

Groupe de travail Réseau
Request for Comments : 2609
 RFCmise à jour : 2165
 Catégorie : En cours de normalisation
 Traduction Claude Brière de L'Isle

E. Guttman
 C. Perkins
 J. Kempf
 Sun Microsystems
 juin 1999

Schéma service: et gabarits de service

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 Internet Society (1999). Tous droits réservés.

Résumé

Le nom du schéma d'URL "service:" est utilisé pour définir les URL (appelés "URL service:" dans le présent document) qui sont principalement destinés à être utilisés par le protocole de localisation de service afin de distribuer les informations d'accès aux services. Ces schémas fournissent un cadre extensible pour les logiciels réseau fondés sur le client pour obtenir les informations de configuration requises pour faire usage des services réseau. Lors de l'enregistrement d'un URL service:, l'URL est accompagné par un ensemble d'attributs bien définis qui définissent le service. Ces attributs portent les informations de configuration au logiciel client, ou les caractéristiques de service significatives pour les utilisateurs finaux.

Le présent document décrit une procédure formelle pour définir et normaliser de nouveaux types de service et des attributs à utiliser avec le schéma "service:". Les descriptions formelles des types de service et des attributs sont des gabarits qui sont compréhensibles pas l'homme et la machine. Ils DEVRAIENT être utilisés par les outils administratifs pour analyser les informations d'enregistrement de service et par les applications client pour fournir des traductions localisées des chaînes d'attributs de service.

Table des Matières

1. Introduction.....	2
1.1 Terminologie.....	2
1.2 Protocole de localisation de service.....	3
2. Syntaxe et sémantique de l'URL Service.....	3
2.1 Syntaxe de l'URL Service.....	3
2.2 Sémantique de l'URL Service.....	5
2.3 Utilisation des URL service:.....	5
2.4 Spécification de la syntaxe d'URL spécifique du type de service.....	6
2.5 Traitement des types de service abstraits.....	6
3. Syntaxe et sémantique des spécifications de type de service.....	7
3.1 Syntaxe des gabarits de type de service.....	7
3.2 Sémantique des gabarits de type de service.....	9
4. Processus pour la normalisation de nouveaux types de services.....	12
5. Considérations relatives à l'IANA.....	13
6. Considérations d'internationalisation.....	13
6.1 Identification de langage et traduction.....	14
7. Considérations pour la sécurité.....	14
Appendice A. Exemples de gabarit de service.....	14
A.1 FOO.....	15
A.2 Type de service abstrait : Net-Transducer.....	16
A.3 Type de service concret : Net-Transducer:Thermometer.....	16
A.4 URL service: et SLP.....	17
Appendice B. Remerciements.....	17
Appendice C. Références.....	18
Appendice D. Adresse des auteurs.....	18
Appendice E. Déclaration complète de droits de reproduction.....	19

1. Introduction

Le présent document décrit un schéma d'URL, appelé URL service:, qui définit des informations d'accès réseau pour les services réseau qui utilisent une notation formelle. Il décrit de plus comment définir un ensemble d'attributs à associer à un URL service:. Ces attributs vont permettre aux programmes d'utilisateurs finaux de choisir entre les services réseau de même type qui ont des capacités différentes. Les attributs sont définis dans un document modèle qui est lisible par les gens et les machines.

Un client utilise des attributs pour choisir un service particulier. Le choix du service se fait en obtenant l'URL service: qui offre la bonne configuration pour le client. Les gabarits de type de service définissent la syntaxe des URL service: pour un type de service particulier, ainsi que les attributs qui accompagnent un URL service: dans un enregistrement de service.

Les gabarits sont utilisés pour les objets distincts suivants :

1. Normalisation : Le gabarit est revu avant sa normalisation. Une fois normalisé, toutes les versions du gabarit sont archivées par l'IANA.
2. Enregistrement de service : Les serveurs qui utilisent le protocole de localisation de service [10] s'enregistrent eux-mêmes et leurs attributs. Ils utilisent les gabarits pour générer les enregistrements de service. En s'enregistrant, le service doit utiliser les valeurs spécifiées pour ses attributs.
3. Présentation client des informations de service : Les applications clients peuvent afficher les informations de service. Le gabarit fournit les informations de type et un texte explicatif qui peut être utile pour produire les interfaces d'utilisateur.
4. Internationalisation : Les entités qui ont accès au gabarit pour un certain type de service dans deux langues différentes peuvent traduire d'une de ces langues à l'autre. Un service peut s'enregistrer dans plus d'une langue en utilisant les gabarits, bien qu'il ait été configuré par un opérateur qui a enregistré les attributs du service dans une seule langue.

Tout le codage de la grammaire suit le BNF augmenté (ABNF) [8] pour la spécification de la syntaxe.

1.1 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 [6].

La terminologie suivante est utilisée pour décrire les URL service:.

schéma de service

Schéma d'URL dont le nom commence par la chaîne "service:" et est suivi par le nom du type de service, construit conformément aux règles du présent document.

