

Groupe de travail Réseau

J. Urpalainen, Nokia

Request for Comments : 5261

Catégorie : Sur la voie de la normalisation

septembre 2008

Traduction Claude Brière de L'Isle

Cadre des opérations Patch du langage de balisage extensible (XML) utilisant les sélecteurs de langage de chemin XML (XPath)

Statut du présent mémoire

Le présent document spécifie un protocole Internet sur la voie de la normalisation pour la communauté de l'Internet. Il appelle à la discussion et à des suggestions pour son amélioration. Prière de se référer à l'édition actuelle des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

Les documents en langage de balisage extensible (XML, *Extensible Markup Language*) sont largement utilisés comme conteneurs pour l'échange et la mémorisation de données arbitraires dans les systèmes d'aujourd'hui. Afin d'envoyer des changements à un document XML, une copie complète de la nouvelle version doit être envoyée, sauf si il y a un moyen pour indiquer seulement les portions qui ont changé. Le présent document décrit un cadre de correction XML qui utilise des sélecteurs de langage de chemin XML (XPath, *XML Path language*). Ces valeurs de sélecteur et le contenu des nouvelles valeurs mises à jour constituent la base des opérations de correction décrites dans ce document. De plus, avec les directives de base <add>, <replace>, et <remove>, un ensemble de corrections peuvent alors être appliquées pour mettre à jour un document XML existant.

Table des matières

1. Introduction.....	2
2. Conventions.....	2
3. Caractéristiques et exigences de base.....	3
4. Opérations Patch.....	3
4.1 Localisation de la cible d'une correction.....	4
4.2 Séparation d'espaces de noms.....	4
4.3 Élément <add>.....	6
4.4 Élément <replace>.....	8
4.5 Élément <remove>.....	9
5. Traitement d'erreur.....	10
5.1 Éléments d'erreur.....	10
6. Usage des opérations Patch.....	11
7. Usage des valeurs de sélecteur.....	12
8. Types de schéma XML des éléments d'opération Patch.....	12
9. Schéma XML des erreurs d'opération Patch.....	14
10. Considérations relatives à l'IANA.....	15
10.1 Enregistrement de sous espace de noms d'URN.....	15
10.2 Type MIME application/patch-ops-error+xml.....	16
10.3 Enregistrement de schéma XML Patch-Ops-Types.....	16
10.4 Enregistrement de schéma XML Patch-Ops-Error.....	16
11. Considérations pour la sécurité.....	16
12. Remerciements.....	16
13. Références.....	17
13.1 Références normatives.....	17
13.2 Références pour information.....	17
Appendice A. Exemples pour information.....	18
A.1 Ajout d'un élément.....	18
A.2 Ajout d'un attribut.....	18
A.3 Ajout d'une déclaration d'espace de noms préfixé.....	18
A.4 Ajout d'un nœud de commentaire avec l'attribut 'pos'.....	19
A.5 Ajout de plusieurs nœuds.....	19
A.6 Remplacement d'un élément.....	20
A.7 Remplacement d'une valeur d'attribut.....	20

A.8 Remplacement d'un URI de déclaration d'espace de noms.....	20
A.9 Remplacement d'un nœud de commentaire.....	21
A.10 Remplacement d'un nœud d'instruction de traitement.....	21
A.11 Remplacement d'un nœud de texte.....	22
A.12 Suppression d'un élément.....	22
A.13 Suppression d'un attribut.....	22
A.14 Suppression d'une déclaration d'espace de noms préfixé.....	23
A.15 Suppression d'un nœud de commentaire.....	23
A.16 Suppression d'un nœud d'instruction de traitement.....	24
A.17 Suppression d'un nœud de texte.....	24
A.18 Plusieurs corrections avec écrasement d'espace de noms.....	24
Adresse de l'auteur.....	25
Déclaration complète de droits de reproduction.....	26

1. Introduction

Les documents en langage de balisage extensible (XML, *Extensible Markup Language*) [xml-20060816] sont largement utilisés comme conteneurs pour l'échange et la mémorisation de données arbitraires dans les systèmes d'aujourd'hui. Pour envoyer des changements à un document XML, une copie entière de la nouvelle version doit être envoyée, sauf si il y a un moyen d'indiquer seulement les portions qui ont changé (les corrections).

Le présent document décrit un cadre de correction XML qui utilise des sélecteurs de langage de chemin XML (XPath, *XML Path language*) [xpath-19991116]. Un sélecteur XPath est utilisé pour pointer la portion spécifique du XML qui est la cible du changement. Ces valeurs de sélecteur et le contenu des nouvelles données mises à jour constituent la base des opérations de correction décrites dans le présent document. En plus de ces valeurs, avec les directives de base <add>, <replace>, et <remove> un ensemble de corrections peut être appliqué pour mettre à jour un document XML cible existant. Avec des opérations de correction, une sémantique simple pour les documents XML en mode données [xmlschema-2-20041028] est réalisée, c'est-à-dire, des modifications comme des ajouts, des suppressions, ou des substitutions d'éléments et attributs peuvent facilement être effectuées. Le présent document ne décrit pas un format XML diff complet, seulement les éléments de base de l'opération de correction qui peuvent être incorporés dans un format complet qui a normalement une sémantique supplémentaire.

Pour un exemple concret, dans le système de présence fondé sur le protocole d'initialisation de session (SIP, *Session Initiation Protocol*) [RFC3903] un format de document XML PIDF partiel [RFC5262] consiste en le format de document existant de format de données d'informations de présence (PIDF, *Presence Information Data Format*) combiné avec les éléments d'opérations de correction décrits dans le présent document. En général, les opérations de correction peuvent être utilisées dans toute application qui échange des documents XML, par exemple, dans le cadre d'événements SIP [RFC3265]. Un autre exemple est XCAP-diff [RFC5874], qui utilise ce cadre pour envoyer des mises à jour partielles de changements aux ressources XCAP [RFC4825].

2. Conventions

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" dans le présent document sont à interpréter comme décrit dans le BCP 14, [RFC2119] et indiquent les niveaux d'exigence pour les mises en œuvre conformes.

Les termes suivants sont utilisés dans ce document :

Document XML cible : un document XML cible qui va être mis à jour par un ensemble de corrections.

Document XML diff : document XML qui contient des éléments d'opération Patch, des déclarations d'espace de noms, et tous les changements au contenu du document qui sont nécessaire afin de transformer un document XML cible en un nouveau document XML corrigé.

Document XML corrigé : document XML qui résulte de l'application d'une ou plusieurs opérations Patch définies dans le document XML diff au document XML cible.

Opération Patch : un seul changement, c'est-à-dire, une correction qui est appliquée pour mettre à jour un document XML cible.

Élément d'opération Patch : élément XML qui représente une seule opération de correction.

Définition de type pour un élément : définition de type de schéma du consortium de la Toile mondiale (W3C, *World Wide Web Consortium*) pour un élément qui décrit un contenu d'opération de correction.

Déclaration d'espace de noms dans la portée : liste de toutes les déclarations d'espace de noms dans la portée au sein d'un nœud de contexte. L'expansion du nom qualifié (Qname, *qualified name*) d'un nœud de contexte se fonde sur la transposition d'un préfixe avec une de ces déclarations. Pour un élément, un lien d'espace de noms peut avoir un préfixe vide.

Contrainte positionnelle : nombre inclus entre des crochets. Elle peut être utilisée comme prédicat d'étape de localisation.

Nœud cible localisé : nœud qui a été trouvé à partir du document XML cible avec l'aide d'une valeur de sélecteur XPath.

Nœud de texte espace : nœud de texte qui contient seulement une ou des espaces.

3. Caractéristiques et exigences de base

Dans ce cadre, les valeurs de sélecteur XPath et le contenu des nouvelles données sont incorporés dans des éléments XML, dont les noms spécifient la modification à effectuer : <add>, <replace>, ou <remove>. Ces éléments (opérations de correction) sont définis par types de schéma avec le langage de schéma du W3C [xmlschema-1-20041028]. Les sélecteurs XPath pointent sur la cible pour un changement et sont exprimés comme des attributs de ces éléments. Le ou les nœuds fils des éléments d'opération Patch contiennent le nouveau contenu de données. En général quand il est applicable, le nouveau contenu DEVRAIT être déplacé non altéré dans le document XML corrigé.

