

Groupe de travail Réseau
Request for Comments : 3403
RFC rendues obsolètes : 2915, 2168
Catégorie : En cours de normalisation

M. Mealling
VeriSign
octobre 2002
Traduction Claude Brière de L'Isle

Système de découverte dynamique de délégation (DDDS) Partie III : base de données du système des noms de domaine (DNS)

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 (2002). Tous droits réservés.

Résumé

Le présent document décrit une base de données du système de découverte dynamique de délégation (DDDS) qui utilise le système des noms de domaines (DNS) comme base de données répartie de règles. Les clés sont les noms de domaines et les règles sont codées en utilisant l'enregistrement de ressource (RR) Pointeur d'autorité de dénomination (NAPTR, *Naming Authority Pointer*).

Comme ce document rend obsolète la [RFC2915], il devient la spécification officielle de l'enregistrement de ressource NAPTR du DNS. Il fait aussi partie d'une série de documents qui sont complètement spécifiés dans "Système de découverte dynamique de délégation (DDDS) Partie I : DDDS complet" [RFC3401]. Il est important de noter qu'il est impossible de lire et comprendre un des documents de cette série sans lire les autres.

Table des Matières

1. Introduction.....	
2. Terminologie.....	
3. Spécification de base de données DDDS.....	
4. Format de RR NAPTR.....	
4.1 Format de paquet.....	
4.2 Traitement des informations supplémentaires.....	
4.3 Format de fichier maître.....	
5. Spécifications d'application.....	
6. Exemples.....	
6.1 Exemple d'URN.....	
6.2 Exemple de E164.....	
7. Avis aux administrateurs du DNS.....	
8. Notes.....	
9. Considérations relatives à l'IANA.....	
10. Considérations sur la sécurité.....	
Déclaration complète de droits de reproduction.....	

1. Introduction

Le système de découverte dynamique de délégation (DDDS, *Dynamic Delegation Discovery System*) est utilisé pour mettre en œuvre une liaison lâche des chaînes aux données, afin de prendre en charge les systèmes de délégation à configuration dynamique. Le DDDS fonctionne en transposant une chaîne unique en données mémorisées au sein d'une base de données du DDDS en appliquant de façon itérative des règles de transformation de chaîne jusqu'à atteindre une condition de fin.

Le présent document décrit la façon dont le système des noms de domaines (DNS, *Domain Name System*) est utilisé comme magasin de données pour les règles qui permettent à une application DDDS de fonctionner. Il ne spécifie aucun scénario particulier d'application ou d'usage. La série entière des documents est spécifiée dans "Système de découverte dynamique de délégation (DDDS) Partie I : DDDS complet" [RFC3401]. Il est important de noter qu'il est impossible de lire et comprendre un document de cette série sans lire les autres documents.

L'enregistrement de ressource (RR, *Resource Record*) du DNS Pointeur d'autorité de désignation (NAPTR) spécifié ici a été à l'origine produit par le groupe de travail URN comme un moyen de coder les ensembles de règles dans le DNS de façon que les sections déléguées des identifiants de ressource uniformes (URI, *Uniform Resource Identifier*) puissent être décomposées d'une façon telle qu'elles puissent être changées et redéléguées au fil du temps. Le résultat était un enregistrement de ressource qui comportait une expression régulière qui pouvait être utilisée par un programme client pour réécrire une chaîne dans un nom de domaine.

Les expressions régulières ont été choisies pour leur compacité afin d'exprimer des ratios permettant que de grandes quantités d'informations soient codées dans un paquet DNS assez petit.

Au fil du temps, ce processus a été généralisé pour d'autres applications et bases de données de règles. Le présent document définit une base de données de règles sans aucune application particulière car il peut y avoir plusieurs applications qui tirent parti de cette base de données de règles particulière.

2. Terminologie

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

Tous les autres termes, en particulier les termes en majuscules, sont tirés de la [RFC3403].

3. Spécification de base de données DDDS

Description générale :

Cette base de données utilise le système des noms de domaines (DNS) comme spécifié dans les [RFC1034] et [RFC1035].

