

The logo for Tirème SARL, featuring the company name in white, uppercase letters on a dark blue rectangular background.

# Infoset, XML Information Set

Rédaction : Pierre Attar

Pour comprendre [Infoset](#), il faut se souvenir qu'[XML](#) définit un format d'échange de données structurées selon leur modèle. Ce format d'échange est, par définition, contenu dans un fichier séquentiel comportant un début et une fin. [Infoset](#) est alors la vue "*désérialisation*" d'un fichier [XML](#) : c'est alors la structure logique d'un document décrit par la syntaxe [XML](#).

La spécification définit le jeu de données abstrait qu'il est nécessaire de construire à partir de la lecture d'un document [XML](#) "bien formé", lors de sa représentation sous forme d'arbre d'objets typés et ordonnés. Cette représentation est celle que doit présupposer n'importe quelle application manipulant de l'information [XML](#) : les chargeurs [XML](#) sont alors supposés reporter, d'une façon ou d'une autre, ces éléments d'informations aux applications.

Le jeu d'informations différencie ce qui est absolument nécessaire (document, élément, attribut, *processing instruction*, entités, caractères, notations, déclarations d'espaces de noms) et ce qui est périphérique ([DTD](#), commentaires, etc.).

## Objectifs

---

L'objectif de cette spécification est de faire partager à des spécifications de plus haut niveau, une vision commune de ce qui est significatif dans un document [XML](#) (par exemple, [DOM Niveau 3](#) est basé sur l'[Infoset](#)). Tel que défini dans la spécification, [Infoset](#) est alors la représentation commune à tous les outils d'un document [XML](#).

Le fait de définir une représentation commune est extrêmement important, cela à plusieurs titres. En tout premier, les développeurs d'applications [XML](#) se noient, aujourd'hui, dans tous les modèles de données proposés représentant ce qui est réellement contenu dans un document, après *désérialisation*. Du coup, écrire des programmes est souvent ardu, il faut savoir quel outil on utilise pour se souvenir des données qu'il manipule. Par ailleurs, avoir un modèle de données partagé permet aux outils de création d'informations [XML](#) de mieux ajuster la façon dont ils utilisent et délivrent les fichiers [XML](#). Enfin, et comme dans le cas de la spécification [Canonical](#), cette représentation commune permettra une simplification des outils, puisque le modèle supprime les différentes formes possibles pour le codage [XML](#) d'une même information logique. Par exemple, [Infoset](#) permet de voir la même information logique dans les codages suivants : `<test type="essai">`, `<test type='essai'>`, `<test type="ess&#97;i">`, `<test type="ess&#x61;i">`, `<test type='ess&#x61;i'>`. Ceci est possible, car il existe dans l'[Infoset](#) une normalisation du codage des caractères.

À noter que [XPath](#) et, donc, [XSLT](#) n'utilisent pas exactement le même jeu d'informations, du fait de la non-disponibilité de cette spécification au moment de l'élaboration de ces deux standards. Cependant, l'annexe B de [XPath](#) explique comment interpréter l'arbre abstrait [XPath](#) en fonction d'[Infoset](#).

## Principes

---

Les différents éléments d'informations reconnus sont les suivants :

- . document (obligatoire),
- . éléments (obligatoire),
- . attributs (obligatoire),
- . processing instructions (obligatoire),
- . références à des entités non lues par un *parser* non validant (obligatoire),
- . caractères (obligatoire),
- . notation (obligatoire),
- . déclaration d'espaces de noms (obligatoire),
- . [DTD](#) (périphérique),
- . entités générales déclarées dans la [DTD](#) (obligatoire pour les entités non *parsées*, sinon, périphérique),
- . début et fin de localisation d'entité incluse (périphérique),
- . début et fin de de section définie comme étant `CDATA` (périphérique),
- . commentaires (périphérique).

À chaque élément d'information est lié un ensemble de propriétés, qui diffère selon l'élément d'information. Par exemple, pour un élément, les propriétés définies s'intéressent à son espace de nom et son nom, à son parent, ses fils et ses attributs, aux déclarations d'espaces de noms sur cet élément et à tous les espaces de noms propagés par ses parents, à l'[URI](#) de base (voir [XML Base](#)), si elle existe, propagée par ses parents.

[Infoset](#) s'intéresse à un document *parsé* et validé. Du coup, aucune information n'est donnée sur la [DTD](#), si ce n'est, de façon optionnelle, sa désignation. Charge aux programmes de récupérer la [DTD](#) elle-même s'ils en ont besoin.

Pour conclure, beaucoup de débats existent sur l'utilité d'une telle spécification. Par exemple, si [Infoset](#) présuppose une organisation sous forme d'arbre, avec des noeuds comportant des parents et des fils, certains rejettent cette vision d'[XML](#), en argumentant que lorsqu'ils échangent des champs de tables issues de bases de données, il n'ont pas besoin de toute cette information liée à des relations dans un arbre.

De fait, la spécification est très controversée et beaucoup de débats ont lieu sur le sujet (pour plus de détails sur les argumentaires, voir la liste de débat [[Monthly Archives for xml-dev](#)], dans ses [[Archives de juillet 2000](#)] et ses [[Archives d'août 2000](#)] 2000).

Il semble cependant important de disposer de ce type de spécification, dès lors que beaucoup de recommandations du [W3C](#) s'intéressent au traitement des documents [XML](#). En effet, quelle serait la validité d'une spécification de traitement, si le modèle de donnée utilisé était laissé au libre choix de l'implémenteur ?

## Recommandations(s)

---


### [L'ensemble d'information XML \(Infoset\)](#)

Recommandation, version 20011024 , du 24-10-2001

Document sur <http://www.a525g.com/programmation/xml-infoset.htm>

### [XML Information Set \(Second Edition\)](#)

Recommandation, version second edition, du 04-02-2004  
Document sur <http://www.w3.org/TR/xml-infoset/>

 *XML Information Set Requirements*  
Note, version 19990218, du 18-02-1999  
Document sur <http://www.w3.org/TR/NOTE-xml-infoset-req>