

Groupe de travail Réseau
Request for Comments : 4790
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

C. Newman, Sun Microsystems
 M. Duerst, Aoyama Gakuin University
 A. Gulbrandsen, Oryx
 mars 2007

Registre de collation de protocoles d'application Internet

Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir 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) The IETF Trust (2007).

Résumé

Beaucoup de protocoles d'application de l'Internet incluent des opérations de recherche ou de tri fondées sur la chaîne. Cependant, l'espace de problème des recherches et du tri sur les chaînes internationales est vaste, non complètement exploré, et est en dehors du domaine d'expertise de l'équipe d'ingénierie de l'Internet (IETF, *Internet Engineering Task Force*). Plutôt que de tenter de résoudre un si vaste problème, la présente spécification crée un cadre abstrait afin que les protocoles d'application puissent précisément identifier une fonction de comparaison, et que le répertoire des fonctions de comparaison puisse être étendu à l'avenir.

Table des matières

1. Introduction.....	2
1.1 Conventions utilisées dans ce document.....	2
2. Définition et objet de la collation.....	2
2.1 Définition.....	2
2.2 Objet.....	2
2.3 Autres termes utilisés dans le document.....	3
2.4 Clés de tri.....	3
3. Syntaxe d'identifiant de collation.....	3
3.1 Syntaxe de base.....	3
3.2 Caractères génériques.....	4
3.3 Direction d'ordre.....	4
3.4 URI.....	4
3.5 Lignes directrices pour les dénominations.....	4
4. Exigences de spécification de collation.....	4
4.1 Interface de collation/serveur.....	4
4.2 Opérations prises en charge.....	5
4.3 Clés de tri.....	6
4.4. Utilisation de tableaux de recherche.....	6
5. Exigences de protocole d'application.....	6
5.1 Codage de caractères.....	6
5.2 Opérations.....	6
5.3 Caractères génériques.....	7
5.4 Comparaison de chaînes.....	7
5.5 Clients déconnectés.....	7
5.6 Codes d'erreur.....	7
5.7 Collation d'octets.....	7
6. Utilisation par les protocoles existants.....	8
7. Enregistrement de collation.....	8
7.1 Procédure d'enregistrement de collation.....	8
7.2 Format d'enregistrement de collation.....	8
7.3 Structure de registre de collation.....	10
7.4. Exemple de sommaire de registre initial.....	10

8. Lignes directrices pour un expert réviseur.....	11
9. Collations initiales.....	11
9.1 Collation numérique ASCII.....	12
9.2 Collation de Casemap ASCII.....	12
9.3 Collation d'octet.....	13
10. Considérations relatives à l'IANA.....	14
11. Considérations sur la sécurité.....	14
12. Remerciements.....	14
13. Références.....	14
13.1 Références normatives.....	14
13.1 Références pour information.....	15
Adresse des auteurs.....	15
Déclaration complète de droits de reproduction.....	15

1. Introduction

La spécification du protocole d'accès à la configuration d'application (ACAP, *Application Configuration Access Protocol*) [RFC2244] introduisait le concept d'un comparateur (qu'on appelle collation dans le présent document) mais n'a pas créé de registre IANA. Avec l'introduction de stringprep [RFC3454] et de l'algorithme de collationnement Unicode [Unicode], il est temps maintenant de créer ce registre et de le remplir avec des valeurs initiales appropriées pour la communauté internationale. La présente spécification remplace et généralise la définition d'un comparateur dans ACAP, et crée un registre de collations.

1.1 Conventions utilisées dans ce document

Les mots clés "DOIT", "NE DOIT PAS", "DEVRAIT", "NE DEVRAIT PAS", et "PEUT" en majuscules dans ce document sont à interpréter comme décrit dans "Mots-clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", [RFC2119].

Les spécifications de syntaxe d'attribut utilisent la notation de forme Backus-Naur augmenté (ABNF, *Augmented Backus-Naur Form*) [RFC4234], incluant le cœur des règles définies dans l'Appendice A. La production ABNF "Language-tag" est importée des étiquettes de langue [RFC4646] et "reg-name" de la syntaxe générique d'URI: [RFC3986].

2. Définition et objet de la collation

2.1 Définition

Une collation est une fonction désignée qui prend en entrée deux chaînes de longueur arbitraire et peut être utilisée pour effectuer une ou plusieurs des trois opérations de comparaison de base : essai d'égalité, correspondance de sous chaîne, et essai de rangement.

2.2 Objet

Les collations sont une abstraction pour les fonctions de comparaison afin que ces fonctions de comparaison puissent être utilisées dans de multiples protocoles. Les détails d'une opération de comparaison particulière peuvent être spécifiées par quelqu'un qui a une expertise appropriée, indépendamment des protocoles d'application qui utilisent cette collation. Ceci est similaire à la façon dont un jeu de caractères [RFC2978] sépare les détails de la transposition d'octet à caractère à partir d'une spécification de protocole, comme MIME [RFC2045], ou la façon dont SASL [RFC4422] sépare les détails d'un mécanisme d'authentification à partir d'une spécification de protocole, comme ACAP [RFC2244].

Voici un petit diagramme pour aider à illustrer la valeur de cette abstraction :

```

+-----+
|Recherche IMAP i18n|--+
+-----+ |
+-----+ | +-----+ | +-----+
|Recherche ACAP i18n|--+--| Registre de |--+--| Spéc. de collation |
+-----+ | | collation | | | stringprep |
+-----+ | +-----+ | +-----+
|...autre protocole |--+
+-----+ +-----+

```

Donc les protocoles d'application IMAP, ACAP, et futurs avec des capacités de recherche internationale spécifient simplement comment faire l'interface avec le registre de collation au lieu que chaque spécification de protocole ait à spécifier toutes les collations qu'elle prend en charge.

2.3 Autres termes utilisés dans le document

Le termes de client, serveur, et protocole sont utilisés dans un sens un peu inhabituel.

Client signifie un utilisateur, ou programme agissant directement au nom d'un utilisateur. Cela peut être un lecteur de messages agissant comme un client IMAP, ou une coquille interactive, où l'utilisateur peut taper directement les commandes/demandes de protocole, ou ce peut être un descriptif ou programme écrit par l'utilisateur.