Le jeu de caractères utilisé pour spécifier les diverses valeurs des enregistrements NAPTR est l'UTF-8 [RFC2279]. Il faut veiller à s'assurer que, dans le cas où l'entrée ou le résultat de l'expression de substitution contient des codets qui sortent de l'équivalence ASCII/Unicode en UTF-8, tout codet UTF-8 est interprété comme une série de codets au lieu d'une série d'octets. Cela pour s'assurer que les caractéristiques internationalisées des expressions régulières étendues POSIX sont capables de correspondre au codet prévu. Les expressions de substitution NE DOIVENT PAS être écrites là où elles dépendent d'un POSIX local spécifique car cela causerait la perte de la capacité d'application universelle des expressions de substitution.

Tous les enregistrements de ressource du DNS ont une durée de vie (TTL, *Time To Live*) associée. Lorsque le nombre de secondes indiqué par le TTL est passé depuis que l'enregistrement a été restitué, il n'est plus valide et une nouvelle interrogation doit être utilisée pour restituer les nouveaux enregistrements. Donc, comme mentionné dans l'algorithme DDDS, il peut y avoir le cas où une certaine règle arrive à expiration. Dans le cas où une application tente de revenir à des ensembles de règles restitués précédemment (soit dans le cas d'un mauvais chemin de délégation, soit d'une défaillance d'un réseau ou serveur) l'application DOIT s'assurer qu'aucun des enregistrements sur lesquels il s'appuie n'est arrivé à expiration. Dans le cas où même un seul enregistrement serait arrivé à expiration, l'application est obligée de recommencer au début de l'algorithme.

Format de clé :

Une clé est un nom de domaine du DNS dont la construction est valide.

Demande de recherche :

Afin de demander un ensemble de règles pour une certaine clé, le client produit une demande, conformément aux règles standard du DNS, pour un enregistrement de ressource NAPTR pour le nom de domaine en question.

Réponse de recherche :

La réponse à une demande pour une certaine clé (nom de domaine) sera une série d'enregistrements NAPTR. Le format d'un enregistrement de ressource NAPTR se trouve décrit à la Section 4.

Procédure d'insertion de règle :

Les règles sont insérées par l'ajout de nouveaux enregistrements dans la zone appropriée du DNS. Si une règle produit une clé qui existe dans une zone particulière, seule l'entité qui a le contrôle administratif de cette zone peut alors spécifier la règle associée à cette clé.

Évitement de collision :

Dans le cas où deux applications peuvent utiliser cette base de données (ce qui est en fait le cas avec ENUM et l'application de résolution d'URI, paragraphe 6.2) il y a des chances de collision entre des règles lorsque deux enregistrements NAPTR apparaissent dans le même domaine mais s'appliquent à plus d'une application. Il y a trois moyens d'éviter les collisions :

- * créer une nouvelle zone au sein du domaine en commun ne contenant que les enregistrements NAPTR qui sont appropriés pour l'application. Par exemple, tous les enregistrements de résolution d'URI existeraient sous "urires.example.com" et tous les enregistrements ENUM seraient sous "enum.example.com". Dans le cas où ce ne serait pas possible à cause du manque de contrôle sur la délégation vers l'amont, on utilise la seconde méthode.
- * écrire l'expression régulière de telle sorte qu'elle contienne assez de la chaîne unique d'application pour la distinguer sans ambiguïté de toute autre. Par exemple, l'application de résolution d'URI serait capable d'utiliser le nom de schéma sur le côté gauche pour ancrer la correspondance de l'expression régulière à ce schéma. Un enregistrement spécifique de ENUM dans cette même zone serait capable d'ancrer le côté gauche de la correspondance avec le caractère "+" qui est défini par ENUM comme étant au début de toute chaîne unique d'application. De cette façon, une chaîne unique d'application ne peut correspondre qu'à un seul enregistrement, et pas aux deux.
- * Si deux applications utilisent des valeurs de fanions ou service différentes, un enregistrement de l'autre application sera ignoré car il ne s'applique pas aux services/fanions en question.

