

Groupe de travail Réseau  
**Request for Comments : 5732**  
**STD : 69**  
RFC rendue obsolète : 4932  
Catégorie : Norme

S. Hollenbeck, VeriSign, Inc.  
août 2009

Traduction Claude Brière de L'Isle

# Transposition d'hôte du protocole d'approvisionnement extensible (EPP)

## Résumé

Le présent document décrit une transposition de protocole d'approvisionnement extensible (EPP, *Extensible Provisioning Protocol*) pour l'approvisionnement et la gestion des noms d'hôte de l'Internet mémorisés dans un répertoire central partagé. Spécifiée en XML, la transposition définit la syntaxe et la sémantique des commandes XML appliquées aux noms de domaines. Le présent document rend obsolète la RFC 4932.

## Statut de ce mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

## Notice de copyright

Copyright (c) 2009 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 sur les documents de l'IETF (<http://trustee.ietf.org/license-info>) en vigueur à la date de publication du présent document. Prière de relire attentivement ces documents, car ils décrivent vos droits et obligations à l'égard du présent document.

## Table des Matières

1. Introduction.....	1
1.1 Relations des objets d'hôte et des objets de domaine .....	2
1.2 Conventions de notation.....	2
2. Attributs d'objet .....	2
2.1 Noms d'hôte.....	2
2.2 Identifiants de client.....	2
2.3 Valeurs d'état.....	2
2.4 Dates et heures.....	3
2.5 Adresses IP.....	3
3. Transposition de commande EPP.....	4
3.1 Commande d'interrogation EPP.....	4
3.2 Commandes de transformation EPP.....	6
3.3 Révision hors ligne d'actions demandées.....	10
4. Syntaxe formelle.....	11
5. Considérations d'internationalisation.....	15
6. Considérations relatives à l'IANA.....	15
7. Considérations pour la sécurité.....	16
8. Remerciements.....	16
9. Références.....	16
9.1 Références normatives.....	16
9.2 Références pour information.....	17
Adresse de l'auteur.....	17

## 1. Introduction

Le présent document décrit une transposition de nom d'hôte Internet pour la version 1.0 du protocole d'approvisionnement extensible (EPP, *Extensible Provisioning Protocol*). Cette transposition est spécifiée en utilisant le langage de balisage extensible (XML, *Extensible Markup Language*) 1.0 tel que décrit dans [xml-20040204] et la notation de schéma XML telle que décrite dans [xmldata-20040204] et [xmldata-20040208]. Le présent document rend obsolète la [RFC4931].

La [RFC5730] donne une description complète des structures de commande et de réponse d'EPP. Une bonne compréhension de la spécification du protocole de base est nécessaire pour comprendre la transposition décrite dans le présent document.

XML est sensible à la casse. Sauf mention contraire, les spécifications et exemples XML fournis dans le présent document DOIVENT être interprétés dans la casse de caractères présentée pour développer une mise en œuvre conforme.

### 1.1 Relations des objets d'hôte et des objets de domaine

Le présent document suppose que les objets de nom d'hôte ont une relation de subordination à l'égard d'un objet de nom de domaine d'ordre supérieur. Par exemple, le nom d'hôte "ns1.example.com" a une relation de subordination au nom de domaine "example.com". Les actions EPP (comme des transferts d'objets) qui ne préservent pas cette relation DOIVENT être explicitement interdits.

Un objet de nom d'hôte peut être créé dans un répertoire pour lequel il n'existe aucun nom de domaine d'ordre supérieur. Par exemple, le nom d'hôte "ns1.example.com" peut être créé dans le répertoire ".example" afin que les domaines DNS dans ".example" puissent être délégués à l'hôte. De tels hôtes sont décrits comme hôtes "externes" dans la présente spécification car le nom de l'hôte n'appartient pas à l'espace de noms du répertoire dans lequel l'hôte est utilisé pour des besoins de délégation.

Qu'un hôte soit externe ou interne se rapporte au répertoire dans lequel l'hôte est utilisé aux fins de délégation. Un hôte interne est subordonné si le nom de l'hôte appartient au domaine au sein du répertoire dans lequel l'hôte est utilisé aux fins de délégation. Par exemple, l'hôte ns1.example1.com est un hôte subordonné du domaine example1.com, mais il n'est pas un hôte subordonné du domaine example2.com.

ns1.example1.com peut être utilisé comme serveur de noms pour example2.com. Dans ce cas, ns1.example1.com DOIT être traité comme hôte interne, sous réserve des règles qui gouvernent les opérations sur les hôtes subordonnés au sein du même répertoire.

### 1.2 Conventions de notation

Les mots clés "DOIT", "NE DOIT PAS", "EXIGÉ", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme décrit dans la [RFC2119].

Dans les exemples, "C:" représente les lignes envoyées par un client de protocole et "S:" représente les lignes retournées par un serveur du protocole. Les sauts à la ligne et les espaces ne sont fournis dans les exemples que pour illustrer les relations entre les éléments et ne sont pas des caractéristiques EXIGÉES de ce protocole. Un client de protocole qui est autorisé à gérer un objet existant est décrit comme un client de "parrainage" dans le présent document.

## 2. Attributs d'objet

Un objet d'hôte EPP a des attributs et des valeurs associées qui peuvent être vues et modifiées par le client parrain ou le serveur. Cette section décrit en détails chaque type d'attribut. La syntaxe formelle des valeurs d'attribut décrites ici se trouve dans la section "Syntaxe formelle" du présent document et dans les références normatives appropriées.

### 2.1 Noms d'hôte

La syntaxe de noms d'hôte décrite dans le présent document DOIT se conformer à la [RFC0952] telle que mise à jour par la [RFC1123]. Au moment de la rédaction, la [RFC3490] décrit une norme d'utilisation de certaines étiquettes de nom ASCII pour représenter des étiquettes de nom non ASCII. Ces exigences de conformité pourront changer à l'avenir par suite des progrès des travaux de développement de normes pour les noms d'hôte internationalisés.

