

Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 7303
 RFC rendue obsolète : 3023
 RFC mise à jour : 6339
 Catégorie : En cours de normalisation
 ISSN : 2070-1721

H. Thompson, University of Edinburgh
 C. Lilley, W3C

juillet 2014
 Traduction Claude Brière de L'Isle

Types de supports XML

Résumé

La présente spécification normalise trois types de supports -- application/xml, application/xml-external-parsed-entity, et application/xml-dtd – pour les utiliser dans les échanges d'entités de réseau qui se rapportent au langage de balisage extensible (XML, *Extensible Markup Language*) tout en définissant text/xml et text/xml-external-parsed-entity comme des alias des application/types respectifs. La présente spécification normalise aussi le suffixe '+xml' pour désigner des types de supports en dehors de ces cinq types lorsque ces types de supports représentent des entités MIME XML.

Statut de ce mémoire

Ceci est un document Internet en cours de normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Plus d'informations sur les normes de l'Internet sont disponibles à la Section 2 de la [RFC5741].

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à <http://www.rfc-editor.org/info/rfc7303>

Notice de droits de reproduction

Copyright (c) 2014 IETF Trust et les personnes identifiées comme auteurs du document. Tous droits réservés.

Le présent document est soumis au BCP 78 et aux dispositions légales de l'IETF Trust qui se rapportent aux documents de l'IETF (<http://trustee.ietf.org/license-info>) en vigueur à la date de publication de ce document. Prière de revoir ces documents avec attention, car ils décrivent vos droits et obligations par rapport à ce document. Les composants de code extraits du présent document doivent inclure le texte de licence simplifié de BSD comme décrit au paragraphe 4.e des dispositions légales du Trust et sont fournis sans garantie comme décrit dans la licence de BSD simplifiée.

Le présent document peut contenir des matériaux provenant de documents de l'IETF ou de contributions à l'IETF publiées ou mises à la disposition du public avant le 10 novembre 2008. La ou les personnes qui ont le contrôle sur les droits de reproduction de ces matériaux peuvent ne pas avoir accordé à l'IETF Trust le droit de permettre des modifications de ces matériaux en dehors du processus de normalisation de l'IETF. En l'absence d'une licence adéquate de la ou des personnes qui contrôlent les droits de reproduction de ces matériaux, le présent document ne peut pas être modifié en dehors du processus de normalisation de l'IETF, et des travaux dérivés ne peuvent être créés en dehors du processus de normalisation de l'IETF, excepté pour le formater pour publication comme RFC ou pour le traduire dans des langues autres que l'anglais.

Table des Matières

1. Introduction.....	2
2. Conventions de notation.....	2
2.1 Langage des exigences.....	2
2.2 Caractères, codages, jeux de caractères.....	3
2.3 Entités MIME, entités XML.....	3
3. Considérations de codage.....	3
3.1 Producteurs MIME XML.....	4
3.2 Consommateurs MIME XML.....	4
3.3 Conversions de BOM et de codage.....	5
4. Types de supports XML.....	5
4.1 Entités MIME XML.....	5
4.2 Utiliser '+xml' lors de l'enregistrement des types de supports fondés sur XML.....	6
4.3 Directives pour l'enregistrement des types de supports fondés sur XML qui n'utilisent pas '+xml'.....	7
5. Identifiants de fragments.....	7
6. URI de base.....	8

7. Versions XML.....	8
8. Exemples.....	8
8.1 Jeu de caractères UTF-8.....	9
8.2 Jeu de caractères UTF-16.....	9
8.3 Jeu de caractères omis et entité MIME de huit bits.....	9
8.4 Jeu de caractères omis et entité MIME de 16 bits.....	9
8.5 Jeu de caractères omis, pas de déclaration de codage interne.....	10
8.6 Jeu de caractères UTF-16BE.....	10
8.7 Jeu de caractères non UTF.....	10
8.8 Exemple incohérent : jeu de caractères et déclaration de codage interne en conflit.....	10
8.9 Exemple incohérent : jeu de caractères et BOM en conflit.....	11
9. Considérations relatives à l'IANA.....	11
9.1 Enregistrement d'application/xml.....	11
9.2 Enregistrement de text/xml.....	12
9.3 Enregistrement de application/xml-external-parsed-entity.....	12
9.4 Enregistrement de text/xml-external-parsed-entity.....	12
9.5 Enregistrement de application/xml-dtd.....	12
9.6 Convention de dénomination '+xml' pour types de supports fondés sur XML.....	13
10. Considérations sur la sécurité.....	13
11. Références.....	14
11.1 Références normatives.....	14
11.2 Références pour information.....	16
Appendice A. Pourquoi utiliser le suffixe '+xml' pour les types MIME fondés sur XML ?.....	17
Appendice B. Cœur des spécifications XML.....	17
Appendice C. Considérations de fonctionnement.....	17
C.1 Considérations générales.....	17
C.2 Considérations pour les producteurs.....	18
C.3 Considérations pour les consommateurs.....	18
Appendice D. Changements depuis la RFC 3023.....	18
Appendice E. Remerciements.....	18
Adresse des auteurs.....	19

1. Introduction

Le World Wide Web Consortium (W3C) a produit les spécifications du langage de balisage extensible (XML, *Extensible Markup Language*) version 1.0 [XML] et version 1.1 [XML1.1]. Pour permettre l'échange d'entités réseau XML, la présente spécification normalise trois types de supports (`application/xml`, `application/xml-external-parsed-entity`, et `application/xml-dtd`), deux alias (`text/xml` et `text/xml-external-parsed-entity`), et une convention de dénomination pour identifier les types de supports MIME fondés sur XML (utilisant '+xml').

XML a été utilisé comme base pour d'autres types de supports, incluant des types dans chaque branche de l'arborescence des types de supports de l'IETF. Pour faciliter le traitement de ces types, et en ligne avec la reconnaissance dans la [RFC6838] des suffixes de noms syntaxiques structurés, un suffixe de '+xml' est enregistré au paragraphe 9.6. Cela permet aux outils génériques fondés sur XML – navigateurs, éditeurs, moteurs de recherche, et autres processeurs – de travailler avec tous les types de supports fondés sur XML.

La présente spécification remplace la [RFC3023]. Les différences majeures sont dans les domaines de l'alignement de `text/xml` et `text/xml-external-parsed-entity` avec respectivement `application/xml` et `application/xml-external-parsed-entity`, l'ajout de XPointer et XML de base comme respectivement identifiants de fragment et URI de base, l'intégration du registre XPointer, et la mise à jour de nombreuses références.

2. Conventions de notation

2.1 Langage des exigences

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2.2 Caractères, codages, jeux de caractères

XML (dans une déclaration XML ou Text utilisant le pseudo attribut de codage) et MIME (dans un champ d'en-tête Content-Type utilisant le paramètre charset) utilisent tous deux un ensemble commun d'étiquettes [IANA-CHARSETS] pour identifier le jeu de caractères MIME (transposant de flux d'octets en séquence de caractères [RFC2978]).

Dans la présente spécification, on utilise les expressions "paramètre de jeux de caractères" et "déclaration de codage" pour se référer à tout jeu de caractères MIME spécifié respectivement par un paramètre MIME de jeu de caractères ou une déclaration de codage XML. On réserve l'expression "codage de caractère" (ou, lorsque le contexte rend l'intention claire, simplement "codage") pour le jeu de caractères MIME réellement utilisé dans une entité MIME XML particulière.

[UNICODE] définit trois "formes de codage", à savoir UTF-8, UTF-16, et UTF-32. Comme UTF-8 peut seulement être mis en série d'une seule façon, la seule étiquette possible pour les documents codés en UTF-8 lorsque mis en série dans des entités MIME est "utf-8". Cependant, les documents XML UTF-16 peuvent être mis en série dans des entités MIME d'une de deux façons : soit gros boutien, étiqueté (facultativement) "utf-16" ou "utf-16be", soit petit boutien, étiqueté (facultativement) "utf-16" ou "utf-16le". Voir au paragraphe 3.3 comment une marque d'ordre des octets (BOM, *Byte Order Mark*) est exigée lorsque la mise en série "utf-16" est utilisée.

UTF-32 a quatre mises en série potentielles, dont deux seulement (UTF-32BE et UTF-32LE) sont des noms donnés dans [UNICODE]. La prise en charge des diverses mises en série varie largement, et des soucis de sécurité sur leur utilisation ont été évoqués (par exemple, dans [Sivonen]). L'utilisation de UTF-32 est NON RECOMMANDÉE pour les entités MIME XML.

2.3 Entités MIME, entités XML

Comme il arrive parfois entre deux communautés, MIME et XML ont tous deux défini le terme d'entité, avec une signification différente. Le paragraphe 2.4 de la [RFC2045] dit : le terme "entité", se réfère spécifiquement aux champs d'en-tête définis par MIME et au contenu soit d'un message soit d'une des parties dans le corps d'une entité multiparties.

La Section 4 de [XML] dit : un document XML peut consister en une ou plusieurs unités de mémorisation. Elles sont appelées entités ; elles ont toutes un contenu et sont toutes (sauf les entités document et le sous ensemble de DTD (déclaration de type de données) externes) identifiées par un nom d'entité.

Dans cette spécification, "entité MIME XML" est défini comme une entité XML encapsulée dans une entité MIME (la seconde définition dans la première).

De plus, XML assure la dénomination et le référencement des entités pour les besoins d'inclusion et/ou substitution. Dans la présente spécification, "déclaration/référence/... d'entité XML" est utilisé pour éviter la confusion lorsque on se réfère à de tels cas.