4. Format de RR NAPTR

4.1 Format de paquet

Le format du paquet du RR NAPTR est donné ci-dessous. Le code de type DNS pour NAPTR est 35.

Le format de paquet de l'enregistrement NAPTR est celui qui suit :

```

                                1 1 1 1 1 1
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+-----+-----+-----+-----+-----+
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
+-----+-----+-----+-----+-----+-----+
/                                     /
/                                     /
/                                     /
/                                     /
/                                     /
/                                     /
/                                     /
+-----+-----+-----+-----+-----+-----+

```

<chaîne de caractère> et <nom de domaine> sont utilisés ici comme défini dans la [RFC1035].

Ordre

C'est un entier de 16 bits non signé qui spécifie l'ordre dans lequel les enregistrements NAPTR DOIVENT être traités pour représenter précisément la liste ordonnée des règles. L'ordre est du plus petit au plus grand. Si deux enregistrements ont la même valeur d'ordre, ils sont alors considérés comme étant la même règle et devraient être choisis sur la base de la combinaison des valeurs de préférence et de services offerts.

Préférence

Bien qu'il soit appelé "préférence" par déférence pour la terminologie du DNS, ce champ est équivalent à la valeur de priorité dans l'algorithme DDDS. C'est un entier non signé de 16 bits qui spécifie l'ordre dans lequel les enregistrements NAPTR qui ont des valeur d'ordre égales DEVRAIENT être traités, les valeurs faibles étant traitées avant les grandes. Ceci est similaire au champ de préférence dans un enregistrement MX, et est utilisé de façon que les administrateurs de domaine puissent diriger les clients vers des hôtes plus capables ou des protocoles plus légers. Un client PEUT chercher les enregistrements qui ont des valeurs de préférence plus élevée si ils ont de bonnes raisons de le faire, comme de ne pas prendre très bien en charge certain protocole ou service.

La différence importante entre Ordre et Préférence est que une fois qu'une correspondance est trouvée, le client NE DOIT PAS prendre en considération les enregistrements qui ont un Ordre différent mais ils PEUVENT traiter les enregistrements qui ont le même Ordre et des Préférences différentes. La seule exception est notée dans la seconde note importante dans la

spécification de l'algorithme DDDS concernant la permission aux clients d'utiliser une détermination de service plus complexe entre les étapes 3 et 4 de l'algorithme. La préférence est utilisée pour donner une meilleure qualité de service aux règles qui sont considérées comme les mêmes du point de vue d'une autorité mais pas du point de vue du simple équilibrage de charge.

Il est important de noter que le DNS contient plusieurs mécanismes d'équilibrage de charge et si entre des services par ailleurs égaux l'équilibrage de charge était nécessaire, des méthodes telles que les enregistrements SRV ou plusieurs enregistrements A devraient alors être utilisées pour réaliser l'équilibrage de charge.

Fanions

C'est une <chaîne de caractères> contenant des fanions pour contrôler les aspects de réécriture et d'interprétation des champs de l'enregistrement. Les fanions sont un seul caractère dans l'ensemble A-Z et 0-9. La casse des caractères alphabétiques n'est pas significative. Le champ peut être vide.

Il dépend de l'application qui spécifie de décider comment elle utilise la base de données pour définir les fanions dans ce champ. Elle doit définir ceux qui sont terminaux et ceux qui ne le sont pas.

Services

C'est une <chaîne de caractères> qui spécifie les paramètres de service applicables à ce chemin de délégation. Il appartient à la spécification d'application de spécifier les valeurs qu'on trouve dans ce champ.

REGEXP

C'est une <chaîne de caractères> contenant une expression de substitution qui est appliquée à la chaîne originale détenue par le client afin de construire le prochain nom de domaine à rechercher. Voir dans la spécification de l'algorithme DDDS la syntaxe de ce champ.

