

Groupe de travail Réseau
Request for Comments : 5733
STD : 69
RFC rendue obsolète : 4933
Catégorie : Norme

S. Hollenbeck, VeriSign, Inc.
août 2009

Traduction Claude Brière de L'Isle

Transposition de contact 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 identifiants d'informations individuelles ou d'organisation sociale (connus comme des "contacts") 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 4933.

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.....	2
1.1 Conventions de notation utilisées dans le document.....	2
2. Attributs d'objet.....	2
2.1 Identifiants de contact et de client.....	2
2.2 Valeurs d'état.....	2
2.3 Noms d'individus et d'organisation.....	3
2.4 Adresse.....	3
2.5 Numéros de téléphone.....	4
2.6 Adresses de messagerie électronique.....	4
2.7 Dates et heures.....	4
2.8 Informations d'autorisation.....	4
2.9 Divulgarion des éléments et attributs des données.....	4
3. Transposition de commande EPP.....	5
3.1 Commande d'interrogation EPP.....	5
3.2 Commandes de transformation EPP.....	9
3.3 Révision hors ligne d'actions demandées.....	15
4. Syntaxe formelle.....	16
5. Considérations d'internationalisation.....	22
6. Considérations relatives à l'IANA.....	23
7. Considérations pour la sécurité.....	23
8. Remerciements.....	23
9. Références.....	23
9.1 Références normatives.....	23
9.2 Références pour information.....	24
Adresse de l'auteur.....	24

1. Introduction

Le présent document décrit une transposition d'identifiant de personne et d'organisation 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-20041028] et [xmldata-20041028]. Le présent document rend obsolète la [RFC4933].

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 Conventions de notation utilisées dans le document

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 de contact 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 Identifiants de contact et de client

Tous les contacts EPP sont identifiés par un identifiant unique pour le serveur. Les identifiants de contact sont des chaînes de caractères d'une longueur minimum spécifiée, d'une longueur maximum spécifiée, et d'un format spécifié. Les identifiants de contact utilisent la syntaxe d'identifiant de client "clIDType" décrite dans la [RFC5730].

2.2 Valeurs d'état

Un objet de contact 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 de contact et par le serveur sur lequel l'objet réside. Un client peut changer l'état d'un objet de contact 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.
- clientTransferProhibited, serverTransferProhibited
Les demandes de transfert 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 de contact 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, 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 "pendingTransfer" NE DOIT PAS être combiné avec l'état "clientTransferProhibited" ou "serverTransferProhibited".

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.

2.3. Noms d'individus et d'organisation

Les noms de personnes et d'organisation associés à un contact sont représentés en utilisant des chaînes de caractères. Ces chaînes ont une longueur spécifiée minimum et une longueur spécifiée maximum. Les noms d'individus et d'organisation PEUVENT être fournis en UTF-8 [RFC3629] ou dans un sous ensemble d'UTF-8 qui peut être représenté en ASCII à 7 bits, selon les besoins locaux.

2.4 Adresse

Tout contact a des informations d'adresse postale associées. Une adresse postale contient des informations de rue FACULTATIVES, des informations de ville FACULTATIVES, des informations de département/région FACULTATIVES, un code postal FACULTATIF, et un identifiant de pays. Les informations d'adresse PEUVENT être fournies en UTF-8 ou dans un sous ensemble d'UTF-8 qui peut être représenté en ASCII à 7 bits, selon les besoins locaux.

2.4.1 Rue, ville et département ou région

Les informations de rue, de ville et de département ou région du contact sont représentées en utilisant des chaînes de caractères. Ces chaînes peuvent avoir une longueur spécifiée minimum et une longueur spécifiée maximum.

2.4.2 Code postal

Les codes postaux des contacts sont représentés en utilisant des chaînes de caractères. Ces chaînes ont une longueur spécifiée minimum et une longueur spécifiée maximum.

2.4.3 Pays

Les identifiants de pays des contacts sont représentés en utilisant les identifiants à deux caractères spécifiés dans [ISO3166-1].

2.5 Numéros de téléphone

La structure du numéro de téléphone du contact est dérivée des structures définies dans [E.164]. Les numéros de téléphone décrits dans cette transposition sont des chaînes de caractères qui DOIVENT commencer par un signe plus ("+", valeur ASCII 0x002B) suivi par un code de pays défini dans [E.164], suivi par un point (".", valeur ASCII 0x002E), suivi par une séquence de chiffres représentant le numéro de téléphone. Un attribut "x" facultatif est fourni pour noter les informations d'extension téléphoniques

2.6 Adresses de messagerie électronique

La syntaxe d'adresse de messagerie électronique est définie dans la [RFC5322]. La présente transposition ne prescrit pas de longueur minimum ou maximum pour les chaînes de caractères utilisées pour représenter les adresses de messagerie électronique.

2.7 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 [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" minuscules.

2.8 Informations d'autorisation

Des informations d'autorisation sont associées aux objets de contact pour faciliter les opérations de transfert. Les informations d'autorisation sont allouées lors de la création d'un objet de contact, et elles peuvent être mises à jour ultérieurement. La présente spécification décrit des informations d'autorisation fondées sur le mot de passe, bien que d'autres mécanismes soient possibles.

2.9 Divulgence des éléments et attributs des données

Le cœur de protocole EPP exige qu'un opérateur de serveur annonce les politiques de collecte de données aux clients ; voir le paragraphe 2.4 de la [RFC5730]. En conjonction avec cette exigence de divulgation, la présente transposition inclut des éléments de données qui permettent à un client d'identifier les éléments qui exigent un traitement exceptionnel de l'opérateur de serveur pour permettre ou interdire la divulgation aux tiers.