3. Considérations de codage

Les enregistrements ci-dessous traitent les questions qui tournent autour du codage de caractère de la même façon, en faisant référence à cette Section.

Jusqu'à trois sources d'informations distinctes sur le codage de caractère peuvent être présentes pour une entité MIME XML : un paramètre "charset" (*jeu de caractères*), une BOM (voir au paragraphe 3.3), et une déclaration de codage XML (voir le paragraphe 4.3.3 de [XML]). Assurer la cohérence entre ces sources exige une coordination entre les auteurs d'entité et les agents MIME (c'est-à-dire les processus qui mettent en paquets, transfèrent, livrent, et/ou reçoivent des entités MIME).

L'utilisation de UTF-8, sans une BOM, est RECOMMANDÉE pour toutes les entités MIME XML.

Certains agents MIME seront ce qu'on appelle "à capacité XML", c'est-à-dire, capables de traiter les entités MIME XML comme XML et de détecter la déclaration de codage XML (ou son absence). Les trois sources d'informations sur le codage leur sont disponibles, et on peut s'attendre à ce qu'elles connaissent la présente spécification.

D'autres agents MIME n'auront pas de capacité XML ; donc, ils ne peuvent rien savoir de la déclaration de codage XML. Non seulement il leur manque une des trois sources d'informations sur le codage, mais il est très probable qu'ils ne connaîtront pas la présente spécification.

Certains agents MIME, comme les mandataires et les transcodeurs, consomment et produisent des entités MIME.

Ce mélange de deux sortes d'agents qui traitent des entités MIME XML augmente la complexité de la tâche de coordination. Les recommandations qu'on donne ci-dessous sont destinées à maximiser l'interopérabilité en face de ceci : d'un côté, en rendant obligatoire une production cohérente et en encourageant des formes de production d'une robustesse maximale et, d'un autre côté, en spécifiant des stratégies de récupération pour maximiser l'interopérabilité des consommateurs lorsque les règles de production sont violées.

3.1 Producteurs MIME XML

Les producteurs MIME à capacité XML DEVRAIENT fournir un paramètre de jeux de caractères et/ou une BOM appropriée avec des entités MIME XML non codées en UTF-8 qui n'ont pas une déclaration de codage. De tels producteurs DEVRAIENT retirer ou corriger une déclaration de codage qui est connue pour être incorrecte (par exemple, par suite d'un transcodage).

Les producteurs MIME à capacité XML DOIVENT fournir une déclaration de texte XML au début des entités XML non UNICODE analysées en externe qui autrement commenceraient par les séquences d'octets hexadécimaux 0xFE 0xFF, 0xFF 0xFE ou 0xEF 0xBB 0xBF, afin d'éviter la détection erronée d'une BOM.

Les producteurs MIME sans capacité XML NE DOIVENT PAS fournir un paramètre de jeux de caractères avec une entité MIME XML sauf si le codage de caractère de l'entité est connu de façon fiable. Noter que ceci est particulièrement pertinent pour la configuration centrale des serveurs de la Toile, où la configuration d'une valeur par défaut pour les paramètres de jeux de caractères va presque certainement violer cette exigence.

Il est RECOMMANDÉ aux producteurs MIME XML de fournir aux utilisateurs les moyens de contrôler quelle valeur, si il en est, est donnée aux paramètres de jeux de caractères pour les entités MIME XML, par exemple, en donnant aux utilisateurs le contrôle de la configuration des transpositions de nom de fichier de serveur de la Toile en en-tête de type de contenu, fichier par fichier, ou sur la base du suffixe.

3.2 Consommateurs MIME XML

Pour les consommateurs MIME XML, la question de la priorité se pose dans des cas où les informations de codage de caractère disponibles ne sont pas cohérentes. Là encore, on doit distinguer les agents à capacité XML et sans capacité XML.

Lorsque un paramètre de jeux de caractères est spécifié pour une entité MIME XML, le composant normatif de la spécification [XML] laisse la question ouverte quant à la façon de déterminer le codage avec lequel tenter de traiter l'entité. Ceci est vrai indépendamment de si l'entité contient ou non des informations de codage dans la bande, c'est-à-dire, soit une BOM (paragraphe 3.3) soit une déclaration de codage XML, les deux, ou aucune. En particulier, dans le cas où il y a des informations dans la bande et qu'elles entrent en conflit avec le paramètre de jeux de caractères, la spécification [XML] ne spécifie pas qui a le dernier mot. Dans son Appendice F (non normatif) elle renvoie la réponse à la présente spécification : "la méthode préférée de traitement de conflit devrait être spécifiée au titre du protocole de couche supérieure utilisé pour livrer XML. En particulier, se référer à la [RFC3023] ou son successeur..."

En conséquence, pour se conformer aux processeurs et contenus déployés et éviter d'entrer en conflit avec cette spécification normative ou d'autres, la présente spécification établit la priorité comme suit : une BOM (paragraphe 3.3) est d'autorité si elle est présente dans une entité MIME XML ; en l'absence d'une BOM, le paramètre de jeux de caractères est d'autorité, si présent.

Chaque fois qu'il est déterminé qu'une source d'informations de codage est d'autorité, les consommateurs DEVRAIENT traiter les entités MIME XML sur la base de ces informations.

Lorsque les producteurs MIME se conforment aux exigences de la Section 3 et du paragraphe 3.1, il n'y aura pas d'incohérence (la déclaration des priorités n'a d'impact pratique que dans le cas d'entités MIME XML non conformes). En présence d'incohérences, aucune stratégie uniforme ne peut donner chaque fois la "bonne" réponse : on en spécifie une ici pour encourager la convergence au fil du temps, d'abord de la part des consommateurs, puis de la part des producteurs.

Pour les consommateurs à capacité XML, on notera que le paragraphe 4.3.3 de [XML] ne fait pas une erreur que le paramètre de jeux de caractères et la déclaration de codage XML (ou l'UTF-8 par défaut en l'absence de déclaration de codage et de BOM) soient incohérentes, bien que de tels consommateurs puissent choisir de produire un avertissement dans ce cas.

Si une entité MIME XML est reçue où le paramètre de jeux de caractères est omis, aucune information n'est fournie sur le codage de caractère par l'en-tête MIME Content-Type. Les consommateurs à capacité XML DOIVENT suivre les exigences du paragraphe 4.3.3 de [XML] qui traitent directement ce cas. Les consommateurs MIME sans capacité XML NE DEVRAIENT PAS supposer un codage par défaut dans ce cas.

3.3 Conversions de BOM et de codage

Le paragraphe 4.3.3 de [XML] spécifie que les entités MIME XML UTF-16 non étiquetées "utf-16le" ou "utf-16be" DOIVENT commencer par une BOM, U+FEFF, qui apparaît comme la séquence d'octets hexadécimaux 0xFE 0xFF (gros boutien) ou 0xFF 0xFE (petit boutien). [XML] déclare de plus que la BOM est une signature de codage et ne fait partie ni du balisage ni des données de caractères du document XML.

À cause de la présence de la BOM, les applications qui convertissent XML de l'UTF-16 en un codage autre que UTF-8 DOIVENT effacer la BOM avant la conversion. De même, lors de la conversion d'un autre codage en UTF-16, soit sans un paramètre de jeux de caractères soit étiqueté "utf-16", la BOM DOIT être ajoutée sauf si le codage d'origine était UTF-8 et qu'une BOM était déjà présente, auquel cas elle DOIT être transcodée dans la BOM UTF-16 appropriée.

Le paragraphe 4.3.3 de [XML] permet aussi que les entités MIME XML UTF-8 commencent par une BOM, qui apparaît comme la séquence d'octets hexadécimaux 0xEF 0xBB 0xBF. Ceci est aussi défini comme étant une signature de codage, et ne faisant partie ni du balisage ni des données de caractère du document XML.

Les applications qui convertissent XML de UTF-8 en un codage autre que UTF-16 DOIVENT effacer la BOM, si présente, avant la conversion. Les applications qui convertissent XML en UTF-8 PEUVENT ajouter une BOM.

En plus du jeu de caractères MIME "utf-16", la [RFC2781] introduit "utf-16le" (petit boutien) et "utf-16be" (gros boutien). Lorsque une entité MIME XML est codée en "utf-16le" ou "utf-16be", elle NE DOIT PAS commencer par la BOM mais DEVRAIT contenir une déclaration de codage XML dans la bande. La conversion de UTF-8 ou UTF-16 (non étiqueté, ou étiqueté avec "utf-16") en "utf-16be" ou "utf-16le" DOIT supprimer la BOM si il en est présente. La conversion d'UTF-16 étiqueté "utf-16le" ou "utf-16be" en UTF-16 sans étiquette ou étiqueté "utf-16" DOIT ajouter la BOM appropriée. La conversion de UTF-16 étiqueté "utf-16le" ou "utf-16be" en UTF-8 PEUT ajouter une BOM UTF-8, mais ceci est NON RECOMMANDÉ.

L'Appendice F de [XML] implique aussi qu'une BOM UTF-32 peut être utilisée en conjonction avec des documents codés en UTF-32. Comme noté ci-dessus, la présente spécification recommande de ne pas utiliser UTF-32. Si il est utilisé, les mêmes considérations que pour UTF-16 s'appliquent par rapport à son caractère de signature (ne faisant pas partie du document) du transcodage de ou vers lui, et du transcodage de ou vers les jeux de caractères MIME "utf-32le" et "utf-32be". Les consommateurs qui ne prennent pas en charge UTF-32 DEVRAIENT néanmoins reconnaître les signatures UTF-32 afin de donner d'utiles messages d'erreur (au lieu de les traiter comme de l'UTF-16 invalide).