Comme établi dans l'algorithme DDDS, les expressions régulières NE DOIVENT PAS être utilisées de façon cumulative, c'est-à-dire qu'elles ne devraient être appliquées qu'à la chaîne originale détenue par le client, et jamais au nom de domaine produit par une précédente réécriture de NAPTR. Cette dernière est tentante dans certaines applications mais l'expérience a montré qu'une telle utilisation est extrêmement sensible aux fautes, très encline à l'erreur, et extrêmement difficile à corriger.

Remplacement

C'est un <nom de domaine> qui est le nom du prochain domaine à interroger selon les valeurs potentielles trouvées dans le champ Fanions. Ce champ est utilisé lorsque l'expression régulière est une opération de simple remplacement. Toute valeur dans ce champ DOIT être un nom de domaine pleinement qualifié. La compression de nom ne doit pas être utilisée pour ce champ.

Ce champ constitue avec le champ REGEXP l'expression de substitution dans l'algorithme DDDS. L'existence de ce champ est due simplement à une optimisation historique spécifique de la compression du DNS. Les champs sont aussi mutuellement exclusifs. Si un enregistrement est retourné avec des valeurs pour les deux champs, c'est considéré comme une erreur et DEVRAIT être ignoré ou signalé comme erreur.

4.2 Traitement des informations supplémentaires

Le traitement de la section supplémentaire exige des serveurs DNS mis à niveau, donc cela prendra de nombreuses années avant que les applications puissent s'attendre à voir des enregistrements pertinents dans la section d'informations supplémentaires.

4.2.1 Traitement de section supplémentaire par les serveurs du DNS

Les serveurs DNS PEUVENT ajouter des ensembles d'enregistrements de ressources (RRset) à la section d'informations supplémentaires qui sont pertinentes pour la réponse et ont la même authenticité que les données dans la section réponse. Généralement cela sera constitué d'enregistrements A et SRV mais les enregistrements exacts dépendent de l'application.

4.2.2 Traitement de section supplémentaire par les résolveurs/applications

Les applications PEUVENT inspecter la section d'informations supplémentaires à la recherche d'enregistrements pertinents mais les applications NE DOIVENT PAS exiger que des enregistrements de quelque type que ce soit se trouvent dans la section d'informations supplémentaire d'une réponse du DNS pour les besoins du fonctionnement des clients. Toutes les applications doivent être capables de traiter les réponses provenant des serveurs de noms qui ne remplissent jamais la partie Informations supplémentaire d'une réponse.

4.3 Format de fichier maître

Le format du fichier maître suit les règles standard de la [RFC1035]. Ordre et Préférence sont des entiers non signés de 16 bits dans la gamme de 0 à 65 535. Les champs Fanions, Services et Regexp sont tous des <chaîne de caractères> entre guillemets car le champ Regexp peut contenir de nombreuses barres obliques inverses et devrait donc être traité avec prudence. Voir à la Section 7 la façon d'entrer correctement et d'esquiver les expressions régulières.

5. Spécifications d'application

La présente base de données DDDS est utilisable par toute application qui fait usage de l'algorithme DDDS. En plus des éléments requis pour spécifier une application DDDS, une application qui souhaite utiliser cette base de données doit aussi définir les valeurs suivantes :

- o À quel domaine appartient la clé qui est produite par la première règle bien connue. Toute application doit s'assurer que ses règles n'entrent pas en collision avec celles d'une autre application qui utilise la même base de données. Par exemple, l'application "foo" pourrait avoir toutes ses premières clés bien connues dans la zone "foo.net".
- o Quelles sont les valeurs admises pour les champs Services et Protocole.
- o Le résultat attendu de la règle terminale de réécriture en plus de la façon dont les fanions sont en fait codés et utilisés.

6. Exemples

6.1 Exemple d'URN

L'enregistrement NAPTR a été à l'origine créé pour être utilisé avec le service de découverte de résolveur (RDS, *Resolver Discovery Service*) [RFC2216] de nom de ressource uniforme (URN, *Uniform Resource Name*). L'exemple qui suit précise comment un certain URN va utiliser l'enregistrement NAPTR pour trouver un service de résolveur qui puisse répondre aux questions sur l'URN. Voir dans la [RFC3402] la spécification "définitive" pour cette application.