Un opérateur de serveur annonce une politique de divulgation par défaut lorsque il établit une session avec un client. Lorsque un objet est créé ou mis à jour, le client peut spécifier les attributs de contact qui exigent un traitement exceptionnel de divulgation en utilisant un élément FACULTATIF <contact:disclose>. Une fois établies, les préférences de divulgation peuvent être révisées en utilisant une interrogation d'informations de contact. Un opérateur de serveur DOIT rejeter toute transaction qui demande des pratiques de divulgation qui ne sont pas conformes à la politique de collecte des informations annoncée avec un code de réponse d'erreur de 2308.

Si il est présent, l'élément <contact:disclose> DOIT contenir un attribut "flag". L'attribut "flag" contient une valeur booléenne de schéma XML. Une valeur de "true" ou "1" (un) note la préférence d'un client pour permettre la divulgation des éléments spécifiés comme exception à la politique de collecte des données déclarée. Une valeur de "false" ou "0" (zéro) note la préférence d'un client pour ne pas permettre la divulgation des éléments spécifié comme exception à la politique de collecte des données déclarée.

L'élément <contact:disclose> DOIT contenir au moins un des éléments fils suivants :

```
<contact:name type="int"/>
<contact:name type="loc"/>
<contact:org type="int"/>
<contact:org type="loc"/>
<contact:addr type="int"/>
<contact:addr type="loc"/>
```

```
<contact:voice/>
<contact:fax/>
<contact:email/>
```

Exemple d'élément `<contact:disclose>`, `flag="0"`:

```
<contact:disclose flag="0">
<contact:email/>
<contact:voice/>
</contact:disclose>
```

Dans cet exemple, l'adresse de messagerie électronique du contact et son numéro de téléphone vocal ne doivent pas être divulgués. Tous les autres éléments sont sujets à divulgation en accord avec la politique de collecte des données du serveur.

Exemple d'élément `<contact:disclose>`, `flag="1"`:

```
<contact:disclose flag="1">
<contact:name type="int"/>
<contact:org type="int"/>
<contact:addr type="int"/>
</contact:disclose>
```

Dans cet exemple, le nom internationalisé du contact, l'organisation, et les informations d'adresse peuvent être divulgués. Tous les autres éléments sont sujets à divulgation en accord avec la politique de collecte des données du serveur.

Les caractéristiques d'identification de client fournies par la commande EPP `<login>` et les informations d'autorisation de contact sont utilisées pour déterminer si un client est autorisé à effectuer des commandes d'interrogation d'informations de contact. Ces caractéristiques déterminent aussi si un client est autorisé à recevoir des données qui sont par ailleurs marquées comme non divulguables dans une réponse d'interrogation.

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 objets de contact via EPP.

3.1 Commande d'interrogation EPP

EPP fournit trois commandes pour restituer les informations d'hôte : `<check>` pour déterminer si un objet de contact peut être provisionné au sein d'un répertoire, `<info>` pour restituer les informations détaillées associées à un objet de contact, et `<transfer>` pour restituer les informations concernant l'état du transfert de l'objet de contact.

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 `<contact:check>` qui identifie l'espace de noms du contact. L'élément `<contact:check>` contient les éléments fils suivants :

- Un ou plusieurs éléments `<contact:id>` qui contiennent l'identifiant unique pour le serveur des objets de contact 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: <contact:check
C: xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
C: <contact:id>sh8013</contact:id>
```

```

C: <contact:id>sah8013</contact:id>
C: <contact:id>8013sah</contact:id>
C: </contact: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 <contact:chkData> qui identifie l'espace de noms du contact. L'élément <contact:chkData> contient un ou plusieurs éléments <contact:cd> qui contiennent les éléments fils suivants :

- Un élément <contact:id> qui identifie l'objet 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 <contact: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. Ce 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: <contact:chkData
S: xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
S: <contact:cd>
S: <contact:id avail="1">sh8013</contact:id>
S: </contact:cd>
S: <contact:cd>
S: <contact:id avail="0">sah8013</contact:id>
S: <contact:reason>In use</contact:reason>
S: </contact:cd>
S: <contact:cd>
S: <contact:id avail="1">8013sah</contact:id>
S: </contact:cd>
S: </contact: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 de contact. En plus des éléments de commande EPP standard, la commande <info> DOIT contenir un élément <contact:info> qui identifie l'espace de noms du contact. L'élément <contact:info> contient les éléments fils suivants :

- Un élément <contact:id> qui contient l'identifiant unique pour le serveur de l'objet de contact à interroger.
- Un élément FACULTATIF <contact:authInfo> qui contient les informations d'autorisation à associer à l'objet de contact. Si cet élément n'est pas fourni ou si les informations d'autorisation sont invalides, la politique du serveur détermine si la commande est rejetée ou si les informations de réponse seront retournées au client.