Les documents XML qui sont équivalents pour les besoins de nombreuses applications PEUVENT différer dans leur représentation physique. Le but du présent document est de décrire un cadre déterministe où la forme canonique avec commentaires [xml-c14n-20010315] d'un document XML détermine une équivalence logique. Par exemple, les nœuds de texte d'espaces DOIVENT être traités correctement afin de satisfaire cette exigence car les espaces sont significatives par défaut [xml-c14n-20010315].

Les spécifications qui font référence à ces types de schéma d'élément DOIVENT définir le format XML diff complet avec un type MIME approprié [RFC3023] et un jeu de caractères, par exemple, UTF-8 [RFC3629]. Par exemple, le format PIDF partiel [RFC5262] inclut ce schéma et décrit des définitions supplémentaires pour produire un format XML diff complet pour les mises à jour d'informations de présence partielles.

Comme le schéma défini dans le présent document ne déclare aucun espace de noms cible, les définitions de type héritent de l'espace de noms cible du schéma incluant. Donc, des déclarations d'espace de noms supplémentaires au sein des documents XML diff peuvent être évitées.

Il est prévu que les applications qui utilisent ces types vont définir les éléments <add>, <replace>, et <remove> sur la base des définitions de type correspondantes dans ce schéma. De plus, une application peut faire référence à seulement un sous ensemble de ces définitions de type. Une future extension pourrait introduire d'autres opérations, par exemple, avec des modèles tournés vers le document [xmlschema-2-20041028], une opération <move> et un algorithme de correction de nœud de texte combiné avec <move> produirait sans doute de plus petits documents XML diff.

Les éléments de document d'instance fondés sur ces définitions de type de schéma DOIVENT être bien formés et DEVRAIENT être valides.

Les types de nœuds du modèle de données XPath 1.0 suivants peuvent être ajoutés, remplacés, ou supprimés avec ce cadre : éléments, attributs, espaces de noms, commentaires, textes, et instructions de traitement. Le prologue XML complet, incluant par exemple des entités XML [xml-20060816] et le nœud racine d'un document XML, ne peut pas être corrigé en accord avec ce cadre. Cependant, la correction des commentaires et des instructions de traitement du nœud racine est permise. Naturellement, la suppression ou l'ajout d'un élément racine d'un document n'est pas permise car tout document XML valide DOIT toujours contenir un seul élément racine. Noter aussi que la prise en charge d'entités externes sort du domaine d'application de ce cadre.

4. Opérations Patch

Un document XML diff contient une collection d'éléments d'opérations de correction, incluant un ou plusieurs éléments `<add>`, `<replace>`, et `<remove>`. Ces opérations de correction vont être appliquées à la suite dans l'ordre du document. Après que la première correction a été appliquée pour mettre à jour un document XML cible, le document XML corrigé devient un nouveau document XML indépendant auquel la nouvelle correction va être appliquée. Cette procédure est répétée jusqu'à ce que toutes les corrections aient été traitées avec succès.

4.1 Localisation de la cible d'une correction

Chaque élément d'opération de correction contient un attribut 'sel'. La valeur de cet attribut est un sélecteur XPath avec un ensemble restreint de la recommandation XPath 1.0 complète. La valeur de 'sel' est utilisée pour localiser un seul nœud cible unique du document XML cible. Ce nœud localisé pointe sur la cible d'un changement et est généralement un élément, qui est par exemple soit lui-même mis à jour, soit un ou des nœuds fils qui lui sont ajoutés. Ce PEUT aussi être, par exemple, un nœud de commentaire, après lequel un ou d'autres nœuds apparentés sont insérés. Dans tous les cas, c'est une condition d'erreur si plusieurs nœuds sont trouvés durant l'évaluation de cette valeur de sélecteur.

Les choix de XPath de l'attribut 'sel' commencent toujours au nœud racine d'un document. Donc, des chemins de localisation relative DEVRAIENT être utilisés afin que le choix du nœud racine de début "/" puisse être omis. Quand il localise des éléments dans une arborescence de documents, un essai de nœud peut être soit un caractère "*", soit un QName [xml-names-20060816]. Un caractère "*" choisit tous les éléments fils du nœud de contexte. Juste après l'essai de nœud, une étape de localisation peut contenir un ou plusieurs prédicats dans un ordre quelconque. Une comparaison de valeurs d'attribut est un des prédicats les plus typiques. La valeur de chaîne du nœud de contexte en cours ou un élément fils peut autrement être utilisé pour identifier des éléments dans l'arborescence. Le caractère ".", qui note un choix de nœud de contexte courant, est une forme abrégée de "self::node()". Finalement, des contraintes de position comme "[2]" peuvent aussi être utilisées comme prédicat supplémentaire.

Une fonction XPath 1.0 "id()" node-set PEUT aussi être utilisée pour identifier des éléments uniques de l'arborescence de documents. Le schéma qui décrit le modèle de contenu du document DOIT alors utiliser un attribut avec l'identifiant de type [xmlschema-2-20041028] ou avec des analyseurs qui ne valident pas XML, un attribut "xml:id" [xml-id-20041109] DOIT avoir été utilisé dans un document d'instance.

4.2 Séparation d'espaces de noms

Le modèle normal pour les préfixes d'espace de noms est qu'ils sont de portée locale. Donc, un document XML diff PEUT avoir différents préfixes pour les espaces de noms utilisés dans le document cible. L'agent qui analyse le document diff DOIT résoudre les préfixes séparément dans les deux documents afin de correspondre aux QNames (nom qualifié) résultants de chacun. Le document XML diff DOIT contenir des déclarations pour tous les espaces de noms utilisés dans le document diff. Les déclarations de documents diff sont toujours utilisées pour déterminer quels espaces de noms s'appliquent dans le document diff.

4.2.1 Espaces de noms utilisés dans les sélecteurs

Un sélecteur dans un document diff peut utiliser des préfixes quand il désigne des éléments. Si il utilise un préfixe, le préfixe doit être recherché dans les déclarations d'espace de noms du document diff.

Par exemple, l'élément d'opération Patch d'un document diff a une déclaration d'espace de noms "xmlns:a='foo:'" dans la portée avec un sélecteur "sel='a:bar'". L'agent qui traite cette correction DOIT alors chercher un élément 'bar' qualifié avec l'espace de noms 'foo:' sans considérer si l'espace de noms 'foo:' a un préfixe alloué dans le document cible ou quel est ce préfixe.

Les espaces de noms par défaut rendent ce modèle un peu plus compliqué. Quand le document diff a une déclaration d'espace de noms par défaut, tout sélecteur d'élément sans préfixe DOIT être évalué en utilisant cet espace de noms.

Par exemple, l'élément d'opération Patch d'un document diff a une déclaration d'espace de noms "xmlns='foo:'" dans la portée avec un sélecteur "sel='bar'". L'agent qui traite cette correction DOIT ensuite chercher un élément 'bar' qualifié avec l'espace de noms 'foo:' sans considérer si l'espace de noms 'foo:' a un préfixe alloué dans le document cible ou quel est ce préfixe.

Les noms non qualifiés sont aussi possibles. Si il n'y a pas d'espace de noms par défaut déclaré, et si un élément nom apparaît sans préfixe, c'est alors un nom d'élément non qualifié. Si cela apparaît dans un sélecteur, il DOIT correspondre à un élément non qualifié dans le document cible.

Par exemple, l'élément d'opération Patch d'un document diff a seulement une déclaration d'espace de noms "xmlns:a='foo:'" dans la portée avec un sélecteur "sel='bar'". Comme l'élément 'bar' n'a pas de préfixe, et qu'il n'y a pas de déclaration d'espace de noms par défaut dans la portée, l'agent qui traite cette correction peut seulement confronter le sélecteur à un élément 'bar' qui n'a pas de préfixe ni d'espace de noms par défaut dans la portée.