Considérons un espace de nom d'URN fondé sur les identifiants de contenu MIME (tout ceci est purement fictif). L'URN pourrait ressembler à :

```
urn:cid:199606121851.1@bar.example.com
```

La première règle bien connue de cette application est d'extraire les caractères entre les premiers et les seconds deux-points. Pour cet URN, ce serait "cid". L'application spécifie aussi que, afin de construire une clé valide pour la base de données, la chaîne "urn.arpa" devrait être ajoutée au résultat de la première règle bien connue. Le résultat est "cid.urn.arpa". Ensuite, le client interroge le DNS sur les enregistrements NAPTR pour le nom de domaine "cid.urn.arpa". Le résultat est un seul enregistrement :

```
cid.urn.arpa.
;;          ordre   préf   fanions   service   regexp   remplacement
IN NAPTR  100      10    ""        ""        "!^urn:cid:.*@([\^\.]+\.)[*]$!2i"  .
```

Comme il n'y a qu'un seul enregistrement, le rangement des réponses ne pose pas de problème. Le champ "remplacement" est vide, de sorte que le schéma fourni dans le champ "regexp" est utilisé. On applique cette "regexp" à l'URN entier pour voir si il correspond, ce qu'il fait. La partie \2 de l'expression de substitution retourne la chaîne "example.com". Comme le champ Fanions est vide, la recherche n'est pas terminale et notre prochain sondage du DNS est sur d'autres enregistrements NAPTR où le nouveau domaine est "example.com".

Noter que la règle n'extrait pas le nom de domaine complet du CID, elle suppose plutôt que le CID vient d'un hôte et elle extrait son domaine. Bien que tous les hôtes, comme "bar", puissent avoir leur propre NAPTR, conserver ces enregistrements pour toutes les machines d'un site pourrait être un fardeau insupportable. Les caractères génériques ne sont pas appropriés ici car ils ne retournent les résultats que lorsque il n'y a pas de nom correspondant exactement déjà dans le système.

L'enregistrement retourné suite à l'interrogation sur "example.com" pourrait ressembler à :

```
example.com.
;;          ordre   préf   fanions   service   regexp   remplacement
IN NAPTR  100      50     "a"      "z3950+N2L+N2C"  ""      cidserver.example.com.
IN NAPTR  100      50     "a"      "rcds+N2C"       ""      cidserver.example.com.
IN NAPTR  100      50     "s"      "http+N2L+N2C+N2R" ""      www.example.com.
```

En continuant avec l'exemple, on note que les valeurs des champs `Ordre` et `Préférence` sont égaux dans tous les enregistrements, de sorte que le client est libre de prendre celui qu'il veut. L'application définit le fanion 'a' pour signifier une recherche terminale et que le résultat de la réécriture sera un nom de domaine pour lequel un enregistrement A devrait être demandé. Une fois que le client a fait cela, il a les informations suivantes : l'hôte, son adresse IP, le protocole, et les services disponibles via ce protocole. Ces bits d'information étant donnés, le client en a assez pour être capable de contacter ce serveur et lui poser ses questions sur l'URN.

On se rappelle que l'expression régulière a utilisé `\2` pour extraire un nom de domaine du CID, et `\.` pour faire correspondre les caractères littéraux "." qui séparent les composants du nom de domaine. Comme "\" est le caractère d'échappement, les occurrences littérales de barre oblique inverse doivent être remplacées par une autre barre oblique inverse. Pour le cas de l'enregistrement "cid.urn.arpa" ci-dessus, l'expression régulière entrée dans le fichier maître devrait être `"!^urn:cid: +@([\.\.]+\.)(.*)$!\2!i"`. Lorsque le code client reçoit en fait l'enregistrement, le schéma aura été converti en `"!^urn:cid: +@([\.\.]+\.)(.*)$!\2!i"`.

