

Groupe de travail Réseau
Request for Comments : 2589
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

Y. Yaacovi, Microsoft
M. Wahl, Innosoft International, Inc.
T. Genovese, Microsoft
mai 1999

Protocole léger d'accès de répertoire (v3) : Extensions pour les services de répertoire dynamique

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) The Internet Society (1999). Tous droits réservés.

1. Résumé

Le présent document définit les exigences pour les services dynamiques de répertoire et spécifie le format des opérations d'extension de demande et de réponse pour la prise en charge de l'interopération client-serveur dans un environnement de répertoires dynamiques.

Le protocole léger d'accès à un répertoire (LDAP, *Lightweight Directory Access Protocol*) [RFC2251] prend en charge un accès léger à des services de répertoire statiques, permettant une recherche et un accès aux mises à jour relativement rapides. Les services de répertoire statiques mémorisent les informations sur les gens qui contribuent à sa précision et sa valeur sur une longue période.

Les services de répertoire dynamiques sont différents en ce qu'ils mémorisent les informations dont la précision et la valeur ne persistent que quand elles sont rafraîchies périodiquement. Ces informations sont mémorisées comme entrées dynamiques dans le répertoire. Une utilisation typique serait un client ou une personne qui serait en ligne – auquel cas elle a une entrée dans le répertoire, ou serait hors ligne – auquel cas son entrée disparaît du répertoire. Bien que le fonctionnement et les attributs du protocole utilisés par les services de répertoire dynamique soient similaires à ceux utilisés pour les services de répertoire statiques, les clients qui mémorisent des informations dynamiques dans le répertoire ont besoin de rafraîchir périodiquement ces informations, afin de les empêcher de disparaître. Si les entrées dynamiques ne sont pas rafraîchies dans un délai donné, elles seront retirées du répertoire. Par exemple, cela va arriver si le client qui les a établies passe hors ligne.

Un mécanisme de contrôle de flux à partir du serveur est aussi décrit, qui permet au serveur d'informer les clients du délai dans lequel ils devraient rafraîchir leur présence.

2. Exigences

Les extensions au protocole doivent permettre d'accéder aux informations dynamiques dans un répertoire de la façon standard LDAP, pour permettre aux clients d'accéder aux informations statiques et dynamiques de la même manière.

Par définition, les entrées dynamiques ne sont pas persistantes et les clients peuvent s'en aller de bon gré ou non. Les extensions proposées doivent offrir un moyen pour qu'un serveur dise si les entrées sont encore valides, et de faire cela d'une façon adaptable. Il doit aussi y avoir un mécanisme pour que les clients rétablissent leur entrée auprès du serveur.

Il doit y avoir un moyen pour que les clients trouvent, de la façon standard de LDAP, si les serveurs prennent en charge les extensions dynamiques.

Finalement, pour permettre aux clients d'utiliser largement les extensions dynamiques, celles-ci doivent être enregistrées selon le fonctionnement LDAP étendu standard.

3. Description de l'approche

Le protocole léger d'accès à un répertoire (LDAP) [RFC2251] permet que des opérations supplémentaires de demandes et réponses soient ajoutées au protocole. Cette proposition en tire parti pour prendre en charge des répertoires qui contiennent des informations dynamiques d'une façon complètement intégrée à LDAP.

L'approche décrite dans cette proposition définit des entrées dynamiques afin de permettre de mettre en œuvre des répertoires avec des informations dynamiques. Une mise en œuvre de répertoires dynamiques doit être capable de prendre en charge les entrées de répertoire dynamiques.

3.1 Entrées dynamiques et classe d'objet `dynamicObject`

Une entrée dynamique est un objet dans l'arborescence de répertoire qui a une durée de vie (TTL, *time-to-live*) associée. Cette durée de vie est établie lors de la création de l'entrée. La durée de vie est automatiquement décrétementée, et lorsque elle arrive à expiration, l'entrée dynamique disparaît. En invoquant l'opération de rafraîchissement étendu (définie ci-dessous) pour rétablir la durée de vie, un client peut faire que l'entrée reste présente un peu plus longtemps.

Une entrée dynamique est créée en incluant la valeur d'`objectClass` donnée à la section 5 dans la liste des attributs lors de l'ajout d'une entrée. Cette méthode est soumise aux restrictions standard de contrôle d'accès.