4.2.2 Écart des exigences de XPath

Les règles de correspondance de préfixe décrites précédemment dans cette section sont différentes de celles requises dans XPath 1.0 et 2.0 [xpath20-20070123]. Dans XPath 1.0, un sélecteur "bar" localise toujours un élément <bar> non qualifié. Dans XPath 2.0, un sélecteur "bar" correspond non seulement à un élément <bar> non qualifié, mais aussi à un élément <bar> qualifié qui est dans la portée d'une déclaration d'espace de noms par défaut. À l'opposé, dans la présente spécification, un sélecteur sans un préfixe correspond seulement à un élément, et il peut correspondre à un élément avec ou sans un préfixe mais seulement si l'espace de noms avec lequel il est qualifié (ou aucun) est une correspondance exacte.

La recommandation XPath 1.0 spécifie les fonctions d'ensemble de nœuds "namespace-uri()" et "local-name()" qui peuvent être utilisées dans les prédicats. Ces fonctions peuvent être utilisées durant les évaluations XPath si il n'y a pas d'autres moyens "d'enregistrer" les préfixes avec les URI d'espace de noms associés. Elles peuvent aussi être utilisées quand on traite des choix où les espaces de noms par défaut sont attachés aux éléments. Cependant, la présente spécification ne permet pas l'usage de ces fonctions.

4.2.3 Espaces de noms et ajout/changement de contenu

Les éléments au sein du contenu des données changé sont aussi dans la portée des déclarations d'espace de noms. Par exemple, quand on ajoute un nouvel élément qualifié d'espace de noms au document XML cible, le document diff DOIT contenir une déclaration d'espace de noms qui s'applique à l'élément. L'agent qui traite le document diff DOIT s'assurer que le document cible contient aussi la même déclaration d'espace de noms. Comme pour XPath, la même déclaration d'espace de noms dans ce contexte signifie que les URI d'espace de noms DOIVENT être égaux, mais les préfixes PEUVENT être différents dans les documents diff et cible.

Par exemple, si un nouvel élément <a:bar> ajouté a une référence de déclaration d'espace de noms à "xmlns:a='foo:'" dans le document diff et si le document cible a seulement une déclaration d'espace de noms "xmlns:b='foo:'" dans la portée au point d'insertion, la référence d'espace de noms DOIT être changée afin qu'un élément <b:bar> existe alors dans le document corrigé. Les mêmes règles s'appliquent bien que des espaces de noms par défaut aient été utilisés dans l'un des documents ou les deux, les URI d'espace de noms déterminent ce que vont être les références correctes (préfixes) dans le document corrigé.

Quand le contenu nouveau ou changé a des éléments qui déclarent de nouveaux espaces de noms (de portée locale) ces déclarations sont copiées sans changement (préfixe et tout le reste) du document XML diff au document XML cible. Des déclarations d'espace de noms par défaut peuvent seulement être ajoutées de cette façon, mais des déclarations d'espace de noms préfixées PEUVENT être ajoutées ou supprimées avec la sémantique d'axe d'espace de noms XPath décrite plus loin (voir le paragraphe 4.3.3).

Un cas d'utilisation très difficile pour ces règles est celui où le document cible a plusieurs déclarations d'espace de noms dans sa portée pour le même espace de noms. Un document cible pourrait déclarer plusieurs préfixes différents pour le même espace de noms. Normalement, l'agent qui applique le document diff choisit *le* préfixe approprié pour ajouter de nouveaux éléments au document cible, mais dans ce cas particulier il y en a plus d'un. Ces exigences créent un comportement déterministe pour ce cas particulier et rare en pratique :

- Si le document diff se trouve utiliser un préfixe qui est un des préfixes déclarés pour le même espace de noms dans le nœud de contexte d'évaluation du document cible, ce préfixe DOIT être utilisé dans le document corrigé résultant. Un préfixe évaluable vide et une déclaration d'espace de noms existante dans la portée par défaut signifient que l'espace de noms par défaut DOIT être choisi. En d'autres termes, les noms développés sont alors égaux dans les documents diff et corrigé.

Dans une opération <add>, le nœud de contexte d'évaluation est l'élément parent du nœud inséré, par exemple, avec un sélecteur "sel='*/ bar'" et sans une directive d'attribut 'pos' (voir au paragraphe 4.3) il est l'élément <bar> de l'élément

document racine. Avec des modifications des éléments, le nœud de contexte d'évaluation est l'élément parent de l'élément modifié, et donc dans l'exemple précédent l'élément document racine.

- Ensuite, le préfixe (aussi vide) du nœud de contexte d'évaluation DOIT être choisi si les URI d'espace de noms sont égaux.
- Enfin, si les deux règles ci-dessus ne s'appliquent toujours pas, tous les préfixes d'espace de noms dans la portée du nœud de contexte d'évaluation sont rangés en ordre alphabétique ascendant. Si il existe une déclaration d'espace de noms par défaut, elle est interprétée comme la première entrée de cette liste. Le préfixe provenant de la liste est alors choisi comme celui qui est le plus proche et juste avant le préfixe comparé si il était inséré dans la liste. Si le préfixe comparé existait avant le premier préfixe, le premier préfixe dans la liste DOIT être choisi (c'est-à-dire, il n'y a pas d'espace de noms par défaut).

Par exemple, si la liste des préfixes dans la portée dans le document cible est "x", "y" et si le préfixe comparé dans le document diff est "xx", le préfixe "x" DOIT alors être choisi. Si un préfixe "a" était évalué, le préfixe "x", première entrée DOIT être choisi. Si il y avait aussi une déclaration d'espace de noms par défaut dans la portée, un préfixe "a" évaluable aurait alors choisi la déclaration par défaut. Noter que les attributs sans préfixe n'héritent pas de la déclaration d'espace de noms par défaut. Quand on ajoute des attributs qualifiés, la déclaration d'espace de noms par défaut n'est alors pas sur la liste des préfixes qui correspondent (voir au paragraphe 4.3.2).

Noter que ces exigences pourraient signifier qu'un document corrigé résultant pourrait contenir des déclarations d'espace de noms non utilisées et/ou superflues. Le document corrigé résultant NE DOIT PAS être "nettoyé" de telle façon que ces déclarations d'espace de noms soient supprimées.

Note : en pratique, l'agent qui construit un document diff peut généralement librement choisir les préfixes appropriés pour les déclarations d'espace de noms et il n'a pas besoin de savoir ou de se soucier des préfixes réels dans le document cible sauf si il y a des déclarations qui se chevauchent. En d'autres termes, le contenu du format diff est normalement indépendant de l'usage des préfixes d'espace de noms des documents cibles. Cependant, il peut être très utile de savoir où les espaces de noms sont déclarés dans le document cible. Le cas d'utilisation le plus typique est cependant celui où l'agent qui génère un diff a à sa disposition les deux documents, le précédent (cible) et le nouveau (corrigé) et les déclarations d'espace de noms sont donc exactement connues. Noter aussi, que dans le cas où le document cible n'est pas exactement connu, il est permis d'utiliser des déclarations d'espace de noms de portée locale, dont la conséquence est un document corrigé plus grand et moins lisible par l'homme.

4.3 Élément <add>

L'élément <add> représente l'ajout d'un contenu nouveau au document XML cible : par exemple, un nouvel élément peut être ajouté dans un élément existant.

Le nouveau contenu de données existe comme nœud fils de l'élément <add>. Quand on ajoute des attributs et des espaces de noms, le nœud fils de l'élément <add> DOIT être un seul nœud texte. Autrement, l'élément <add> PEUT contenir tout mélange de nœuds d'élément, texte, commentaire ou instruction de traitement dans n'importe quel ordre. Tous les enfants de l'élément <add> sont alors copiés dans un document XML cible. La procédure de fusion d'espace de noms décrite s'applique aux éléments ajoutés, qui incluent tous leurs attributs, espace de noms et nœuds descendants.

Le type d'élément <add> a trois attributs : 'sel', 'type', et 'pos'.