Serveur signifie un programme qui effectue les services demandés par le client. Ce peut être un serveur traditionnel comme un serveur HTTP, ou ce peut être un interpréteur Sieve [RFC3028] fonctionnant sur un descriptif Sieve écrit par un utilisateur. Un serveur a besoin d'utiliser les opérations fournies par les collations afin de satisfaire les demandes du client.

Le protocole décrit comment le client dit au serveur ce qu'il veut faire, et (si applicable) comment le serveur dit au client le résultat. IMAP est un protocole par sa définition, et il est donc le langage de Sieve.

2.4 Clés de tri

Une composante d'une collation est une transformation, qui transforme une chaîne en une clé de tri, qui est alors utilisée lors du tri.

La transformation peut aller d'une transposition d'identité (par exemple, la collation i;octet du paragraphe 9.3) à une transposition qui rend la chaîne illisible à un humain.

C'est un détail de mise en œuvre des collations ou des serveurs. Un protocole NE DEVRAIT PAS l'exposer aux clients, car certaines collations laissent le format des clés de tri au choix de la mise en œuvre, et les mises en œuvre conformes actuelles sont connues pour utiliser des formats différents.

3. Syntaxe d'identifiant de collation

3.1 Syntaxe de base

L'identifiant de collation lui-même est une seule chaîne ASCII. L'identifiant NE DOIT PAS faire plus de 254 caractères, et obéit à la grammaire suivante :

collation-char = ALPHA / DIGIT / "-" / ";" / "=" / "."

collation-id = collation-scope ";" collation-core-name*collation-arg

collation-scope = Language-tag / "vnd-" reg-name

collation-core-name = ALPHA *(ALPHA / DIGIT / "-")

collation-arg = ";" ALPHA *(ALPHA / DIGIT) "=" 1*(ALPHA / DIGIT / ".")

Note : la production ABNF "Language-tag" est importée de Étiquettes de langues [RFC4646] et "reg-name" de la syntaxe générique de URI: [RFC3986].

Il y a un identifiant spécial appelé "default". Pour les protocoles qui ont une collation par défaut, "default" se réfère à cette collation. Pour les autres protocoles, l'identifiant "default" DOIT ne correspondre à aucune collation, et les serveurs DEVRAIENT le traiter de la même façon que des collations non existantes.

3.2 Caractères génériques

La chaîne qu'un client utilise pour choisir une collation PEUT contenir un ou plusieurs caractères génériques ("*") qui correspondent à zéro, un ou plusieurs caractères de collation (*collation-char*). Les caractères génériques NE DOIVENT PAS être adjacents. Si la chaîne à caractères génériques correspond à plusieurs collations, le serveur DEVRAIT tenter de choisir une collation largement utile de préférence à une d'utilité plus étroite.

collation-wild = ("*" / (ALPHA ["*"])) *(collation-char ["*"]) ; NE DOIT PAS excéder 254 caractères au total.

3.3 Direction d'ordre

Quand il est utilisé comme un élément de protocole pour le rangement, l'identifiant de collation PEUT être muni d'un préfixe "+" ou "-" pour spécifier explicitement une direction d'ordre. "+" n'a pas d'effet sur l'opération de rangement, tandis que "-" inverse le résultat de l'opération de rangement. En général, l'ordre de collation est utilisé quand un client demande une collation, et collation-selected (*collation choisie*) est utilisé quand le serveur informe le client de la collation choisie.

collation-selected = ["+" / "-"] collation-id

collation-order = ["+" / "-"] collation-wild

3.4 URI

Certains protocoles sont conçus pour utiliser des URI [RFC3986] pour se référer aux collations plutôt que de simples jetons. Une section spéciale de l'espace d'URL de l'IANA est réservée à un tel usage. La forme "collation-uri" est utilisée pour se référer à une collation désignée spécifique (l'enregistrement de collation peut n'être en fait pas présent). La forme "collation-auri" est un nom abstrait pour un rangement, un schéma de collation ou une collation privée de fabricant.

collation-uri = "http://www.iana.org/assignments/collation/"collation-id ".xml"

collation-auri = ("http://www.iana.org/assignments/collation/"collation-order ".xml") / other-uri

other-uri = <absoluteURI> ; excluant l'espace de noms collation de l'IANA.

3.5 Lignes directrices pour les dénominations

Bien que la présente spécification ne fasse pas d'exigence absolue sur la structure des identifiants de collation, la cohérence de dénomination est importante, donc on fournit les lignes directrices initiales suivantes.

Les identifiants de collation avec une audience internationale commencent normalement par "i;". Les identifiants de collation destinés à un langage particulier ou local commencent normalement par une étiquette de langue [RFC4646] suivie par un ";". Après le premier ";" est normalement le nom de l'algorithme général de collation, suivi par une série de modifications de l'algorithme séparées par le délimiteur ";". Les modifications paramétrées vont utiliser "=" pour délimiter le paramètre de la valeur. Les numéros de version de tout tableau de recherche utilisé par l'algorithme DEVRAIENT être présents comme modifications paramétrées.

Les identifiants de collation de la forme *;vnd-hostname;* sont réservés pour des collations spécifiques de fabricant créées par le possesseur du nom d'hôte suivant le préfixe "vnd-" (par exemple, vnd-exemple.com pour le fabricant exemple.com). L'enregistrement de ces collations (ou l'espace de noms dans sa totalité) avec l'utilisation prévue du "Vendeur", est encouragé quand une spécification publique ou une mise en œuvre libre est disponible, mais n'est pas exigé.

4. Exigences de spécification de collation

4.1 Interface de collation/serveur

La collation elle-même définit ce sur quoi elle opère. La plupart des collations sont supposées opérer sur des chaînes de caractères. La collation `i;octet` (paragraphe 9.3) opère sur des chaînes d'octets. L'opération `i;ascii-numeric` (paragraphe 9.1) opère sur des nombres.