URL service:

URL construit conformément à la définition du schéma de service. Il fournit normalement au moins ce qui suit : le nom d'un protocole d'accès, et une adresse localisant ce service. L'URL service: peut inclure des informations de chemin d'URL spécifiques du type de service, ainsi que des informations d'attribut codées conformément à la grammaire de l'URL. L'URL service: est utilisé par le protocole de localisation de service pour enregistrer et découvrir la localisation des services. Il peut être utilisé aussi par d'autres protocoles et dans d'autres documents.

type de service

Nom qui identifie la sémantique par laquelle le reste de l'URL service: doit être compris. Il peut noter un protocole réseau particulier, ou un service abstrait associé à divers protocoles. Si le type de service note un protocole particulier, le nom du type de service DEVRAIT alors recevoir le nom d'un accès particulièrement bien connu [2] par convention ou être le nom des numéros alloués pour le service [1].

type de service abstrait

Nom de type de service qui est associé à divers protocoles différents. Un exemple est donné à l'Annexe A. La Section 2 expose diverses façons de traiter les types abstraits.

enregistrement de service

Un URL service: et facultativement un ensemble d'attributs composent un enregistrement de service. Cet enregistrement

est fait par ou au nom d'un certain service. La syntaxe et les attributs d'URL doivent se conformer au gabarit de service pour le service enregistré.

gabarit de service

Description formelle des attributs de service et du schéma de service associés à un type de service particulier.

1.2 Protocole de localisation de service

Le protocole de localisation de service version 2 [10] permet d'enregistrer et découvrir les URL service:, bien qu'ils puissent aussi être utilisés dans d'autres contextes.

Les applications clients découvrent les enregistrements de service en produisant des interrogations sur les services d'un type particulier, spécifiant les attributs des URL service: à retourner. Les clients retrouvent les attributs d'un service particulier en fournissant son URL service:. Les attributs pour tous les enregistrements de service d'un type particulier peuvent aussi être retrouvés.

Les services peuvent s'enregistrer eux-mêmes, ou les enregistrements peuvent être faits en leur nom. Ces enregistrements contiennent un URL service:, et éventuellement des attributs et des signatures numériques.

1.2.1 Compatibilité avec SLPv1

Le présent document adopte les conventions de codage de SLPv2.

La compatibilité avec SLPv1 [15] est possible, si les conventions suivantes sont observées :

1. Les types de service abstraits ne doivent pas être utilisés. SLPv2 spécifie l'utilisation des URL Service avec des types de service abstraits. SLPv1 ne les prend pas en charge. Donc, un gabarit de service qui doit servir à la fois SLPv1 et SLPv2 ne doit pas utiliser des types de service abstraits.
2. La syntaxe pour représenter des valeurs opaques dans le présent document est cohérente avec SLPv2. La syntaxe doit être convertie pour être utilisée avec SLPv1. Au lieu d'une séquence de "\FF" puis "\" HEXDIG HEXDIG pour chaque octet dans la valeur opaque, SLPv1 utilise la notation radix-64.
3. Les caractères d'échappement sont significativement différents dans SLPv1 et SLPv2. Au lieu d'utiliser la notation d'échappement d'octet pour les caractères d'échappement, SLPv1 utilise la convention HTML '&' '#' 1*CHIFFRE `;'.

2. Syntaxe et sémantique de l'URL Service

La présente section décrit la syntaxe et la sémantique des URL service:.

2.1 Syntaxe de l'URL Service

La syntaxe de l'URL service: DOIT se conformer à un 'URI absolu' comme défini par [5]. La syntaxe d'un URL service: diffère assez d'un 'URI générique' pour qu'il soit préférable de le traiter comme un URI opaque sauf si un analyseur spécifique est disponible pour l'URL service:.

Tous les URL service: ont la même syntaxe jusqu'à la 'partie-url' (*url-part*). La syntaxe d'un URL service: dépend du type de service qui suit le schéma de service. Tous les URL service: ont une portion de point d'accès au service qui indique l'adresse du service auquel il faut accéder.

La syntaxe du champ <sap> dépend de la définition du type de service. Le champ <sap> est le point d'accès au service, et décrit comment accéder au service. De plus, bien que les caractères aussi bien majuscules que minuscules soient reconnus dans le champ <type-de-service> pour des raisons pratiques, le nom est replié en minuscules. Les types de service ne sont donc pas distinguables sur la base de la casse, de sorte que, par exemple, "http" et "HTTP" désignent le même type de service. Ceci est cohérent avec la pratique générale des URL, comme esquissé dans [5].