La valeur de l'attribut facultatif 'type' est seulement utilisée quand on ajoute des attributs et des espaces de noms. Alors, le nœud cible visé DOIT être un élément dans lequel les nouveaux attributs et déclarations d'espace de noms sont insérés. Quand la valeur de cet attribut 'type' est égale à "@attr", la chaîne "attr" est le nom de l'attribut réel ajouté. La valeur de ce nouvel attribut 'attr' est le contenu de nœud texte de l'élément <add>. Les attributs préfixés les moins fréquemment utilisés (c'est-à-dire, d'espace de noms qualifiés) peuvent aussi être ajoutés. Si la valeur de l'attribut 'type' est égale à "namespace::pref", "pref" est la chaîne de préfixe réelle à utiliser pour la déclaration d'espace de noms dans le document corrigé et le contenu du nœud texte de l'élément <add> contient l'URI d'espace de noms correspondant.

Note : l'attribut 'type' est donc aussi un sélecteur XPath, mais il localise seulement des attributs et des espaces de noms. L'axe d'attribut "attribute" a une forme abrégée "@" à la différence de l'axe "namespace", qui n'a pas de forme abrégée. Des doubles deux-points "::" sont utilisés comme séparateur d'axes dans XPath.

La valeur de l'attribut facultatif 'pos' indique le positionnement du nouveau contenu de données. Il n'est pas utilisé quand on ajoute des attributs ou espaces de noms. Quand ni l'attribut 'type' ni l'attribut 'pos' n'existent, les enfants de l'élément <add>

sont alors ajoutés au dernier nœud fils de l'élément cible localisé. Quand la valeur de l'attribut 'pos' est "prepend" le ou les nouveaux nœuds sont ajoutés comme premiers nœuds fils de l'élément cible localisé. Avec la valeur de "before", le ou les nouveaux nœuds ajoutés DOIVENT être les nœuds apparentés immédiatement précédents, et avec "after", les nœuds apparentés immédiatement suivants du nœud cible localisé.

Voici des exemples qui décrivent les cas d'utilisation de ces attributs d'élément <add>. Les nœuds ne sont pas des espaces de noms qualifiés et les préfixes ne sont donc pas utilisés, et le contenu XML diff complet n'est pas montré dans ces exemples, seulement les éléments de l'opération Patch. Des exemples complets sont donnés à l'Appendice A.

4.3.1 Ajout d'un élément

Exemple d'ajout d'un élément :

```
<add sel="doc"><foo id="ert4773">Ceci est un nouvel enfant</foo></add>
```

Une fois que l'élément <doc> a été trouvé dans le document XML cible, un nouvel élément <foo> est ajouté comme dernier nœud fils de l'élément <doc>. Le nœud cible localisé, l'élément <doc> est naturellement l'élément racine du document XML cible. Le nouvel élément <foo> contient un attribut 'id' et un nœud texte fils.

4.3.2 Ajout d'un attribut

Exemple d'ajout d'un attribut :

```
<add sel="doc/foo[@id='ert4773']" type="@utilisateur">Bob</add>
```

Cette opération ajoute un nouvel attribut 'utilisateur' à l'élément <foo> qui a été localisé en utilisant un prédicat de valeur d'attribut 'id'. La valeur de ce nouvel attribut 'utilisateur' est "Bob".

Un document XML corrigé similaire est réalisé quand on utilise un analyseur validant XML, si la valeur de sélecteur 'sel' avait été 'id("ert4773")' et si le type de données de l'attribut 'id' est "ID" [xmlschema-2-20041028].

Noter qu'avec des attributs qualifiés d'espace de noms, les règles de correspondance de préfixe dans l'attribut 'type' sont évalués avec des règles similaires à celles décrites au paragraphe 4.2.3. Noter aussi qu'alors, la possible déclaration d'espace de noms par défaut de l'élément de contexte n'est pas applicable.

Note : comme la valeur du sélecteur 'sel' PEUT contenir des guillemets, des formes d'échappement """ ou "'" peuvent être utilisées dans les valeurs d'attributs. Cependant, il est souvent plus approprié d'utiliser le caractère apostrophe (') comme montré dans ces exemples. Une autre solution est aussi d'échanger les apostrophes et les guillemets.

4.3.3 Ajout d'une déclaration d'espace de noms préfixée

Exemple d'ajout d'une déclaration d'espace de noms préfixée :

```
<add sel="doc" type="namespace::pref">urn:ns:xxx</add>
```

Cette opération ajoute une nouvelle déclaration d'espace de noms à l'élément <doc>. Le préfixe de ce nouveau nœud d'espace de noms est donc "pref" et l'URI d'espace de noms est "urn:ns:xxx".

4.3.4 Ajout de nœuds avec l'attribut "pos"

Exemple d'ajout d'un nœud de commentaire :

```
<add sel="doc/foo[@id='ert4773']" pos="before"><!-- commentaire --></add>
```

Cette opération ajoute un nouveau nœud de commentaire juste avant l'élément <foo> comme un nœud apparenté immédiatement précédant. C'est aussi un exemple de la façon dont une directive d'attribut 'pos' peut être utilisée.

4.3.5 Ajout de plusieurs nœuds

Une certaine complexité apparaît quand ce qu'on appelle des nœuds texte d'espace existent dans un document XML cible. Le modèle de données XPath 1.0 exige qu'un nœud texte NE DOIT PAS avoir un autre nœud texte comme nœud apparenté immédiat. Par exemple, si une opération add est telle que :

```
<add sel="doc">
  <foo id="ert4773">Ceci est un nouvel enfant</foo></add>
```

L'élément <add> a alors deux nœuds fils : un nœud texte espace (un saut à la ligne et deux espaces) et un élément <foo>. Si le dernier fils existant de l'élément <doc> est un nœud texte, son contenu et le contenu du nœud texte espace DOIVENT alors être combinés. Autrement, les nœuds texte (espace) peuvent être ajoutés juste comme des éléments, et donc, la forme canonique du document XML corrigé reste facilement déterministe. Comme plusieurs nœuds apparentés peuvent être insérés avec une seule opération <add>, un style "pretty printing" peut facilement être maintenu.

Voici un autre exemple sur le traitement des nœuds texte :

```
<add sel="*/foo/text()[2]" pos="after">new<bar/>elem</add>
```

Le second nœud texte fils de l'élément <foo> est d'abord localisé. Le nouveau contenu ajouté contient deux nœuds texte et un élément. Comme il ne peut pas y avoir de nœuds texte immédiatement apparentés, le contenu du nœud texte cible localisé et le contenu du premier nouveau nœud texte DOIVENT être combinés. Essentiellement, si la valeur 'pos' avait été "before", le contenu du second nouveau nœud texte aurait été effectivement ajouté devant le nœud texte cible localisé.

Note : il faut noter que les nœuds texte PEUVENT contenir des sections CDATA, dont les dernières ne sont pas traitées comme des nœuds séparés. Quand ces sections CDATA existent dans les nouveaux nœuds texte, elles DEVRAIENT être transportées sans changement dans le document XML corrigé.

Bien que les entités XML [xml-20060816] ne puissent pas être corrigées avec ce cadre, des références à d'autres que des entités internes prédéfinies peuvent exister dans les nœuds texte ou les attributs quand le prologue XML contient ces déclarations. Ces références peuvent alors être préservées si les deux documents XML diff et XML cible ont des déclarations identiques dans leur prologue. Autrement, les références peuvent être remplacées par du texte identique pour autant que la règle de "l'équivalence canonique" soit respectée.

4.4 Élément <replace>

L'élément <replace> représente une opération de remplacement : par exemple, un élément existant est mis à jour par un nouvel élément ou une valeur d'attribut est remplacée par une nouvelle valeur. Cette opération <replace> met toujours à jour un seul nœud ou contenu de nœud à la fois.

Le type d'élément <replace> a seulement un attribut 'sel'. Si le nœud cible localisé est un élément, un commentaire ou une instruction de traitement, alors le fils de l'élément <replace> DOIT aussi être du même type. Autrement, l'élément <replace> DOIT avoir un contenu de texte, ou il PEUT être vide quand on remplace une valeur d'attribut ou un contenu de nœud texte.

4.4.1 Remplacement d'un élément

Exemple de remplacement d'un élément :

```
<replace sel="doc/foo[@a='1']"><bar a="2"/></replace>
```

Cela va mettre à jour l'élément <foo> qui a un attribut 'a' de valeur "1". L'élément cible localisé est remplacé par l'élément <bar>. Donc tous les nœuds descendants, déclarations d'espace de noms, et attributs de l'élément remplacé <foo>, si il en existait, sont supprimés.