### 2.2 Identifiants de client

Tous les clients EPP sont identifiés par un identifiant unique pour le serveur. Les identifiants de client se conforment à la syntaxe "clIDType" décrite dans la [RFC5730].

### 2.3 Valeurs d'état

Un objet d'hôte DOIT toujours avoir au moins une valeur d'état associée. Les valeurs d'état PEUVENT être seulement réglées par le client qui parraine un objet d'hôte et par le serveur sur lequel l'objet réside. Un client peut changer l'état d'un objet hôte

en utilisant la commande EPP <update>. Chaque valeur d'état PEUT être accompagnée d'une chaîne de texte lisible par l'homme qui décrit la raison de l'état appliqué à cet objet.

Un client NE DOIT PAS changer les valeurs d'état établies par le serveur. Un serveur PEUT altérer ou outrepasser les valeurs d'état établies par un client, sous réserve des politiques locales de serveur. L'état d'un objet PEUT changer par suite d'une commande de transformation initiée par un client ou d'une action effectuée par un opérateur de serveur.

Les valeurs d'état qui peuvent être ajoutées ou retirées par un client sont munies du préfixe "client". Les valeurs d'état correspondantes qui peuvent être ajoutées ou retirées par un serveur sont munies du préfixe "server". Les valeurs d'état qui ne commencent pas par "client" ou "server" sont gérées par le serveur.

Descriptions des valeurs d'état :

- clientDeleteProhibited, serverDeleteProhibited  
Les demandes de suppression de l'objet DOIVENT être rejetées.
- clientUpdateProhibited, serverUpdateProhibited  
Les demandes de mise à jour de l'objet (autres que de supprimer cet état) DOIVENT être rejetées.
- linked  
L'objet d'hôte a au moins une association active avec un autre objet, comme un objet de domaine. Les serveurs DEVRAIENT fournir des services pour déterminer les associations d'objet existantes.
- ok  
C'est la valeur d'état normale pour un objet qui n'a pas d'opération ou interdiction en cours. Cette valeur est établie et supprimée par le serveur lorsque d'autres valeurs d'état sont ajoutées ou supprimées.
- pendingCreate, pendingDelete, pendingTransfer, pendingUpdate  
Une commande de transformation a été traitée pour l'objet (ou dans le cas d'une commande <transfer>, pour l'objet de domaine supérieur de l'objet hôte) mais l'action n'a pas été achevée par le serveur. Les opérateurs de serveur peuvent différer l'achèvement de l'action pour diverses raisons, comme de permettre une revue humaine ou l'action d'un tiers. Une commande de transformation qui est traitée, mais dont l'action demandée est en cours, est notée avec le code de réponse 1001.

Lorsque l'action demandée a été achevée, la valeur d'état pendingCreate, pendingDelete, pendingTransfer, ou pendingUpdate DOIT être supprimée. Tous les clients impliqués dans la transaction DOIVENT en être notifiés en utilisant un message de service qui dit que l'action a été achevée et que l'état de l'objet a changé.

L'état "ok" PEUT seulement être combiné avec l'état "linked".

L'état "linked" PEUT être combiné avec tout état.

L'état "pendingDelete" NE DOIT PAS être combiné avec l'état "clientDeleteProhibited" ou "serverDeleteProhibited".

L'état "pendingUpdate" NE DOIT PAS être combiné avec l'état "clientUpdateProhibited" ou "serverUpdateProhibited".

Les valeurs d'état pendingCreate, pendingDelete, pendingTransfer, et pendingUpdate NE DOIVENT PAS être combinées les unes avec les autres.

Les autres combinaisons d'état non expressément interdites PEUVENT être utilisées.

## 2.4 Dates et heures

Les valeurs d'attribut de date et heure DOIVENT être représentées en temps universel coordonné (UTC, *Universal Coordinated Time*) en utilisant le calendrier grégorien. La forme étendue de date-heure utilisant les caractères majuscules "T" et "Z" définis dans [xmldate-20041028] DOIT être utilisée pour représenter les valeurs de date-heure, car le schéma XML ne prend pas en charge les formes tronquées de date-heure ou les caractères "T" et "Z" minuscules.

## 2.5 Adresses IP

La syntaxe des adresses IPv4 décrite dans le présent document DOIT se conformer à la [RFC0791]. La syntaxe des adresses

IPv6 décrite dans ce document DOIT se conformer à la [RFC4291]. Les considérations pratiques sur la publication des informations d'adresse IPv6 dans les fichiers de zone sont documentées dans la [RFC2874] et la [RFC3596]. Un serveur PEUT rejeter les adresses IP qui n'ont pas été allouées à une utilisation publique par l'IANA. Lorsque un objet hôte est provisionné pour être utilisé comme serveur de noms du DNS, les adresses IP DEVRAIENT n'être exigées qu'autant que nécessaire pour générer les enregistrements glue du DNS.

### 3. Transposition de commande EPP

Une description détaillée de la syntaxe et de la sémantique d'EPP se trouve dans la [RFC5730]. Les transpositions de commandes décrites ici sont spécifiquement à utiliser pour le provisionnement et la gestion des noms d'hôte de l'Internet via EPP.

#### 3.1 Commande d'interrogation EPP

EPP fournit deux commandes pour restituer les informations d'hôte : <check> pour déterminer si un objet hôte peut être provisionné au sein d'un répertoire, et <info> pour restituer les informations détaillées associées à un objet hôte.

##### 3.1.1 Commande EPP <check>

La commande EPP <check> est utilisée pour déterminer si un objet peut être provisionné au sein d'un répertoire. Elle fournit une indication qui permet à un client d'anticiper le succès ou l'échec de l'approvisionnement d'un objet en utilisant la commande <create>, car les exigences d'approvisionnement d'objet sont en fin de compte une affaire de politique de serveur.

En plus des éléments standard de commande EPP, la commande <check> DOIT contenir un élément <host:check> qui identifie l'espace de noms d'hôte. L'élément <host:check> contient les éléments fils suivants :

- Un ou plusieurs éléments <host:name> qui contiennent les noms pleinement qualifiés des objets d'hôte à interroger.