L'ABNF pour un URL service: est :

```
URL service: = "service:" service-type ":" sap
service-type = abstract-type ":" url-scheme / concrete-type
```

```

abstract-type = type-name [ "." naming-auth ]
concrete-type = protocol [ "." naming-auth ]
type-name     = resname
naming-auth   = resname
url-scheme    = resname
; Nom d'un schéma d'URL reconnu, normalisé soit par la pratique courante, soit par l'approbation d'un organisme de
normalisation.
resname       = ALPHA [ 1*(ALPHA / DIGIT / "+" / "-") ]
sap           = site [url-part]
site          = ipsite / atsite / ipxsite
ipxsite       = "/" [ [ user "@" ] hostport ]
hostport      = host [ ":" port ]
host          = hostname / hostnumber
hostname      = *( domainlabel "." ) toplabel
alphanum      = ALPHA / DIGIT
domainlabel   = alphanum / alphanum *[alphanum / "-"] alphanum
toplabel      = ALPHA / ALPHA *[ alphanum / "- " ] alphanum
hostnumber    = ipv4-number
ipv4-number   = 1*3DIGIT 3("." 1*3DIGIT)
port          = 1*DIGIT
; Un numéro d'accès doit être inclus si le champ Protocole n'a pas un numéro d'accès alloué par l'IANA.
user          = *[ uchar / ";" / "+" / "&" / "=" ]
ipxsite       = "/ipx/" ipx-net ":" ipx-node ":" ipx-socket
ipx-net       = 8 HEXDIGIT
ipx-node      = 12 HEXDIGIT
ipx-socket    = 4 HEXDIGIT
atsite        = "/at/" at-object ":" at-type "" at-zone
at-object     = 1*31apple-char
at-type       = 1*31apple-char
at-zone       = 1*31apple-char
apple-char    = ALPHA / DIGIT / safe / escaped
; Les valeurs AppleAscii [7] qui ne sont pas dans la gamme interdite doivent être échappées.
; Note : Les valeurs échappées NE correspondent PAS ici aux valeurs UTF-8 : ce sont des octets AppleAscii.
url-part      = [ url-path ] [ attr-list ]
url-path      = 1 * ( "/" *xchar )
; Chaque type de service doit définir sa propre syntaxe cohérente avec [5].
attr-list     = 1 * ( ";" attr-assign )
attr-assign   = attr-id / attr-id "=" attr-value
safe          = "$" / "-" / "_" / "." / "~"
extra         = "!" / "*" / "" / "(" / ")" / "," / "+"
uchar         = unreserved / escaped xchar      = unreserved / reserved / escaped
escaped       = 1*(\' HEXDIG HEXDIG)
reserved     = ";" / "/" / "?" / ":" / "@" / "&" / "=" / "+"
unreserved   = ALPHA / DIGIT / safe / extra

```

Les adresses IPX [14] se composent d'un numéro de réseau, de nœud et de prise. Le numéro de réseau IPX est un nombre de quatre octets, dans l'ordre du réseau et exprimé en hexadécimal, qui signifie un sous-réseau IPX. L'élément nœud représente une carte d'interface réseau. C'est un numéro de six octets, exprimé en hexadécimal, qui est habituellement le même que l'identifiant de nœud de la carte d'interface. L'élément 'prise' représente un point d'accès spécifique du service, pour un réseau et nœud IPX donnés. Les prises IPX sont analogues aux accès TCP/IP, et ne doivent pas être confondues avec les prises Berkeley.

Les adresses AppleTalk [9] se composent d'un objet, d'un type et d'une zone. L'objet est une chaîne lisible par l'homme. Le type est un identifiant, non destiné à la lecture humaine. La zone se réfère au nom de zone AppleTalk, qui est aussi lisible par l'homme. Les caractères qui composent ces noms sont tirés du jeu de caractères AppleAscii [7]. Donc, ils ne sont pas équivalents aux échappements de caractères ASCII ou UTF-8. Les caractères "=" et "*" sont réservés et ne peuvent pas être inclus dans les noms, même sous forme binaire.

En dehors de la grammaire AppleTalk, certains caractères doivent être esquivés avant utilisation. Pour l'échappement de tout caractère, on fait précéder les deux chiffres qui indiquent sa valeur ASCII par '%'.

2.2 Sémantique de l'URL Service

Les informations spécifiques du schéma de service qui suivent l'identifiant du schéma d'URL "service:" donnent les informations nécessaires pour accéder au service. Comme décrit au paragraphe précédent, la forme d'un URL service: est la suivante :

URL service: = "service:" type-de-service ":" site chemin-d'url

où <site> a une des formes suivantes pourrait se référer à un <ipsite>, <ipxsite> ou <atsite> si l'URL service localise un point d'accès au service respectivement IP, IPX ou AppleTalk.

Comme exposé à la Section 1, le <type-de-service> peut être un nom de protocole concret, ou un nom de type abstrait.

Le champ <ipsite> est normalement soit un nom de domaine (DNS) soit une adresse de protocole de réseau IP pour le service, et éventuellement un numéro d'accès. Noter que l'utilisation des noms d'hôtes du DNS est préférée pour faciliter les dénumérotages. Le champ <site> peut être nul si les autres informations dans l'URL service ou les attributs de service sont suffisants pour utiliser le service.

Le champ <sap> permet que plus d'informations soient fournies (au moyen de <chemin-d'url> et de <attr-list>) qui peuvent localiser de façon univoque le service ou la ressource si le <site> n'est pas suffisant pour cela. Pour les adresses IP, le champ <site> commence par "//". Les autres familles d'adresse acceptées sont IPX [14] et AppleTalk [9].

Un champ <attr-list> apparaît à la fin du champ <url-part>, mais il n'est jamais exigé qu'il existe dans un enregistrement de localisation de service.

Le champ <attr-list> se compose d'une liste de points-virgules (";") séparés par des allocations d'attribut de la forme :

attr-id "=" attr-value

ou pour les attributs mots-clés : attr-id

Les attributs font partie des URL service: lorsque les attributs sont exigés pour accéder à un service particulier. Par exemple, un service ACAP [13] pourrait exiger que le client s'authentifie auprès de lui au moyen de Kerberos. Inclure un attribut dans l'enregistrement de service permet au client ACAP d'utiliser le mécanisme d'authentification SASL [11] correct. L'enregistrement du serveur ACAP pourrait ressembler à :

service:acap://some.where.net;authentication=KERBEROSV4

Noter qu'il peut y avoir d'autres attributs d'un serveur ACAP qu'il n'est pas approprié d'inclure dans l'URL. Par exemple, la liste des utilisateurs qui ont accès au serveur est utile pour choisir un serveur ACAP, mais n'est pas requise pour qu'un client utilise le service enregistré.