4.4.2 Remplacement d'une valeur d'attribut

Exemple de remplacement d'une valeur d'attribut :

```
<replace sel="doc/@a">nouvelle valeur</replace>
```


Cela va remplacer le contenu de l'attribut 'a' de l'élément <doc> par la valeur "nouvelle valeur". Si l'élément <replace> est vide, l'attribut 'a' DOIT alors rester dans le document XML corrigé apparaissant comme <doc a=""/>.

4.4.3 Remplacement d'un URI de déclaration d'espace de noms

Exemple de remplacement d'un URI d'espace de noms :

```
<replace sel="doc/namespace::pref">urn:new:xxx</replace>
```

Cela va remplacer la valeur d'URI du nœud d'espace de noms préfixé 'pref' par "urn:new:xxx". Le nœud parent de la déclaration d'espace de noms DOIT être l'élément <doc>, autrement, une erreur se produit.

4.4.4 Remplacement d'un nœud de commentaire

Exemple de remplacement d'un nœud commentaire :

```
<replace sel="doc/comment()[1]"><!-- C'est le nouveau contenu --></replace>
```

Cela va remplacer un nœud commentaire. Le nœud cible localisé est le premier nœud fils de commentaire de l'élément <doc>.

4.4.5 Remplacement d'un nœud d'instruction de traitement

Exemple de remplacement d'un nœud d'instructions de traitement :

```
<replace sel='doc/instruction de traitement("test")'><?test bar="foobar"?></replace>
```

Cela va remplacer le nœud d'instructions de traitement "test" dont le parent est l'élément <doc>.

4.4.6 Remplacement d'un nœud de texte

Exemple de remplacement d'un nœud texte :

```
<replace sel="doc/foo/text()[1]">Ceci est le nouveau contenu de texte</replace>
```

Cela va remplacer le premier nœud texte fils de l'élément <foo>. La contrainte positionnelle "[1]" n'est généralement pas nécessaire car le contenu de l'élément est rarement de type mixte [xmlschema-1-20041028] lorsque plusieurs nœuds texte apparentés existent.

Si un nœud texte est mis à jour et si l'élément <replace> est vide, le nœud texte DOIT alors être supprimé parce qu'un nœud texte DOIT toujours avoir au moins un caractère de données.

4.5 Élément <remove>

L'élément <remove> représente une opération de suppression de, par exemple, un élément ou attribut existant.

Le type d'élément <remove> a deux attributs: 'sel' et 'ws'. La valeur de l'attribut facultatif 'ws' est utilisée pour supprimer les possibles nœuds texte d'espace qui existent soit comme nœuds apparentés suivant ou précédant immédiatement le nœud cible localisé. L'usage de l'attribut 'ws' est permis seulement quand on supprime d'autres types que des nœuds texte, attribut et espace de noms. Si la valeur de 'ws' est "before", l'objet est de supprimer le nœud apparenté précédent immédiat qui DOIT être un nœud texte espace et si la valeur est "after", le nœud suivant correspondant. Si la valeur de 'ws' est "both", (les deux nœuds texte espace précédant et suivant DOIVENT être supprimés.

4.5.1 Suppression d'un élément

Exemple de la suppression d'un élément incluant tous ses nœuds descendants, attributs, et espace de noms :

```
<remove sel="doc/foo[@a='1']" ws="after"/>
```

Cela va supprimer l'élément <foo> ainsi que le nœud texte espace apparenté immédiatement suivant de l'élément <foo>. Si le nœud apparenté immédiatement suivant n'est pas un nœud texte espace, une erreur se produit.

4.5.2 Suppression d'un attribut

Exemple de la suppression d'un nœud attribut :

```
<remove sel="doc/@a"/>
```

Cela va supprimer le nœud attribut 'a' de l'élément <doc>.

4.5.3 Suppression d'une déclaration d'espace de nom préfixée

Exemple de la suppression d'un nœud espace de noms préfixé :

```
<remove sel="doc/foo/namespace::pref"/>
```

Cela va supprimer le nœud espace de noms préfixé 'pref' de l'élément <foo>. Naturellement, ce préfixe NE DOIT PAS être associé à un nœud avant la suppression de ce nœud d'espace de noms. Aussi, le nœud parent de cette déclaration d'espace de noms DOIT être l'élément <foo>.

4.5.4 Suppression d'un nœud de commentaire

Exemple de la suppression d'un nœud commentaire :

```
<remove sel="doc/comment()[1]"/>
```

Cela va supprimer le premier nœud commentaire fils de l'élément <doc>.

4.5.5 Suppression d'un nœud d'instruction de traitement

Exemple de la suppression d'un nœud d'instruction de traitement :

```
<remove sel="doc/processing-instruction('test')"/>
```

Cela va supprimer le nœud d'instructions de traitement fils "test" de l'élément <doc>.

4.5.6 Suppression d'un nœud de texte

Exemple de la suppression d'un nœud texte :

```
<remove sel="doc/foo/text()[1]"/>
```

Cela va supprimer le premier nœud texte fils de l'élément <foo>.

Quand on supprime un élément, un commentaire, ou un nœud d'instructions de traitement qui a des nœuds texte apparentés précédant et suivant immédiatement sans la directive 'ws', le contenu de ces deux nœuds texte DOIT être combiné. Le dernier nœud texte disparaît donc du document.

5. Traitement d'erreur

C'est une condition d'erreur qu'une des opérations Patch ne puisse pas être satisfaite sans ambiguïté. En d'autres termes, une fois qu'une opération Patch particulière échoue, c'est une condition d'erreur et le traitement d'opérations Patch ultérieures n'aurait pas de sens.

Un nouveau format d'erreur MIME est défini pour les applications qui exigent un traitement d'erreur déterministe quand une correction ne peut pas être appliquée. Il est prévu que ces éléments d'erreur puissent être utilisés dans d'autres types MIME qui permettent des éléments d'extension.

5.1 Éléments d'erreur

L'élément racine du document d'erreur est `<patch-ops-error>`. Le contenu de cet élément est une condition d'erreur spécifique. Chaque condition d'erreur est représentée par un élément différent. Cela permet que différentes conditions d'erreur fournissent des données différentes sur la nature de l'erreur. Tous les éléments d'erreur prennent en charge un attribut "phrase", qui peut contenir du texte destiné au rendu à l'utilisateur humain. L'attribut facultatif "xml:lang" PEUT être utilisé pour décrire le langage de l'attribut "phrase". La plupart des éléments de condition d'erreur sont supposés contenir l'élément d'opération Patch qui a causé l'échec de la correction.

Les éléments d'erreur suivants sont définis par la présente spécification :

`<invalid-attribute-value>` : les contraintes de validité des valeurs d'attribut 'sel', 'type', 'ws', ou 'pos' PEUVENT être indiquées avec cette erreur, c'est-à-dire, un contenu non admis a été utilisé. Aussi, cette erreur peut être utilisée pour indiquer si un contenu d'attribut ajouté ou modifié n'est pas valide, par exemple, des sections CDATA ont été utilisées quand un nouvel attribut été destiné à être ajouté.

`<invalid-character-set>` : la correction n'a pas pu être appliquée parce que le document diff et le document corrigé utilisent des jeux de caractères différents.

`<invalid-diff-format>` : cela indique que le corps diff de la demande n'était pas un document XML bien formé ou un document XML valide en accord avec ce schéma.

`<invalid-entity-declaration>` : une référence d'entité a été trouvée mais la déclaration correspondante n'a pas pu être localisée ou résolue.

`<invalid-namespace-prefix>` : l'URI d'espace de noms pour le préfixe donné na pas pu être localisé ou résolu, par exemple, au sein de l'attribut 'sel', un préfixe a été utilisé mais sa déclaration manque d'après le document cible.

`<invalid-namespace-uri>` : la valeur de l'URI d'espace de noms n'est pas valide ou le document cible n'a pas cette déclaration.

`<invalid-node-types>` : les types de nœud d'une opération `<replace>` ne correspondaient pas, c'est-à-dire, par exemple, le sélecteur 'sel' localise un élément mais le contenu remplaçable est de type texte. Aussi, une opération `<replace>` peut localiser un élément unique, mais le contenu remplaçable a plusieurs nœuds.