L'opération étendue traitée ici, permet à un client de rafraîchir une entrée dynamique en invoquant, à un certain intervalle, les opérations de rafraîchissement qui contiennent le nom de cette entrée. Les entrées dynamiques seront traitées de la même façon que les entrées non dynamiques lors du traitement des opérations de recherche, comparaison, ajout, suppression, modification et `modifyDN`. Cependant, si les clients arrêtent d'envoyer des opérations de rafraîchissement pour une entrée, le serveur va alors automatiquement et sans notification retirer cette entrée du répertoire. Ce retrait sera traité de la même façon que si l'entrée avait été supprimée par une opération du protocole LDAP.

Il n'y a aucun moyen de changer une entrée statique en une entrée dynamique et vice-versa. Si le client utilise `Modify` avec une classe d'objet de `dynamicObject` sur une entrée statique, le serveur doit retourner une erreur de service soit `"objectClassModsProhibited"` (si le serveur ne permet pas du tout les modifications de `objectClass`) soit `"objectClassViolation"` (si le serveur permet en général les modifications de classe d'objet).

Un serveur dynamique peut être retiré par le client avec l'opération `delete` (*supprimer*). Cette opération sera soumise aux restrictions de contrôle d'accès.

Une entrée non dynamique ne peut pas être ajoutée en étant subordonnée à une entrée dynamique : si cela est tenté, le serveur doit retourner une mise à jour appropriée ou une erreur de service.

La prise en charge d'attributs dynamiques d'un objet par ailleurs statique sort du domaine d'application du présent document.

3.2 Réunions dynamiques (conférences)

La définition de `dynamicObject` lui associe une durée de vie, et c'est tout. Bien que l'objet dynamique le plus commun soit un objet `"person"`, il n'y a pas de type spécifique associé au `dynamicObject` tel que défini ici. Par l'utilisation des attributs de l'objet dynamique, on peut faire représenter pratiquement n'importe quoi à cet objet.

Précisément, les réunions (conférences) peuvent être représentées par des objets dynamiques. Alors que la prise en charge de toutes les caractéristiques d'une réunion exige une sémantique et un traitement particuliers de la part du serveur (ce qui sort du domaine d'application du présent document) les extensions décrites ici assurent la prise en charge de base des réunions. Un objet réunion peut être rafraîchi par les participants à la réunion, et lorsque il ne l'est pas, l'entrée disparaît. Le seul type de réunion qui est naturellement pris en charge par les extensions dynamiques est la réunion qui est sous le contrôle de son créateur.

3.2.1 Réunions sous le contrôle du créateur

Les réunions sous le contrôle du créateur sont créées par un client qui règle l'attribut durée de vie pour l'entrée, et il est de la responsabilité du client de rafraîchir l'entrée `"meeting"`, afin qu'elle ne disparaisse pas. D'autres peuvent se joindre à la réunion, en modifiant les attributs appropriés, mais ils n'ont pas l'autorisation de rafraîchir l'entrée. Lorsque le client qui a

créé l'entrée s'en va, il peut supprimer l'entrée "meeting", ou elle peut disparaître lors de l'arrivée à expiration de sa durée de vie. Ceci est cohérent avec le modèle courant de dynamicObject décrit ci-dessus.

4. Ajouts au protocole

4.1 Demande Refresh

Refresh est une opération du protocole qui est envoyée par un client pour dire au serveur que le client est encore en vie et que l'entrée de répertoire dynamique est toujours précise et en vigueur. Le client envoie une demande Refresh périodiquement sur la base de la valeur de la période de rafraîchissement du client (CRP, *client refresh period*). Le serveur peut demander que le client change cette valeur. Tant que le serveur reçoit une demande Refresh dans la période de temporisation, l'entrée de répertoire est assurée de persister sur le serveur. Les mises en œuvre de client devraient être conscientes que comme le réseau intervenant entre le client et le serveur n'est pas fiable, un paquet de demande Refresh peut être retardé ou perdu dans le transit. Si cela se produit, l'entrée peut disparaître, et le client aura besoin de le détecter et de rajouter l'entrée.

Un client peut demander cette opération en transmettant une unité de données de protocole (PDU, *Protocol Data Unit*) LDAP contenant une ExtendedRequest. Une ExtendedRequest LDAP est définie comme suit :

```
ExtendedRequest ::= [APPLICATION 23] SEQUENCE {
    requestName      [0] LDAPOID,
    requestValue     [1] OCTET STRING OPTIONAL }
```

Le champ requestName doit être réglé à la chaîne "1.3.6.1.4.1.1466.101.119.1".