Les attributs associés à l'URL service: ne sont normalement pas inclus dans l'URL service:. Ils sont mémorisés et restitués en utilisant d'autres mécanismes. L'URL service: est identifié de façon univoque avec un agent de service ou ressource particulier, et est utilisé lors de l'enregistrement ou de la demande des informations d'attribut. Le protocole de localisation de service spécifie comment de telles informations sont enregistrées par les services réseau et obtenus par le logiciel client.

2.3 Utilisation des URL service:

L'URL service: est destiné à permettre à des systèmes client/serveur et d'homologue à homologue arbitraires d'utiliser un mécanisme dynamique de découverte de points d'accès de service normalisés.

Il est prévu que les URL service: soient choisis conformément aux aptitudes des attributs associés. Une application client peut obtenir les URL de plusieurs services du même type et distinguer celui qui est préférable parmi eux au moyen de leurs attributs. Le client utilise l'URL service: pour communiquer directement avec un service.

Les attributs sont spécifiés avec une syntaxe de gabarit de service formelle qui est décrite à la Section 3. Si un enregistrement d'URL service: comporte des attributs, l'agent d'enregistrement DEVRAIT aussi garder trace des attributs qui caractérisent le service.

Les enregistrements peuvent être comparés à la spécification formelle d'attribut définie dans le gabarit par le client ou l'agent qui représente le client. L'enregistrement de service est normalement fait en utilisant le protocole de localisation de service (SLP) [10]. SLP donne un mécanisme pour que les URL service: soient obtenus de façon dynamique,

conformément aux attributs du service.

Il est aussi possible d'obtenir des URL service: à partir de documents et en utilisant d'autres protocoles. Dans ce cas, l'URL ne peut pas être accompagné par les attributs de service. Le contexte dans lequel apparaît l'URL devraient rendre clair, si possible, quand il est approprié d'utiliser le service. Par exemple, dans un message électronique, un service pourrait être recommandé pour être utilisé lorsque l'utilisateur est dans une succursale. Ou, un document HTML pourrait inclure un URL service: comme pointeur sur un service, décrivant en texte ce que fait le service et qui est autorisé à l'utiliser.

2.4 Spécification de la syntaxe d'URL spécifique du type de service

Lorsque un type de service est spécifié, la spécification inclut la définition de la syntaxe pour tous les URL qui sont enregistrés par les services de ce type particulier. Par exemple, le type de service "lpr" peut être défini avec les URL service: sous la forme suivante :

```
service:printer:lpr://<adresse d'imprimante>/<nom de file d'attente>
```

La section de l'URL après l'adresse de l'imprimante : "/" <nom de file d'attente> est spécifique du type de service lpr et correspond au champ <chemin-d'url> de la syntaxe générale d'URL service:. Cette partie est spécifiée lorsque le type de service lpr est spécifié.

2.5 Traitement des types de service abstraits

Un type de service abstrait est un type de service qui peut être mis en œuvre chez divers agents de service différents.

Afin d'enregistrer un URL service: pour un type de service abstrait on utilise la règle de grammaire 'abstract-type' décrite au paragraphe 3.1. Il va en résulter un URL qui comporte assez d'informations pour utiliser le service, à savoir, les informations de protocole, d'adresse et de chemin. À l'opposé cependant, dans les URL service: 'concrets', le type de service n'est pas suffisant pour déterminer l'accès au service. Un type de service abstrait note plutôt une classe de type de services. Les paragraphes qui suivent exposent ce point plus en détail.

Les gabarits de service concrets héritent de tous les attributs définis dans le gabarit de service abstrait d'où a été déduit le gabarit de service concret. L'attribut défini dans le gabarit de service abstrait NE DOIT PAS être défini aussi dans le gabarit de service concret. Cela simplifie l'interprétation des gabarits.

Par exemple, si un gabarit de service abstrait pour le type de service 'Abstrait' définit un attribut FOO, tous les gabarits de service concrets dérivés du gabarit de service abstrait, tels que 'Abstrait:Concret' vont implicitement inclure la définition de l'attribut FOO. Ce gabarit dérivé NE DOIT PAS redéfinir FOO, conformément à la règle ci-dessus.

Un autre exemple est décrit à l'Appendice A.

2.5.1 Annonce des types de service abstraits

Certains services peuvent faire usage de plusieurs protocoles qui sont d'utilisation courante et sont des services distincts de leur propre chef. Dans ces cas, un type de service abstrait est approprié. Ce qui est essentiel est que toutes les informations requises pour le service soient clairement définies.

Par exemple, supposons un service réseau développé pour un chargement dynamique de pilotes d'appareils. Le client exige les trois éléments d'informations suivants avant qu'il puisse être chargé avec succès et instancier le pilote :

1. Le protocole utilisé pour charger le code du pilote, par exemple, "ftp", "http" ou "tftp"
2. Un nom de chemin identifiant où est situé le code du pilote, par exemple "/systemhost/drivers/diskdrivers.drv",
3. Le nom du pilote, par exemple, "scsi".

Il est tentant de former les deux premiers éléments dans un URL et d'incorporer cela dans un URL service:. Comme exemple de ce qui devrait être évité, service:ftp://x3.bean.org/drivers/diskdrivers.drv;driver=scsi est un URL service: qui semble indiquer où obtenir le pilote.