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: <contact:info
C:   xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
C:   <contact:id>sh8013</contact:id>
C:   <contact:authInfo>
C:     <contact:pw>2fooBAR</contact:pw>
C:   </contact:authInfo>
C: </contact: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 <contact:infData> qui identifie l'espace de noms du contact. L'élément <contact:infData> contient les éléments fils suivants :

- Un élément <contact:id> qui contient l'identifiant unique pour le serveur de l'objet de contact.
- Un élément <contact:roid> qui contient l'identifiant d'objet de répertoire alloué à l'objet de contact lors de la création de l'objet.
- Un ou plusieurs éléments <contact:status> qui décrivent l'état de l'objet de contact.
- Un ou plusieurs éléments <contact:postalInfo> qui contiennent les informations d'adresse postale. Deux éléments sont fournis afin que les informations d'adresse puissent être fournies dans les deux formats international et local ; un attribut "type" est utilisé pour identifier les deux formes. Si c'est la forme internationale (type="int") qui est fournie, le contenu de l'élément DOIT être représenté dans un sous-ensemble de l'UTF-8 qui puisse être représenté dans le jeu de caractères US-ASCII à 7 bits. Si c'est la forme locale (type="loc") qui est fournie, le contenu de l'élément PEUT être représenté en UTF-8 sans restriction. L'élément <contact:postalInfo> contient les éléments fils suivants :
 - Un élément <contact:name> qui contient le nom de l'individu ou du rôle représenté par le contact.
 - Un élément FACULTATIF <contact:org> qui contient le nom de l'organisation à laquelle appartient le contact.
 - Un élément <contact:addr> qui contient les informations d'adresse associées au contact. Un élément <contact:addr> contient les éléments fils suivants :
 - Un, deux, ou trois éléments FACULTATIFS <contact:street> qui contiennent le numéro et la rue du contact.
 - Un élément <contact:city> qui contient la ville du contact.
 - Un élément FACULTATIF <contact:sp> qui contient l'état ou la province du contact.
 - Un élément FACULTATIF <contact:pc> qui contient le code postal du contact.
 - Un élément <contact:cc> qui contient le code de pays du contact.
- Un élément FACULTATIF <contact:voice> qui contient le numéro de téléphone vocal du contact.
- Un élément FACULTATIF <contact:fax> qui contient le numéro de télécopie du contact.
- Un élément <contact:email> qui contient l'adresse de messagerie électronique du contact.
- Un élément <contact:clID> qui contient l'identifiant du client parrain.
- Un élément <contact:crID> qui contient l'identifiant du client qui a créé l'objet de contact.
- Un élément <contact:crDate> qui contient la date et l'heure de la création de l'objet de contact.
- Un élément <contact:upID> qui contient l'identifiant du client qui a le dernier mis à jour l'objet de contact. Cet élément NE DOIT PAS être présent si l'objet de contact n'a jamais été modifié.
- Un élément <contact:upDate> qui contient la date et l'heure de la plus récente modification de l'objet de contact. Cet élément NE DOIT PAS être présent si l'objet de contact n'a jamais été modifié.
- Un élément <contact:trDate> qui contient la date et l'heure du plus récent transfert réussi d'objet de contact. Cet élément NE DOIT PAS être fourni si l'objet de contact n'a jamais été transféré.
- Un élément <contact:authInfo> qui contient les informations d'autorisation associées à l'objet de contact. Cet élément NE DOIT PAS être fourni si le client qui interroge n'est pas le client parrain actuel.
- Un élément FACULTATIF <contact:disclose> qui identifie les éléments qui exigent un traitement exceptionnel de la part du serveur/opérateur pour permettre ou interdire la divulgation aux tiers. Voir au paragraphe 2.9 la description des éléments fils contenus dans l'élément <contact:disclose>.

Exemple de réponse <info> pour un client autorisé :

```
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 ave succès</msg>
S: </result>S: <resData>
S: <contact:infData
S: xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
S: <contact:id>sh8013</contact:id>
S: <contact:roid>SH8013-REP</contact:roid>
S: <contact:status s="linked"/>
S: <contact:status s="clientDeleteProhibited"/>
S: <contact:postalInfo type="int">
S: <contact:name>John Doe</contact:name>
S: <contact:org>Example Inc.</contact:org>
S: <contact:addr>
S: <contact:street>123 Example Dr.</contact:street>
S: <contact:street>Suite 100</contact:street>
S: <contact:city>Dulles</contact:city>
S: <contact:sp>VA</contact:sp>
S: <contact:pc>20166-6503</contact:pc>
S: <contact:cc>US</contact:cc>
S: </contact:addr>S: </contact:postalInfo>
S: <contact:voice x="1234">+1.7035555555</contact:voice>
S: <contact:fax>+1.7035555556</contact:fax>
S: <contact:email>jdoe@example.com</contact:email>
S: <contact:clID>ClientY</contact:clID>
S: <contact:crID>ClientX</contact:crID>
S: <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S: <contact:upID>ClientX</contact:upID>
S: <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S: <contact:trDate>2000-04-08T09:00:00.0Z</contact:trDate>
S: <contact:authInfo>
S: <contact:pw>2fooBAR</contact:pw>
S: </contact:authInfo>
S: <contact:disclose flag="0">
S: <contact:voice/>
S: <contact:email/>
S: </contact:disclose>
S: </contact: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 commande EPP <transfer> fournit une opération d'interrogation qui permet à un client de déterminer en temps réel l'état des demandes de transfert en cours et achevées. En plus des éléments standard de commande EPP, la commande <transfer> DOIT contenir un attribut "op" avec la valeur "query", et un élément <contact:transfer> qui identifie l'espace de noms du contact. L'élément <contact:transfer> contient les éléments fils suivants :

- Un élément <contact:name> qui contient le nom pleinement qualifié de l'objet de contact à interroger.
- Un élément FACULTATIF <contact:authInfo> qui contient les informations d'autorisation associées à l'objet de contact. Si cet élément n'est pas fourni ou si les informations d'autorisation sont invalides, la politique du serveur détermine si la commande est rejetée ou si les informations en réponse seront retournées au client.

Exemple de commande d'interrogation <transfer> :