Le champ requestValue va contenir comme valeur le codage en DER du type de données ASN.1 suivant :

```
SEQUENCE {
    entryName [0] LDAPDN,
    requestTtl [1] ENTIER
}
```

Le champ entryName est la représentation de chaîne UTF-8 du nom de l'entrée dynamique [RFC2253]. Cette entrée doit déjà exister.

Le requestTtl est une durée en secondes (entre 1 et 31 557 600) que demande le client, pendant laquelle l'entrée va exister dans le répertoire avant d'être automatiquement retirée. Les serveurs ne sont pas obligés d'accepter cette valeur et peuvent retourner une valeur de TTL différente au client. Les clients doivent être capables d'utiliser cette valeur dictée par le serveur comme CRP.

4.2 Réponse à Refresh

Si un serveur met en œuvre cette extension, il va alors retourner, lorsque la demande est faite, une PDU LDAP contenant une ExtendedResponse. Une ExtendedResponse LDAP est définie comme suit :

```
ExtendedResponse ::= [APPLICATION 24] SEQUENCE {
    COMPONENTS OF LDAPResult,
    responseName [10] LDAPOID OPTIONAL,
    response     [11] OCTET STRING OPTIONAL }
```

Le champ responseName contient la même chaîne que celle présente dans la demande.

Le champ de réponse va contenir comme valeur le codage en DER du type de données ASN.1 suivant :

```
SEQUENCE {
    responseTtl [1] ENTIER
}
```

Le champ responseTtl est la durée en secondes que choisit le serveur comme champ de durée de vie pour cette entrée. Elle

ne doit pas être inférieure à celle que demandait le client, et elle peut être supérieure. Cependant, pour permettre aux serveurs de conserver un répertoire relativement précis, et pour empêcher les clients d'abuser des extensions dynamiques, il est permis aux serveurs de raccourcir une valeur de durée de vie demandée par un client, jusqu'à un minimum de 86 400 secondes (24 heures).

Si l'opération a réussi, le champ `errorCode` dans la partie `standardResponse` d'une `ExtendedResponse` sera réglé à succès.

En cas d'erreur, le champ `responseTtl` aura la valeur de 0, et le champ `errorCode` va contenir une valeur appropriée, comme suit : si l'entrée désignée par `entryName` n'a pas pu être localisée, le champ `errorCode` va contenir "noSuchObject". Si l'entrée n'est pas dynamique, le champ `errorCode` va contenir "objectClassViolation". Si le demandeur n'a pas la permission de rafraîchir l'entrée, le champ `errorCode` va contenir "insufficientAccessRights". Si le champ `requestTtl` est trop grand, le champ `errorCode` va contenir "sizeLimitExceeded".

Si un serveur ne met pas en œuvre cette extension, il va retourner une PDU LDAP contenant une `ExtendedResponse`, qui contient seulement l'élément `standardResponse` (les éléments `responseName` et `response` seront absents). L'élément `LDAPResult` va indiquer le code de résultat `protocolError`.

Il est permis d'invoquer cette demande lorsque LDAP est porté par un transport sans connexion.

Lorsque on utilise un transport en mode connexion, il n'est pas exigé que cette opération soit sur la même connexion. Un client peut ouvrir plusieurs connexions, ou clore une connexion et la rouvrir ensuite.

4.3 X.500/DAP Modify(97)

Les serveurs X.500/DAP peuvent transposer les opérations de demande et réponse Refresh en opération X.500/DAP Modify(97).

5. Ajouts de schémas

Toutes les entrées dynamiques doivent avoir la valeur `dynamicObject` dans leur attribut `objectClass`. Cette classe d'objet est définie comme suit (en utilisant la notation `ObjectClassDescription` de la [RFC2252]) :

(1.3.6.1.4.1.1466.101.119.2 NAME 'dynamicObject'

DESC "Cette classe, si elle est présente dans une entrée, indique que cette entrée a une durée de vie limitée et peut disparaître automatiquement quand sa durée de vie atteint 0. Il n'y a pas d'attribut obligatoire de cette classe, cependant si le client n'a pas fourni une valeur pour l'attribut `entryTtl`, le serveur va en fournir une".