4. Types de supports XML

4.1 Entités MIME XML

Dans la spécification XML, les entités MIME XML peuvent être classées en quatre types. Dans la terminologie XML, elles sont appelées "entités de document", "sous ensembles de DTD externes", "entités analysées externes", et "entités de paramètre externes". L'usage approprié pour les types enregistrés ci-dessous est le suivant :

entités de document : les types de support application/xml ou text/xml, ou un type de support plus spécifique (voir le paragraphe 9.6) DEVRAIENT être utilisés.

sous ensembles de DTD externes : le type de support application/xml-dtd DEVRAIT être utilisé. Les types de support application/xml et text/xml NE DOIVENT PAS être utilisés.

entités analysées externes : les types de support application/xml-external-parsed-entity ou text/xml-external-parsed-entity DEVRAIENT être utilisés. Les types de support application/xml et text/xml NE DOIVENT PAS être utilisés sauf si les entités analysées sont aussi des "entités de document" bien formées.

entités de paramètre externes : le type de support application/xml-dtd DEVRAIT être utilisé. Les types de support application/xml et text/xml NE DOIVENT PAS être utilisés.

Noter que la [RFC3023] (que la présente spécification rend obsolète) recommandait l'utilisation de text/xml et text/xml-external-parsed-entity pour, respectivement, les entités de document et les entités analysées externes, mais décrivait un traitement du codage de caractère qui différait de la pratique de mise en œuvre courante. Ces types de support sont toujours d'utilisation courante, et la présente spécification aligne le traitement du codage de caractère sur la pratique de l'industrie.

Noter que la [RFC2376] (qui est obsolète) permettait l'utilisation de application/xml et text/xml pour les quatre types, bien qu'en pratique cela ait probablement été rare.

Ni les sous ensembles de déclarations de types de données (DTD) externes ni les entités de paramètre externes ne s'analysent comme documents XML, et bien que certaines entités de document XML puissent être utilisées comme entités analysées externes et vice versa, il y a de nombreux cas où les deux ne sont pas interchangeables. XML a aussi des entités non analysées, des entités analysées internes, et des entités de paramètre internes, mais ce ne sont pas des entités MIME XML.

Comparée à la [RFC2376] ou la [RFC3023], la présente spécification altère le traitement du codage de caractère de text/xml et text/xml-external-parsed-entity, en ne les traitant pas différemment des types application/ respectifs. Cependant, application/xml et application/xml-external-parsed-entity sont toujours RECOMMANDÉS, pour éviter de possibles confusions sur la base de la distinction antérieure. L'ancienne confusion autour de la question des jeux de caractères par défaut pour les deux types text/ ne se produit plus parce que la [RFC7231] change la [RFC2616] en supprimant le ISO-8859-1 par défaut et en ne définissant pas de défaut du tout ; la [RFC6657] met à jour la [RFC2046] pour supprimer le US-ASCII [ASCII] par défaut.

Voir la Section 3 sur l'approche maintenant unifiée de paramètre de jeux de caractères qui en résulte.

XML fournit un cadre général pour définir des séquences de données structurées. Il est souvent approprié de définir de nouveaux types de supports qui utilisent XML mais définissent une application spécifique de XML, à cause de considérations d'affichage, d'édition, de sécurité ou d'informations de démarrage spécifiques du domaine. De plus, de tels types de supports peuvent permettre seulement UTF-8 et/ou UTF-16 et interdire les autres jeux de caractères. La présente spécification n'interdit pas ces types de supports ; en fait, on s'attend à ce qu'ils prolifèrent.

Cependant, il est RECOMMANDÉ aux développeurs de ces types de supports d'utiliser la présente spécification comme base de leur enregistrement. Voir au paragraphe 4.2 des recommandations plus détaillées sur l'utilisation du suffixe '+xml' pour l'enregistrement de ces types de supports.

Un document XML étiqueté comme application/xml ou text/xml, ou avec un type de support '+xml', pourrait contenir des déclarations d'espace de noms, des instructions de traitement reliées à une feuille de style, des informations de schéma, ou autres déclarations qui pourraient être utilisées pour suggérer comment le document doit être traité. Par exemple, un document peut avoir l'espace de noms XHTML et une référence à une feuille de style en cascade (CSS, *Cascading Style Sheet*). Un tel document peut être traité par des applications qui vont utiliser ces informations pour donner au document le traitement approprié. L'Appendice B fait la liste des spécification XML centrales qui, avec [XML] lui-même, montrent comment déterminer la sémantique de niveau langage d'un document XML et suggérer comment localiser les informations sur sa sémantique de niveau application.

4.2 Utiliser '+xml' lors de l'enregistrement des types de supports fondés sur XML

Au paragraphe 9.6, la présente spécification met à jour l'enregistrement de la [RFC6839] pour les types MIME fondés sur XML (types '+xml').

Lorsque un nouveau type de support est introduit pour un format fondé sur XML, le nom du type de support DEVRAIT se terminer par '+xml' sauf si le traitement XML générique n'est pas approprié pour les documents de ce nouveau type. Cette convention permettra aux applications qui peuvent traiter XML générique de détecter que l'entité MIME est supposée être un document XML, de vérifier cette hypothèse en invoquant un processeur XML, et ensuite en traitant le document XML en conséquence. Les applications peuvent vérifier que les types représentent des entités MIME XML en comparant les quatre derniers caractères du sous type de la chaîne '+xml'. (Cependant, noter que quatre des cinq types de supports définis dans la présente spécification -- text/ xml, application/xml, text/xml-external-parsed-entity, et application/xml-external-parsed-entity -- représentent aussi des entités MIME XML bien que ne se terminant pas par '+xml'.)

Note : Le paragraphe 5.3.2 de la [RFC7231] ne prend en charge aucune forme d'en-tête Accept qui ne satisferait qu'aux types '+xml'. En particulier, les en-têtes Accept de la forme "Accept: */*+xml" ne sont pas permis, et ne fonctionneront pas pour cela.

Les types de supports qui satisfont à la convention de dénomination '+xml' DEVRAIENT, pour la cohérence, définir le paramètre de jeux de caractères, car le traitement XML générique traite par définition toutes les entités MIME XML de façon uniforme en ce qui concerne les informations de codage de caractère. Cependant, il y a des cas où le paramètre de jeux de caractères n'a pas besoin d'être défini. Par exemple, lorsque un type de support fondé sur XML est restreint à UTF-8, il n'est pas nécessaire de définir le paramètre de jeux de caractères. UTF-8 est par défaut pour XML. Lorsque un type de support fondé sur XML est restreint à UTF-8 et UTF-16, il n'est pas déraisonnable d'omettre le paramètre de jeux de caractères. Ni UTF-8 ni UTF-16 n'exigent de déclaration de codage XML.

Le traitement XML générique n'est pas toujours approprié pour les types de supports fondés sur XML. Par exemple, les auteurs de certains de ces types de supports peuvent souhaiter que les types restent entièrement opaques sauf pour les applications qui

sont spécifiquement conçues pour traiter ce type de supports. En NE suivant PAS la convention de dénomination '+xml', de tels types de supports peuvent éviter le traitement XML générique. Comme le traitement générique sera cependant utile dans de nombreux cas, y compris dans des situations qu'il est difficile de prédire, la convention '+xml' doit être préférée sauf si il y a une raison impérieuse de ne pas le faire.

Le processus d'enregistrement pour les types de supports '+xml' spécifiques est décrit dans la [RFC6838]. Les nouveaux enregistrements de types de supports fondés sur XML dans l'IETF doivent suivre ces lignes directrices. Lorsque d'autres organisations enregistrent des types de supports fondés sur XML via la politique d'enregistrement de l'IANA "spécification exigée" [RFC5226], le réviseur de supports pertinent doit vérifier qu'elles utilisent la convention '+xml', afin de s'assurer d'une interopérabilité maximum de ces documents fondés sur XML. Seuls les sous types de supports qui représentent des entités MIME XML sont autorisés à s'enregistrer avec le suffixe '+xml'.

En plus des changements décrits ci-dessus, le contrôle des changements a été passé au World Wide Web Consortium (W3C).

4.3 Directives pour l'enregistrement des types de supports fondés sur XML qui n'utilisent pas '+xml'

Les enregistrements de nouveaux types de supports fondés sur XML qui n'utilisent pas le suffixe '+xml' DEVRAIENT, en spécifiant le paramètre de jeux de caractères et les considérations de codage, les définir comme : "même [paramètre de jeux de caractères / considérations de codage] de application/xml que spécifié dans la RFC 7303".

Définir le paramètre de jeux de caractères est RECOMMANDÉ, car cette information peut être utilisée par les processeurs XML pour déterminer d'autorité le codage de caractère de l'entité MIME XML en l'absence de BOM. Si il y a des raisons de ne pas suivre cet avis, elles DEVRAIENT être incluses au titre de l'enregistrement. Comme indiqué ci-dessus, deux de ces raisons sont "UTF-8 seulement" ou "seulement UTF-8 ou UTF-16".

Ces enregistrements DEVRAIENT spécifier que le type de supports fondé sur XML enregistré a toutes les considérations de sécurité décrites dans la présente spécification plus toutes considérations supplémentaires spécifiques de ce type de supports.