6.2 Exemple de E164

Le groupe de travail ENUM de l'IETF a spécifié un service qui permet aux numéros de téléphone d'être transposés en URI [RFC2916]. La chaîne unique d'application pour l'application ENUM est le numéro de téléphone E.164 dont les tirets ont été retirés. La première règle bien connue est de retirer tous les caractères du numéro de téléphone puis d'utiliser le numéro entier comme première clé. Par exemple, le numéro de téléphone "770-555-1212" représenté comme un numéro E.164 serait "+1- 770-555-1212". Converti en clé, ce serait "17705551212".

L'application ENUM n'utilise à présent que cette base de données. Elle spécifie que, afin de convertir la première clé en une forme valide pour cette base de données, des points soient insérés entre chaque chiffre, que la clé entière soit inversée et qu'ensuite "e164.arpa" soit ajouté à la fin. Le numéro de téléphone ci-dessus se lirait alors "2.1.2.1.5.5.5.0.7.7.1.e164.arpa.". Ce nom de domaine est ensuite utilisé pour restituer les règles de réécriture comme des enregistrements NAPTR.

Pour cet exemple le numéro de téléphone pourrait recevoir en retour les enregistrements NAPTR suivants :

```
$ORIGIN 2.1.2.1.5.5.5.0.7.7.1.e164.arpa.
IN NAPTR 100 10 "u" "sip+E2U" "!^.*$!sip:information@foo.se!i" .
IN NAPTR 102 10 "u" "smtp+E2U" "!^.*$!mailto:information@foo.se!i" .
```

Les applications ENUM [RFC2916] et Résolution d'URI [RFC3404] utilisent toutes deux le fanion "u". Ce fanion déclare que la règle est terminale et que le résultat est un URI qui contient les informations nécessaires pour contacter de service de téléphone. ENUM utilise aussi le même format pour ses paramètres de service. Ceux-ci déclarent que les protocoles disponibles utilisés pour accéder aux services de téléphone sont soit le protocole d'initiation de session, soit la messagerie SMTP.

7. Avis aux administrateurs du DNS

Attention aux expressions régulières. Elles sont non seulement difficiles à faire tenir correctement par elles-mêmes, mais il y a l'interaction précédemment mentionnée avec le DNS. Toute barre oblique inverse dans une "regex" doit être entrée deux fois dans un fichier de zone afin d'apparaître une seule fois dans une réponse d'interrogation. Plus sérieux, la nécessité de la double barre oblique inverse n'a probablement pas été vérifiée par toutes les mises en œuvre de serveur du DNS.

Pour atténuer les problèmes de fichier de zone, les administrateurs devraient encourager ceux qui écrivent les règles de réécriture à utiliser la caractéristique "délimiteur par défaut" de l'expression régulière. Dans la spécification DDDS, l'expression régulière commence par le caractère qui va être le délimiteur. Donc, si le premier caractère de l'expression régulière est, par exemple, un point d'exclamation (!) l'expression régulière peut normalement être écrite avec moins de barres obliques inverses.

8. Notes

Un client DOIT traiter plusieurs enregistrements NAPTR dans l'ordre spécifié par le champ "ordre", il NE DOIT PAS simplement utiliser le premier enregistrement qui fournit une combinaison de paramètres de service connus.

Lorsque plusieurs RR ont le même "ordre" et que tous les autres critères sont égaux, le client devrait utiliser la valeur du champ Préférence pour choisir le prochain NAPTR à considérer. Cependant, parce que ce sera souvent le cas que les protocoles ou services préférés existent, les clients peuvent utiliser ce critère supplémentaire pour trier les enregistrements.

Si la recherche échoue après une réécriture, les clients sont vivement encouragés à faire rapport de l'échec, plutôt que de revenir pour poursuivre sur d'autres chemins de réécriture.

9. Considérations relatives à l'IANA

Les valeurs des champs Services et Fanions seront déterminées par l'application qui fait usage de cette base de données DDDS. Ces valeurs peuvent exiger un mécanisme d'enregistrement et donc nécessiter des ressources de l'IANA. La présente spécification ne le fait pas par elle-même.