SUP top AUXILIARY)

De plus, l'entrée dynamique doit avoir l'attribut de fonctionnement suivant. Il est décrit en utilisant la notation `AttributeTypeDescription` de la [RFC2252] :

(1.3.6.1.4.1.1466.101.119.3 NAME 'entryTtl'

DESC "Cet attribut de fonctionnement est entretenu par le serveur et apparaît dans toute entrée dynamique. L'attribut est absent lorsque l'entrée ne contient pas la classe d'objet `dynamicObject`. La valeur de cet attribut est la durée en secondes pendant laquelle l'entrée va continuer d'exister avant de disparaître du répertoire. En l'absence d'opérations de rafraîchissement intermédiaires, les valeurs retournées par la lecture de l'attribut dans deux recherches successives sont garanties non croissantes. La plus petite valeur permise est 0, qui indique que l'entrée peut disparaître sans avertissement. L'attribut est marqué NO-USER-MODIFICATION car il ne peut être changé qu'en utilisant l'opération refresh".

SYNTAXE 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE

NO-USER-MODIFICATION USAGE dSAOperation)

Pour permettre aux serveurs de prendre en charge les entrées dynamiques dans seulement une partie de l'arborescence des informations d'annuaire (DIT, *Directory Information Tree*) on définit l'attribut de fonctionnement suivant. Il est décrit en utilisant la notation `AttributeTypeDescription` de la [RFC2252] :

(1.3.6.1.4.1.1466.101.119.4 NAME 'dynamicSubtrees'

DESC "Cet attribut de fonctionnement est tenu par le serveur et est présent dans le DSE racine, si le serveur prend en charge les extensions dynamiques décrites dans ce mémoire. L'attribut contient une liste de toutes les sous arborescences du répertoire pour lequel le serveur accepte les extensions dynamiques".

SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 NO-USER-MODIFICATION

USAGE dSAOperation)

6. Exigences pour le client et le serveur

6.1 Exigences pour le client

Les clients peuvent découvrir si un serveur prend en charge les extensions dynamiques en vérifiant le champ `supportedExtension` dans le DSE racine, pour voir si l'IDENTIFIANT D'OBJET décrit à la section 4 est présent. Comme les serveurs peuvent choisir, pour prendre en charge les extensions dynamiques, parmi seulement quelques unes des sous-arborescences de la DIT, les clients doivent vérifier l'attribut de fonctionnement `dynamicSubtrees` dans le DSE racine pour savoir si les extensions dynamiques sont acceptées sur une sous-arborescence spécifique.

Une fois qu'une entrée dynamique a été créée, les clients sont chargés d'invoquer l'opération d'extension `refresh`, afin de garder cette entrée présente dans le répertoire.

Les clients ne doivent pas s'attendre à ce qu'une entrée dynamique soit présente dans la DIT après l'arrivée à expiration de son délai de vie ; cependant ils ne doivent pas exiger que le serveur retire l'entrée immédiatement (certains serveurs peuvent ne traiter les entrées arrivées à expiration qu'à un intervalle donné). Si le client souhaite s'assurer que l'entrée n'existe pas, il devrait produire une `RemoveRequest` pour cette entrée.

Initialement, un client a besoin de savoir à quelle fréquence il devrait envoyer des demandes de rafraîchissement au serveur. Cette valeur est définie comme la période de rafraîchissement du client (CRP, *Client Refresh Period*) et elle est établie par le serveur sur la base du `entryTtl`.

Comme l'opération LDAP `AddRequest` est laissée inchangée et n'est pas modifiée par la présente proposition pour retourner cette valeur, le client doit produire une opération d'extension `Refresh` immédiatement après un `Add` qui a créé une entrée dynamique. La réponse `Refresh` va retourner la CRP (dans `responseTtl`) au client.

Les clients ne doivent pas produire de demande de rafraîchissement pour les entrées dynamiques qu'ils n'ont pas créées. Si un client anonyme tente de le faire, il est permis à un serveur de retourner "insufficientAccessRights" (50) dans la `RefreshResponse`, obligeant le client à se lier d'abord. Prière de noter que les serveurs qui permettent à des clients anonymes de créer et rafraîchir des entrées dynamiques ne seront pas capables de mettre cette disposition en application.

Les clients devraient toujours être prêts à traiter le cas dans lequel leur entrée arrive à expiration. Dans un tel cas, l'opération `Refresh` va échouer avec un code d'erreur comme `noSuchObject`, et le client est supposé recréer son entrée.

Les clients devraient être prêts à subir des échecs d'opérations de rafraîchissement avec `protocolError`, même si l'ajout et les demandes précédentes de rafraîchissement avaient réussi. Cela peut arriver si un mandataire tombe en panne entre le client et le serveur, et qu'un autre mandataire est utilisé, qui ne prend pas en charge l'opération d'extension `Refresh`.