Ces enregistrements DEVRAIENT aussi faire référence à la présente spécification en spécifiant des numéros magiques, des URI de base, et l'utilisation de la BOM.

Ces enregistrements PEUVENT faire référence à l'enregistrement application/xml dans ce document en spécifiant les considérations d'interopérabilité et d'identifiant de fragment, si ces considérations ne sont pas outrepassées par des questions spécifiques de ce type de supports.

5. Identifiants de fragments

Les identifiants de ressource universels (URI, *Uniform Resource Identifier*) peuvent contenir des identifiants de fragment (voir le paragraphe 3.5 de la [RFC3986]). Spécifier la syntaxe et la sémantique des identifiants de fragment est dévolu par la [RFC3986] à l'enregistrement de type de supports approprié.

La syntaxe et la sémantique des identifiants de fragment pour les types de supports XML définis dans la présente spécification se fondent sur la recommandation [XPFR] du W3C. Elle permet des noms simples et des constructions plus complexes sur la base de schémas désignés. Lorsque la syntaxe d'une partie identifiant de fragment de tout URI ou identifiant de ressource internationalisé (IRI, *Internationalized Resource Identifier*) ([RFC3987]) avec un type de supports restitué gouverné par la présente spécification se conforme à la syntaxe spécifiée dans [XPFR], les applications conformes DOIVENT interpréter de tels identifiants de fragment comme désignant ce qui est spécifié par le [XPFR] ainsi que par toutes autres spécifications gouvernant les schémas XPointer utilisés dans ces identifiants que les applications prennent en charge. Les applications conformes DOIVENT prendre en charge le schéma 'element' comme défini dans [XPE], mais n'ont pas besoin de prendre en charge d'autres schémas.

Si une erreur XPointer est rapportée dans la tentative de traitement de la partie, la présente spécification ne définit pas d'interprétation pour la partie.

Un registre des schémas XPointer [XPtrReg] est tenu par le W3C. Les processeurs génériques d'entités MIME XML NE DEVRAIENT PAS mettre en œuvre de schémas XPointer non enregistrés ([XPtrRP] décrit les exigences et procédures pour l'enregistrement des schémas). Voir au paragraphe 4.2 les exigences supplémentaires qui s'appliquent lorsque un type de support fondé sur XML suit la convention de dénomination '+xml'.

Si [XPFR] et [XPE] sont inappropriés pour certains types de supports fondés sur XML, ils NE DEVRAIENT PAS suivre la

convention de dénomination '+xml'.

Lorsque un URI a un identifiant de fragment, il est codé par un sous ensemble limité du répertoire des caractères US-ASCII, voir les détails dans [XPFR].

6. URI de base

Une entité MIME XML de type `application/xml`, `text/xml`, `application/xml-external-parsed-entity`, ou `text/xml-external-parsed-entity` PEUT utiliser l'attribut `xml:base`, comme décrit dans [XMLBase], pour incorporer un URI de base dans cette entité pour l'utiliser à résoudre des références d'URI relatives (voir le paragraphe 5.1 de la [RFC3986]).

Noter que l'URI de base lui-même peut être incorporé dans une entité MIME différente, car la valeur par défaut pour l'attribut `xml:base` peut être spécifiée dans un sous ensemble de DTD externe ou une entité de paramètre externe. Comme les processeurs XML conformes n'ont pas toujours besoin de lire et traiter les entités externes, l'effet d'une telle valeur par défaut externe est incertain ; donc, son utilisation N'EST PAS RECOMMANDÉE.

7. Versions XML

"`application/xml`", "`application/xml-external-parsed-entity`", "`application/xml-dtd`", "`text/xml`", et "`text/xml-external-parsed-entity`" sont à utiliser avec [XML]. Dans tous les exemples présentés ici où `version="1.0"` est montré, on comprend que `version="1.1"` peut aussi apparaître, pourvu que le contenu se conforme à [XML1.1].

L'exigence normative de la présente spécification sur les documents et processeurs XML est de respecter les exigences du paragraphe 4.3.3 de [XML].

Sauf pour des éclaircissements mineurs, cette section est en substance identique à la première édition de XML 1.0, et pour XML 1.1 à la première ou seconde édition [XML1.1]. Donc, les références données ici à [XML] peuvent être interprétées comme faisant référence à toute version ou édition existante de XML, ou toute édition ou version suivante qui ne fait pas de changement incompatible à cette section.

Les spécifications et recommandations qui se fondent ou font référence à la présente RFC DEVRAIENT indiquer toutes limitations sur les versions ou éditions particulières de XML à utiliser.

8. Exemples

Cette section n'est pas normative. Noter en particulier que tout le langage de la [RFC2119] reproduit ici résume les conséquences des déclarations normatives déjà faites ci-dessus, et n'a pas de force normative indépendante, et en conséquence elles ne sont pas données en majuscules.

Les exemples ci dessous donnent l'en-tête MIME Content-Type, incluant le paramètre de jeux de caractères, si il est présent et la déclaration XML ou la déclaration Text (qui inclut la déclaration de codage) à l'intérieur de l'entité MIME XML. Pour les exemples en UTF-16, le caractère de marque d'ordre d'octet (BOM) codé de façon appropriée en UTF-16 est noté par "{BOM}", et la déclaration XML ou Text est supposée venir au début de l'entité MIME XML, suivant immédiatement la BOM codée. Noter que d'autres en-têtes MIME peuvent être présents, et l'entité MIME XML va normalement contenir d'autres données en plus de la déclaration XML ; les exemples se concentrent sur l'en-tête Content-Type et la déclaration de codage pour la clarté de l'exposé.

Bien qu'ils montrent un type de contenu de '`application/xml`', tous les exemples ci-dessous s'appliquent à tous les cinq types de supports déclarés à la Section 9, ainsi qu'à tout type de supports déclaré en utilisant la convention '+xml' (à l'exception des exemples qui impliquent le paramètre de jeux de caractères pour tous ces types de support qui ne permettent pas son utilisation). Voir le tableau des entités MIME XML (paragraphe 4.1, alinéa 1) pour la discussion des types qui sont appropriés pour les différentes variétés d'entité MIME XML.

8.1 Jeu de caractères UTF-8

Content-Type: `application/xml; charset=utf-8`


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

ou

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

UTF-8 est le codage recommandé à utiliser avec tous les types de supports définis dans la présente spécification. Comme le paramètre de jeux de caractères est fourni et qu'il n'y a pas de BOM pour l'écraser, les processeurs conformes MIME et XML doivent traiter l'entité enclose comme codée en UTF-8.

Si elle est envoyée en utilisant un transport à 7 bits (par exemple, SMTP [RFC5321]) en général, une entité MIME XML UTF-8 doit utiliser un codage de transfert de contenu de quoted-printable ou de base64. Pour un transport à 8 bits pur (par exemple, 8BITMIME ESMTP ou NNTP) ou un transport binaire pur (par exemple, BINARY ESMTP ou HTTP) aucun codage de transfert de contenu n'est nécessaire (ou même possible, dans le cas de HTTP).

8.2 Jeu de caractères UTF-16

Content-Type: application/xml; charset=utf-16

```
{BOM}<?xml version="1.0" encoding="utf-16"?>
```

ou

```
{BOM}<?xml version="1.0"?>
```

Pour les trois types d'application/supports définis ci-dessus, si ils sont envoyés en utilisant un transport à 7 bits (par exemple, SMTP) ou un transport à 8 bits pur (par exemple, 8BITMIME ESMTP ou NNTP) l'entité MIME XML doit être codée en quoted-printable ou base64 ; pour un transport binaire pur (par exemple, BINARY ESMTP ou HTTP) aucun codage de transfert de contenu n'est nécessaire (ou même possible, dans le cas de HTTP).

Comme décrit dans la [RFC2781], la famille UTF-16 ne doit pas être utilisée avec les types de supports en dessous du type de niveau supérieur "text" sauf sur HTTP ou HTTPS (voir pour les détails le paragraphe A.2 de HTTP [RFC7231]). Donc, un des deux types de text/media définis ci-dessus ne peut être utilisé avec cet exemple que lorsque l'entité MIME XML est transmise via HTTP ou HTTPS, qui utilise un mécanisme de style MIME et sont des protocoles binaires purs et donc n'effectuent pas de transformations de CR et LF et permettent les octets NUL. Comme HTTP est binaire pur, aucun codage de transfert de contenu n'est nécessaire (ou même possible).

8.3 Jeu de caractères omis et entité MIME de huit bits

Content-Type: application/xml

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

Comme le paramètre de jeux de caractères n'est pas fourni dans l'en-tête Content-Type et qu'il n'y a pas de BOM, pour l'écraser, les processeurs XML conformes doivent traiter le codage "iso-8859-1" comme d'autorité. Les processeurs MIME conformes sans capacité XML ne devraient faire aucune hypothèse sur le codage de caractères de l'entité MIME XML.

8.4 Jeu de caractères omis et entité MIME de 16 bits

Content-Type: application/xml

```
{BOM}<?xml version="1.0" encoding="utf-16"?>
```

ou

```
{BOM}<?xml version="1.0"?>
```

Cet exemple montre une entité MIME de 16 bits sans paramètre de jeu de caractères. Cependant, comme il y a une BOM, les processeurs conformes doivent traiter cette entité comme codée en UTF-16.