Un type de service abstrait DEVRAIT plutôt être utilisé. Le type de service n'est pas "ftp", comme l'exemple l'indique. C'est plutôt "device-drivers". L'URL service: qui devrait être utilisé, en conformité aux règles posées dans [6], est le suivant :

```
service:device-drivers:ftp://x3.bean.org/drivers/diskdrivers.drv;
driver=scsi;platform=sys3.2-rs3000
```

D'autres URL pour le même service utilisant d'autres protocoles sont aussi pris en charge, comme dans :

```
service:device-drivers:tftp://x2.bean.org/vol3/disk/drivers.driv;  
driver=scsi;platform=sys3.2-rs3000
```

```
service:device-drivers:http://www.bean.org/drivers/drivpak.driv;  
driver=scsi;platform=sys3.2-rs3000
```

En utilisant SLP, une recherche du type de service "device-drivers" peut retourner les trois URL énumérés ci-dessus. Le client choisit le protocole d'accès le plus approprié pour la ressource désirée.

L'exigence fondamentale est que le type de service abstrait DOIT être bien spécifié. Cette exigence est imposée de telle sorte que le code programme ou les utilisateurs humains aient assez d'informations pour accéder au service. Dans tous les cas, un type abstrait bien spécifié va inclure soit un protocole d'accès et une adresse réseau où le service est disponible, soit un URL incorporé pour un schéma d'URL normalisé qui décrit comment accéder au service. Dans l'exemple ci-dessus, il y a trois autres exigences supplémentaires : un chemin d'URL est inclus pour les protocoles d'accès qui indique le document à télécharger, et deux attributs sont inclus pour caractériser le pilote.

3. Syntaxe et sémantique des spécifications de type de service

Les spécifications de type de service sont documentées dans une syntaxe formelle qui définit les propriétés importantes pour l'enregistrement de service. Ces propriétés sont :

1. des informations générales sur la spécification de type de service lui-même,
2. la syntaxe de la partie spécifique du type de service de l'URL service,
3. la définition des attributs associés à un service.

Le document de spécification du type de service est le gabarit de type de service.

Les paragraphes qui suivent décrivent la syntaxe et la sémantique des gabarits de type de service.

3.1 Syntaxe des gabarits de type de service

Les documents de gabarit de service sont codés sous une forme simple. Ils peuvent être traduits dans toute langue ou jeu de caractères, mais le gabarit utilisé pour la normalisation DOIT être codé dans la sous-gamme 0x00 à 0x7F de l'UTF-8 [16] (qui correspond au codage de caractères ASCII) et être en anglais.

Un document de gabarit commence par un bloc de texte allouant des valeurs à cinq éléments d'identification du document. Les cinq éléments d'identification peuvent apparaître dans n'importe quel ordre au sein du bloc, mais par convention, l'élément "type-de-gabarit", qui alloue le nom du type de service se produit au tout début du document afin de fournir le contexte pour le reste du document. L'élément de définition d'attribut est produit après les éléments d'identification du document.

Tous les éléments se terminent pas une ligne blanche. Les caractères réservés sont ";", "%", "=", ",", "#", LF, et CR. Les caractères réservés DOIVENT être esquivés. La séquence d'échappement est la même que décrit en [5].

Le gabarit de service est codé dans le jeu de caractères UTF-8, mais soumis au titre d'un travail en cours, qui est codé en caractères ASCII. Tous les caractères qui sont en dehors de la gamme ASCII DOIVENT être esquivés en utilisant la syntaxe '\' HEXDIG HEXDIG. Par exemple, la lettre e accent aigu sera représentée par "\c3a9". Malheureusement, cela va nuire à la lisibilité du gabarit de service dans la soumission du gabarit de service. Heureusement que certains outils dans le domaine public vont émerger pour traduire les caractères UTF-8 échappés en caractères lisibles par l'homme.

Les valeurs dans les listes de valeurs sont séparées par des virgules. Une liste de valeurs est terminée par une nouvelle ligne non précédée d'une virgule. Si la nouvelle ligne est précédée d'une virgule, la liste de valeurs est interprétée comme continuant sur la ligne suivante.

Les identifiants d'attribut, les noms de type d'attribut, et les fanions sont tous insensibles à la casse. Pour faciliter la présentation, des caractères majuscules et minuscules peuvent être utilisés pour les représenter dans le document de gabarit. Les nouvelles lignes sont significatives dans la grammaire. Elles délimitent un élément d'un autre, ainsi qu'elles séparent les parties internes des éléments.

newline = CR / (CR LF)

3.2 Sémantique des gabarits de type de service

Le gabarit de type de service définit la syntaxe des attributs du service et de l'URL service: pour un type de service particulier. La définition d'attribut inclut le type d'attribut, les valeurs par défaut, les valeurs admises et les autres informations.

Noter que 'template-type', 'template-version', 'template-description' et 'template-url-syntax' ont été définis comme attributs. Ces attributs PEUVENT accompagner tout enregistrement de service en utilisant SLPv2.

3.2.1 Définition d'un gabarit de service

Il y a quatre éléments qui sont inclus dans le gabarit de service. La sémantique de chaque élément est résumée ci-dessous.