La présente spécification définit l'interface de collation en termes de chaînes d'octets. Cependant, les mises en œuvre peuvent choisir d'utiliser à la place des chaînes de caractères. De telles mises en œuvre peuvent n'être pas capables de mettre en œuvre par exemple, `i;octet`. Comme `i;octet` n'est actuellement de mise en œuvre obligatoire pour aucun protocole, cela ne devrait pas poser de problème.

4.2 Opérations prises en charge

Une spécification de collation DOIT déclarer lesquelles des trois opérations de base sont supportées (égalité, sous chaîne, rangement) et comment effectuer chacune des opérations supportées sur toute paires de chaînes de caractères d'entrée, incluant des chaînes vides. Les collations doivent être déterministes, c'est-à-dire, étant donnée une collation avec un identifiant spécifique, et deux chaînes d'entrée fixées, le résultat DOIT être le même pour la même opération.

En général, les opérations de collation devraient se comporter comme leur nom le suggère. Bien qu'une collation puisse être nouvelle, les opérations ne le sont pas, donc les opérations d'une nouvelle collation devraient être similaires à celles des plus anciennes collations. Par exemple, une collection date/heure ne devrait pas fournir une opération "sous chaîne" qui transformerait une sous chaîne IMAP SEARCH en, par exemple, une recherche date-gamme.

Une conséquence non évidente des règles pour chaque opération de collation est que, pour toute collation seule, soit aucune, soit toutes les opérations peuvent retourner "indéfini". Par exemple, il n'est pas possible d'avoir une opération égalité qui ne retourne jamais "indéfini", et une opération de sous chaîne qui le fasse occasionnellement.

4.2.1 Validité

L'essai de validité prend une chaîne comme argument. Il retourne valide si sa chaîne d'entrée est une entrée valide pour les autres opérations de la collation, et invalide sinon. (En d'autres termes, une chaîne est valide si elle est égale à elle-même en accord avec l'opération d'égalité de la collation.)

L'essai de validité est fourni par toutes les collations. Il NE DOIT PAS être mentionné séparément dans l'enregistrement de la collation.

4.2.2 Égalité

L'essai d'égalité retourne toujours "match" ou "no-match" quand il lui est fourni une entrée valide, et PEUT retourner "indéfini" si une ou les deux chaînes d'entrée ne sont pas valides.

L'essai d'égalité DOIT être réflexif et symétrique. Pour une entrée valide, il DOIT être transitif.

Si une collation fournit une sous chaîne ou un essai d'ordre, elle DOIT aussi fournir un essai d'égalité. Les essais de sous chaîne et/ou d'ordre DOIVENT être cohérents avec l'essai d'égalité.

Les valeurs de retour de l'essai d'égalité sont appelées "match", "no-match" et "indéfini" dans ce document.

4.2.3 Sous chaîne

L'opération de correspondance de sous chaîne détermine si la première chaîne est une sous chaîne de la seconde chaîne, c'est-à-dire, si une ou plusieurs sous chaînes de la seconde chaîne sont égales à la première, comme défini par l'opération d'égalité de la collation.

Une collation qui prend en charge la correspondance de sous chaîne va automatiquement prendre en charge deux cas

particuliers de correspondance de sous chaîne : correspondance de préfixe et de suffixe, si ces cas particuliers sont supportés par le protocole d'application. Il retourne "match" ou "no-match" quand une entrée valide est fournie et retourne "indéfini" quand une entrée invalide est fournie.

Les protocoles d'application PEUVENT retourner des informations de position pour les correspondances de sous chaînes. Si cela est fait, les informations de position DEVRAIENT inclure le décalage de début et le décalage de fin pour chaque correspondance. Ceci est important parce que des collations plus sophistiquées peuvent faire correspondre des chaînes de longueur inégale (par exemple, un caractère pré composé accentué peut correspondre à un caractère accentué décomposé). En général, les correspondances en chevauchement DEVRAIENT être rapportées (comme quand "ana" se produit deux fois dans "banana") bien qu'il y ait des cas où une collation peut décider de ne pas le faire. Par exemple, dans une collation qui traite toutes les séquences d'espaces comme identiques, l'opération de sous chaîne pourrait être définie de telle façon que " 1 " (SP "1" SP) soit rapporté juste une fois dans " 1 " (SP SP "1" SP SP), et non quatre fois (SP SP "1" SP, SP "1" SP, SP "1" SP SP et SP SP "1" SP SP) car les quatre correspondances sont, en un sens, la même correspondance.

Une chaîne est une sous chaîne d'elle-même. La chaîne vide est une sous chaîne de toutes les chaînes.

Noter que l'opération sous chaîne de certaines collations peut correspondre à des chaînes de longueur inégale. Par exemple, un caractère pré composé accentué peut correspondre à un caractère accentué décomposé. L'algorithme de collation Unicode [Unicode] discute cela plus en détails.

Les valeurs retournées de l'opération sous chaîne sont appelées "match", "no-match", et "indéfini" dans ce document.

4.2.4 Ordre

L'opération de rangement détermine comment deux chaînes sont ordonnées. Elle DOIT être réflexive. Pour une entrée valide, elle DOIT être transitive et trichotomique.

L'ordre retourne "moins" si la première chaîne est mentionnée avant la seconde chaîne, selon la collation ; "plus grand", si la seconde chaîne est mentionnée avant la première chaîne; et "égal", si les deux chaînes sont égales, comme défini par l'opération égalité de la collation. Si une des chaînes ou les deux sont invalides, le résultat du rangement est "indéfini".

Quand la collation est utilisée avec un préfixe "+", le comportement est le même que quand elle est utilisée sans préfixe. Quand la collation est utilisée avec un préfixe "-", le résultat de l'opération de rangement de la collation DOIT être inversé.

Les valeurs retournées par l'opération de rangement sont appelées "moins", "égal", "plus grand", et "indéfini" dans le présent document.

4.3 Clés de tri

Une spécification de collation DEVRAIT décrire l'algorithme de transformation interne pour générer les clés de tri. Cet algorithme peut être appliqué aux chaînes individuelles, et le résultat peut être mémorisé pour potentiellement optimiser de futures opérations de comparaison. Une collation PEUT spécifier que la clé de tri est générée par la fonction d'identité. La clé de tri peut n'avoir pas de signification pour un humain. La clé de tri peut n'être pas une entrée valide pour la collation.