```

C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">

```

```

C: <command>
C: <transfer op="query">
C: <contact:transfer
C:   xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
C:   <contact:id>sh8013</contact:id>
C:   <contact:authInfo>
C:     <contact:pw>2fooBAR</contact:pw>
C:   </contact:authInfo>
C: </contact:transfer>
C: </transfer>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>

```

Lorsque une commande d'interrogation `<transfer>` a été traitée avec succès, l'élément EPP `<resData>` DOIT contenir un élément fils `<contact:trnData>` qui identifie l'espace de noms du contact. L'élément `<contact:trnData>` contient les éléments fils suivants :

- Un élément `<contact:id>` qui contient l'identifiant unique pour le serveur du contact interrogé.
- Un élément `<contact:trStatus>` qui contient l'état de la plus récente demande de transfert.
- Un élément `<contact:reID>` qui contient l'identifiant du client qui a demandé le transfert de l'objet.
- Un élément `<contact:reDate>` qui contient la date et l'heure de demande du transfert.
- Un élément `<contact:acID>` qui contient l'identifiant du client qui DEVRAIT agir sur une demande de transfert EN COURS. Pour tous les autres types d'état, la valeur identifie le client qui a fait l'action indiquée.
- Un élément `<contact:acDate>` qui contient la date et l'heure d'une réponse exigée ou achevée. Pour une demande en cours, la valeur identifie la date et l'heure à laquelle une réponse est exigée avant qu'une action de réponse automatique DEVRAIT être faite par le serveur. Pour tous les autres types d'état, la valeur identifie la date et l'heure à laquelle la demande a été achevée.

Exemple de réponse d'interrogation `<transfer>` :

```

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:     <contact:trnData
S:       xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
S:       <contact:id>sh8013</contact:id>
S:       <contact:trStatus>pending</contact:trStatus>
S:       <contact:reID>ClientX</contact:reID>
S:       <contact:reDate>2000-06-06T22:00:00.0Z</contact:reDate>
S:       <contact:acID>ClientY</contact:acID>
S:       <contact:acDate>2000-06-11T22:00:00.0Z</contact:acDate>
S:     </contact:trnData>
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 d'interrogation `<transfer>` ne peut être traitée pour une raison quelconque.

3.2 Commandes de transformation EPP

EPP fournit quatre commandes pour transformer les objets de contact : `<create>` pour créer une instance d'objet de contact, `<delete>` pour supprimer une instance d'objet de contact, `<transfer>` pour gérer les changements de parrainage d'un objet de contact, et `<update>` pour changer les informations associées à un objet de contact. Le présent document ne définit pas de

transposition pour la commande EPP <renew>.

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 qu'il 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 de contact. En plus des éléments de commande EPP standard, la commande <create> DOIT contenir un élément <contact:create> qui identifie l'espace de noms du contact. L'élément <contact:create> contient les éléments fils suivants :

- Un élément <contact:id> qui contient l'identifiant unique pour le serveur pour le contact à créer.
- Un ou deux éléments FACULTATIFS <contact:postalInfo> qui contiennent les informations d'adresse postale. Deux éléments sont fournis afin que les informations d'adresse puissent être fournies dans les deux formats international et local ; un attribut "type" est utilisé pour identifier les deux formes. Si c'est la forme internationale (type="int") qui est fournie, le contenu de l'élément DOIT être représenté dans un sous-ensemble de l'UTF-8 qui puisse être représenté dans le jeu de caractères US-ASCII à 7 bits. Si c'est la forme locale (type="loc") qui est fournie, le contenu de l'élément PEUT être représenté en UTF-8 sans restriction. L'élément <contact:postalInfo> contient les éléments fils suivants :
 - o Un élément <contact:name> qui contient le nom de l'individu ou le rôle représenté par le contact.
 - o Un élément FACULTATIF <contact:org> qui contient le nom de l'organisation à laquelle appartient le contact.
 - o Un élément <contact:addr> qui contient les informations d'adresse associées au contact. Un élément <contact:addr> contient les éléments fils suivants :
 - * Un, deux ou trois éléments FACULTATIFS <contact:street> qui contiennent le numéro et la rue du contact.
 - * Un élément <contact:city> qui contient la ville du contact.
 - * Un élément FACULTATIF <contact:sp> qui contient l'état ou la province du contact.
 - * Un élément FACULTATIF <contact:pc> qui contient le code postal du contact.
 - * Un élément <contact:cc> qui contient le code de pays du contact.
- Un élément FACULTATIF <contact:voice> qui contient le numéro de téléphone vocal du contact.
- Un élément FACULTATIF <contact:fax> qui contient le numéro de télécopie du contact.
- Un élément <contact:email> qui contient l'adresse de messagerie électronique du contact.
- Un élément <contact:authInfo> qui contient les informations d'autorisation à associer à l'objet de contact. Cette transposition inclut un mécanisme d'authentification fondé sur le mot de passe, mais le schéma permet que de nouveaux mécanismes soient définis dans de nouveaux schémas.
- Un élément FACULTATIF <contact:disclose> qui permet à un client d'identifier les éléments qui exigent un traitement exceptionnel du serveur/opérateur pour permettre ou interdire la divulgation à des tiers. Voir au paragraphe 2.9 la description des éléments fils contenus dans l'élément <contact:disclose>.