- template-type (*type de gabarit*)

C'est le nom du schéma du service. Le nom de schéma consiste en le nom du type de service et un nom facultatif d'autorité de désignation, séparé du nom du type de service par un point. Voir au paragraphe 3.2.2 les conventions qui gouvernent les noms de type de service.

- template-version (*version de gabarit*)

C'est le numéro de version de la spécification du type de service.

- template-description (*description de gabarit*)

C'est une description du service qui convient pour être incluse dans un texte lisible par l'homme.

- template-url-syntax (*syntaxe de gabarit d'URL*)

La syntaxe de la partie d'URL spécifique du type de service de l'URL service:.

- définitions d'attribut

Une collection de zéro, une ou plusieurs définitions des attributs associés au service dans les enregistrements de service.

Chacun des paragraphes suivants traite d'un de ces éléments.

3.2.2 Type de service

Le schéma de service comporte le nom du type de service et le nom facultatif d'une autorité de désignation séparé du nom du type de service par un point. Le schéma de service est une chaîne qui est ajoutée à l'identifiant du schéma d'URL 'service:', et est la valeur de l'élément "template-type" dans le document de gabarit. Si le nom de l'autorité de désignation est absent, il est supposé être IANA.

3.2.3 Numéro de version

Le numéro de version du gabarit de type de service est la valeur de l'élément "template-version". Un projet de proposition commence à 0.0, et le numéro mineur est augmenté de un par révision. Un gabarit normalisé commence à 1.0. Les ajouts d'attributs facultatifs ajoutent un au numéro mineur, et les ajouts d'attributs exigés, les changements de définition, ou les suppressions d'attributs ajoutent un au numéro majeur. L'intention est qu'un vieux gabarit de service définisse toujours précisément, même si c'est incomplet, les attributs d'un enregistrement de service si le gabarit ne diffère de l'enregistrement que dans sa version mineure. Voir à la Section 4 plus de détails sur la façon dont on utilise l'attribut template-version.

3.2.4 Description

La description est un bloc de texte lisible par l'homme dans la langue du gabarit et c'est la valeur de l'élément "template-description". Elle devrait être suffisante pour identifier le service aux lecteurs humains et fournir une brève description informative de ce que fait le service.

Si le type de service correspond à un protocole particulier, la spécification du protocole doit être citée. Il n'est pas nécessaire que le protocole soit normalisé. Le gabarit peut se référer à une spécification propriétaire, et renvoyer le lecteur du gabarit à une personne de contact pour des informations complémentaires.

3.2.5 Syntaxe de la partie d'URL spécifique du type de service

La syntaxe de la partie spécifique du type de service de l'URL service: est fournie dans le document de gabarit comme la valeur de l'élément "template-url-syntax". Le champ <url-path> de l'URL service: est conçu pour fournir des informations supplémentaires pour localiser un service lorsque le champ <addr-spec> n'est pas suffisant. Le champ <url-path> distingue les URL d'un type de service particulier de ceux d'un autre type de service. Par exemple, dans le cas du type de service lpr, l'<url-path> peut être défini de telle sorte qu'il doive inclure le nom de la queue, mais d'autres types de services peuvent ne pas exiger cette information.

La syntaxe du champ <url-path> DOIT accompagner la définition d'un nouveau type de service, sauf si le schéma d'URL a déjà été normalisé et n'est pas un URL service:. La syntaxe est incluse dans le document de gabarit comme un ABNF [8] suivant les règles de la syntaxe d'URL décrites dans [5]. Il n'est pas exigé qu'un schéma de service prenne en charge un <url-path>. Le champ <url-path> peut être très simple, ou même omis. Si le schéma d'URL a déjà été normalisé, l'élément "template-url-syntax" DEVRAIT inclure une référence au document de normalisation approprié. Les types de service abstraits peuvent reporter ce champ dans les documents de gabarit qui décrivent leurs instances concrètes.

3.2.6 Définition d'attribut

Le gros du gabarit est normalement consacré à la définition des attributs spécifiques du type de service. Une définition d'attribut spécifie avec précision le type de l'attribut, les autres restrictions portant sur l'attribut (si il est multivaleurs, facultatif, etc.) du texte lisible par l'homme qui décrit l'attribut, et la liste des valeurs par défaut et admises. La seule information obligée est le type de l'attribut, le reste étant facultatif. L'enregistrement, le désenregistrement et l'utilisation des attributs dans les interrogations peuvent être accomplis en utilisant le protocole de localisation de service [10] ou d'autres moyens, et la discussion de ces sujets sort du domaine d'application du présent document.

Les attributs sont utilisés pour porter les informations sur un certain service afin de différencier différents services du même type. Ils portent des informations à utiliser dans le choix d'un service particulier avec lequel établir la communication, soit par un programme qui offre une interface humaine soit par programmation. Les attributs peuvent être codés dans différents jeux de caractères et dans différents langages. La procédure pour ce faire est décrite à la Section 6.

Une définition d'attribut commence par la spécification de l'identifiant de l'attribut et se termine par une seule ligne vide. Les définitions d'attributs ont cinq composants (dans l'ordre d'apparition dans la définition) :

1. un identifiant d'attribut qui agit comme le nom de l'attribut,
2. les descripteurs de l'attribut (type et fanions),
3. une liste facultative de valeurs qui sont allouées par défaut à l'attribut,
4. un bloc facultatif de texte lisible par l'homme qui fournit une brève description informative de l'attribut,
5. une liste facultative des valeurs admises qui restreignent la ou les valeur que l'attribut peut prendre.