`<invalid-patch-directive>` : une directive Patch n'a pas pu être exécutée parce que les directives données n'ont pas été comprises.

`<invalid-root-element-operation>` : l'élément racine du document ne peut pas être supprimé ou un autre élément apparenté pour l'élément racine du document ne peut pas être ajouté.

`<invalid-xml-prolog-operation>` : échec de correction relatif aux nœuds de prologue XML.

`<invalid-whitespace-directive>` : une opération `<remove>` exige la suppression d'un nœud d'espace qui n'existe pas dans le document cible.

`<unlocated-node>` : un seul nœud unique (normalement un élément) n'a pas pu être localisé avec la valeur d'attribut 'sel'. Aussi, la localisation de plusieurs nœuds peut conduire à cette erreur.

`<unsupported-id-function>` : la fonction `nodeset id()` n'est pas prise en charge, et donc les attributs avec le type ID ne sont pas connus.

`<unsupported-xml-id>` : l'attribut `xml:id` comme attribut d'identifiant dans les documents XML n'est pas pris en charge.

Des éléments d'erreur supplémentaires peuvent être indiqués dans l'élément racine `<patch-ops-error>` à partir de tout espace de noms. Cependant, l'IETF PEUT spécifier des éléments d'erreur supplémentaires dans l'espace de noms "urn:ietf:params:xml:ns:patch-ops-error".

Par exemple, le document suivant indique qu'on a tenté d'ajouter un nouvel élément `<note>` avec une espace dans un document, mais l'élément parent n'a pas pu être localisé :

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

```

<patch-ops-error
xmlns:p="urn:ietf:params:xml:ns:pidf-diff"
xmlns="urn:ietf:params:xml:ns:patch-ops-error">
<unlocated-node
phrase="un nœud unique ne peut pas être localisé avec la fonction id()."
><p:add sel='id("ert4773")">
  <p:note>texte ajouté</p:note>
</p:add></unlocated-node>
</patch-ops-error>

```

6. Usage des opérations Patch

Un document XML diff DEVRAIT contenir seulement les nœuds qui ont été modifiés car l'intention est d'essayer de réduire les exigences de bande passante/mémorisation. Cependant, quand il y a une grande collection de changements, il peut être souhaitable de changer plutôt tout le contenu du document. Comment cela va être fait en pratique sort du domaine d'application du présent document.

Certaines applications PEUVENT exiger que l'historique complet des versions soit indiqué bien que l'historique ait des changements superflus. Le présent cadre ne rend obligatoire aucun comportement spécifique, les applications PEUVENT décider elles-mêmes de la sémantique appropriée. Aussi, en pratique, les applications sont libres de choisir les algorithmes appropriés quand elles génèrent le contenu du document diff.

7. Usage des valeurs de sélecteur

Il appartient à l'application de décider quelles sortes de valeurs de sélecteur utiliser. Les sélecteurs d'élément positionnel comme `"*/*[3]/*[2]"` donnent les plus courts sélecteurs, mais il faut faire attention quand on les utilise. Quand il y a plusieurs suppressions d'éléments apparentés, les indices d'élément positionnel changent après chaque mise à jour. De même, ces indices changent quand de nouveaux éléments sont insérés dans l'arborescence. Utiliser des noms avec de possibles prédicats d'attribut comme `"doc[@sel='foo']"` est généralement plus facile pour une application, si elle est par exemple un outil auto diff, mais cela conduit à de plus grands documents diff.

8. Types de schéma XML des éléments d'opération Patch

Les types de schéma pour les élément d'opération Patch :

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE schema [
  <!ENTITY nname "\i\c*">
  <!ENTITY qname "(&nname;)?&nname;">
  <!ENTITY aname "@&qname;">
  <!ENTITY pos "\[d+]">
  <!ENTITY attr "\[&aname;=?(.*)*\]\/\[&aname;=&quot;(.)*&quot;]">
  <!ENTITY valueq "\[(&qname;|\.)=&quot;(.)*&quot;]">
  <!ENTITY value "\[(&qname;|\.)=(.*)*\]\/\[&valueq;">
  <!ENTITY cond "&attr;|&value;|&pos;">
  <!ENTITY step "(&qname;|*)(&cond;)*">
  <!ENTITY piq "processing-instruction\/((&quot;&nname;&quot;))">
  <!ENTITY pi "processing-instruction\/(('&nname;')?)\/&piq;">
  <!ENTITY id "id\/((&nname;')?)\/id\/((&quot;&nname;&quot;)?)">
  <!ENTITY com "comment\/()">
  <!ENTITY text "text\/()">
  <!ENTITY nspa "namespace::&nname;">
  <!ENTITY cnodes "(&text;|&pos;|&com;|&pi;|&step;)">
  <!ENTITY child "&cnodes;|&step;">
  <!ENTITY last "(&child;|&aname;|&nspa;)">
]>
<xsd:schema

```

```

xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

<xsd:simpleType name="xpath">
  <xsd:restriction base="xsd:string">
    <xsd:pattern
      value="(/?)((&id;)(/(&step;)*(/&last;))?(/(&step;)/)*(&last;))"/>
    </xsd:restriction>
  </xsd:simpleType>

<xsd:simpleType name="xpath-add">
  <xsd:restriction base="xsd:string">
    <xsd:pattern
      value="(/?)((&id;)(/(&step;)*(/&child;))?(/(&step;)/)*(&child;))"/>
    </xsd:restriction>
  </xsd:simpleType>

<xsd:simpleType name="pos">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="before"/>
    <xsd:enumeration value="after"/>
    <xsd:enumeration value="prepend"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="type">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="&aname;|&nspa;"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="add">
  <xsd:complexContent mixed="true">
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:any processContents="lax" namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="sel" type="xpath-add" use="required"/>
      <xsd:attribute name="pos" type="pos"/>
      <xsd:attribute name="type" type="type"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="replace">
  <xsd:complexContent mixed="true">
    <xsd:restriction base="xsd:anyType">
      <xsd:sequence>
        <xsd:any processContents="lax" namespace="##any" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="sel" type="xpath" use="required"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:simpleType name="ws">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="before"/>
    <xsd:enumeration value="after"/>
    <xsd:enumeration value="both"/>
  </xsd:restriction>

```

```

</xsd:simpleType>

<xsd:complexType name="remove">
  <xsd:attribute name="sel" type="xpath" use="required"/>
  <xsd:attribute name="ws" type="ws"/>
</xsd:complexType>

</xsd:schema>

```

9. Schéma XML des erreurs d'opération Patch

Définitions des erreurs d'opération Patch.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  targetNamespace="urn:ietf:params:xml:ns:patch-ops-error"
  xmlns:tns="urn:ietf:params:xml:ns:patch-ops-error"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <!-- Cette importation amène l'attribut de langage XML xml:lang-->
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <!-- Élément document ROOT pour signaler les erreurs de patch-ops -->
  <xsd:element name="patch-ops-error">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:any namespace="##any" processContents="lax"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
  </xsd:element>

  <!-- Éléments d'erreur patch-ops : non destinés à être utilisés comme éléments de document racine -->
  <xsd:element name="invalid-attribute-value"
    type="tns:patch-error"/>
  <xsd:element name="invalid-character-set"
    type="tns:patch-error-simple"/>
  <xsd:element name="invalid-diff-format"
    type="tns:patch-error-simple"/>
  <xsd:element name="invalid-entity-declaration"
    type="tns:patch-error"/>
  <xsd:element name="invalid-namespace-prefix"
    type="tns:patch-error"/>
  <xsd:element name="invalid-namespace-uri"
    type="tns:patch-error"/>
  <xsd:element name="invalid-node-types"
    type="tns:patch-error"/>
  <xsd:element name="invalid-patch-directive"
    type="tns:patch-error"/>
  <xsd:element name="invalid-root-element-operation"
    type="tns:patch-error"/>
  <xsd:element name="invalid-xml-prolog-operation"
    type="tns:patch-error"/>
  <xsd:element name="invalid-whitespace-directive"
    type="tns:patch-error"/>
  <xsd:element name="unlocated-node"