Exemple de commande <check> :

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C: <command>
C: <check>
C: <host:check
C:   xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C:   <host:name>ns1.example.com</host:name>
C:   <host:name>ns2.example.com</host:name>
C:   <host:name>ns3.example.com</host:name>
C: </host:check>
C: </check>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Lorsque une commande <check> a été traitée avec succès, l'élément EPP <resData> DOIT contenir un élément fils <host:chkData> qui identifie l'espace de noms de l'hôte. L'élément <host:chkData> contient un ou plusieurs éléments <host:cd> qui contiennent les éléments fils suivants :

- Un élément <host:name> qui contient le nom pleinement qualifié de l'objet d'hôte interrogé. Cet élément DOIT contenir un attribut "avail" dont la valeur indique la disponibilité de l'objet (peut il être provisionné ou non) au moment où la commande <check> a été achevée. Une valeur de "1" ou "vrai" signifie que l'objet peut être provisionné. Une valeur de "0" ou "faux" signifie que l'objet ne peut pas être provisionné.
- Un élément FACULTATIF <host:reason> qui PEUT être fourni lorsque un objet ne peut pas être provisionné. Si il est présent, cet élément contient un texte spécifique du serveur pour aider à expliquer pourquoi l'objet ne peut pas être provisionné. Ces texte DOIT être représenté dans le langage de réponse négocié précédemment avec le client ; un attribut FACULTATIF "lang" PEUT être présent pour identifier le langage si la valeur négociée est quelque chose d'autre que la valeur par défaut de "en" (anglais).

Exemple de réponse <check> :

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```

S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S: <result code="1000">
S: <msg>Commande achevée avec succès</msg>
S: </result>
S: <resData>
S: <host:chkData
S: xmlns:host="urn:ietf:params:xml:ns:host-1.0">
S: <host:cd>
S: <host:name avail="1">ns1.example.com</host:name>
S: </host:cd>
S: <host:cd>
S: <host:name avail="0">ns2.example2.com</host:name>
S: <host:reason>In use</host:reason>
S: </host:cd>
S: <host:cd>
S: <host:name avail="1">ns3.example3.com</host:name>
S: </host:cd>
S: </host:chkData>
S: </resData>
S: <trID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54322-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>

```

Une réponse d'erreur DOIT être retournée si une commande <check> ne peut être traitée pour une raison quelconque.

### 3.1.2 Commande EPP <info>

La commande EPP <info> est utilisée pour restituer les informations associées à un objet hôte. En plus des éléments de commande EPP standard, la commande <info> DOIT contenir un élément <host:info> qui identifie l'espace de noms de l'hôte. L'élément <host:info> contient les éléments fils suivants :

- Un élément <host:name> qui contient le nom pleinement qualifié de l'objet d'hôte pour lequel les informations sont demandées.

Exemple de commande <info> :

```

C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C: <command>
C: <info>
C: <host:info
C: xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C: <host:name>ns1.example.com</host:name>
C: </host:info>
C: </info>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>

```

Lorsque une commande <info> a été traitée avec succès, l'élément EPP <resData> DOIT contenir un élément fils <host:infData> qui identifie l'espace de noms de l'hôte. L'élément <host:infData> contient les éléments fils suivants :

- Un élément <host:name> qui contient le nom pleinement qualifié de l'objet d'hôte.
- Un élément <host:roid> qui contient l'identifiant d'objet de répertoire alloué à l'objet hôte lors de la création de l'objet.
- Un ou plusieurs éléments <host:status> qui décrivent l'état de l'objet d'hôte.
- Zéro, un ou plusieurs éléments <host:addr> qui contiennent les adresses IP associées à l'objet hôte.
- Un élément <host:clID> qui contient l'identifiant du client parrain.
- Un élément <host:crID> qui contient l'identifiant du client qui a créé l'objet hôte.
- Un élément <host:crDate> qui contient la date et heure de la création de l'objet hôte.
- Un élément <host:upID> qui contient l'identifiant du client qui a le dernier mis à jour l'objet hôte. Cet élément NE DOIT

PAS être présent si l'objet hôte n'a jamais été modifié.

- Un élément <host:upDate> qui contient la date et heure de la plus récente modification de l'objet hôte. Cet élément NE DOIT PAS être présent si l'objet hôte n'a jamais été modifié.
- Un élément <host:trDate> qui contient la date et l'heure du plus récent transfert réussi d'objet hôte. Cet élément NE DOIT PAS être fourni si l'objet hôte n'a jamais été transféré. Noter que les objets d'hôte NE DOIVENT PAS être transférés directement ; les objets d'hôte DOIVENT être transférés implicitement lorsque l'objet de domaine de niveau supérieur de l'objet hôte est transféré. Les objets hôtes qui sont soumis à un transfert lors du transfert d'un objet de domaine sont énumérés dans la réponse à une commande EPP <info> effectuée sur l'objet de domaine.

Exemple de réponse <info> :

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S: <result code="1000">
S: <msg>Commande achevée avec succès</msg>
S: </result>
S: <resData>
S: <host:infData
S: xmlns:host="urn:ietf:params:xml:ns:host-1.0">
S: <host:name>ns1.example.com</host:name>
S: <host:roid>NS1_EXAMPLE1-REP</host:roid>
S: <host:status s="linked"/>
S: <host:status s="clientUpdateProhibited"/>
S: <host:addr ip="v4">192.0.2.2</host:addr>
S: <host:addr ip="v4">192.0.2.29</host:addr>
S: <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
S: <host:clID>ClientY</host:clID>
S: <host:crID>ClientX</host:crID>
S: <host:crDate>1999-04-03T22:00:00.0Z</host:crDate>
S: <host:upID>ClientX</host:upID>
S: <host:upDate>1999-12-03T09:00:00.0Z</host:upDate>
S: <host:trDate>2000-04-08T09:00:00.0Z</host:trDate>
S: </host:infData>
S: </resData>
S: <trID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54322-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>
```

Une réponse d'erreur DOIT être retournée si une commande <info> ne peut être traitée pour une raison quelconque.