4.4 Utilisation de tableaux de recherche

Certaines collations utilisent des tableaux de recherche personnalisables, par exemple, parce que les tableaux dépendent de particularités locales, et peuvent être modifiés après le chargement du logiciel. Les collations qui utilisent plus d'un tableau de recherche personnalisable dans un format documenté DOIVENT allouer des numéros aux tableaux qu'elles utilisent. Cela permet à une commande de protocole d'application d'accéder aux tableaux utilisés par une collation de serveur, afin que clients et serveurs utilisent les mêmes tableaux.

5. Exigences de protocole d'application

Cette section décrit les exigences et problèmes qu'un protocole d'application doit prendre en compte si il offre la recherche, la confrontation de sous chaînes et/ou le tri, et permet l'utilisation de caractères hors du jeu de caractères US-ASCII.

5.1 Codage de caractères

La spécification du protocole doit s'assurer qu'il est clair sur quels caractères (plutôt que juste les octets) sont utilisés dans les collations. Cela peut être fait en spécifiant le protocole lui-même en termes de caractères (par exemple, dans le cas d'un langage d'interrogation) en spécifiant un seul codage de caractères pour le protocole (par exemple, UTF-8 [RFC3629]), ou en décrivant avec soin les problèmes pertinents de l'étiquetage et de la conversion du codage de caractères. Dans ce dernier cas, les détails à considérer incluent comment traiter les jeux de caractères inconnus, tous les jeux de caractères dont la mise en œuvre est obligatoire, tous les problèmes d'ordre des octets qui pourraient s'appliquer, et tous les codages de transfert qui doivent être pris en charge.

5.2 Opérations

Le protocole doit spécifier quelles opérations définies dans la présente spécification (confrontation pour égalité, correspondance de sous chaîne, et ordre) peuvent être invoquées dans le protocole, et comment elles sont invoquées. Il peut y avoir plus d'une façon d'invoquer une opération.

Le protocole DOIT fournir un mécanisme pour que le client choisisse la collation à utiliser avec la confrontation pour égalité, la correspondance de chaîne, et l'ordre.

Si un protocole a besoin d'un rangement total et si la collation choisie ne le fournit pas parce que l'opération d'ordre retourne "indéfini" au moins une fois, le repli recommandé est de trier toutes les chaînes invalides après les valides, et d'utiliser i;octet pour ordonner les chaînes invalides.

Bien que la fonction de sous chaîne de la collation fournisse une liste des correspondances, un protocole n'a pas besoin de fournir tout cela au client. Il peut fournir seulement la première sous chaîne qui correspond, ou même juste l'information que la recherche de sous chaîne a trouvé une correspondance. De cette façon, les collations peuvent être utilisées avec des protocoles qui sont définis de telle façon que "x est une sous chaîne de y" retourne vrai ou faux.

Si le protocole fournit des informations de position pour le résultat d'une correspondance de sous chaîne, ces informations de position DEVRAIENT spécifier complètement la ou les sous chaînes dans le résultat qui correspond, indépendamment de la longueur de la chaîne de recherche. Par exemple, retourner les deux décalages de début et de fin de la correspondance devrait suffire, comme le feraient le décalage de début et une longueur. Retourner juste le décalage de début n'est pas acceptable. Cette règle est nécessaire parce que des collations évoluées peuvent traiter des chaînes de longueur différente comme égales (par exemple, des caractères accentués pré composés et décomposés).

5.3 Caractères génériques

Le protocole DOIT spécifier si il permet l'utilisation de caractères génériques dans les identifiants de collation. Si le protocole permet les caractères génériques, alors : le protocole DOIT spécifier comment les comparaisons se comportent en l'absence d'une négociation explicite de collation, ou quand une collation de "défaut" est demandée. Le protocole PEUT spécifier que la collation par défaut utilisée dans de telles circonstances est sensible à la configuration du serveur.

Le protocole DEVRAIT donner un moyen de faire la liste des collations disponibles correspondant à un ou des schémas de caractères génériques donnés.

5.4 Comparaison de chaînes

Si un protocole compare des chaînes d'une façon non triviale, utiliser une collation peut être approprié. Par exemple, de nombreux protocoles utilisent des chaînes indépendantes de la casse. Dans de nombreux cas, une simple transposition ASCII de majuscules/minuscules fonctionne bien. Dans d'autres cas, il peut être mieux d'utiliser une collation spécifiable ; par exemple, de façon qu'un serveur puisse traiter "i" et "I" comme équivalents en Italie, et différents en Turquie (le turc a aussi un "I" majuscule avec un point et un "i" minuscule sans point).

Les concepteurs de protocoles devraient considérer, dans chaque cas, si il faut utiliser une collation spécifiable. Les mots clés ont souvent d'autres besoins que les variables d'utilisateur, et les arguments de recherche peuvent être là aussi différents.

5.5 Clients déconnectés

Si le protocole prend en charge les clients déconnectés, et si on se sert d'une collation qui peut utiliser des tableaux configurables (par exemple, pour prendre en charge des extensions spécifiques locales) alors le client peut n'être pas capable de reproduire les opérations de collation du serveur lorsque il est hors ligne.

Un mécanisme pour télécharger de tels tableaux a été discuté. Ce mécanisme n'est pas inclus dans la présente spécification, car le problème n'est pas encore bien compris.

5.6 Codes d'erreur

La spécification du protocole devrait envisager d'allouer des codes d'erreur de protocole pour les circonstances suivantes :

- o Le client demande l'utilisation d'une collation par identifiant ou schéma, mais aucune collation mise en œuvre ne correspond à ce schéma.
- o Le client tente d'utiliser une collation pour une opération qui n'est pas prise en charge par cette collation -- par exemple, tenter d'utiliser la collation "i;ascii-numeric" pour la correspondance de sous chaîne.
- o Le client utilise une collation de correspondance d'égalité ou de sous chaîne, et le résultat est une erreur. Il peut être approprié de distinguer entre les deux chaînes d'entrée, en particulier quand l'une est fournie par le client et l'autre est mémorisée par le serveur. Il peut aussi être approprié de distinguer le cas spécifique d'une chaîne UTF-8 invalide.