```

```

        type="tns:patch-error"/>
<xsd:element name="unsupported-id-function"
        type="tns:patch-error"/>
<xsd:element name="unsupported-xml-id"
        type="tns:patch-error"/>

<!-- Type simple d'erreur patch-ops -->
<xsd:complexType name="patch-error-simple">
  <xsd:attribute name="phrase" type="xsd:string"/>
  <xsd:attribute ref="xml:lang"/>
  <xsd:anyAttribute processContents="lax"/>
</xsd:complexType>

<!-- Type d'erreur qui inclut l'opération Patch -->
<xsd:complexType name="patch-error">
  <xsd:sequence>
    <xsd:any namespace="##any" processContents="lax"/>
  </xsd:sequence>
  <xsd:attribute name="phrase" type="xsd:string"/>
  <xsd:attribute ref="xml:lang"/>
  <xsd:anyAttribute processContents="lax"/>
</xsd:complexType>

</xsd:schema>

```

10. Considérations relatives à l'IANA

L'IANA a réalisé les actions suivantes :

- o enregistré un nouvel URN d'espace de noms XML conformément aux procédures de la [RFC3688].
- o enregistré un nouveau type MIME 'application/patch-ops-error+xml' conformément aux procédures de la [RFC4288] et aux lignes directrices de la [RFC3023].
- O enregistré deux schémas XML conformément aux procédures de la [RFC3688].

10.1 Enregistrement de sous espace de noms d'URN

La présente spécification enregistre un nouvel espace de noms XML, conformément aux lignes directrices de la [RFC3688].

URI : l'URI pour cet espace de noms est urn:ietf:params:xml:ns:patch-ops-error

Contact d'enregistrement : IETF, groupe de travail SIMPLE (simple@ietf.org),
Jari Urpalainen (jari.urpalainen@nokia.com).

XML :

DÉBUT

```

<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html;charset=iso-8859-1"/>
  <title>Patch-Ops Error Namespace</title>
</head>
<body>
  <h1>Namespace for Patch-Ops Error Documents</h1>
  <h2>urn:ietf:params:xml:ns:patch-ops-error</h2>
  <p>See <a
href="http://www.rfc-editor.org/rfc/rfc5261.txt">RFC5261</a>.</p>

```

```
</body>  
</html>  
FIN
```

10.2 Type MIME application/patch-ops-error+xml

Nom de type de support MIME : application

Nom de sous type MIME : patch-ops-error+xml

Paramètres obligatoires : aucun

Paramètres facultatifs : les mêmes que le paramètre charset application/xml spécifié dans la [RFC3023].

Considérations de codage : les mêmes que les considérations de codage de application/xml spécifiées dans la [RFC3023].

Considérations de sécurité : voir la Section 10 de la [RFC3023].

Considérations d'interopérabilité : aucune.

Spécification publiée : RFC 5261

Applications qui utilisent ce type de supports : le présent type de document a été utilisé pour prendre en charge le transport des erreur de Patch-Ops dans la RFC 5261.

Informations supplémentaires :

Numéro magique : aucun

Extension de fichier : .xer

Code de type de fichier Macintosh : "TEXT"

Personne et adresse de messagerie pour plus d'informations: Jari Urpalainen, jari.urpalainen@nokia.com

Usage prévu : COMMUN

Auteur/contrôleur des changements : IETF

10.3 Enregistrement de schéma XML Patch-Ops-Types

Ce paragraphe enregistre un nouveau schéma XML, dont le seul contenu est montré à la Section 8.

URI : urn:ietf:params:xml:schema:patch-ops

Contact d'enregistrement :

IETF, groupe de travail SIMPLE, <simple@ietf.org>

Jari Urpalainen, <jari.urpalainen@nokia.com>

10.4 Enregistrement de schéma XML Patch-Ops-Error

Ce paragraphe enregistre un nouveau schéma XML, dont le seul contenu est montré à la Section 9.

URI : urn:ietf:params:xml:schema:patch-ops-error

Contact d'enregistrement :

IETF, groupe de travail SIMPLE, <simple@ietf.org>

Jari Urpalainen, <jari.urpalainen@nokia.com>

11. Considérations pour la sécurité

Les considérations sur la sécurité dépendent beaucoup de l'application qui utilise ce cadre. Comme chaque application va avoir des besoins, des modèles de menace, et des caractéristiques de sécurité, différents, il va être nécessaire de les examiner application par application.

Cependant, ce cadre utilise un sous ensemble limité de XPath 1.0. Les applications peuvent donc être vulnérables à des attaques d'injection XPath qui peuvent révéler un contenu interdit d'un document XML. Les attaques d'injection sont plus probables avec des ressources partageables où l'accès à une ressource est limité à seulement quelques parties spécifiques pour un utilisateur, contrairement au cas d'utilisation normal de ce cadre. Pour se défendre contre ces attaques, l'entrée DOIT être sécurisée, ce qui peut être fait, par exemple, par la validation des formats diff avec ces schémas restrictifs.

12. Remerciements

L'auteur tient à remercier Lisa Dusseault de ses efforts incluant les arrangements de réunions informelles, de commentaires et d'assistance à l'édition. L'auteur tient aussi à remercier Eva Leppanen, Mikko Lonnfors, Aki Niemi, Jonathan Rosenberg, Miguel A. Garcia, Anat Angel, Stephane Bortzmeyer, Dave Crocker, Joel Halpern, Jeffrey Hutzelman, David Ward, et Chris Newman de leurs précieux commentaires et Ted Hardie de ses apports et son soutien.

13. Références

13.1 Références normatives

- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par [RFC8174](#))
- [RFC3023] M. Murata, S. St.Laurent et D. Kohn, "Types de supports XML", janvier 2001. (*Obsolète, voir [RFC7303](#)*)
- [RFC3629] F. Yergeau, "[UTF-8, un format de transformation](#) de la norme ISO 10646", STD 63, novembre 2003.
- [RFC3688] M. Mealling, "[Registre XML de l'IETF](#)", BCP 81, janvier 2004.
- [RFC4288] N. Freed et J. Klensin, "Spécifications du [type de support et procédures d'enregistrement](#)", [BCP 13](#), décembre 2005.
- [xml-20060816] Maler, E., Paoli, J., Bray, T., Yergeau, F., and C. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", World Wide Web Consortium Recommendation REC-xml-20060816, août 2006, <<http://www.w3.org/TR/2006/REC-xml-20060816>>.
- [xpath-19991116] DeRose, S. and J. Clark, "XML Path Language (XPath) Version 1.0", World Wide Web Consortium Recommendation REC-xpath-19991116, novembre 1999, <<http://www.w3.org/TR/1999/REC-xpath-19991116>>.
- [xml-names-20060816] Hollander, D., Bray, T., Layman, A., and R. Tobin, "Namespaces in XML 1.0 (Second Edition)", World Wide Web Consortium Recommendation REC-xml-names-20060816, août 2006, <<http://www.w3.org/TR/2006/REC-xml-names-20060816>>.
- [xmlschema-1-20041028] Beech, D., Thompson, H., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-1-20041028, octobre 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>>.
- [xml-c14n-20010315] Boyer, J., "Canonical XML Version 1.0", World Wide Web Consortium Recommendation REC-xml-c14n-20010315, mars 2001, <<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>>.
- [xmlschema-2-20041028] Malhotra, A. and P. Biron, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, octobre 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.
- [xml-id-20041109] Veillard, D., Walsh, N., and J. Marsh, "xml:id Version 1.0", W3C LastCall WD-xml-id-20041109, novembre 2004.

13.2 Références pour information

- [RFC3265] A.B. Roach, "[Notification d'événement spécifique](#) du protocole d'initialisation de session (SIP)", juin 2002. (MàJ par [RFC6446](#)) (*Remplacée par la RFC6665*)
- [RFC3903] A. Niemi, "[Extension au protocole d'initialisation de session](#) (SIP) pour la publication d'état d'événement", octobre 2004.
- [RFC4825] J. Rosenberg, "[Protocole d'accès de configuration \(XCAP\)](#) du langage de balisage extensible (XML)", mai 2007. (*P.S.*)

- [RFC5262] M. Lonnfors et autres, "[Extension de format de données](#) d'information de présence (PIDF) pour présence partielle", septembre 2008. (P.S.)
- [RFC5874] J. Rosenberg, J. Urpalainen, "Format de document en langage de balisage extensible (XML) pour indiquer un changement des ressources du protocole d'accès de configuration XML (XCAP)", mai 2010. (P. S.)
- [xpath20-20070123] Berglund, A., Fernandez, M., Chamberlin, D., Boag, S., Robie, J., Kay, M., and J. Simeon, "XML Path Language (XPath) 2.0", World Wide Web Consortium Recommendation REC-xpath20-20070123, January 2007, <<http://www.w3.org/TR/2007/REC-xpath20-20070123>>.