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: <contact:create
C: xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
C: <contact:id>sh8013</contact:id>
C: <contact:postalInfo type="int">
C: <contact:name>John Doe</contact:name>
C: <contact:org>Example Inc.</contact:org>
C: <contact:addr>
C: <contact:street>123 Example Dr.</contact:street>
C: <contact:street>Suite 100</contact:street>
C: <contact:city>Dulles</contact:city>
```

```

C: <contact:sp>VA</contact:sp>
C: <contact:pc>20166-6503</contact:pc>
C: <contact:cc>US</contact:cc>
C: </contact:addr>
C: </contact:postalInfo>
C: <contact:voice x="1234">+1.7035555555</contact:voice>
C: <contact:fax>+1.7035555556</contact:fax>
C: <contact:email>jdoe@example.com</contact:email>
C: <contact:authInfo>
C: <contact:pw>2fooBAR</contact:pw>
C: </contact:authInfo>
C: <contact:disclose flag="0">
C: <contact:voice/>
C: <contact:email/>
C: </contact:disclose>
C: </contact: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 `<contact:creData>` qui identifie l'espace de noms du contact. L'élément `<contact:creData>` contient les éléments fils suivants :

- Un élément `<contact:id>` qui contient l'identifiant unique pour le serveur pour le contact à créer.
- Un élément `<contact:crDate>` qui contient la date et l'heure de la création de l'objet de contact.

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: <contact:creData
S: xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
S: <contact:id>sh8013</contact:id>
S: <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S: </contact:creData>
S: </resData>
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 `<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 de contact. En plus des éléments de commande EPP standard, la commande `<delete>` DOIT contenir un élément `<contact:delete>` qui identifie l'espace de noms du contact. L'élément `<contact:delete>` contient les éléments fils suivants :

- Un élément `<contact:id>` qui contient l'identifiant unique pour le serveur pour le contact à supprimer.

Un objet de contact NE DEVRAIT PAS être supprimé si il est associé à d'autres objets connus. Un contact associé NE DEVRAIT PAS être supprimé tant que les associations à d'autres objets connus n'ont pas été rompues. Un serveur DEVRAIT notifier aux clients que des relations d'objets existent en envoyant un code de réponse d'erreur 2305 lorsque une commande `<delete>` est tentée et échoue à cause de l'existence de relations d'objets.

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: <contact:delete
C:   xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
C:   <contact:id>sh8013</contact:id>
C: </contact: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 de contact, de sorte qu'il n'y a pas de transposition définie pour la commande EPP <renew>.

3.2.4 Commande EPP <transfer>

La commande EPP <transfer> fournit une opération de transformation qui permet à un client de gérer les demandes de transfert du parrainage d'un objet de contact. En plus des éléments de commande EPP standard, la commande <transfer> DOIT contenir un élément <contact:transfer> qui identifie l'espace de noms du contact. L'élément <contact:transfer> contient les éléments fils suivants :

- Un élément <contact:id> qui contient l'identifiant unique au serveur de l'objet de contact pour lequel une demande de transfert est à créer, approuver, rejeter, ou annuler.
- Un élément <contact:authInfo> qui contient les informations d'autorisation associées à l'objet de contact.

Chaque commande EPP <transfer> DOIT contenir un attribut "op" qui identifie l'opération de transfert à effectuer, comme défini dans la [RFC5730].

Exemple de commande de demande <transfer> :

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C: <command>
C: <transfer op="request">
C: <contact:transfer
C:   xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
C:   <contact:id>sh8013</contact:id>
```

```

C: <contact:authInfo>
C: <contact:pw>2fooBAR</contact:pw>
C: </contact:authInfo>
C: </contact:transfer>
C: </transfer>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>

```

Lorsque une commande <transfer> a été traitée avec succès, l'élément EPP <resData> DOIT contenir un élément fils <contact:trnData> qui identifie l'espace de noms du contact. L'élément <contact:trnData> contient les mêmes éléments fils que définis pour une réponse d'interrogation <transfer>.

Exemple de réponse <transfer> :

```

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: <contact:trnData
S: xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
S: <contact:id>sh8013</contact:id>
S: <contact:trStatus>pending</contact:trStatus>
S: <contact:reID>ClientX</contact:reID>
S: <contact:reDate>2000-06-08T22:00:00.0Z</contact:reDate>
S: <contact:acID>ClientY</contact:acID>
S: <contact:acDate>2000-06-13T22:00:00.0Z</contact:acDate>
S: </contact:trnData>
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 <transfer> ne peut pas être traitée pour une raison quelconque.

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

- Un élément <contact:id> qui contient l'identifiant unique pour le serveur pour l'objet de contact à mettre à jour.
- Un élément FACULTATIF <contact:add> qui contient les valeurs d'attribut à ajouter à l'objet.
- Un élément FACULTATIF <contact:rem> qui contient les valeurs d'attribut à retirer à l'objet.
- Un élément FACULTATIF <contact:chg> qui contient les valeurs d'attribut d'objet à changer.

Au moins un élément <contact:add>, <contact:rem>, ou <contact: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 <contact:add> et <contact:rem> contiennent les éléments fils suivants :

- Un ou plusieurs éléments <contact:status> qui contiennent les valeurs d'état à associer ou retirer à l'objet. Lorsque on spécifie une valeur à retirer, seule la valeur de l'attribut est significative ; le texte de l'élément n'est pas obligé de correspondre à une valeur à retirer.