### 3.1.3 Commande d'interrogation EPP <transfer>

La sémantique du transfert ne s'applique pas directement aux objets d'hôte, de sorte qu'il n'y a pas de transposition définie pour la commande d'interrogation EPP <transfer>.

## 3.2 Commandes de transformation EPP

EPP fournit trois commandes pour transformer les objets d'hôte : <create> pour créer une instance d'objet hôte, <delete> pour supprimer une instance d'objet hôte, et <update> pour changer les informations associées à un objet hôte. Le présent document ne définit pas de transposition hôte-objet pour les commandes EPP <renew> et <transfer>.

Les commandes de transformation sont normalement traitées et achevées en temps réel. Les opérateurs de serveurs PEUVENT recevoir et traiter les commandes de transformation mais différer d'achever l'action demandée si une révision humaine ou de tiers est nécessaire avant que l'action demandée puisse être achevée. Dans de telles situations le serveur DOIT retourner un code de réponse 1001 au client pour noter que la commande a été reçue et traitée mais que l'action demandée est en cours. Le serveur DOIT aussi gérer l'état de l'objet qui fait l'objet de la commande pour refléter l'initiation et l'achèvement de l'action demandée. Une fois l'action achevée, tous les clients impliqués dans la transaction DOIVENT être notifiés en utilisant un message de service de ce que l'action a été achevée et de ce que l'état de l'objet a changé. D'autres méthodes de notification

PEUVENT être utilisées en plus du message de service requis.

Les opérateurs de serveurs DEVRAIENT confirmer qu'un client est autorisé à effectuer une commande de transformation sur un objet donné. Toute tentative de transformer un objet par un client non autorisé DOIT être rejetée, et le serveur DOIT retourner un code de réponse 2201 au client pour noter que le client n'a pas les privilèges pour exécuter la commande demandée.

### 3.2.1 Commande EPP <create>

La commande EPP <create> fournit une opération de transformation qui permet à un client de créer un objet d'hôte. En plus des éléments de commande EPP standard, la commande <create> DOIT contenir un élément <host:create> qui identifie l'espace de noms de l'hôte. L'élément <host:create> contient les éléments fils suivants :

- Un élément <host:name> qui contient le nom pleinement qualifié de l'objet d'hôte à créer.
- Zéro, un ou plusieurs éléments <host:addr> qui contiennent les adresses IP à associer à l'hôte. Chaque élément PEUT contenir un attribut "ip" pour identifier le format d'adresse IP. La valeur d'attribut "v4" est utilisée pour noter le format d'adresse IPv4. La valeur d'attribut "v6" est utilisée pour noter le format d'adresse IPv6. Si l'attribut "ip" n'est pas spécifié, "v4" est la valeur d'attribut par défaut.

Les hôtes peuvent être provisionnés pour une utilisation comme serveur de noms dans le système des noms de domaines (DNS, *Domain Name System*) décrit dans les [RFC1034] et [RFC1035]. Les hôtes provisionnés comme serveurs de noms peuvent être soumis à des politiques de serveur-opérateur qui exigent ou interdisent la spécification d'adresses IP, selon le nom de l'hôte et l'espace de noms dans lequel le serveur va être utilisé comme serveur de noms. Lorsque il est provisionné pour être utilisé comme serveur de noms, les adresses IP ne sont EXIGÉES qu'autant que nécessaire pour produire les enregistrements glue du DNS. Par exemple, si le serveur est d'autorité pour l'espace de noms "com" et si le nom du serveur est "ns1.example.net", le serveur n'est pas obligé de produire les enregistrements glue DNS pour le serveur de noms, et les adresses IP pour le serveur ne sont pas exigées par le DNS.

Si le nom d'hôte existe dans un espace de noms pour lequel le serveur est d'autorité, le domaine d'ordre supérieur de l'hôte DOIT être connu du serveur avant que l'objet hôte puisse être créé.

Exemple de commande <create> :

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C: <command>
C: <create>C: <host:create
C:   xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C:   <host:name>ns1.example.com</host:name>
C:   <host:addr ip="v4">192.0.2.2</host:addr>
C:   <host:addr ip="v4">192.0.2.29</host:addr>
C:   <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
C: </host:create>
C: </create>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Lorsque une commande <create> a été traitée avec succès, l'élément EPP <resData> DOIT contenir un élément fils <host:creData> qui identifie l'espace de noms de l'hôte. L'élément <host:creData> contient les éléments fils suivants :

- Un élément <host:name> qui contient le nom pleinement qualifié de l'objet d'hôte.
- Un élément <host:crDate> qui contient la date et heure de la création de l'objet d'hôte.

Exemple de réponse <create> :

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S: <result code="1000">
S: <msg>Commande achevée avec succès</msg>
S: </result>
S: <resData>
S: <host:creData
```

```
S: xmlns:host="urn:ietf:params:xml:ns:host-1.0">
S: <host.name>ns1.example.com</host.name>
S: <host.crDate>1999-04-03T22:00:00.OZ</host.crDate>
S: </host.creData>
S: </resData>
S: <trID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54322-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>
```

Une réponse d'erreur DOIT être retournée si une commande <create> ne peut être traitée pour une raison quelconque.

### 3.2.2 Commande EPP <delete>

La commande EPP <delete> fournit une opération de transformation qui permet à un client de supprimer un objet d'hôte. En plus des éléments de commande EPP standard, la commande <delete> DOIT contenir un élément <host:delete> qui identifie l'espace de noms de l'hôte. L'élément <host:delete> contient les éléments fils suivants :

- Un élément <host.name> qui contient le nom pleinement qualifié de l'objet d'hôte à supprimer.

Un objet de nom d'hôte NE DEVRAIT PAS être supprimé si l'objet hôte est associé à un autre objet. Par exemple, si l'objet hôte est associé à un objet domaine, l'objet hôte NE DEVRAIT PAS être supprimé tant que l'association existante n'a pas été rompue. Supprimer un objet hôte sans rompre d'abord les associations existantes peut causer l'échec de la résolution du DNS pour les objets de domaine qui se réfèrent à l'objet d'hôte supprimé.

Exemple de commande <delete> :

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C: <command>
C: <delete>
C: <host:delete
C: xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C: <host.name>ns1.example.com</host.name>
C: </host:delete>
C: </delete>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Lorsque une commande <delete> a été traité avec succès, un serveur DOIT répondre par une réponse EPP sans élément <resData>.

Exemple de réponse <delete> :

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S: <result code="1000">
S: <msg>Commande achevée avec succès</msg>
S: </result>
S: <trID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54321-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>
```

Une réponse d'erreur DOIT être retournée si une commande <delete> ne peut être traitée pour une raison quelconque.