10. Considérations sur la sécurité

L'enregistrement NAPTR, comme tout autre enregistrement du DNS, peut être signé et validé conformément aux procédures spécifiées dans DNSSEC.

La présente base de données fabrique des identifiants à partir d'autres espaces de noms qui sont sujets aux mêmes attaques que les noms de domaine normaux. Comme ce problème n'a pas été facile à résoudre auparavant, cela peut être considéré ou non comme un problème.

Les expressions régulières devraient subir une vérification de bonne santé, et non passées aveuglément à quelque chose comme PERL car un code arbitraire peut être inclus et traité ultérieurement.

Références

- [IEEE] "IEEE Standard for Information Technology - Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities (Vol. 1)", IEEE Std 1003.2-1992, janvier 1993.
- [RFC1034] P. Mockapetris, "Noms de domaines - [Concepts et facilités](#)", STD 13, novembre 1987.
- [RFC1035] P. Mockapetris, "Noms de domaines - [Mise en œuvre](#) et spécification", STD 13, novembre 1987.
- [RFC2119] S. Bradner, "[Mots clés](#) à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2141] R. Moats, "Syntaxe des URN", mai 1997.
- [RFC2168] R. Daniel, M. Mealling, "Résolution des identifiants de ressource uniformes avec le système des noms de domaines", juin 1997. (*Obsolète, voir [RFC3401](#), [RFC3402](#), [RFC3403](#), [RFC3404](#) (MàJ par [RFC2915](#)) (Exp.)*)
- [RFC2169] R. Daniel, "Convention triviale pour l'utilisation de HTTP dans la résolution d'URN", juin 1997. (*Exp.*)
- [RFC2234] D. Crocker et P. Overell, "[BNF augmenté](#) pour les spécifications de syntaxe : ABNF", novembre 1997. (*Obsolète, voir [RFC5234](#)*)
- [RFC2276] K. Sollins, "Principes d'architecture de la résolution de nom de ressource uniforme", janvier 1998. (*MàJ par [RFC3401](#) (Information)*)
- [RFC2279] F. Yergeau, "UTF-8, un [format de transformation](#) de la norme ISO 10646", janvier 1998. (*Obsolète, voir [RFC3629](#) (D.S.)*)
- [RFC2396] T. Berners-Lee, R. Fielding et L. Masinter, "[Identifiants de ressource uniformes](#) (URI) : Syntaxe générique", août 1998. (*Obsolète, voir [RFC3986](#)*)
- [RFC2782] A. Gulbrandsen, P. Vixie et L. Esibov, "RR DNS pour la [spécification de la localisation](#) des services (DNS SRV)", février 2000.
- [RFC2916] P. Faltstrom, "Numéros E.164 et DNS", septembre 2000. (*Obsolète, voir [RFC3761](#) (P.S.)*)
- [RFC3401] M. Mealling, "Système de découverte dynamique de délégation (DDDS) Partie I : [DDDS complet](#)", octobre 2002. (*Info.*)
- [RFC3402] M. Mealling, "Système de découverte dynamique de délégation (DDDS) Partie II : [l'algorithme](#)", octobre 2002.
- [RFC3403] M. Mealling, "Système de découverte dynamique de délégation (DDDS) Partie III : [base de données](#) du système de noms de domaines (DNS)", octobre 2002. (*P.S.*)
- [RFC3404] M. Mealling, "Système de découverte dynamique de délégation (DDDS) Partie IV : [Identifiants de ressource uniformes](#) (URI)", octobre 2002. (*P.S.*)

[RFC3405] M. Mealling, "Système de découverte dynamique de délégation (DDDS) Partie V : [Procédures d'allocation URI.ARPA](#)", octobre 2002. ([BCP0065](#))

Adresse de l'auteur

Michael Mealling
VeriSign
21345 Ridgetop Circle
Sterling, VA 20166
USA
mél : michael@neonym.net
URI : <http://www.verisignlabs.com>

Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2002). 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 droits de reproduction 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 de normes pour 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 ou 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'édition des RFC est actuellement fourni par la Internet Society.