Un élément <contact:chg> contient les éléments fils FACULTATIFS suivants. Au moins un élément fils DOIT être présent :

- Un ou deux éléments FACULTATIFS <contact:postalInfo> qui contiennent les informations d'adresse postale. Deux éléments sont fournis afin que les informations d'adresse puissent être fournies dans les deux formats international et local ; un attribut "type" est utilisé pour identifier les deux formes. Si c'est la forme internationale (type="int") qui est fournie, le

contenu de l'élément DOIT être représenté dans un sous ensemble de l'UTF-8 qui puisse être représenté dans le jeu de caractères US-ASCII à 7 bits. Si c'est la forme locale (type="loc") qui est fournie, le contenu de l'élément PEUT être représenté en UTF-8 sans restriction. L'élément <contact:postalInfo> contient les éléments fils suivants :

- o Un élément <contact:name> qui contient le nom de l'individu ou le rôle représenté par le contact.
- o Un élément <contact:org> qui contient le nom de l'organisation à laquelle appartient le contact.
- o Un élément <contact:addr> qui contient les informations d'adresse associées au contact. Un élément <contact:addr> contient les éléments fils suivants :
 - * Un, deux ou trois éléments FACULTATIFS <contact:street> qui contiennent le numéro et la rue du contact.
 - * Un élément <contact:city> qui contient la ville du contact.
 - * Un élément FACULTATIF <contact:sp> qui contient l'état ou la province du contact.
 - * Un élément FACULTATIF <contact:pc> qui contient le code postal du contact.
 - * Un élément <contact:cc> qui contient le code de pays du contact.
- Un élément <contact:voice> qui contient le numéro de téléphone vocal du contact.
- Un élément <contact:fax> qui contient le numéro de télécopie du contact.
- Un élément <contact:email> qui contient l'adresse de messagerie électronique du contact.
- Un élément <contact:authInfo> qui contient les informations d'autorisation associées à l'objet de contact. Cette transposition inclut un mécanisme d'authentification fondé sur le mot de passe, mais le schéma permet que de nouveaux mécanismes soient définis dans de nouveaux schémas.
- Un élément <contact:disclose> qui permet à un client d'identifier les éléments qui exigent un traitement exceptionnel du serveur/opérateur pour permettre ou interdire la divulgation à des tiers. Voir au paragraphe 2.9 la description des éléments fils contenus dans l'élément <contact:disclose>.

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: <contact:update
C: xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
C: <contact:id>sh8013</contact:id>
C: <contact:add>
C: <contact:status s="clientDeleteProhibited"/>
C: </contact:add>
C: <contact:chg>
C: <contact:postalInfo type="int">
C: <contact:org/>
C: <contact:addr>
C: <contact:street>124 Example Dr.</contact:street>
C: <contact:street>Suite 200</contact:street>
C: <contact:city>Dulles</contact:city>
C: <contact:sp>VA</contact:sp>
C: <contact:pc>20166-6503</contact:pc>
C: <contact:cc>US</contact:cc>
C: </contact:addr>
C: </contact:postalInfo>
C: <contact:voice>+1.7034444444</contact:voice>
C: <contact:fax/>
C: <contact:authInfo>
C: <contact:pw>2fooBAR</contact:pw>
C: </contact:authInfo>
C: <contact:disclose flag="1">
C: <contact:voice/>
C: <contact:email/>
C: </contact:disclose>
C: </contact:chg>
C: </contact: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: <contact:creData
S: xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
S: <contact:id>sh8013</contact:id>
S: <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S: </contact:creData>
S: </resData>
S: <trID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54321-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>
```

L'état de l'objet de contact 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 <contact:panData> qui identifie l'espace de noms du domaine. L'élément <contact:panData> contient les éléments fils suivants :

- Un élément <contact:id> qui contient l'identifiant unique au serveur de l'objet de contact. L'élément <contact: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 <contact:paTRID> qui contient l'identifiant de transaction du client et l'identifiant de transaction du serveur retournés avec l'original de la réponse au traitement de la commande. L'identifiant de transaction du client est FACULTATIF et ne sera retourné que si le client a fourni un identifiant dans la commande <create> d'origine.
- Un élément <contact:paDate> qui contient la date et l'heure de l'achèvement 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>Action en cours achevée avec succès.</msg>
S: </msgQ>
S: <resData>
S: <contact:panData
S: xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
S: <contact:id paResult="1">sh8013</contact:id>
S: <contact:paTRID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54321-XYZ</svTRID>
S: </contact:paTRID>
S: <contact:paDate>1999-04-04T22:00:00.0Z</contact:paDate>
S: </contact: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 expresses 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:contact-1.0"
  xmlns:contact="urn:ietf:params:xml:ns:contact-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 de contact du protocole d'approvisionnement extensible v1.0
```

```
</documentation>
```

```
</annotation>
```

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

```
<element name="check" type="contact:mIDType"/>
```

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

```
<element name="delete" type="contact:sIDType"/>
```

```
<element name="info" type="contact:authIDType"/>
```

```
<element name="transfer" type="contact:authIDType"/>
```

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

```
<!-- types d'utilitaires. -->
```

```
<simpleType name="ccType">
```

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

```
<length value="2"/>
```

```
</restriction>
```

```
</simpleType>
```

```
<complexType name="e164Type">
```

```
<simpleContent>
```

```
<extension base="contact:e164StringType">
```

```
<attribute name="x" type="token"/>
```

```
</extension>
```

```
</simpleContent>
```

```
</complexType>
```

```
<simpleType name="e164StringType">
```