### 3.2.3 Commande EPP <renew>

La sémantique du renouvellement ne s'applique pas aux objets d'hôte, de sorte qu'il n'y a pas de transposition définie pour la commande EPP <renew>.

### 3.2.4 Commande EPP <transfer>

La sémantique de transfert ne s'applique pas directement aux objets d'hôte, de sorte qu'il n'y a pas de transposition définie pour la commande EPP <transfer>. Les objets d'hôte sont subordonnés à un objet de domaine supérieur existant, et à ce titre ils sont soumis à transfert lorsque un objet de domaine est transféré.

### 3.2.5 Commande EPP <update>

La commande EPP <update> fournit une opération de transformation qui permet à un client de modifier les attributs d'un objet d'hôte. En plus des éléments de commande EPP standard, la commande <update> DOIT contenir un élément <host:update> qui identifie l'espace de noms de l'hôte. L'élément <host:update> contient les éléments fils suivants :

- Un élément <host:name> qui contient le nom pleinement qualifié de l'objet d'hôte à mettre à jour.
- Un élément FACULTATIF <host:add> qui contient les valeurs d'attribut à ajouter à l'objet.
- Un élément FACULTATIF <host:rem> qui contient les valeurs d'attribut à retirer à l'objet.
- Un élément FACULTATIF <host:chg> qui contient les valeurs d'attribut d'objet à changer.

Au moins un élément <host:add>, <host:rem>, ou <host:chg> DOIT être fourni si la commande n'est pas étendue. Tous ces éléments PEUVENT être omis si une extension <update> est présente. Les éléments <host:add> et <host:rem> contiennent les éléments fils suivants :

- Un ou plusieurs éléments <host:addr> qui contiennent les adresses IP à associer à, ou à retirer de, l'objet hôte. Les restrictions d'adresse IP décrites dans la transposition de commande <create> s'appliquent aussi ici.
- Un ou plusieurs éléments <host:status> qui contiennent les valeurs d'état à associer à, ou à retirer de l'objet. Lorsque on spécifie une valeur à retirer, seule la valeur de l'attribut est significative ; le texte de l'élément ne doit pas obligatoirement correspondre à une valeur à retirer.

Un élément <host:chg> contient les éléments fils suivants :

- Un élément <host:name> qui contient un nouveau nom d'hôte pleinement qualifié par lequel l'objet hôte sera connu.

Les changements de noms d'hôte PEUVENT exiger que l'ajout ou la suppression d'adresses IP soit accepté par le serveur. Une association d'adresse IP PEUT être soumise à des politiques de serveur pour provisionner les hôtes comme serveurs de noms.

Les changements de noms d'hôte peuvent avoir un impact sur les objets associés qui se réfèrent à l'objet hôte. Un changement de nom d'hôte NE DEVRAIT PAS exiger de mise à jour supplémentaire des objets associés pour préserver les associations existantes, à une exception près : changer un objet d'hôte externe qui a des associations avec des objets qui sont parrainés par un client différent. Les tentatives de mise à jour directe de tels hôtes DOIVENT échouer avec le code d'erreur EPP de 2305. Le changement peut être provisionné en créant un nouvel hôte externe avec un nouveau nom et tous nouveaux attributs nécessaires, et ensuite en mettant à jour les autres objets parrainés par le client.

Exemple de commande <update> :

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C: <command>
C: <update>
C: <host:update
C:   xmlns:host="urn:ietf:params:xml:ns:host-1.0">
C:   <host:name>ns1.example.com</host:name>
C:   <host:add>
C:     <host:addr ip="v4">192.0.2.22</host:addr>
C:     <host:status s="clientUpdateProhibited"/>
C:   </host:add>
C:   <host:rem>
C:     <host:addr ip="v6">1080:0:0:8:800:200C:417A</host:addr>
C:   </host:rem>
C:   <host:chg>
C:     <host:name>ns2.example.com</host:name>
C:   </host:chg>
C: </host:update>
```

```
C: </update>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Lorsque une commande <update> a été traitée avec succès, un serveur DOIT répondre par une réponse EPP sans élément <resData>.

Exemple de réponse <update> :

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S: <result code="1000">
S: <msg>Commande achevée avec succès</msg>
S: </result>
S: <trID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54321-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>
```

Une réponse d'erreur DOIT être retournée si une commande <update> ne peut être traitée pour une raison quelconque.

### 3.3 Révision hors ligne d'actions demandées

Les commandes sont traitées par un serveur dans l'ordre de leur réception du client. Bien qu'une réponse immédiate confirmant la réception et le traitement de la commande soit produite par le serveur, un opérateur de serveur PEUT effectuer une révision hors ligne des commandes de transformation demandées avant d'achever l'action demandée. Dans de telles situations, la réponse du serveur DOIT clairement noter que la commande de transformation a été reçue et traitée mais que l'action demandée est en cours. L'état de l'objet correspondant DOIT clairement refléter le traitement de l'action en cours. Le serveur DOIT notifier au client quand le traitement hors ligne de l'action est achevé.

On inclut ici des exemples qui décrivent une commande <create> qui exige une révision hors ligne. Noter le code de résultat et le message retournés en réponse à la commande <create>.

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S: <result code="1001">
S: <msg>Commande achevée avec succès ; action en cours</msg>
S: </result>
S: <resData>
S: <host:creData
S: xmlns:host="urn:ietf:params:xml:ns:host-1.0">
S: <host:name>ns1.example.com</host:name>
S: <host:crDate>1999-04-03T22:00:00.0Z</host:crDate>
S: </host:creData>
S: </resData>
S: <trID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54322-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>
```

L'état de l'objet d'hôte après le retour de cette réponse DOIT inclure "pendingCreate". L'opérateur de serveur revoit la demande hors ligne, et informe le client du résultat de la revue soit en mettant en file d'attente un message de service pour restitution via la commande <poll>, soit en utilisant un mécanisme hors bande pour informer le client de la demande.