5.7 Collation d'octets

La collation i;octet (paragraphe 9.3) n'est utilisable qu'avec les protocoles fondés sur des chaînes d'octets. Les clients et serveurs NE DOIVENT PAS utiliser i;octet avec d'autres protocoles.

Si le protocole permet l'utilisation de collations avec des structures de données autres que des chaînes, le protocole DOIT décrire le comportement par défaut pour une collation avec ces structures de données.

6. Utilisation par les protocoles existants

Cette section est pour information.

ACAP [RFC2244] et Sieve [RFC3028] sont toutes deux des spécifications sur la voie de la normalisation qui utilisaient des collations avant la création de la présente spécification et son registre. Ces normes ne satisfont pas toutes les exigences de protocole d'application décrites à la Section 5.

Ces protocoles permettent l'utilisation de la collation i;octet (paragraphe 9.3) fonctionnant directement sur des données UTF-8, utilisées dans ces protocoles.

Dans Sieve, toutes les correspondances sont soit vraies, soit fausses. Par conséquent, les serveurs Sieve doivent traiter les résultats "indéfini" et "no-match" des opérations d'égalité et de correspondance de sous chaîne comme fausses, et seulement "match" comme vraies.

Dans ACAP et Sieve, il n'y a pas de chaînes invalides. Dans les termes du présent document, les chaînes invalides sont triées après les chaînes valides.

IMAP [RFC3501] fait aussi des collations, bien que ce ne soit explicite que quand l'extension COMPARATOR [RFC5255] est utilisée. L'opération incorporée de sous chaîne de IMAP et le rangement fourni par l'extension SORT [RFC5256] ne peuvent pas satisfaire aux exigences du présent document.

D'autres protocoles peuvent être dans une position similaire.

Dans IMAP, la collation par défaut est i;ascii-casemap, parce que ses opérations sont comprises comme correspondant aux opérations incorporées dans IMAP.

7. Enregistrement de collation

7.1 Procédure d'enregistrement de collation

L'IETF va créer une liste de diffusion, `collation@ietf.org`, qui peut être utilisée pour la discussion publique des propositions de collations avant leur enregistrement. L'utilisation de la liste de diffusion est fortement encouragée. L'IESG appointera un expert désigné pour surveiller la liste de diffusion `collation@ietf.org` et revoir les enregistrements.

La procédure d'enregistrement commence quand un gabarit d'enregistrement complété est envoyé à `iana@iana.org` et `collation@ietf.org`. L'expert désigné est supposé dire à l'IANA et au soumettant de l'enregistrement dans les deux semaines si l'enregistrement est approuvé, approuvé avec des changements mineurs, ou rejeté avec la cause du rejet. Quand un enregistrement est rejeté avec une cause, il peut être re-soumis si les problèmes mentionnés dans la cause du rejet sont réglés. Les décisions de l'expert désigné peuvent être soumises en appel au directeur de la zone Applications de l'IESG, puis à l'IESG. Cela suit la procédure normale d'appel pour les décisions de l'IESG.

Les enregistrements de collation dans une RFC sur la voie de la normalisation, un BCP, ou une RFC expérimentale approuvée par l'IESG sont la propriété de l'IETF, et les changements à l'enregistrement suivent les procédures normales pour la mise à jour de tels documents. Les enregistrements de collation dans les autres RFC sont la propriété des auteurs de la RFC. Les autres enregistrements de collation sont la propriété des individus mentionnés dans le champ contact de l'enregistrement, et l'IANA conservera cette information.

Si l'enregistrement est un changement d'une collation existante, il DOIT être approuvé par le propriétaire. Au cas où le propriétaire ne peut pas être contacté pendant une période de un mois, et si l'expert désigné estime le changement nécessaire, l'IESG PEUT réallouer la propriété à une partie appropriée.

7.2 Format d'enregistrement de collation

L'enregistrement d'une collation est fait par l'envoi d'un document XML bien formé à `collation@ietf.org` et `iana@iana.org`.

7.2.1 Gabarit d'enregistrement

Voici un gabarit d'enregistrement :

```
<?xml version='1.0'?>
<!DOCTYPE collation SYSTEM 'collationreg.dtd'>
<collation rfc="YYYY" scope="global" intendedUse="common">
  <identifiant>identifiant de collation</identifiant>
  <titre>titre technique pour la collation</titre>
  <operations>equality order subchaîne</operations>
  <specification>référence de la spécification</specification>
  <owner>adresse de messagerie électronique du propriétaire ou IETF</owner>
  <submitter>adresse de messagerie électronique du soumettant</submitter>
  <version>1</version>
</collation>
```

7.2.2 Élément Collation

La racine du document d'enregistrement DOIT être un élément `<collation>`. L'élément collation contient les autres éléments de l'enregistrement, qui sont décrits dans les paragraphes suivants, dans l'ordre donné ici.

L'élément `<collation>` PEUT inclure un attribut `"rfc="` si la spécification est dans une RFC. L'attribut `"rfc="` donne seulement le numéro de la RFC, sans aucun préfixe, comme "RFC", ou suffixe, comme ".txt".

L'élément `<collation>` DOIT inclure un attribut `"scope="`, qui DOIT avoir une des valeurs "global", "local", ou "autre".

L'élément `<collation>` DOIT inclure un attribut `"intendedUse="`, qui doit avoir une des valeurs "common", "limited", "vendor", ou "deprecated". Les spécifications de collation destinées à l'usage "common" sont supposées faire référence à des normes d'organismes de normalisation ayant une expérience significative du traitement des détails des jeux de caractères internationaux.

Il faut savoir que de futures révisions de la présente spécification pourront ajouter des types de fonctions supplémentaires, ainsi que des attributs, valeurs, et éléments XML supplémentaires. Tout système qui analyse automatiquement ces documents XML DOIT prendre cela en compte pour préserver la future compatibilité.

7.2.3 Élément Identifier

L'élément <identifier> donne l'identifiant précis de la collation, par exemple, i;ascii-casemap. L'élément <identifier> est obligatoire.