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

```
<pattern value="(\+[0-9]{1,3}\.[0-9]{1,14})?"/>
```

```
<maxLength value="17"/>
```

```
</restriction>
```

```
</simpleType>
```

```
<simpleType name="pcType">
```

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

```
<maxLength value="16"/>
```

```
</restriction>
```

```
</simpleType>
```

```
<simpleType name="postalLineType">
```

```
<restriction base="normalizedString">
```

```
<minLength value="1"/>
```

```
<maxLength value="255"/>
```

```
</restriction>
```

```
</simpleType>

<simpleType name="optPostalLineType">
  <restriction base="normalizedString">
    <maxLength value="255"/>
  </restriction>
</simpleType>

<!-- Éléments fils de la commande <create>. -->
<complexType name="createType">
  <sequence>
    <element name="id" type="eppcom:clIDType"/>
    <element name="postalInfo" type="contact:postalInfoType"
      maxOccurs="2"/>
    <element name="voice" type="contact:e164Type"
      minOccurs="0"/>
    <element name="fax" type="contact:e164Type"
      minOccurs="0"/>
    <element name="email" type="eppcom:minTokenType"/>
    <element name="authInfo" type="contact:authInfoType"/>
    <element name="disclose" type="contact:discloseType"
      minOccurs="0"/>
  </sequence>
</complexType>

<complexType name="postalInfoType">
  <sequence>
    <element name="name" type="contact:postalLineType"/>
    <element name="org" type="contact:optPostalLineType"
      minOccurs="0"/>
    <element name="addr" type="contact:addrType"/>
  </sequence>
  <attribute name="type" type="contact:postalInfoEnumType"
    use="required"/>
</complexType>

<simpleType name="postalInfoEnumType">
  <restriction base="token">
    <enumeration value="loc"/>
    <enumeration value="int"/>
  </restriction>
</simpleType>

<complexType name="addrType">
  <sequence>
    <element name="street" type="contact:optPostalLineType"
      minOccurs="0" maxOccurs="3"/>
    <element name="city" type="contact:postalLineType"/>
    <element name="sp" type="contact:optPostalLineType"
      minOccurs="0"/>
    <element name="pc" type="contact:pcType"
      minOccurs="0"/>
    <element name="cc" type="contact:ccType"/>
  </sequence>
</complexType>

<complexType name="authInfoType">
  <choice>
    <element name="pw" type="eppcom:pwAuthInfoType"/>
    <element name="ext" type="eppcom:extAuthInfoType"/>
  </choice>
</complexType>
```

```
<complexType name="discloseType">
  <sequence>
    <element name="name" type="contact:intLocType"
      minOccurs="0" maxOccurs="2"/>
    <element name="org" type="contact:intLocType"
      minOccurs="0" maxOccurs="2"/>
    <element name="addr" type="contact:intLocType"
      minOccurs="0" maxOccurs="2"/>
    <element name="voice" minOccurs="0"/>
    <element name="fax" minOccurs="0"/>
    <element name="email" minOccurs="0"/>
  </sequence>
  <attribute name="flag" type="boolean" use="required"/>
</complexType>

<complexType name="intLocType">
  <attribute name="type" type="contact:postalInfoEnumType"
    use="required"/>
</complexType>

<!-- Éléments fils des commandes qui exigent un seul identifiant. -->
<complexType name="sIDType">
  <sequence>
    <element name="id" type="eppcom:clIDType"/>
  </sequence>
</complexType>

<!-- Éléments fils des commandes qui acceptent plusieurs identifiants. -->
<complexType name="mIDType">
  <sequence>
    <element name="id" type="eppcom:clIDType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!-- Éléments fils des commandes <info> et <transfer>. -->
<complexType name="authIDType">
  <sequence>
    <element name="id" type="eppcom:clIDType"/>
    <element name="authInfo" type="contact:authInfoType"
      minOccurs="0"/>
  </sequence>
</complexType>

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

<!-- Éléments de données qui peuvent être ajoutés ou supprimés. -->
<complexType name="addRemType">
  <sequence>
    <element name="status" type="contact:statusType"/>
  </sequence>
</complexType>
```

```
    maxOccurs="7"/>
  </sequence>
</complexType>

<!-- Éléments de données qui peuvent être changés. -->
<complexType name="chgType">
  <sequence>
    <element name="postalInfo" type="contact:chgPostalInfoType"
      minOccurs="0" maxOccurs="2"/>
    <element name="voice" type="contact:e164Type"
      minOccurs="0"/>
    <element name="fax" type="contact:e164Type"
      minOccurs="0"/>
    <element name="email" type="eppcom:minTokenType"
      minOccurs="0"/>
    <element name="authInfo" type="contact:authInfoType"
      minOccurs="0"/>
    <element name="disclose" type="contact:discloseType"
      minOccurs="0"/>
  </sequence>
</complexType>

<complexType name="chgPostalInfoType">
  <sequence>
    <element name="name" type="contact:postalLineType"
      minOccurs="0"/>
    <element name="org" type="contact:optPostalLineType"
      minOccurs="0"/>
    <element name="addr" type="contact:addrType"
      minOccurs="0"/>
  </sequence>
  <attribute name="type" type="contact:postalInfoEnumType"
    use="required"/>
</complexType>

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

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

<complexType name="checkType">
  <sequence>
    <element name="id" type="contact:checkIDType"/>
    <element name="reason" type="eppcom:reasonType"
      minOccurs="0"/>
  </sequence>
</complexType>

<complexType name="checkIDType">
  <simpleContent>
    <extension base="eppcom:clIDType">
      <attribute name="avail" type="boolean"