Omettre le paramètre de jeu de caractères n'est pas recommandé en conjonction avec les types de supports sous le type de

niveau supérieur "application" lorsque utilisé avec des transports autres que HTTP ou HTTPS. Les types de supports sous le type de niveau supérieur "text" ne devraient pas être utilisés pour MIME à 16 bits avec des transports autres que HTTP ou HTTPS (voir la discussion à l'alinéa 7 du paragraphe 8.2).

8.5 Jeu de caractères omis, pas de déclaration de codage interne

Content-Type: application/xml

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

Dans cet exemple, le paramètre de jeux de caractères a été omis, il n'y a pas de déclaration de codage interne, et il n'y a pas de BOM. Comme il n'y a ni BOM ni paramètre de jeux de caractères, le processeur XML suit les exigences du paragraphe 4.3.3, et applique facultativement le mécanisme décrit à l'Appendice F (qui n'est pas normatif) de [XML] pour déterminer un codage de UTF-8. Bien que l'entité MIME XML ne contienne pas de déclaration de codage, étant donné que le codage est en fait UTF-8, c'est une entité MIME XML conforme.

Un processeur MIME conforme sans capacité XML ne devrait faire aucune hypothèse sur le codage de caractère de l'entité MIME XML.

Voir le paragraphe 8.1 pour les questions relatives au transport des entités MIME XML UTF-8.

8.6 Jeu de caractères UTF-16BE

Content-Type: application/xml; charset=utf-16be

```
<?xml version='1.0' encoding='utf-16be'?>
```

On observe que, comme exigé pour ce codage, il n'y a pas de BOM. Comme le paramètre de jeux de caractères est fourni et qu'il n'y a pas de BOM qui l'outrepasse, les processeurs conformes à MIME et XML doivent traiter l'entité enclose comme codée en UTF-16BE.

Voir aussi les considérations supplémentaires dans l'exemple UTF-16 du paragraphe 8.2.

8.7 Jeu de caractères non UTF

Content-Type: application/xml; charset=iso-2022-kr

```
<?xml version="1.0" encoding="iso-2022-kr"?>
```

Cet exemple montre l'utilisation d'un codage de caractère non UTF (dans le cas présent, du Hangul, mais cet exemple est destiné à couvrir tous les codages de caractère qui ne sont pas de la famille UTF-8). Comme le paramètre de jeu de caractères est fourni et qu'il n'y a pas de BOM pour l'outrepasser, les processeurs conformes doivent traiter l'entité enclose comme codée selon la RFC 1557.

Comme ISO-2022-KR [RFC1557] a été défini comme n'utilisant que 7 bits de données, aucun codage de transfert de contenu n'est nécessaire avec tout transport : pour les jeux de caractères qui ont besoin de 8 bits ou plus, les considérations exposées aux paragraphes 8.1 et 8.2 vont s'appliquer.

8.8 Exemple incohérent : jeu de caractères et déclaration de codage interne en conflit

Content-Type: application/xml; charset=iso-8859-1

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

Bien que le paramètre de jeu de caractères soit fourni dans l'en-tête Content-Type et qu'il n'y ait pas de BOM et que le paramètre de jeu de caractères diffère de la déclaration de codage XML, les processeurs MIME et XML conformes vont interopérer. Comme le paramètre de jeu de caractères est d'autorité en l'absence de BOM, les processeurs conformes vont traiter l'entité enclose comme codée en iso-8859-1. C'est-à-dire que la déclaration de codage "UTF-8" sera ignorée.

Les processeurs conformes qui génèrent des entités MIME XML ne doivent pas étiqueter les informations de codage de

caractère en conflit entre le type de contenu MIME et la déclaration XML sauf si ils ont des informations définitives sur le codage réel, par exemple, par suite d'un transcodage systématique. En particulier, l'ajout par les serveurs d'un paramètre de jeu de caractères explicite par défaut pour l'ensemble du site a fréquemment conduit à des problèmes d'interopérabilité pour les documents XML.

8.9 Exemple incohérent : jeu de caractères et BOM en conflit

Content-Type: application/xml; charset=iso-8859-1

```
{BOM}<?xml version="1.0"?>
```

Bien que le paramètre de jeu de caractères soit fourni dans l'en-tête Content-Type, il y a une BOM, de sorte que les processeurs MIME et XML peuvent ne pas interopérer. Comme le paramètre BOM est d'autorité pour les processeurs XML conformes, ils vont traiter l'entité enclose comme codée en UTF-16. C'est-à-dire que le paramètre de jeu de caractères "iso-8859-1" sera ignoré. Les processeurs MIME sans capacité XML peuvent par ailleurs ne pas connaître les BOM et traiter l'entité comme codée en iso-8859-1.

Les processeurs conformes qui génèrent des entités MIME XML ne doivent pas étiqueter les informations de codage de caractère conflictuelles entre l'en-tête MIME Content-Type et une BOM d'entité initiale.

9. Considérations relatives à l'IANA

9.1 Enregistrement d'application/xml

Nom de type : application

Nom de sous type : xml

Paramètres exigés : aucun

Paramètres facultatifs : charset (voir la Section 3)

Considérations de codage : selon le type de codage de caractère utilisé, les entités MIME XML peuvent consister en données de 7 bits, 8 bits, ou binaires [RFC6838]. Pour les transports à 7 bits, les données de 7 bits, par exemple, données codées en US-ASCII, n'exigent pas de codage de transfert de contenu, mais les données à 8 bits ou binaires, par exemple, les données UTF-8 ou UTF-16, DOIVENT avoir un codage de transfert de contenu en "quoted-printable" ou "base64". Pour le transport en 8 bits pur (par exemple, 8BITMIME ESMTP [RFC6152] ou NNTP [RFC3977]) les données à 7 bits ou 8 bits, par exemple, les données US-ASCII ou UTF-8, n'exigent pas de codage de transfert de contenu, mais les données binaires, par exemple, les données avec un codage UTF-16, DOIVENT recevoir un codage de transfert de contenu en base64. Pour les transports en binaire pur (par exemple, BINARY ESMTP [RFC3030] ou HTTP [RFC7230]) aucun codage de transfert de contenu n'est nécessaire (ou même possible, dans le cas de HTTP) pour les données à 7 bits, 8 bits, ou binaires.

Considérations de sécurité : voir la Section 10.

Considérations d'interopérabilité : XML s'est révélé interopérable sur les applications aussi bien génériques et que spécifiques d'une tâche et pour l'import/export à partir de plusieurs outils de rédaction et d'édition XML. Les processeurs de validation fournissent une interopérabilité maximum, parce qu'ils doivent traiter tous les aspects de XML. Bien qu'un processeur non valideur puisse être plus efficace, il ne peut pas traiter tous les aspects. Pour plus d'informations, voir le paragraphe 2.9 "Déclaration de document autonome" et la Section 5 "Conformité" de [XML]. En pratique, les questions de jeu de caractères se sont révélées être la plus grosse source de problèmes d'interopérabilité. L'utilisation de UTF-8, et une bonne attention aux directives de la Section 3, sont le meilleur moyen d'éviter de tels problèmes.

Spécification publiée : Langage de balisage extensible (XML) 1.0 (cinquième édition) [XML] ou ses éditions/versions suivantes.

Applications qui utilisent ce type de support : XML est neutre quant aux appareils, plateformes, et fabricants, et est pris en charge par les applications génériques et spécifiques d'une tâche et une large gamme d'outils XML génériques (éditeurs, analyseurs, agents de la Toile, ...).

Informations supplémentaires :

Numéros magiques : aucun. Bien qu'on ne puisse pas compter qu'une séquence d'octets soit toujours présente, les entités MIME XML dans les jeux de caractères compatibles ASCII (incluant UTF-8) commencent souvent par l'hexadécimal 3C 3F 78 6D 6C ("<?xml"), et celles en UTF-16 commencent souvent par l'hexadécimal FE FF 00 3C 00 3F 00 78 00 6D 00 6C ou FF FE 3C 00 3F 00 78 00 6D 00 6C 00 (la BOM suivie par "<?xml"). Pour plus d'informations, voir l'Appendice F de [XML].

Extensions de fichier : .xml

Codes de type de fichier Macintosh : "TEXT"

URI de base : voir la Section 6

Adresse personnelle et de messagerie pour plus d'informations : voir la Section "Adresse des auteurs.

Utilisation prévue : COMMUNE

Auteur : voir la Section "Adresse des auteurs.

Contrôleur des changements : la spécification XML est un produit du travail du groupe de travail XML Core du World Wide Web Consortium. Le W3C a le contrôle des changements sur la RFC 7303.

9.2 Enregistrement de text/xml

Les informations d'enregistrement pour text/xml sont en tous points les mêmes que celles données pour application/xml ci-dessus (paragraphe 9.1) sauf que le "Nom de type" est "text".

9.3 Enregistrement de application/xml-external-parsed-entity

Nom de type : application

Nom de sous type : xml-external-parsed-entity

Paramètres exigés : aucun

Paramètres facultatifs : charset. Voir la Section 3.

Considérations de codage : les mêmes que pour application/xml (paragraphe 9.1).

Considérations de sécurité : voir la Section 10.

Considérations d'interopérabilité : les entités XML analysées externes sont aussi interopérables que les documents XML, bien qu'elles aient une structure moins étroitement contrainte et donc doivent être référencées par les documents XML pour un traitement approprié par les processeurs XML. De même, les documents XML ne peuvent pas être utilisés de façon fiable comme entités analysées externes parce que il est interdit que les entités analysées externes aient des déclarations de document ou DTD autonomes. Identifier les entités analysées externes XML avec leur propre type de contenu améliore l'interopérabilité des documents XML et des entités analysées externes XML.