Le message de service DOIT contenir un texte qui décrit la notification dans l'élément fils <msg> de l'élément <msgQ> de la

réponse. De plus, l'élément EPP <resData> DOIT contenir un élément fils <host:panData> qui identifie l'espace de noms de l'hôte. L'élément <host:panData> contient les éléments fils suivants :

- Un élément <host:name> qui contient le nom pleinement qualifié de l'objet d'hôte. L'élément <host:name> contient un attribut EXIGÉ "paResult". Une valeur booléenne positive indique que la demande a été approuvée et achevée. Une valeur booléenne négative indique que la demande a été refusée et que l'action demandée n'a pas été effectuée.
- Un élément <host:paTRID> qui contient l'identifiant de transaction du client et l'identifiant de transaction du serveur retournés avec la réponse d'origine pour traiter la commande. L'identifiant de transaction du client est FACULTATIF et ne sera retourné que si le client a fourni un identifiant avec la commande <create> d'origine.
- Un élément <host:paDate> contenant la date et l'heure qui décrivent quand s'est terminée la revue de l'action demandée.

Exemple de message de service "revue achevée" :

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S: <result code="1301">
S: <msg>Commande achevée avec succès ; accuser réception pour sortir de la file d'attente</msg>
S: </result>
S: <msgQ count="5" id="12345">
S: <qDate>1999-04-04T22:01:00.0Z</qDate>
S: <msg>l'action en cours est achevée avec succès.</msg>
S: </msgQ>
S: <resData>
S: <host:panData
S: xmlns:host="urn:ietf:params:xml:ns:host-1.0">
S: <host:name paResult="1">ns1.example.com</host:name>
S: <host:paTRID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54322-XYZ</svTRID>
S: </host:paTRID>
S: <host:paDate>1999-04-04T22:00:00.0Z</host:paDate>
S: </host:panData>
S: </resData>
S: <trID>
S: <clTRID>BCD-23456</clTRID>
S: <svTRID>65432-WXY</svTRID>
S: </trID>
S: </response>
S:</epp>
```

#### 4. Syntaxe formelle

Une transposition d'objet EPP est spécifiée en notation de schéma XML. La syntaxe formelle présentée ici est une représentation de la transposition d'objet qui convient pour les instances de validation automatique de XML EPP. Les étiquettes DÉBUT et FIN ne font pas partie du schéma ; elles sont utilisées pour noter le début et la fin du schéma pour les besoins d'enregistrement d'URI.

Copyright (c) 2009 IETF Trust et les personnes identifiées comme auteurs du code. Tous droits réservés.

La redistribution et l'utilisation en forme source et binaire, avec ou sans modification, sont permises sous réserve du respect des conditions suivantes :

- o Les redistributions de code source doivent conserver la notice de droits de reproduction, cette liste de conditions et le déclinatoire de responsabilité qui suit.
- o Les redistributions en forme binaire doivent reproduire la notice de droits de reproduction ci-dessus, cette liste de conditions et le déclinatoire de responsabilité qui suit dans la documentation et/ou autres matériaux fournis avec la distribution.
- o Ni le nom de la Internet Society, de l'IETF ou de l'IETF Trust, ni les noms des contributeurs, ne peuvent être utilisés pour endosser ou promouvoir des produits dérivés du présent logiciel sans permission préalable écrite spécifique.

Le présent logiciel est fourni "tel quel" par les détenteurs des droits de reproduction et les contributeurs et toutes garanties expressees ou implicites, y compris, mais sans s'y limiter, les garanties implicites de commercialisation et de convenance pour un objet particulier sont déclinées. En aucun cas le détenteur des droits de reproduction ou les contributeurs ne pourront

cependant être tenus pour responsables d'aucun dommage direct, indirect, incident, spécial, exemplaire, ou conséquent (incluant, sans s'y limiter, la fourniture de biens ou services, la perte d'usage, de données, ou de profits, ou une interruption d'affaires) causés et selon aucune théorie de responsabilité, que ce soit par contrat, responsabilité stricte, ou tort (incluant la négligence ou autrement) résultant de quelque façon que ce soit de l'utilisation du présent logiciel, même si il est informé de la possibilité de tels dommages.

## DÉBUT

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

```
<schema targetNamespace="urn:ietf:params:xml:ns:host-1.0"
  xmlns:host="urn:ietf:params:xml:ns:host-1.0"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
```

```
<!-- Importation des types d'éléments communs. -->
```

```
<import namespace="urn:ietf:params:xml:ns:eppcom-1.0"/>
```

```
<import namespace="urn:ietf:params:xml:ns:epp-1.0"/>
```

```
<annotation>
```

```
<documentation>
```

```
Schéma d'approvisionnement d'hôte du protocole d'approvisionnement extensible v1.0
```

