br/>et patata</html:div></resume>
  </dansRevue>
</bibliographie>
```

# En conclusion

## Bonne pratique

L'utilisation des espaces de noms peut parfois être un peu compliquée. Quand un espace de nom est très clairement majoritaire dans le document XML (que ce soit une instance ou un schéma), on le définit comme espace de nom par défaut, sinon une bonne pratique est de lier (dans un premier temps) tous les espaces de noms à des préfixes et de ne définir un espace de nom par défaut que lorsqu'il y en a un qui se *détache* (c.a.d. quand on tape majoritairement toujours le même préfixe).