Spécification publiée : la même que pour application/xml (paragraphe 9.1).

Applications qui utilisent ce type de support : les mêmes que pour application/xml (paragraphe 9.1).

Informations supplémentaires :

Numéros magiques : les mêmes que pour application/xml (paragraphe 9.1).

Extensions de fichier : .xml ou .ent

Codes de type de fichier Macintosh : "TEXT"

URI de base : voir la Section 6

Adresse personnelle et de messagerie pour plus d'informations : voir la Section "Adresse des auteurs.

Utilisation prévue : COMMUNE

Auteur : voir la Section "Adresse des auteurs.

Contrôleur des changements : la spécification XML est un produit du travail du groupe de travail XML Core du World Wide Web Consortium. Le W3C a le contrôle des changements sur la RFC 7303.

9.4 Enregistrement de text/xml-external-parsed-entity

Les informations d'enregistrement pour text/xml-external-parsed-entity sont en tous points les mêmes que celles données pour application/xml-external-parsed-entity ci-dessus (paragraphe 9.3) sauf que le "Nom de type" est "text".

9.5 Enregistrement de application/xml-dtd

Nom de type : application

Nom de sous type : xml-dtd

Paramètres exigés : aucun

Paramètres facultatifs : charset. Voir la Section 3.

Considérations de codage : les mêmes que pour application/xml (paragraphe 9.1).

Considérations de sécurité : voir la Section 10.

Considérations d'interopérabilité : les DTD XML se sont montrés interopérables par les outils de rédaction de DTD et les valideurs XML, entre autres.

Spécification publiée : la même que pour application/xml (paragraphe 9.1).

Applications qui utilisent ce type de support : les outils de rédaction de DTD traitent les sous ensembles de DTD externes ainsi que les entités de paramètres externes. Les valideurs XML peuvent aussi accéder aux sous ensembles de DTD externes et aux entités de paramètres externes.

Informations supplémentaires :

Numéros magiques : les mêmes que pour application/xml (paragraphe 9.1).

Extensions de fichier : .dtd ou .mod

Codes de type de fichier Macintosh : "TEXT"

Adresse personnelle et de messagerie pour plus d'informations : voir la Section "Adresse des auteurs.

Utilisation prévue : COMMUNE

Auteur : voir la Section "Adresse des auteurs.

Contrôleur des changements : la spécification XML est un produit du travail du groupe de travail XML Core du World Wide Web Consortium. Le W3C a le contrôle des changements sur la RFC 7303.

9.6 Convention de dénomination '+xml' pour types de supports fondés sur XML

Ce paragraphe se substitue à l'enregistrement précédent du suffixe '+xml' [RFC6839].

La présente spécification recommande l'utilisation des conventions de dénomination '+xml' pour identifier les types de supports fondés sur XML, en ligne avec la reconnaissance dans la [RFC6838] de suffixes de noms de syntaxe structurés. Cela permet l'utilisation de processeurs et technologies XML génériques sur une grande variété de types de documents XML différents à un coût minimum, en utilisant les cadres existants pour l'enregistrement de type de supports.

Voir au paragraphe 4.2 des directives sur quand et comment enregistrer un sous type de supports qui est fondé sur '+xml', et au paragraphe 4.3 sur l'enregistrement de sous type de supports pour XML mais n'utilisant pas '+xml'.

9.6.1 Enregistrement du suffixe de syntaxe structurée '+xml'

Nom : Extensible Markup Language (XML)

+suffix : +xml

Référence : RFC 7303

Considérations de codage : les mêmes qu'au paragraphe 9.1.

Considérations d'identifiant de fragment : les enregistrements qui utilisent cette convention '+xml' DOIVENT aussi faire référence au présent document, spécifiquement sa Section 5, en spécifiant la syntaxe et la sémantique des identifiants de fragments, et ils PEUVENT restreindre la syntaxe à un sous ensemble spécifié de schémas, sauf qu'ils NE DOIVENT PAS interdire les noms nus ou les pointeurs de schéma 'element'. Ils PEUVENT de plus exiger la prise en charge d'autres schémas enregistrés. Ils PEUVENT aussi ajouter de la syntaxe supplémentaire (qui NE DOIT PAS se chevaucher avec la syntaxe de [XPFR]) ainsi que sa sémantique associée, et ils PEUVENT ajouter de la sémantique pour les noms nus de XPointers qui, comme mentionné à la Section 5, ne s'appliquera que lorsque le présent document ne définit pas une interprétation. En pratique, ces contraintes impliquent que pour un identifiant de fragment adressé dans une instance d'un type "xxx/yyy+xml" spécifique, il y a trois cas :

- pour les identifiants de fragment qui satisfont à la syntaxe définie dans [XPFR], où l'identifiant de fragment se résout selon les règles qui y sont spécifiées, ils sont traités comme il y est spécifié ;

- pour les identifiants de fragment qui satisfont à la syntaxe définie dans [XPFR], où l'identifiant de fragment ne se résout pas selon les règles qui y sont spécifiées, ils sont traités comme spécifié dans "xxx/yyy+xml" ;

- pour les identifiants de fragment qui ne satisfont pas à la syntaxe définie dans [XPFR], ils sont traités comme spécifié dans "xxx/yyy+xml". Un identifiant de fragment de la forme "xywh=160,120,320,240", comme défini dans [MediaFragments], qui peut être utilisé dans un URI pour une image codée en XML, va tomber dans cette catégorie.

Considérations d'interopérabilité : les mêmes qu'au paragraphe 9.1. Voir ci-dessus, et aussi Section 3, des directives sur l'utilisation du paramètre 'charset'.

Considérations de sécurité : voir la Section 10.

Contact : voir la Section "Adresse des auteurs.

Auteur : voir la Section "Adresse des auteurs.

Contrôleur des changements : la spécification XML est un produit du travail du groupe de travail XML Core du World Wide Web Consortium. Le W3C a le contrôle des changements sur la RFC 7303.

10. Considérations sur la sécurité

Les entités MIME XML contiennent des informations qui peuvent être analysées et ensuite traitées par le receveur. Ces entités peuvent contenir, et les receveurs peuvent permettre, que des commandes explicites de niveau système soient exécutées tout en traitant les données. Dans la mesure où une application receveuse exécute des chaînes de commandes arbitraires à partir de l'intérieur des entités MIME XML, il peut y avoir un risque.

En général, toutes les informations mémorisées hors du contrôle direct de l'utilisateur – incluant des feuilles de style CSS, des

transformations XSL, des déclarations d'entités XML, et des DTD – peuvent être une source d'insécurité, par des moyens évidents ou plus subtils. Par exemple, une petite modification "d'attaque d'embrouillage" faite à une feuille de style "maîtresse" pourrait faire disparaître des mots dans des positions critiques dans les documents de l'utilisateur, sans modifier directement le document ou la feuille de style référencée. Donc, la sécurité de tout document XML est dans une dépendance vitale à l'égard de tous les documents référencés de façon récurrente par ce document.

Les listes d'entités XML et les DTD pour XHTML 1.0 [XHTML], par exemple, sont probablement un ensemble de ressources largement exploité. Elles vont être utilisées et être de confiance pour de nombreux développeurs, dont peu vont être au fait du niveau de sécurité sur les serveurs du W3C, ou sur tout répertoire d'une confiance similaire.

La plus simple attaque suppose d'ajouter des déclarations qui cassent la validation. Ajouter des déclarations étrangères à une liste de caractères des entités XML peut effectivement "casser le contrat" utilisé par les documents. Un petit changement qui produit une erreur fatale dans une DTD pourrait arrêter le traitement XML sur une grande échelle. Des déclarations étrangères sont très visibles, mais des tours plus sophistiqués, comme de changer des attributs de facultatifs en exigés, peut être difficile à découvrir. Peut-être l'option la plus dangereuse disponible aux attaquants, quand sont impliquées des sous ensembles de DTD externes ou des entités de paramètres externes ou d'autres spécifiés par défaut en externe, est de redéfinir des valeurs par défaut pour des attributs : par exemple, si les développeurs se sont appuyés sur des attributs par défaut pour la sécurité, un changement relativement mineur peut exposer d'énormes quantités d'informations.

À part les possibilités structurelles, une autre option, "maquillage d'entité XML," peut être utilisée pour insérer du texte dans des documents, les vandaliser et peut-être convoyer un message inattendu. Parce que XML permet plusieurs déclarations d'entités XML, et que la première déclaration prend l'avantage, il est possible d'insérer du contenu malveillant là où est utilisée une référence d'entité XML, comme en insérant le texte complet de Winnie l'ourson à la place de chaque occurrence de —.

Les considérations de sécurité vont varier selon le domaine d'utilisation. Par exemple, les enregistrement médicaux en XML vont avoir des considérations de confidentialité et de sécurité plus strictes que des métadonnées de bibliographie XML. De même, l'utilisation de XML comme syntaxe de contrôle des paramètres nécessite un examen cas par cas de la sécurité.