7.2.4 Élément Title

L'élément <title> donne le titre de la collation. L'élément <title> est obligatoire.

7.2.5 Élément Operations

L'élément <operations> fait la liste des opérations parmi les trois ("égalité", "ordre" ou "sous chaîne") que la collation fournit, séparées par une seule espace. L'élément <operations> est obligatoire.

7.2.6 Élément Specification

L'élément <specification> décrit où trouver la spécification. L'élément <specification> est obligatoire. Il PEUT avoir un attribut URI. Il peut y avoir plus d'un élément <specification> ; dans ce cas, ils forment ensemble la spécification.

Si il est découvert que des parties d'une spécification de collation sont en conflit, une nouvelle révision de la collation est nécessaire, et la liste de diffusion collation@ietf.org devrait en être notifiée.

7.2.7 Élément Submitter

L'élément <submitter> donne une adresse de messagerie électronique de la [RFC2822] de la personne qui a soumis l'enregistrement. Il est facultatif si l'élément <owner> contient une adresse de messagerie électronique.

Il peut y avoir plus d'un élément <submitter>.

7.2.8 Élément Owner

L'élément <owner> contient soit les quatre lettres "IETF", soit une adresse de messagerie électronique du propriétaire de l'enregistrement. L'élément <owner> est obligatoire. Il peut y avoir plus d'un élément <owner>. Si il en est ainsi, tous les propriétaires sont égaux. Chaque propriétaire peut parler pour tous.

7.2.9 Élément Version

L'élément <version> DOIT être inclus quand l'enregistrement va probablement être révisé, ou a été révisé d'une façon telle que le résultat change pour une ou plusieurs chaînes d'entrée. L'élément <version> est facultatif.

7.2.10 Élément Variable

L'élément <variable> spécifie une variable facultative pour contrôler le comportement de la collation, par exemple si elle est sensible à la casse. L'élément <variable> est facultatif. Quand <variable> est utilisé, il doit contenir les éléments <name> et <default>, et il peut contenir un ou plusieurs éléments <value>.

7.2.10.1 Élément Name

L'élément <name> spécifie la valeur de nom d'une variable. L'élément <name> est obligatoire.

7.2.10.2 Élément Default

L'élément <default> spécifie la valeur par défaut d'une variable. L'élément <default> est obligatoire.

7.2.10.3 Élément Value

L'élément <value> spécifie une valeur légale d'une variable. L'élément <value> est facultatif. Si un ou plusieurs éléments <value> sont présents, seules ces valeurs sont légales. Si aucune n'est présente, les valeurs légales de la variable ne forment pas un ensemble énuméré, et les règles DOIVENT être spécifiées dans une RFC accompagnant l'enregistrement.

7.3 Structure de registre de collation

Une fois l'enregistrement approuvé, l'IANA va mémoriser chaque document d'enregistrement XML dans un URL de la forme <http://www.iana.org/assignments/collation/collation-id.xml>, où collation-id est le contenu de l'élément identifiant dans l'enregistrement. Le soumettant et l'expert désigné sont tous deux responsables de la vérification que le XML est bien formé. Le document d'enregistrement devrait éviter d'utiliser de nouveaux éléments. Si ils sont nécessaires, il est important qu'ils soient cohérents avec les autres enregistrements.

L'IANA va aussi tenir un résumé textuel du registre sous le nom <http://www.iana.org/assignments/collation/collation-index.html>. Ce résumé est divisé en quatre sections. La première section est pour les collations destinées à un usage commun. Cette section est destinée aux enregistrements de collations publiés dans des RFC approuvées par l'IESG, ou pour des collations de portée locale provenant du principal organisme de normalisation pour cette localisation. L'expert désigné est invité à rejeter les enregistrements de collation avec un usage prévu de "commun" si l'expert estime qu'il devrait être "limité", car il est souhaitable de garder petit et de haute qualité le nombre d'enregistrements "communs". La seconde section est réservée aux collations d'utilisation limitée. La troisième section est réservée aux collations spécifiques de fabricants enregistrés. La dernière section est réservée aux collations déconseillées.

7.4 Exemple de sommaire de registre initial

Ce qui suit est un exemple de la façon dont l'IANA pourrait structurer le registre initial du fichier summary.html :

Collation	Fonctions	Portée	Référence
Collations d'utilisation commune :			
i;ascii-casemap	e, o, s	Locale	[RFC 4790]
Collations d'utilisation limitée :			
i;octet	e, o, s	Autre	[RFC 4790]
i;ascii-numeric	e, o	Autre	[RFC 4790]
Collations de fabricant :			
Collations déconseillées :			

Références

[RFC4790] Newman, C., Duerst, M., Gulbrandsen, A., "Internet Application Protocol Collation Registry", RFC 4790, Sun Microsystems, mars 2007.

8. Lignes directrices pour un expert réviseur

L'expert réviseur appointé par l'IESG a une très grande latitude pour ce registre. Bien qu'un certain nombre de collations soit attendu (en particulier des personnalisations de l'UCA pour utilisation localisée) une explosion de collations (en particulier de collations d'utilisation commune) n'est pas désirable pour une large interopérabilité. Cependant, il est important que l'expert réviseur fournisse la cause du rejet d'un enregistrement, et, lorsque possible, il décrive l'action corrective qui permettrait à l'enregistrement de se faire. Le tableau suivant inclut des exemples de raisons de rejet d'un enregistrement avec sa cause :

- o L'enregistrement n'est pas un document XML bien formé.
- o L'enregistrement a un usage prévu de "commun", mais il n'est pas prouvé que la collation va être largement déployée, donc elle devrait être marquée comme "limité".
- o L'enregistrement a un usage prévu de "commun", mais il est redondant avec la fonctionnalité d'une collation précédemment enregistrée comme "commun".
- o L'enregistrement a un usage prévu de "commun", mais la spécification n'est pas assez détaillée pour permettre une mise

- en œuvre interopérable par les autres.
- o L'identifiant de collation n'identifie pas avec précision les numéros de version des tableaux pertinents à utiliser.
- o L'enregistrement ne satisfait pas aux exigences marquées "DOIT" de la Section 4.
- o L'identifiant de collation ne satisfait pas à la syntaxe de la Section 3.
- o La spécification de collation référencée dans l'enregistrement est vague ou a des caractéristiques facultatives sans qu'un comportement clair soit spécifié.
- o La spécification référencée ne traite pas de façon adéquate les considérations de sécurité spécifiques de cette collation.
- o Les opérations de l'enregistrement sont sans nécessité différentes des opérations traditionnelles.
- o Le XML de l'enregistrement est sans nécessité différent de celui des collations déjà enregistrées.