Appendice A. Exemples pour information

Tous les exemples qui suivent supposent un document XML diff imaginaire incluant ces éléments d'opération Patch.

A.1 Ajout d'un élément

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <note>Ceci est un échantillon de document</note>
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <add sel="doc"><foo id="ert4773">Ceci est un nouvel enfant</foo></add>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <note>Ceci est un échantillon de document</note>
  <foo id="ert4773">Ceci est un nouvel enfant</foo></doc>
```

A.2 Ajout d'un attribut

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <note>Ceci est un échantillon de document</note>
  <foo id="ert4773">Ceci est un nouvel enfant</foo></doc>
```

Document XML diff:

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <add sel="doc/foo[@id='ert4773']" type="@utilisateur">Bob</add>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <note>Ceci est un échantillon de document</note>
```

```
<foo id="ert4773" utilisateur="Bob">Ceci est un nouvel enfant</foo></doc>
```

A.3 Ajout d'une déclaration d'espace de noms préfixé

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <note>Ceci est un échantillon de document</note>
  <foo id="ert4773">Ceci est un nouvel enfant</foo></doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <add sel="doc" type="namespace::pref">urn:ns:xxx</add>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc xmlns:pref="urn:ns:xxx">
  <note>Ceci est un échantillon de document</note>
  <foo id="ert4773">Ceci est un nouvel enfant</foo></doc>
```

A.4 Ajout d'un nœud de commentaire avec l'attribut 'pos'

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <note>Ceci est un échantillon de document</note>
  <foo id="ert4773">Ceci est un nouvel enfant</foo></doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <add sel="doc/foo[@id='ert4773']" pos="before"><!-- commentaire --></add>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <note>Ceci est un échantillon de document</note>
  <!-- commentaire --><foo id="ert4773">Ceci est un nouvel enfant</foo></doc>
```

A.5 Ajout de plusieurs nœuds

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <note>Ceci est un échantillon de document</note>
</doc>
```

Document XML diff :

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

```
<add sel="doc">
<foo id="ert4773">Ceci est un nouvel enfant</foo></add>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
<note>Ceci est un échantillon de document</note>
<foo id="ert4773">Ceci est un nouvel enfant</foo></doc>
```

A.6 Remplacement d'un élément

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
<foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
<replace sel="doc/foo[@a='1']"><bar a="2"/></replace>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
<bar a="2"/>
</doc>
```

A.7 Remplacement d'une valeur d'attribut

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc a="test">
<foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
<replace sel="doc/@a">nouvelle valeur</replace>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc a="nouvelle valeur">
<foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

A.8 Remplacement d'un URI de déclaration d'espace de noms

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc xmlns:pref="urn:test">
  <foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <replace sel="doc/namespace::pref">urn:new:xxx</replace>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc xmlns:pref="urn:new:xxx">
  <foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

A.9 Remplacement d'un nœud de commentaire

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc xmlns:pref="urn:test">
  <foo a="1">Ceci est un échantillon de document</foo>
  <!-- commentaire -->
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <replace sel="doc/comment()[1]"><!-- Ceci est le nouveau contenu --></replace>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc xmlns:pref="urn:test">
  <foo a="1">Ceci est un échantillon de document</foo>
  <!-- Ceci est le nouveau contenu -->
</doc>
```

A.10 Remplacement d'un nœud d'instruction de traitement

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
  <?test foo="bar"?>
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <replace sel='doc/processing-instruction("test")'
    ><?test bar="foobar"?></replace>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
  <?test bar="foobar"?>
</doc>
```

A.11 Remplacement d'un nœud de texte

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <replace sel="doc/foo/text()[1]">Ceci est le nouveau contenu de texte</replace>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est le nouveau contenu de texte</foo>
</doc>
```

A.12 Suppression d'un élément

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <remove sel="doc/foo[@a='1']" ws="after"/>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
</doc>
```

A.13 Suppression d'un attribut

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc a="foo">
  <foo a="1">Ceci est un échantillon de document</foo>
```

```
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <remove sel="doc/@a"/>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

A.14 Suppression d'une déclaration d'espace de noms préfixé

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1" xmlns:pref="urn:test"
    >Ceci est un échantillon de document</foo>
  <!-- commentaire -->
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <remove sel="doc/foo/namespace::pref"/>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
  <!-- commentaire -->
</doc>
```

A.15 Suppression d'un nœud de commentaire

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
  <!-- commentaire -->
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <remove sel="doc/comment()[1]" ws="after"/>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

A.16 Suppression d'un nœud d'instruction de traitement

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
  <?test?>
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <remove sel='doc/processing-instruction("test")'/>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

A.17 Suppression d'un nœud de texte

Exemple de document XML cible :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1">Ceci est un échantillon de document</foo>
</doc>
```

Document XML diff :

```
<?xml version="1.0" encoding="UTF-8"?>
<diff>
  <remove sel="doc/foo/text()[1]"/>
</diff>
```

Document XML résultant :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <foo a="1"/>
</doc>
```

A.18 Plusieurs corrections avec écrasement d'espace de noms

Exemple de document XML cible où existent des éléments d'espace de noms qualifiés :

```
<?xml version="1.0" encoding="UTF-8"?>
<doc xmlns="urn:ietf:params:xml:ns:xxx"
  xmlns:z="urn:ietf:params:xml:ns:yyy">
  <note>Ceci est un échantillon de document</note>
  <elem a="foo">
```



```

    <child/>
  </elem>
  <elem a="bar">
    <z:child/>
  </elem>
</doc>

```

Document XML diff imaginaire où le préfixe "p" correspond à l'espace de noms cible de ce schéma imaginaire :

```

<?xml version="1.0" encoding="UTF-8"?>
<p:diff xmlns="urn:ietf:params:xml:ns:xxx"
  xmlns:y="urn:ietf:params:xml:ns:yyy"
  xmlns:p="urn:ietf:params:xml:ns:diff">

  <p:add sel="doc/elem[@a='foo']"> <!-- Ceci est un nouvel enfant -->
  <child id="ert4773">
    <y:node/>
  </child>
</p:add>
<p:replace sel="doc/note/text()">Document corrigé</p:replace>
<p:remove sel="*/elem[@a='bar']/y:child" ws="both"/>
<p:add sel="*/elem[@a='bar']" type="@b">nouvel attribut</p:add>
</p:diff>

```

Une forme possible du document XML résultant après application des corrections :

```

<?xml version="1.0" encoding="UTF-8"?>
<doc xmlns="urn:ietf:params:xml:ns:xxx"
  xmlns:z="urn:ietf:params:xml:ns:yyy">
  <note>Document corrigé</note>
  <elem a="foo">
    <child/>
    <!-- Ceci est un nouvel enfant -->
    <child id="ert4773">
      <z:node/>
    </child>
  </elem>
  <elem a="bar" b="new attr"/>
</doc>

```

Les préfixes d'élément <node> et <child> supprimé dans le document XML diff sont différents de ce que sont les déclarations d'espace de noms "identical" dans le document XML cible. Si le document XML cible avait utilisé une déclaration d'espace de noms préfixée au lieu de celle par défaut, le document XML diff pourrait quand même avoir été le même. Les nouveaux éléments qualifiés ajoutés auraient juste hérité de ce préfixe.

Adresse de l'auteur

Jari Urpalainen
 Nokia
 Itamerenkatu 11-13
 Helsinki 00180
 Finland

téléphone : +358 7180 37686
 mél : jari.urpalainen@nokia.com

Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2008).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations y contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY, le IETF TRUST et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci-encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.