XML peut aussi avoir les mêmes soucis de sécurité que le texte en clair. Comme le texte en clair, XML peut contenir des séquences d'échappement qui, lorsque elles sont affichées, ont le potentiel de changer l'environnement du processeur d'affichage d'une façon nuisible aux opérations suivantes. Les effets possibles incluent, mais sans s'y limiter, de verrouiller le clavier, de changer les paramètres d'affichage de sorte que le texte affiché soit ensuite illisible, ou même de changer les paramètres d'affichage pour obscurcir délibérément ou déformer le matériel affiché ensuite de sorte que sa signification soit perdue ou altérée. Les processeurs d'affichage DEVRAIENT soit filtrer de tels matériaux dans le texte affiché soit s'assurer de rétablir tous les réglages importants après l'achèvement d'une certaine opération d'affichage.

Avec certains terminaux, l'envoi d'une séquence de caractères particulière au processeur d'affichage peut changer le résultat de la frappe ultérieure des touches au clavier. Si cela est possible, l'affichage d'un objet de texte contenant de telles séquences de caractères pourrait reprogrammer les clés pour effectuer une action illicite ou dangereuse lors de la frappe ultérieure de la touche par l'utilisateur. Dans certains cas, non seulement les clés peuvent être programmées, mais elles peuvent être déclenchées à distance, rendant possible qu'une opération d'affichage de texte effectuée directement une action non désirée. À ce titre, la capacité de programmer les touches DEVRAIT être bloquée soit par filtrage, soit en désactivant entièrement la capacité de programmer les touches du clavier.

Noter qu'il est aussi possible de construire des documents XML qui utilisent ce que XML appelle des "références d'entité [XML-]" pour construire des expansions répétées de texte. Les expansions récurrentes sont interdites par [XML] et les processeurs XML sont obligés de les détecter. Cependant, même les expansions non récurrentes peuvent causer des problèmes aux ressources de calcul finies des ordinateurs, si elles sont effectuées de nombreuses fois. Par exemple, considérer le cas où une entité XML A consiste en 100 copies de l'entité XML B, qui à son tour consiste en 100 copies de l'entité XML C, et ainsi de suite.

11. Références

11.1 Références normatives

[IANA-CHARSETS] IANA, "Character Sets Registry", 2013, < <http://www.iana.org/assignments/character-sets/> >.

[RFC2045] N. Freed et N. Borenstein, "[Extensions de messagerie Internet](#) multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. (*D. S., MàJ par 2184, 2231, 5335.*)

- [RFC2046] N. Freed et N. Borenstein, "[Extensions de messagerie Internet](#) multi-objets (MIME) Partie 2 : Types de support", novembre 1996. (*D. S., MàJ par 2646, 3798, 5147, 6657.*)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2781] P. Hoffman et F. Yergeau, "[UTF-16, un codage de la norme ISO 10646](#)", février 2000.
- [RFC2978] N. Freed et J. Postel, "Procédures d'enregistrement des jeux de caractère par l'IANA", BCP 19, octobre 2000.
- [RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "[Identifiant de ressource uniforme](#) (URI) : Syntaxe générique", STD 66, janvier 2005.
- [RFC3987] M. Duerst et M. Suignard, "[Identifiant de ressource internationalisé](#) (IRI)", janvier 2005.
- [RFC6657] A. Melnikov, J. Reschke, "Mise à jour de MIME concernant le traitement du paramètre "charset" dans les types de support textuels", juillet 2012. (*MàJ la RFC2046*) (*P.S.*)
- [RFC6838] N. Freed, J. Klensin et T. Hansen, "Spécifications de type de support et procédures d'enregistrement", BCP 13, janvier 2013.
- [RFC6839] T. Hansen et A. Melnikov, "Suffixes syntaxiques structurés de type de support supplémentaires", janvier 2013. (*Info*)
- [RFC7230] R. Fielding, et J. Reschke, "Protocole de transfert Hypertexte (HTTP/1.1) : syntaxe et acheminement de message", juin 2014. (*Remplace RFC 2145, 2616, MàJ RFC 2617, 2618*) (*P.S.*)
- [RFC7231] R. Fielding, et J. Reschke, "Protocole de transfert Hypertexte (HTTP/1.1) : sémantique et contenu", juin 2014. (*Remplace RFC2616, MàJ RFC2617*) (*P.S.*)
- [UNICODE] The Unicode Consortium, "The Unicode Standard, Version 7.0.0", (Mountain View, CA. 2014 ISBN 978-1-936213-09-2), <<http://www.unicode.org/versions/Unicode7.0.0/>>.
- [XML] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation REC-xml, novembre 2008, <<http://www.w3.org/TR/2008/REC-xml-20081126/>>. Dernière version disponible à <<http://www.w3.org/TR/xml>>.
- [XML1.1] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., Yergeau, F., and J. Cowan, "Extensible Markup Language (XML) 1.1 (Second Edition)", W3C Recommendation REC-xml, septembre 2006, <<http://www.w3.org/TR/2006/REC-xml11-20060816/>>. Dernière version à <<http://www.w3.org/TR/xml11>>.
- [XMLBase] Marsh, J. and R. Tobin, "XML Base (Second Edition)", W3C Recommendation REC-xmlbase-20090128, janvier 2009, <<http://www.w3.org/TR/2009/REC-xmlbase-20090128/>>. Dernière version disponible à <<http://www.w3.org/TR/xmlbase>>.
- [XPE] Grosso, P., Maler, E., Marsh, J., and N. Walsh, "XPointer element() Scheme", W3C Recommendation REC-XPointer-Element, mars 2003, <<http://www.w3.org/TR/2003/REC-xptr-element-20030325/>>. Dernière version disponible à <<http://www.w3.org/TR/xptr-element>>.
- [XPFR] Grosso, P., Maler, E., Marsh, J., and N. Walsh, "XPointer Framework", W3C Recommendation REC-XPointer-Framework, mars 2003, <<http://www.w3.org/TR/2003/REC-xptr-framework-20030325/>>. Dernière version disponible à <<http://www.w3.org/TR/xptr-framework>>.
- [XPtrReg] Hazael-Massieux, D., "XPointer Registry", 2005, <<http://www.w3.org/2005/04/xpointer-schemes/>>.
- [XPtrRP] Hazael-Massieux, D., "XPointer Scheme Name Registry Policy", 2005, <<http://www.w3.org/2005/04/xpointer-policy.html>>.

11.2 Références pour information

- [ASCII] American National Standards Institute, "Coded Character Set -- 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.

- [AWWW] Jacobs, I. et N. Walsh, "Architecture of the World Wide Web, Volume One", W3C Recommendation REC-webarch-20041215, décembre 2004, < <http://www.w3.org/TR/2004/REC-webarch-20041215/> >. Dernière version disponible à < <http://www.w3.org/TR/webarch/> >.
- [FYN] Mendelsohn, N., "The Self-Describing Web", W3C TAG Finding selfDescribingDocuments-2009-02-07, février 2009, < <http://www.w3.org/2001/tag/doc/selfDescribingDocuments-2009-02-07.html> >. Dernière version disponible à < <http://www.w3.org/2001/tag/doc/selfDescribingDocuments.html> >.
- [Infoset] Cowan, J. et R. Tobin, "XML Information Set (Second Edition)", W3C Recommendation REC-xml-infoset-20040204, février 2004, < <http://www.w3.org/TR/2004/REC-xml-infoset-20040204/> >. Dernière version disponible à < <http://www.w3.org/TR/xml-infoset/> >.
- [MediaFragments] Troncy, R., Mannens, E., Pfeiffer, S., et D. Van Deursen, "Media Fragments URI 1.0 (basic)", W3C Recommendation media-frag, septembre 2012, < <http://www.w3.org/TR/2012/REC-media-frag-20120925/> >. Dernière version disponible à < <http://www.w3.org/TR/media-frag/> >.
- [RFC1557] U. Choi, K. Chon et H. Park, "Codage des caractères coréens pour les messages Internet", décembre 1993. (*Info.*)
- [RFC2376] E. Whitehead et M. Murata, "Types de support XML", juillet 1998.
- [RFC2616] R. Fielding et autres, "[Protocole de transfert hypertexte -- HTTP/1.1](#)", juin 1999. (*D.S., MàJ par 2817, 6585*)
- [RFC3023] M. Murata, S. St.Laurent et D. Kohn, "Types de support XML", janvier 2001. (*Obsolète, voir RFC7303*)
- [RFC3030] G. Vaudreuil, "Extensions de service SMTP pour la [transmission de grands messages MIME binaires](#)", décembre 2000. (*P.S.*)
- [RFC3977] C. Feather, "[Protocole de transfert de nouvelles](#) du réseau (NNTP)", octobre 2006. (*P.S.*)
- [RFC5226] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, mai 2008. (*Remplace RFC2434*)
- [RFC5321] J. Klensin, "[Protocole simple de transfert de messagerie](#)", octobre 2008. (*D.S.*)
- [RFC6152] J. Klensin et autres, "Extension du service SMTP pour le transport MIME à 8 bits", mars 2011. (*Remplace la RFC1652, aussi STD0071*)
- [Sivonen] Sivonen, H. and others, "Mozilla bug: Remove support for UTF-32 per HTML5 spec", octobre 2011, <https://bugzilla.mozilla.org/show_bug.cgi?id=604317#c6>.
- [TAGMIME] Bray, T., Ed., "Internet Media Type registration, consistency of use", avril 2004, <<http://www.w3.org/2001/tag/2004/0430-mime>>.
- [XHTML] Pemberton, S. and al, "XHTML 1.0: The Extensible HyperText Markup Language", W3C Recommendation xhtml1, décembre 1999, < <http://www.w3.org/TR/2000/REC-xhtml1-20000126/> >. Dernière version disponible à <<http://www.w3.org/TR/xhtml1/>>.
- [XMLModel] Grosso, P. and J. Kosek, "Associating Schemas with XML documents 1.0 (Third Edition)", W3C Working Group Note NOTE-xml-model-20121009, octobre 2012, <<http://www.w3.org/TR/2012/NOTE-xml-model-20121009/>>. Dernière version disponible à < <http://www.w3.org/TR/xml-model> >.
- [XMLNS10] Bray, T., Hollander, D., Layman, A., Tobin, R., and H. Thompson, "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation REC-xml-names-20091208, décembre 2009, < <http://www.w3.org/TR/2009/REC-xml-names-20091208/> >. Dernière version disponible à < <http://www.w3.org/TR/xml-names> >.
- [XMLNS11] Bray, T., Hollander, D., Layman, A., and R. Tobin, "Namespaces in XML 1.1 (Second Edition)", W3C Recommendation REC-xml-names11-20060816, août 2006, < <http://www.w3.org/TR/2006/REC-xml-names11-20060816/> >. Dernière version disponible à < <http://www.w3.org/TR/xml-names11> >.
- [XMLSS] Clark, J., Pieters, S., and H. Thompson, "Associating Style Sheets with XML documents 1.0 (Second Edition)", W3C Recommendation REC-xml-stylesheet-20101028, octobre 2010, < <http://www.w3.org/TR/2010/REC-xml-stylesheet-20101028/> >. Dernière version disponible à < <http://www.w3.org/TR/xml-stylesheet> >.