9. Collations initiales

Cette section enregistre les trois collations qui étaient à l'origine définies dans la [RFC2244], et sont mises en œuvre dans la plupart des moteurs de la [RFC3028]. Le comportement de ces collations n'est peut-être pas idéal, comme `i;ascii-casemap` qui n'accepte pas les entrées non ASCII. La compatibilité avec le code largement déployé a été jugée plus importante que de corriger les collations. Certains aspects de ces collations sont nécessaires pour conserver la compatibilité avec le code largement déployé.

9.1 Collation numérique ASCII

9.1.1 Description de la collation numérique ASCII

La collation `"i;ascii-numeric"` est une collation simple destinée à être utilisée avec des nombres décimaux entiers non signés de taille arbitraire, mémorisés comme des chaînes d'octets. Les chiffres US-ASCII (0x30 à 0x39) représentent les chiffres des nombres. Avant de convertir de chaîne à entier, la chaîne d'entrée est tronquée au premier caractère non chiffre. Toute entrée est valide ; les chaînes qui ne commencent pas par un chiffre représentent l'infini positif.

La collation prend en charge l'égalité et le rangement, mais ne prend pas en charge l'opération de sous chaîne.

L'opération égalité retourne `"match"` si les deux chaînes représentent le même nombre (c'est-à-dire, les zéros en tête et les non chiffres en queue ne sont pas regardés) et `"no-match"` si les deux chaînes représentent des nombres différents.

L'opération de rangement retourne `"moins"` si la première chaîne représente un plus petit nombre que la seconde, `"égal"` si elles représentent le même nombre, et `"plus grand"` si la première chaîne représente un plus grand nombre que la seconde.

Exemples : `"0"` est moins que `"1"`, et `"1"` est moins que `"4294967298"`. `"4294967298"`, `"04294967298"`, et `"4294967298b"` sont tous égaux. `"04294967298"` est moins que `"."`, `" "`, `"x"`, et `"y"` sont égaux.

9.1.2 Enregistrement de collation numérique ASCII

```
<?xml version='1.0'?>
<!DOCTYPE collation SYSTEM 'collationreg.dtd'>
<collation rfc="4790" scope="other" intendedUse="limited">
  <identifieur>i;ascii-numeric</identifieur>
  <title>ASCII Numeric</title>
  <operations>equality order</operations>
  <specification>RFC 4790</specification>
  <owner>IETF</owner>
  <submitter>chris.newman@sun.com</submitter>
</collation>
```

9.2 Collation de Casemap ASCII

9.2.1 Description de collation de Casemap ASCII

La collation `"i;ascii-casemap"` est une collation simple qui opère sur des chaînes d'octets et traite les lettres US-ASCII sans considération de la casse. Elle fournit les opérations d'égalité, sous chaîne, et rangement. Toute entrée est valide. Noter que

les lettres en dehors de l'ASCII ne sont pas traitées comme insensibles à la casse.

Ses opérations d'égalité, rangement, et sous chaîne sont comme pour `i;octet`, sauf qu'au début, les lettres en minuscules (valeurs d'octet 97 à 122) dans chaque chaîne d'entrée sont changées en majuscules (valeurs d'octet 65 à 90).

On devrait faire attention quand on utilise des fonctions fournies par le système d'exploitation pour mettre en œuvre cette collation, car elle n'est pas sensible aux particularités locales. Des fonctions comme `strcasemp` et `toupper` sont parfois sensibles aux particularités locales, et peuvent de façon inappropriée transposer des lettres minuscules autres que de a à z en majuscules.

La collation `i;ascii-casemap` convient bien pour l'utilisation avec de nombreux protocoles Internet et langages informatiques. L'utilisation avec le langage naturel est souvent inappropriée ; même si la collation prend apparemment en charge des langages comme le swahili et l'anglais, en utilisation réelle, elle tend à mal trier un certain nombre de types de chaînes :

- o les noms de personne et de lieu qui contiennent du non ASCII,
- o les mots comme "naïve" (si épilé avec un accent, le caractère accentué pourrait pousser le mot au mauvais endroit dans une liste triée),
- o des noms comme "Lloyd" (qui, en gallois est trié après "Lyon", à la différence de l'anglais),
- o des chaînes contenant des symboles Euro et Livre sterling, des marques de citation autres que "", des tirets/traits d'union, etc.

9.2.2 Enregistrement de collation de Casemap ASCII

```
<?xml version='1.0'?>
<!DOCTYPE collation SYSTEM 'collationreg.dtd'>
<collation rfc="4790" scope="local" intendedUse="common">
  <identifier>i;ascii-casemap</identifier>
  <title>ASCII Casemap</title>
  <operations>equality order subchain</operations>
  <specification>RFC 4790</specification>
  <owner>IETF</owner>
  <submitter>chris.newman@sun.com</submitter>
</collation>
```

9.3 Collation d'octet

9.3.1 Description de collation d'octet

La collation "`i;octet`" est simple et rapide et destinée à être utilisée sur des chaînes d'octets binaires plutôt que des données de caractères. Les protocoles qui veulent rendre cette collation disponible doivent le faire en la permettant explicitement. Si elle n'est pas explicitement permise, elle NE DOIT PAS être utilisée. Elle ne retourne jamais un résultat "indéfini". Elle fournit les opérations égalité, sous chaîne, et rangement.

L'algorithme de rangement est le suivant :

1. Si les deux chaînes sont la chaîne vide, retourner le résultat "égal".
2. Si la première chaîne est vide et pas la seconde, retourner le résultat "moins".
3. Si la seconde chaîne est vide et pas la première, retourner le résultat "plus grand".
4. Si les deux chaînes commencent avec la même valeur d'octet, supprimer le premier octet des deux chaînes et répéter cet algorithme à partir de l'étape 1.
5. Si la valeur non signée (de 0 à 255) du premier octet de la première chaîne est moins que la valeur non signée du premier octet de la seconde chaîne, retourner "moins".
6. Si cette étape est atteinte, retourner "plus grand".