6.2 Exigences pour le serveur

Les serveurs sont chargés de retirer les entrées dynamiques lorsque elles arrivent à expiration. Les serveurs ne sont pas obligés de le faire immédiatement.

Les serveurs doivent mettre en application les règles structurelles énumérées à la section 3.

Le serveur doit s'assurer que l'attribut de fonctionnement décrit à la section 5 est présent dans les entrées dynamiques.

Le serveur peut permettre à des usagers anonymes de rafraîchir des entrées. Cependant, pour éliminer la possibilité d'un usage malveillant de l'opération `Refresh`, les serveurs peuvent exiger que le client qui rafraîchit se lie d'abord. Une mise en œuvre de serveur peut réaliser cela en présentant des listes de contrôle d'accès (ACL, *Access Control List*) sur l'attribut `entryTtl`, et en retournant `insufficientAccessRights` (50) dans la `RefreshResponse`, si le client n'est pas autorisé à rafraîchir l'entrée. Faire ceci peut cependant avoir des conséquences sur les performances du serveur et peut impacter l'adaptabilité du serveur.

Les serveurs peuvent exiger qu'un client qui tente de créer une entrée dynamique ait une permission de retrait pour cette entrée.

Les serveurs qui mettent en œuvre les extensions dynamiques doivent avoir l'IDENTIFIANT D'OBJET, décrit à la section 4 pour la demande et la réponse, présente comme une valeur du champ `supportedExtension` dans le DSE racine. Ils doivent aussi avoir, comme valeurs d'attribut `attributeTypes` et `objectClasses` de leurs sous-entrées de sous-schéma, la `AttributeTypeDescription` et `ObjectClassDescription` de la section 5.

Les serveurs peuvent limiter la prise en charge des extensions dynamiques à seulement quelques sous-arborescences de la DIT. Les serveurs indiquent pour quelles sous-arborescences ils acceptent les extensions, en spécifiant les OID pour les sous-arborescences acceptées dans l'attribut `dynamicSubtrees` décrit à la section 5. Si un serveur prend en charge les extensions dynamiques pour tous les contextes de désignation qu'il détient, l'attribut `dynamicSubtrees` peut être absent.

7. Questions de mise en œuvre

7.1 Mémorisation des informations dynamiques

Les informations dynamiques sont supposées changer très souvent. De plus, on s'attend à ce que les demandes Refresh arrivent très souvent au serveur. Les bases de données fondées sur le disque qu'utilisent très souvent les bases de données des services de répertoires statiques sont vraisemblablement inappropriées pour mémoriser les informations dynamiques. On recommande que les mises en œuvre de serveur mémorisent les entrées dynamiques dans des mémoires suffisantes pour éviter les recherches. Ceci n'est pas une exigence.

On s'attend à ce que les serveurs LDAP soient capables de mémoriser des entrées statiques et dynamiques. Si un serveur LDAP ne prend pas en charge les entrées dynamiques, il devrait répondre par un code d'erreur tel que `objectClassViolation`.

7.2 Comportement de rafraîchissement du client

Dans certains cas, le client peut ne pas obtenir de réponse à la demande Refresh. Cela peut arriver par suite d'une panne de serveur après la réception de la demande Refresh, de la fin de temporisation de la prise TCP/IP dans le cas d'une connexion, ou de perte du paquet UDP dans le cas sans connexion.

Il est recommandé que dans un tel cas, le client réessaie immédiatement l'opération Refresh, et si cette demande Refresh n'obtient elle non plus pas de réponse, il revienne au cycle original de Refresh, c'est-à-dire, envoi d'une demande Refresh à sa période de rafraîchissement de client (CRP).

7.3 Configuration de l'heure de rafraîchissement

On recommande que les serveurs donnent aux administrateurs la capacité de configurer la période de rafraîchissement de client (CRP) par défaut, et aussi des valeurs de CRP minimum et maximum. Ceci, tout en permettant aux administrateurs de demander que le serveur ne change pas la CRP de façon dynamique, va permettre aux administrateurs d'établir des valeurs de CRP qui vont mettre en application un faible trafic de rafraîchissement, ou – à l'autre extrême, un répertoire très à jour.

8. Duplication

La duplication n'est que partiellement traitée dans le présent mémoire. Des travaux distincts sont en cours dans l'IETF sur la duplication des répertoires statiques et dynamiques.