3.2.6.1 Identifiant d'attribut

Une définition d'attribut commence par la spécification de l'identifiant de l'attribut. L'identifiant de l'attribut fonctionne comme le nom de l'attribut. Noter que les caractères utilisés pour composer un identifiant d'attribut sont restreints à ceux qui sont considérés comme non interdits pour l'inclusion dans un URL selon [5]. La raison en est que les services peuvent afficher des attributs importants dans leurs enregistrements d'URL service:. Chaque identifiant d'attribut doit être unique dans le gabarit. Comme les identifiants ont leur casse normalisée, les caractères majuscules et minuscules sont égaux.

3.2.6.2 Type d'attribut

Les attributs peuvent avoir un des cinq types différents suivants : chaîne, entier, booléen, opaque, ou mot-clé. La spécification du type de l'attribut est séparée de l'identifiant de l'attribut par un signe égal ("=") et suit le signe égal sur la même ligne. Les types chaîne, entier signé, et booléen ont la sémantique standard de langage de programmation ou de base de données. Les entiers sont restreints aux valeurs signées qui peuvent être représentée dans 32 bits. Le jeu de caractères utilisé pour représenter les chaînes n'est pas spécifié au moment de la définition du gabarit mais est plutôt déterminé par l'enregistrement du service. Les booléens ont la syntaxe standard. Les opaques ont des valeurs d'échappement d'octet qui peuvent être utilisées pour représenter toutes autres sortes de données. Les mots-clés sont des attributs qui n'ont pas d'autre caractéristique que leur existence (et éventuellement le texte descriptif de leur définition).

Les attributs mot-clé et booléen imposent des restrictions sur les parties suivantes de la définition de l'attribut. Les définitions d'attribut mot-clé DOIVENT n'avoir pas de fanion d'information qui suive la définition de type, ni aucune liste de valeurs par défaut ou admises. Les attributs booléens sont seulement d'une seule valeur, c'est-à-dire que des attributs booléen à plusieurs valeurs ne sont pas admis.

3.2.6.3 Fanions d'attribut

Les fanions déterminent les autres caractéristique d'un attribut. À l'exception des attributs mots-clés, qui ne peuvent pas avoir de fanion, les fanions suivent le type de l'attribut sur la même ligne que l'identifiant d'attribut, et sont séparé l'un de l'autre par une espace. Les fanions peuvent apparaître dans n'importe quel ordre après le type de l'attribut. D'autres informations ne doivent pas suivre les fanions, et seul un identifiant de fanion d'un type de fanion particulier est admis par définition d'attribut.

La sémantique des fanions est la suivante :

- o ou O (*pour optionnel*)
Indique que l'attribut est facultatif. Si ce fanion est absent, l'attribut est exigé dans tous les enregistrements de service.
- m ou M (*pour multivaleurs*)
Indique que l'attribut peut prendre plusieurs valeurs. Si ce fanion est présent, toute valeur d'un attribut multivaleurs a le même type que celui spécifié dans la partie type de la définition de l'attribut. Les attributs booléens ne doivent pas comporter ce fanion.
- l ou L (*pour littéral*)
Indique que l'attribut est littéral, c'est-à-dire, n'est pas destiné à être traduit dans d'autres langues. Si ce fanion est présent, l'attribut n'est pas considéré comme lisible par l'homme et ne devrait pas être traduit lorsque le gabarit est traduit. Voir la Section 6 pour plus d'informations sur la traduction.
- x ou X
Indique que les clients DEVRAIENT inclure l'attribut indiqué dans les demandes de services. Négliger d'inclure cet attribut ne va pas suffisamment différencier le service. Si les services sont obtenus sans choisir cet attribut, ils risquent d'être inutiles pour le client.

Les valeurs des attributs multivaleurs sont un ensemble non ordonné. Les suppressions de valeurs individuelles d'un attribut multivaleurs ne sont pas acceptées, et la suppression de l'attribut cause le retrait du jeu de valeurs entier.

3.2.6.4 Valeur ou liste de valeurs par défaut

Si la définition de l'attribut comporte une valeur par défaut ou, dans le cas des attributs multivaleurs, une liste de valeurs par défaut, il commence sur la seconde ligne de la définition d'attribut et continue sur les lignes suivantes jusqu'à ce qu'une ligne se termine sans une virgule. Par conséquent, les nouvelles lignes ne peuvent pas être incorporées dans les valeurs sauf par échappement. Voir au paragraphe 2.1.

Les types et définitions d'attribut particuliers restreignent la liste de valeurs par défaut. Les attributs mots-clés ne doivent pas avoir une liste de valeurs par défaut. Si la définition d'un attribut facultatif a une liste de valeurs admises, une valeur ou liste de valeurs par défaut n'est pas facultative mais exigée. Le motif en est que définir un attribut avec une liste de valeurs est destiné à restreindre les valeurs que l'attribut peut prendre, et exiger une valeur ou liste de valeurs par défaut assure que la valeur par défaut est membre de l'ensemble des valeurs admises.