```
</documentation>
```

```
</annotation>
```

```
<!-- Éléments fils trouvés dans les commandes EPP. -->
```

```
<element name="check" type="host:mNameType"/>
```

```
<element name="create" type="host:createType"/>
```

```
<element name="delete" type="host:sNameType"/>
```

```
<element name="info" type="host:sNameType"/>
```

```
<element name="update" type="host:updateType"/>
```

```
<!-- Éléments fils de la commande <create>. -->
```

```
<complexType name="createType">
```

```
<sequence>
```

```
<element name="name" type="eppcom:labelType"/>
```

```
<element name="addr" type="host:addrType"
```

```
  minOccurs="0" maxOccurs="unbounded"/>
```

```
</sequence>
```

```
</complexType>
```

```
<complexType name="addrType">
```

```
<simpleContent>
```

```
<extension base="host:addrStringType">
```

```
<attribute name="ip" type="host:ipType"
  default="v4"/>
```

```
</extension>
```

```
</simpleContent>
```

```
</complexType>
```

```
<simpleType name="addrStringType">
```

```
<restriction base="token">
```

```
<minLength value="3"/>
```

```
<maxLength value="45"/>
```

```
</restriction>
```

```
</simpleType>
```

```
<simpleType name="ipType">
```

```
<restriction base="token">
```

```
<enumeration value="v4"/>
```

```
<enumeration value="v6"/>
```

```
</restriction>
</simpleType>

<!-- Éléments fils des commandes <delete> et <info>. -->
<complexType name="sNameType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
  </sequence>
</complexType>

<!-- Éléments fils des commandes qui acceptent plusieurs noms. -->
<complexType name="mNameType">
  <sequence>
    <element name="name" type="eppcom:labelType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!-- Éléments fils de la commande <update>. -->
<complexType name="updateType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="add" type="host:addRemType"
      minOccurs="0"/>
    <element name="rem" type="host:addRemType"
      minOccurs="0"/>
    <element name="chg" type="host:chgType"
      minOccurs="0"/>
  </sequence>
</complexType>

<!-- Éléments de données qui peuvent être ajoutés ou supprimés. -->
<complexType name="addRemType">
  <sequence>
    <element name="addr" type="host:addrType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="status" type="host:statusType"
      minOccurs="0" maxOccurs="7"/>
  </sequence> </complexType>

<!-- Éléments de données qui peuvent être changés. -->
<complexType name="chgType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
  </sequence>
</complexType>

<!-- Éléments fils de réponse. -->
<element name="chkData" type="host:chkDataType"/>
<element name="creData" type="host:creDataType"/>
<element name="infData" type="host:infDataType"/>
<element name="panData" type="host:panDataType"/>

<!-- Éléments de réponse <check>. -->
<complexType name="chkDataType">
  <sequence>
    <element name="cd" type="host:checkType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="checkType">
```

```
<sequence>
  <element name="name" type="host:checkNameType"/>
  <element name="reason" type="eppcom:reasonType"
    minOccurs="0"/>
</sequence>
</complexType>
```

```
<complexType name="checkNameType">
  <simpleContent>
    <extension base="eppcom:labelType">
      <attribute name="avail" type="boolean"
        use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

```
<!-- Éléments de réponse <create>. -->
<complexType name="creDataType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="crDate" type="dateTime"/>
  </sequence>
</complexType>
```

```
<!-- Éléments de réponse <info>. -->
<complexType name="infDataType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="roid" type="eppcom:roidType"/>
    <element name="status" type="host:statusType"
      maxOccurs="7"/>
    <element name="addr" type="host:addrType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="clID" type="eppcom:clIDType"/>
    <element name="crID" type="eppcom:clIDType"/>
    <element name="crDate" type="dateTime"/>
    <element name="upID" type="eppcom:clIDType"
      minOccurs="0"/>
    <element name="upDate" type="dateTime"
      minOccurs="0"/>
    <element name="trDate" type="dateTime"
      minOccurs="0"/>
  </sequence>
</complexType>
```

```
<!-- L'état est une combinaison d'attributs et d'un message facultatif lisible par l'homme qui peut être exprimé dans un langage
autre que l'anglais. -->
```

```
<complexType name="statusType">
  <simpleContent>
    <extension base="normalizedString">
      <attribute name="s" type="host:statusValueType"
        use="required"/>
      <attribute name="lang" type="language"
        default="en"/>
    </extension>
  </simpleContent>
</complexType>
```

```
<simpleType name="statusValueType">
  <restriction base="token">
    <enumeration value="clientDeleteProhibited"/>
    <enumeration value="clientUpdateProhibited"/>
  </restriction>
</simpleType>
```

```
<enumeration value="linked"/>
<enumeration value="ok"/>
<enumeration value="pendingCreate"/>
<enumeration value="pendingDelete"/>
<enumeration value="pendingTransfer"/>
<enumeration value="pendingUpdate"/>
<enumeration value="serverDeleteProhibited"/>
<enumeration value="serverUpdateProhibited"/>
</restriction>
</simpleType>

<!-- Éléments de réponse de notification d'action en cours. -->
<complexType name="panDataType">
  <sequence>
    <element name="name" type="host:paNameType"/>
    <element name="paTRID" type="epp:trIDType"/>
    <element name="paDate" type="dateTime"/>
  </sequence>
</complexType>
<complexType name="paNameType">
  <simpleContent>
    <extension base="eppcom:labelType">
      <attribute name="paResult" type="boolean"
        use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!-- Fin de schéma. -->
</schema>
FIN
```

## 5. Considérations d'internationalisation

EPP est représenté en XML, qui fournit la prise en charge native des informations de codage en utilisant le jeu de caractères Unicode et ses représentations plus compactes incluant UTF-8. Les processeurs conformes à XML reconnaissent UTF-8 et UTF-16. Bien que XML comporte des dispositions pour identifier et utiliser d'autres codages de caractères par l'utilisation d'un attribut "encoding" dans une déclaration <?xml?>, l'utilisation de UTF-8 est RECOMMANDÉE dans les environnements où existent des incompatibilités de prise en charge d'analyseur de codage.

Toutes les valeurs de date et heure présentées via EPP DOIVENT être exprimées en temps universel coordonné en utilisant le calendrier grégorien. Le schéma XML permet l'utilisation d'identifiants de zone horaire pour indiquer les décalages au méridien zéro, mais cette option NE DOIT PAS être utilisée avec EPP. La forme étendue de date-heure utilisant les caractères majuscules "T" et "Z" définie dans [xmlschema-2-20041028] DOIT être utilisée pour représenter les valeurs de date-heure, car le schéma XML ne prend pas en charge les formes tronquées de date-heure ou les caractères "T" et "Z" en minuscules.