```

```
    use="required"/>
  </extension>
</simpleContent>
</complexType>

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

<!-- Éléments de réponse <info>. -->
<complexType name="infDataType">
  <sequence>
    <element name="id" type="eppcom:clIDType"/>
    <element name="roid" type="eppcom:roidType"/>
    <element name="status" type="contact:statusType"
      maxOccurs="7"/>
    <element name="postalInfo" type="contact:postalInfoType"
      maxOccurs="2"/>
    <element name="voice" type="contact:e164Type"
      minOccurs="0"/>
    <element name="fax" type="contact:e164Type"
      minOccurs="0"/>
    <element name="email" type="eppcom:minTokenType"/>
    <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"/>
    <element name="authInfo" type="contact:authInfoType"
      minOccurs="0"/>
    <element name="disclose" type="contact:discloseType"
      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="contact:statusValueType"
        use="required"/>
      <attribute name="lang" type="language"
        default="en"/>
    </extension>
  </simpleContent>
</complexType>

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

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

<!-- Éléments de réponse de notification d'action en cours. -->
<complexType name="panDataType">
  <sequence>
    <element name="id" type="contact:paCLIDType"/>
    <element name="paTRID" type="epp:trIDType"/>
    <element name="paDate" type="dateTime"/>
  </sequence>
</complexType>

<complexType name="paCLIDType">
  <simpleContent>
    <extension base="eppcom:clIDType">
      <attribute name="paResult" type="boolean"
        use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!-- Éléments de réponse <transfer>. -->
<complexType name="trnDataType">
  <sequence>
    <element name="id" type="eppcom:clIDType"/>
    <element name="trStatus" type="eppcom:trStatusType"/>
    <element name="reID" type="eppcom:clIDType"/>
    <element name="reDate" type="dateTime"/>
    <element name="acID" type="eppcom:clIDType"/>
    <element name="acDate" type="dateTime"/>
  </sequence>
</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 [RFC2781]. 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.

Les personnes, organisations, et autres entités ont souvent besoin de représenter les informations sociales à la fois dans un jeu de caractères compris de tous et un jeu de caractères local optimisé. La présente spécification fournit des caractéristiques permettant la représentation des informations sociales à la fois dans un sous ensemble de l'UTF-8 pour une large lisibilité et en UTF-8 sans restriction pour l'optimisation locale.

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 de contact :

URI : urn:ietf:params:xml:ns:contact-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 de contact :

URI : urn:ietf:params:xml:schema:contact-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

Les informations d'autorisation décrites au paragraphe 2.8 sont EXIGÉES pour créer un objet de contact. Ces informations sont utilisées dans certaines interrogations et opérations de transfert comme moyen supplémentaire de déterminer que le client a l'autorisation d'effectuer la commande. Manquer à protéger les informations d'autorisation contre la divulgation accidentelle peut avoir pour résultat des opérations de transfert non autorisées et une divulgation d'informations non autorisée. Le client et le serveur DOIVENT s'assurer que les informations d'autorisation sont mémorisées et échangées avec des mécanismes de chiffrement de haute qualité pour assurer des services de confidentialité.

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

8. Remerciements

La RFC 3733 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 4933 et le présent document sont des soumissions individuelles, fondées sur le travail effectué dans la RFC 3733.

Des suggestions spécifiques incorporées dans le présent document ont été fournies par Chris Bason, Eric Brunner-Williams, Jordyn Buchanan, Robert Burbidge, Dave Crocker, Ayesha Damaraju, Anthony Eden, Sheer El-Showk, Dipankar Ghosh, Klaus Malorny, Dan Manley, Michael Mealling, Patrick Mevzek, Asbjorn Steira, et Rick Wesson.

9. Références

9.1 Références normatives

[ISO3166-1] International Organization for Standardization, "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes", ISO Standard 3166, novembre 2006.

[E.164] Union Internationale des Télécommunications, "Plan de numérotage des télécommunications publiques internationales", Recommandation UIT-T E.164, février 2005.

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

[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

[RFC5322] P. Resnick, éd., "[Format du message Internet](#)", octobre 2008. (Remplace [RFC2822](#)) (MàJ [RFC4021](#)) (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

[RFC4933] S. Hollenbeck, "Transposition de contact avec le protocole d'approvisionnement extensible (EPP)", mai 2007. (Remplace [RFC3733](#)) (*Remplacée par la RFC5733*, STD 69)

Appendice A. Changements depuis la RFC 4933

1. On a changé "Le présent document rend obsolète la RFC3733" en "Le présent document rend obsolète la RFC4933".
2. On a remplacé la référence à la RFC0822 par la référence à la RFC5322.
3. On a remplacé la référence à la RFC3733 par la référence à la RFC4933.
4. On a remplacé la référence à la RFC4930 par la référence à la RFC5730.
5. On a mis à jour la référence à ISO 3166-1.
6. On a supprimé l'état pendingRenew du paragraphe 2.2 parce que le présent document ne définit pas de transposition pour la commande EPP <renew>.
7. On a modifié le texte du paragraphe 3.2.2 pour inclure le code de réponse 2305.
8. On a mis à jour la Section 5.
9. On a ajouté "D'autres méthodes de notification PEUVENT être utilisées en plus du message de service requis" au paragraphe 3.2.
10. On a ajouté le texte du code de réponse 2201 au paragraphe 3.2.
11. On a ajouté le texte de la licence BSD à la section du schéma XML.

Adresse de l'auteur

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