Cet algorithme est en gros équivalent à la fonction de bibliothèque C `memcmp`, avec en plus les vérifications appropriées de longueur.

L'opération de confrontation retourne "match" si l'algorithme de tri retournerait "égal". Autrement, l'opération de confrontation retourne "no-match".

L'opération de sous chaîne retourne "match" si la première chaîne est la chaîne vide, ou si il existe une sous chaîne de la seconde chaîne de longueur égale à la longueur de la première chaîne, qui résulterait en un résultat "match" de la fonction d'égalité. Autrement, l'opération de sous chaîne retourne "no-match".

9.3.2 Enregistrement de collation d'octet

Cette collation est définie avec `intendedUse="limited"` parce que elle ne peut être utilisée que par des protocoles qui la permettent explicitement.

```
<?xml version='1.0'?>
<!DOCTYPE collation SYSTEM 'collationreg.dtd'>
<collation rfc="4790" scope="global" intendedUse="limited">
  <identifiant>i;octet</identifiant>
  <titre>Octet</titre>
  <operations>equality order subchaîne</operations>
  <specification>RFC 4790</specification>
  <owner>IETF</owner>
  <submitter>chris.newman@sun.com</submitter>
</collation>
```

10. Considérations relatives à l'IANA

La Section 7 définit comment enregistrer les collations auprès de l'IANA. La Section 9 définit une liste de collations prédéfinies qui ont été enregistrées par l'IANA.

11. Considérations sur la sécurité

Les collations vont normalement être utilisées avec des chaînes UTF-8. Donc, les considérations de sécurité pour UTF-8 [RFC3629], stringprep [RFC3454], et Unicode TR-36 [Unicode Sec] s'appliquent aussi, et sont normatives pour la présente spécification.

12. Remerciements

Les auteurs tiennent à remercier tous ceux qui ont contribué à ce document, à savoir Brian Carpenter, John Cowan, Dave Cridland, Mark Davis, Spencer Dawkins, Lisa Dusseault, Lars Eggert, Frank Ellermann, Philip Guenther, Tony Hansen, Ted Hardie, Sam Hartman, Kjetil Torggrim Homme, Michael Kay, John Klensin, Alexey Melnikov, Jim Melton, et Abhijit Menon-Sen.

13. Références

13.1 Références normatives

- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par [RFC8174](#))
- [RFC3454] P. Hoffman et M. Blanchet, "[Préparation de chaînes internationalisées](#) ("stringprep")", décembre 2002. (P.S.)
- [RFC3629] F. Yergeau, "[UTF-8, un format de transformation](#) de la norme ISO 10646", STD 63, novembre 2003.
- [RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "[Identifiant de ressource uniforme](#) (URI) : Syntaxe générique", STD 66, janvier 2005. (P.S. ; MàJ par [RFC8820](#))
- [RFC4234] D. Crocker et P. Overell, "[BNF augmenté pour les spécifications de syntaxe](#) : ABNF", octobre 2005.

(Remplace RFC2234, remplacée par RFC5234)

- [RFC4646] A. Phillips, M. Davis, "[Étiquettes d'identification des langues](#)", BCP0047 septembre 2006. (Remplacée par [RFC5646](#))
- [Unicode] Davis, M. and K. Whistler, "Unicode Collation Algorithm version 14", mai 2005, <<http://www.unicode.org/reports/tr10/tr10-14.html>>.
- [Unicode Sec] Davis, M. and M. Suignard, "Unicode Security Considerations", février 2006, <<http://www.unicode.org/reports/tr36/>>.

13.1 Références pour information

- [RFC2045] N. Freed et N. Borenstein, "[Extensions de messagerie Internet](#) multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. (D. S., MàJ par [2184](#), [2231](#), [5335](#).)
- [RFC2244] C. Newman, J. G. Myers, "[ACAP – Protocole d'accès à la configuration d'application](#)", novembre 1997. (P.S.)
- [RFC2822] P. Resnick, "[Format de message Internet](#)", avril 2001. (Remplace la RFC0822, STD 11, Remplacée par RFC5322)
- [RFC2978] N. Freed et J. Postel, "Procédures d'enregistrement des jeux de caractère par l'IANA", BCP 19, octobre 2000.
- [RFC3028] T. Showalter, "Sieve : Langage de filtrage de messagerie", janvier 2001. (Obsolète, voir la [RFC5228](#)) (P.S.)
- [RFC3501] M. Crispin, "Protocole d'[accès au message Internet - version 4rev1](#)", mars 2003. (P.S. ; MàJ par [RFC4466](#), [4469](#), [4551](#), [5032](#), [5182](#), [7817](#), [8314](#), [8437](#), [8474](#))
- [RFC4422] A. Melnikov et K. Zeilenga, éd, "[Authentification simple et couche de sécurité](#) (SASL)", juin 2006. (P.S.)
- [RFC5255] C. Newman et autres, "Internationalisation du protocole d'accès au message Internet", juin 2008. (P.S.)
- [RFC5256] M. Crispin, K. Murchison, "Protocole d'accès au message Internet - extensions SORT et THREAD", juin 2008. (MàJ par [RFC5957](#)) (P.S.)

Adresse des auteurs

Chris Newman
Sun Microsystems
1050 Lakes Drive
West Covina, CA 91790
USA
mél : chris.newman@sun.com

Martin Duerst
Aoyama Gakuin University
5-10-1 Fuchinobe
Sagamihara, Kanagawa 229-8558
Japan
mél : duerst@it.aoyama.ac.jp

Arnt Gulbrandsen
Oryx Mail Systems GmbH
Schweppermannstr. 8
81671 Munich
Germany
mél : arnt@oryx.com

Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2007).

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

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

Propriété intellectuelle

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

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

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

Remerciement

Le financement de la fonction d'édition des RFC est fourni par la Internet Society.