[XMLid] Marsh, J., Veillard, D., and N. Walsh, "xml:id Version 1.0", W3C Recommendation REC-xml-id-20050909, septembre 2005, < <http://www.w3.org/TR/2005/REC-xml-id-20050909/> >. Dernière version disponible à <<http://www.w3.org/TR/xml-id>>.

Appendice A. Pourquoi utiliser le suffixe '+xml' pour les types MIME fondés sur XML ?

La [RFC3023] contient une discussion de la nouvelle (à l'époque) utilisation d'un suffixe, une pratique qui s'est depuis largement répandue. Ceux qui sont intéressés par une perspective historique de ce sujet se reporteront à l'Appendice A de la [RFC3023].

Le processus d'enregistrement pour les nouveaux types de support '+xml' est décrit dans la [RFC6838].

Appendice B. Cœur des spécifications XML

Les spécifications suivantes décrivent chacune un aspect clé de la sémantique de document XML :

- Espaces de noms en XML 1.0 [XMLNS10] / Espaces de noms en XML 1.1 [XMLNS11]
- Ensemble d'informations XML [Infoset]
- Identifiant xml:id [XMLid]
- XML de base [XMLBase]
- Association de feuille de style aux documents XML [XMLSS]
- Association de schémas aux documents XML [XMLModel]

Le groupe d'architecture technique du W3C a produit deux documents qui sont aussi pertinents :

- La Toile auto descriptive (Self-Describing Web) [FYN] discute des principes globaux de la façon dont la sémantique des document est déterminée sur la Toile.
- Architecture de la Toile mondiale, volume un [AWWW], paragraphe 4.5.4, discute du rôle spécifique des documents d'espace de noms XML dans ce processus.

Appendice C. Considérations de fonctionnement

Cette section donne un résumé informel des considérations de fonctionnement majeures qui surviennent lors de l'échange d'entités MIME XML sur un réseau.

C.1 Considérations générales

L'existence à la fois d'agents à capacité XML et sans capacité XML traitant des entités MIME XML peut compromettre l'interopérabilité. Les mandataires de transcodage génériques font courir un risque particulier à cet égard. On trouvera des conseils détaillés sur le traitement des BOM lors du transcodage au paragraphe 3.3.

La présente spécification exige des consommateurs XML qu'ils traitent les BOM comme d'autorité : c'est en principe une rétro incompatibilité. En pratique, il existe déjà de sérieuses questions d'interopérabilité lorsque des BOM sont utilisées. Rendre les BOM d'autorité, en conjonction avec la désapprobation de la forme de codage UTF-32 et l'exigence d'inclure une déclaration de codage XML dans certains cas (paragraphe 3.1) est destiné à améliorer en pratique l'interopérabilité autant que possible au fil du temps.

La présente spécification établit la Section 5 comme base de l'interprétation des URI pour les entités MIME XML qui incluent des identifiants de fragment, rend la prise en charge obligatoire seulement pour les fragments abrégés ("nom simple") de schéma 'element' et déconseille la prise en charge de schémas XPointer non enregistrés par des processeurs d'entité MIME XML. En conséquence, les URI vont mieux interopérer si ils utilisent seulement des identifiants de fragment de noms simples et de schéma 'element', avec des schémas enregistrés qui varient largement quant au degré de prise en charge à trouver dans les outils génériques. Les auteurs de schéma XPointer ne peuvent s'attendre à une prise en charge par les outils génériques que si ils enregistrent leurs schémas.

C.2 Considérations pour les producteurs

L'interopérabilité pour toutes les entités MIME XML est maximisée par l'utilisation de UTF-8, sans BOM. Lorsque UTF-8

n'est pas utilisé, un paramètre de jeu de caractères et/ou une BOM améliore l'interopérabilité, en particulier lorsque des consommateurs sans capacité XML peuvent être impliqués.

Dans le cas très rare où la substance du contenu d'une entité analysée externe XML non UNICODE commence par la séquence d'octets hexadécimaux 0xFE 0xFF, 0xFF 0xFE ou 0xEF 0xBB 0xBF, l'inclusion d'une déclaration de texte XML va devancer la détection erronée d'une BOM.

L'utilisation de UTF-32 pour des entités MIME XML met l'interopérabilité en grand danger.

Les configurations de serveur de la Toile qui fournissent des paramètres de jeux de caractères fautifs risquent de mal représenter les entités MIME XML. Permettre aux utilisateurs de contrôler la valeur des paramètres de jeu de caractères améliore l'interopérabilité.

Fournir un paramètre de jeu de caractères erroné est pire que de n'en pas fournir du tout. En particulier, les processeurs génériques tels que les transcodeurs, lorsque ils traitent sur la base de paramètres de jeu de caractères erroné, si ils n'échouent pas d'eux-mêmes, vont probablement produire des résultats arbitrairement bogués à partir desquels l'original n'est pas récupérable.

C.3 Considérations pour les consommateurs

Les consommateurs d'entités MIME XML peuvent maximiser l'interopérabilité en :

1. prenant une BOM comme d'autorité si elle est présente dans une entité MIME XML ;
2. en l'absence de BOM, prenant un paramètre de jeu de caractères comme d'autorité si il est présent.

Supposer un codage de caractère par défaut en l'absence d'un paramètre de jeu de caractères nuit à l'interopérabilité.

Bien que la prise en charge de UTF-32 ne soit pas exigée par [XML] lui-même, et que la présente spécification déconseille son utilisation, les consommateurs qui vérifient les BOM UTF-32 peuvent ainsi éviter de traiter à tort des entités UTF-32 comme des entités (invalides) UTF-16.

Appendice D. Changements depuis la RFC 3023

Il y a des différences nombreuses et significatives entre la présente spécification et la [RFC3023], qu'elle rend obsolète. Cet appendice résume seulement les différences majeures :

XPointer ([XPFr] et [XPE]) ont été ajoutés comme syntaxe d'identifiant de fragment pour tous les types de supports XML, et le registre XPointer ([XPtrReg]) mentionné.

[XMLBase] a été ajouté comme mécanisme pour spécifier les URI de base.

Le langage concernant les jeux de caractères a été mis à jour pour correspondre à un TAG W3C trouvant un enregistrement de type de support Internet, cohérent avec l'utilisation [TAGMIME].

La priorité est maintenant donnée à une BOM si elle est présente.

De nombreuses références sont mises à jour, et l'existence de XML 1.1 et sa pertinence pour cette spécification sont reconnues.

Un certain nombre de justifications et de contextualisations qui étaient appropriées lorsque XML était nouveau ont été supprimées, incluant la totalité de l'Appendice A d'origine.

Appendice E. Remerciements

Murata Makoto et Alexey Melnikov ont fait des contributions importantes et précoces à l'effort de révision de la [RFC3023].

La présente spécification reflète les apports de nombreux participants aux listes de diffusion ietf-xml-mime@imc.org, xml-mime@ietf.org, et apps-discuss@ietf.org, mais les erreurs sont de la responsabilité des auteurs. Un merci particulier à Mark Baker, James Clark, Dan Connolly, Martin Duerst, Ned Freed, Yaron Goland, Bjoern Hoehrmann, Rick Jelliffe, Murray S. Kucherawy, Larry Masinter, David Megginson, S. Moonesamy, Keith Moore, Chris Newman, Gavin Nicol, Julian Reschke, Marshall Rose, Jim Whitehead, Erik Wilde, et aux participants à l'activité XML et au TAG du W3C.

Jim Whitehead et Simon St. Laurent étaient les éditeurs, respectivement, de la [RFC2376] et de la [RFC3023].

Adresse des auteurs

Henry S. Thompson
University of Edinburgh
mél : ht@inf.ed.ac.uk
URI : <http://www.ltg.ed.ac.uk/~ht/>

Chris Lilley
World Wide Web Consortium
2004, Route des Lucioles - B.P. 93 06902
Sophia Antipolis Cedex
France
mél : chris@w3.org
URI : <http://www.w3.org/People/chris/>