Il est admis de dupliquer une entrée dynamique ou une entrée statique avec des attributs dynamiques. Comme le `entryTtl` est exprimé en temps relatif (combien de secondes jusqu'à ce que l'entrée arrive à expiration) la duplication signifie que l'entrée dupliquée va être "périmée" à la fin du temps de duplication.

9. Localisation

Il n'y a pas de problème de localisation pour ce fonctionnement étendu.

10. Considérations pour la sécurité

Les règles de sécurité standard de LDAP s'appliquent aux extensions décrites dans le présent document, et il n'y a pas de problème de sécurité particulier pour ces extensions. Prière de noter cependant que les serveurs peuvent permettre à des clients anonymes de rafraîchir des entrées qu'ils n'ont pas créées. Il est aussi permis aux serveurs de contrôler un accès de rafraîchissement à une entrée en demandant aux clients de se lier avant de produire une `RefreshRequest`. Cela aura des implications sur les performances et l'adaptabilité du serveur.

Il faut aussi faire attention lors de l'utilisation des informations obtenues des serveurs de répertoire qui ont été fournis par le client, car elles peuvent être maintenant périmées. Dans de nombreux réseaux, par exemple, les adresses IP sont allouées automatiquement lorsque un hôte se connecte au réseau, et elles peuvent être réallouées si cet hôte se déconnecte ensuite. Une adresse IP obtenue du répertoire peut ne plus être allouée à l'hôte qui a placé l'adresse dans le répertoire. Ce problème n'est pas spécifique de LDAP ou des répertoires dynamiques.

11. Remerciements

Les concepts inclus dans le présent document se fondent sur ceux discutés dans le groupe de travail ASID de l'IETF.

12. Adresses des auteurs

Yoram Yaacovi
Microsoft
One Microsoft way
Redmond, WA 98052
USA
téléphone : +1 206-936-9629
mél : yoramy@microsoft.com

Mark Wahl
Innosoft International, Inc.
8911 Capital of Texas Hwy #4140
Austin, TX 78759
USA
mél : M.Wahl@innosoft.com

Tony Genovese
Microsoft
One Microsoft way
Redmond, WA 98052
USA
téléphone : +1 206-703-0852
mél : tonyg@microsoft.com

13. Bibliographie

[RFC2251] M. Wahl, T. Howes et S. Kille, "[Protocole léger d'accès à un répertoire](#) (v3)", décembre 1997.

[RFC2252] M. Wahl, A. Coulbeck, T. Howes, S. Kille, "[Protocole léger d'accès à un répertoire](#) (v3) : Définitions de syntaxe d'attribut", décembre 1997. (*Obsolète, voir [RFC4510](#), [RFC4517](#), [RFC4523](#), [RFC4512](#)*) (*MàJ par [RFC3377](#)*) (*P.S.*)

[RFC2253] M. Wahl, S. Kille et T. Howes, "[Protocole léger d'accès à un répertoire](#) (LDAPv3) : Représentation de chaîne UTF-8 des noms distinctifs", décembre 1997.

14. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (1999). Tous droits réservés.

Ce document et les traductions de celui-ci peuvent être copiés et diffusés, et les travaux dérivés qui commentent ou expliquent autrement ou aident à sa mise en œuvre peuvent être préparés, copiés, publiés et distribués, partiellement ou en totalité, sans restriction d'aucune sorte, à condition que l'avis de copyright ci-dessus et ce paragraphe soit inclus sur toutes ces copies et œuvres dérivées. Toutefois, ce document lui-même ne peut être modifié en aucune façon, par exemple en supprimant le droit d'auteur ou les références à l'Internet Society ou d'autres organisations Internet, sauf si c'est nécessaire à l'élaboration des normes Internet, auquel cas les procédures pour les droits de reproduction définis dans les processus des normes pour l'Internet doivent être suivies, ou si nécessaire pour le traduire dans des langues autres que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Société Internet, ses successeurs ou ayants droit.

Ce document et les renseignements qu'il contient sont fournis "TELS QUELS" et l'INTERNET SOCIETY et l'INTERNET ENGINEERING TASK FORCE déclinent toute garantie, expresse ou implicite, y compris mais sans s'y limiter, toute garantie que l'utilisation de l'information ici présente n'enfreindra aucun droit ou aucune garantie implicite de commercialisation ou d'adaptation à un objet particulier.

Remerciement

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