La syntaxe des noms de domaines et d'hôtes décrite dans le présent document DOIT se conformer à la [RFC0952] et à la [RFC1123]. Au moment de la rédaction du présent document, la [RFC3490] décrit une utilisation standard de certaines étiquettes de noms ASCII pour représenter des étiquettes de noms non ASCII. Ces exigences de conformité pourraient changer par suite des travaux en cours pour développer des normes pour les noms de domaines internationalisés.

## 6. Considérations relatives à l'IANA

Le présent document utilise des URN pour décrire les espaces de noms XML et les schémas XML conformes à un mécanisme de registre décrit dans la [RFC3688]. Deux allocations d'URI ont été enregistrées par l'IANA.

Demande d'enregistrement pour l'espace de nom d'hôte :

URI : urn:ietf:params:xml:ns:host-1.0

Contact : voir la section "Adresse de l'auteur" du présent document.

XML : aucun. Les URI d'espace de noms ne représentent pas une spécification XML.

Demande d'enregistrement pour le schéma XML d'hôte :

URI : urn:ietf:params:xml:schema:host-1.0

Contact : voir la section "Adresse de l'auteur" du présent document.

XML : voir la section "Syntaxe formelle" du présent document.

## 7. Considérations pour la sécurité

La transposition d'objet décrite dans le présent document ne fournit aucun autre service de sécurité ni n'introduit aucune considérations supplémentaires à celles décrites dans la [RFC5730] ou celles causées par les couches de protocole utilisées par EPP.

## 8. Remerciements

La RFC 3732 a été produite par le groupe de travail PROVREG, qui a suggéré des améliorations et fourni de nombreux commentaires précieux. L'auteur souhaite remercier de leurs efforts les présidents du groupe de travail Edward Lewis et Jaap Akkerhuis pour leurs contributions au traitement et à la rédaction. La RFC 4932 et le présent document sont des soumissions individuelles, fondées sur le travail effectué dans la RFC 3732.

Des suggestions spécifiques incorporées dans le présent document ont été fournies par Chris Bason, Jordyn Buchanan, Dave Crocker, Anthony Eden, Sheer El-Showk, Klaus Malorny, Dan Manley, Michael Mealling, Patrick Mevzek, and Rick Wesson.

## 9. Références

### 9.1 Références normatives

[RFC0791] J. Postel, éd., "Protocole Internet - Spécification du [protocole du programme Internet](#)", STD 5, septembre 1981.

[RFC0952] K. Harrenstien, M. Stahl, E. Feinler, "Spécification du tableau des hôtes de l'Internet du DOD", octobre 1985

[RFC1034] P. Mockapetris, "Noms de domaines - [Concepts et facilités](#)", STD 13, novembre 1987.

[RFC1035] P. Mockapetris, "Noms de domaines – [Mise en œuvre](#) et spécification", STD 13, novembre 1987. (*MàJ par la RFC6604*)

[RFC1123] R. Braden, éditeur, "Exigences pour les hôtes Internet – [Application et prise en charge](#)", STD 3, octobre 1989

[RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997

[RFC3688] M. Mealling, "[Registre XML de l'IETF](#)", BCP 81, janvier 2004

[RFC4291] R. Hinden, S. Deering, "[Architecture d'adressage IP version 6](#)", février 2006. (*MàJ par 5952 et 6052*) (*D.S.*)

[RFC5730] S. Hollenbeck, "[Protocole d'approvisionnement extensible \(EPP\)](#)", STD0069, août 2009. (*Remplace la RFC4930*)

[xml-20040204] Sperberg-McQueen, C., Maler, E., Yergeau, F., Paoli, J., and T. Bray, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium FirstEdition REC-xml-20040204, février 2004, <<http://www.w3.org/TR/2004/REC-xml-20040204>>.

[xmlschema-1-20041028] Maloney, M., Thompson, H., Mendelsohn, N., and D. Beech, "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>>.

[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>>.

## 9.2 Références pour information

- [RFC2781] P. Hoffman et F. Yergeau, "UTF-16, un codage de la norme ISO 10646", février 2000
- [RFC2874] M. Crawford, C. Huitema, "Extensions de DNS pour la prise en charge de l'agrégation et du rénumérotage d'adresse IPv6", juillet 2000. (*MàJ par RFC3152, RFC3226, RFC3363, RFC3364*) (*Expérimentale*)
- [RFC3490] P. Faltstrom et autres, "Internationalisation des noms de domaine dans les applications (IDNA)", mars 2003. (*Remplacée par les RFC5890 et 5891, P.S.*)
- [RFC3596] S. Thomson et autres, "[Extensions au DNS pour la prise en charge de IPv6](#)", octobre 2003. (*D.S.*)
- [RFC4932] S. Hollenbeck, "Transposition d'hôte avec le protocole d'approvisionnement extensible (EPP)", mai 2007. (Remplace la [RFC3732](#)) (*Remplacée par la RFC5732, STD 69*)

## Appendice A. Changements depuis la RFC 4932

1. On a changé "Le présent document rend obsolète la RFC 3732" en "Le présent document rend obsolète la RFC 4932".
2. On a remplacé la référence à la RFC 1886 par la référence à la 3596.
3. On a supprimé la référence à la RFC 3152 car elle et la RFC 1886 ont été rendues obsolètes par la RFC 3596.
4. On a remplacé la référence à la RFC 3732 par la référence à la 4932.
5. On a remplacé la référence à la RFC 4930 par la référence à la 5730.
6. Au paragraphe 3.2, on a ajouté "D'autres méthodes de notification PEUVENT être utilisées en plus du message de service exigé".
7. Au paragraphe 3.2, on a ajouté le texte du code de réponse 2201.
8. Ajout du texte de licence BSD à la section de schéma XML.

### Adresse de l'auteur

Scott Hollenbeck  
VeriSign, Inc.  
21345 Ridgetop Circle  
Dulles, VA 20166-6503  
USA  
mél : [shollenbeck@verisign.com](mailto:shollenbeck@verisign.com)