

# NameSpace, Espace de nom

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [DTD - Schema](#)

Les espaces de noms d'[XML](#) permettent de qualifier de manière unique des éléments et des attributs. On sait alors à quel domaine de définition se rapporte un objet et comment il doit être interprété, selon sa spécification.

Le seul fait d'utiliser l'espace de noms `xml` pour un attribut `lang` (voir [XML Language](#)) permet, par exemple, de définir la langue de codage d'un document (ou d'une portion de document), ainsi que la façon dont l'attribut se propage sur un sous-arbre.

Par ailleurs, différencier des espaces de noms permet de faire coopérer, dans un même document, des objets ayant le même nom, mais une signification différente, souvent liée à un modèle de contenu différent. Par exemple, il devient possible, avec les espaces de noms, de faire coopérer, dans un même document, des [OASIS Tables](#) et [HTML](#) : les deux spécifications utilisent toutes les deux un élément `table` de plus haut niveau qui sera, par exemple, codé `oasis:table` et `html:table`.

Cette spécification est une avancée importante car, à partir du moment où beaucoup de formats s'expriment selon [XML](#) (voir [SVG](#), par exemple), les risques de "collision de noms" deviennent plus importants et cette spécification prend toute son importance.

Le seul regret, dans l'attente des [Schema](#) et de leur implémentation dans les outils, est que cette spécification ne vienne qu'au-dessus de [XML](#) dans le cadre d'une architecture en couches et que, donc, elle ne soit pas prise en compte dans les [DTD](#).

## Objectifs

---

L'objectif des espaces de noms est de donner aux éléments et aux attributs des noms uniques, sur Internet. Cette identification permet d'avoir une connaissance mutuelle d'objets, entre plusieurs [DTD](#), sans pour autant qu'il y ait contradiction entre l'interprétation des éléments et, surtout, leurs contenus.

L'utilité des espaces de noms est de pouvoir avancer sur les notions de documents composites, quelle que soit la nature des contenus des documents. Par exemple, supposons un document qui contienne ses propres paragraphes (nommés  $\mathcal{P}$ ) et qui doive par ailleurs comporter des fragments de contenus [HTML](#). [HTML](#) définissant ses propres paragraphes (aussi nommés  $\mathcal{P}$ ), il peut y avoir contradiction entre les deux définitions. La

notion d'espace de noms lève cette contradiction et permet aux deux éléments de coopérer. L'un, par exemple celui issu du modèle [HTML](#), sera préfixé par son espace de noms : `html:P`.

Cette solution des espaces de noms est très intéressante, dès lors que l'on s'intéresse aux processus de traitements d'un document [XML](#). En effet, grâce aux espaces de noms, il est possible d'avoir des traitements factorisés et dans l'exemple, de faire confiance à un système de consultation [HTML](#), afin d'interpréter directement les objets de l'espace de noms [HTML](#). Il devient alors possible de redéfinir des contextes de traitement, sans pour autant être obligé de modifier tous les attributs des objets de ce contexte.

Différents exemples, dans différents domaines, permettent de mieux comprendre l'utilité des espaces de noms :

- pour le [W3C](#), les espaces de noms servent à faire coopérer ses différentes recommandations ; par exemple, à pouvoir avoir une feuille de styles utilisant en même temps [XSLT](#), [HTML](#) et [XSL](#), sans risques de collision de noms ;

- pour le monde documentaire, si on utilise des modèles de fragments complexes, comme ceux définis pour les graphiques vectoriels ([SVG](#)), pour les notations typographiques mathématiques ([MathML](#)) ou, encore, pour les tableaux ([OASIS Tables](#)), la notion de factorisation de traitement prend toute sa valeur, car afficher un graphique, un tableau ou une équation n'est pas une opération triviale. L'intérêt est donc de pouvoir partager une variété vaste d'outils ;

- cette factorisation de compréhension et, donc, de traitements s'applique ensuite à tous les domaines. Par exemple, dans les applications EDI, il est intéressant de pouvoir construire une base de données qui contienne les éléments nécessaires à un EDI particulier, ces éléments coopérant avec d'autres, plus localisés à des *process* internes d'entreprise. Il en va de même, et de façon plus générale, dans le monde du commerce électronique ;

- enfin, tous les *portails* s'occupant de syndication de données sont intéressés par les espaces de noms. Cela leur permet, d'une part, de savoir d'où viennent les données qu'ils présentent, et, d'autre part, de se reposer sur le site source pour connaître les moyens de les traiter (par exemple, pour les afficher). Ainsi, un site commercial sera capable d'avoir des documents mixtes contenant une information *prix*, selon le modèle du portail, et une information *commerciale*, directement issue du modèle de l'entreprise source, utilisant les outils de cette entreprise pour l'affichage.

## Principes

---

On peut définir un espace de noms sur n'importe quel élément d'un document : son champs d'utilisation est alors celui du sous-arbre délimité par l'élément. Pour ce faire, il suffit d'ajouter à l'élément, un attribut identifiant l'espace.

Par exemple, `<monélément xmlns:mutuxml="http://www.mutu-xml.org/DTDProjet" version="1.0">` définit un espace de noms dénommé `http://www.mutu-xml.org/DTDProjet`; cette déclaration définit aussi que tous les éléments de cet espace de noms seront préfixés par `mutuxml`. Ainsi, il sera possible d'écrire :

```
<monélément xmlns:mutuxml="http://www.mutu-xml.org/DTDProjet/1.0">
  <p>un paragraphe ... de l'espace de noms par défaut</p>
  <mutuxml:p>un paragraphe ... de l'espace de noms mutuxml</p>
</monélément>
```

... où deux éléments `p` cohabitent, appartenant à deux espaces de noms distincts.

Dans cet exemple, la seule donnée identifiant l'espace de noms est l'URI ! Si, par exemple, un autre document définit un espace de noms avec : `xmlns:mualisationxml="http://www.mutu-xml.org/DTDProjet/1.0"`, le

paragraphe `<mutuxml:p>` de l'exemple ci-devant est bien le même type d'objet que `<mutualisationxml:p>`, de cette nouvelle définition. Le préfixe est donc peu signifiant, sauf dans le champ de codage syntaxique défini par l'élément sur lequel l'espace est défini.

Pour finir, il est fondamental de noter qu'un espace de noms n'a qu'une valeur lexicale : il sert à "désambigüiser" des noms d'objets (éléments ou attributs). En effet, la recommandation n'oblige pas à lier des mots d'un espace de noms à un modèle, qu'il soit sous forme de **DTD** ou de **Schema**. Ceci est en outre intéressant, car cela peut s'appliquer aussi aux documents "bien formés". Ceci est gênant dès lors qu'une application utilise des modèles documentaires sous forme de **DTD**, car il existe peu de choses définies quant à l'intégration de modèles liés à un espace de noms.

## Recommandations(s)

---

### ■ ■ *Les espaces de nommage dans XML 1.1*

Recommandation, version 20040204, du 04-02-2004

Document sur <http://www.yoyodesign.org/doc/w3c/xml-names11/>

### 🇺🇸 *Namespaces in XML*

Recommandation, version 1.1 (Third Edition), du 08-12-2009

Document sur <http://www.w3.org/TR/xml-names/>

### 🇺🇸 *Namespaces in XML 1.1 Requirements*

Projet en cours, version 1.1, du 03-04-2002

Document sur <http://www.w3.org/TR/xml-names11-req/>

### 🇺🇸 *Uniform Resource Names (URN) Namespace Definition Mechanisms*

Projet en cours, version 200202, du 02-2002

Document sur <http://www.ietf.org/rfc/rfc3406.txt>