La valeur ou liste de valeurs par défaut indique quelles valeurs sont données à l'attribut si aucune valeur n'est allouée à l'attribut lorsque un service est enregistré. Si la définition d'un attribut facultatif ne comporte pas de valeurs ou liste de valeurs par défaut, les valeurs par défaut suivantes sont allouées :

1. les valeurs de chaîne reçoivent la chaîne vide,
2. les valeurs d'entier reçoivent zéro,
3. les valeurs booléennes reçoivent faux,
4. les valeurs opaques reçoivent un ensemble d'octets ne contenant pas de valeurs,
5. les attributs multivaleurs sont initialisés avec une seule valeur.

Pour les besoins de la traduction des attributs non littéraux, les listes de valeurs par défaut sont prises comme étant un ensemble ordonné, et les traductions DOIVENT conserver cet ordre.

3.2.6.5 Texte descriptif

Immédiatement après la liste de valeurs par défaut, si il en est, un bloc de texte descriptif DEVRAIT être inclus dans la définition d'attribut. Ce texte est destiné à être lisible par l'homme, et devrait inclure une brève description informative de l'attribut. Il peut aussi donner des informations supplémentaires, telles que la description des valeurs admises. Ce texte est principalement destiné à l'affichage par les outils interactifs de navigation. Le texte descriptif est séparé de la définition environnante par un caractère dièse ("#") au début de la ligne. Le texte ne devrait cependant pas être traité comme un

commentaire par l'analyse et les autres outils, car il fait partie intégrante de la définition de l'attribut. Au sein du bloc de texte descriptif, le texte est transféré mot à mot, y compris le formatage et les sauts à la ligne, afin que tout le formatage soit préservé.

3.2.6.6 Liste des valeurs admises

Finalement, la définition de l'attribut se conclut par une liste facultative des valeurs admises. La liste des valeurs admises, s'il en est, suit le texte descriptif, ou, si celui-ci est absent, la liste des valeurs initiales. La syntaxe de la liste des valeurs admises est identique à celles des valeurs initiales. La liste des valeurs admises est aussi terminée par une ligne qui ne se finit pas par une virgule. Si la liste des valeurs admises est présente, son allocation aux attributs est restreinte aux membres de la liste.

Comme avec la liste des valeurs par défaut, la liste des valeurs admises est aussi considérée comme un ensemble ordonné pour les besoins de traduction.

3.2.6.7 Conclusion d'une définition d'attribut

Une définition d'attribut se conclut par une seule ligne vide.

4. Processus pour la normalisation de nouveaux types de services

De nouveaux types de services peuvent être suggérés simplement en fournissant un gabarit de type de service et une brève description de la façon d'utiliser le service. Le gabarit DOIT avoir son attribut de gabarit "template-version" réglé à 0.0.

Les numéros de révision MAJEURS viennent avant le '.', les numéros de révision MINEURS viennent après le '.'.

Le numéro de version mineur s'incrémente de un à chaque changement jusqu'à ce qu'il arrive à 1.0. Il n'est pas garanti qu'une version du gabarit de service soit rétro compatible avant qu'il atteigne 1.0.

Une fois qu'un gabarit de service a atteint 1.0, la définition est "gelée" pour cette version. De nouveaux gabarits doivent être définis, bien sûr, pour préciser cette définition, mais les règles suivantes doivent être respectées :

Un numéro de version MINEUR signifie l'introduction d'un changement compatible. Un numéro de version MAJEUR signifie l'introduction d'un changement incompatible. Ceci est formalisé par les règles suivantes :

- Tout nouvel attribut facultatif défini pour le gabarit augmente le numéro de version mineur de un. Tous les autres attributs pour la version doivent continuer d'être pris en charge comme avant. Un client qui prend en charge 1.x peut toujours utiliser les versions de 1.y (où $x < y$) car il ignore les attributs dont il ne sait rien.
- L'ajout d'un attribut exigé, la suppression de la prise en charge d'un attribut ou le changement de la définition d'un attribut exige de changer le numéro de version majeur d'un gabarit de service. Une application client peut n'être pas capable de faire usage de ces informations, ou elle peut avoir besoin d'obtenir le plus récent gabarit de service pour aider l'utilisateur à interpréter les informations de service.

Le gabarit est soumis à une liste de diffusion spéciale (voir à la Section 5). Le présent document doit inclure un marquage 'Le gabarit commence ici' et 'le gabarit se termine ici', dans le texte, afin qu'il soit trivial de couper et coller le gabarit à partir de la soumission.

La liste de diffusion sera disponible à svrloc-list@iana.org . Idéalement, les experts en mise en œuvre et déploiement du protocole concerné sont consultés afin d'ajouter ou supprimer des attributs ou changer leur définition pour rendre le gabarit aussi utile que possible. La liste de diffusion sera entretenue même lorsque le groupe de travail SVRLOC deviendra dormant pour les besoins de la discussion des gabarits de service.

Toutes les versions publiées du gabarit doivent être disponibles en ligne, y compris celles qui sont obsolètes.

Une fois le consensus réalisé, le gabarit devrait être republié avec les éventuelles corrections, en ayant son numéro de version réglé à 1.0. Les gabarits avec des numéros de version inférieurs à 1.0 ne sont pas soumis à l'IANA. À partir de ce point, les gabarits sont soumis. Voir à la Section 5 les détails sur la façon dont les gabarits sont soumis à un registre IANA des gabarits.

5. Considérations relatives à l'IANA

Il est de la responsabilité de l'IESG (par exemple, du directeur de la zone Applications) de nommer un expert désigné (voir [12]). Tout un chacun peut demander des éclaircissements sur un gabarit de service. Cela sollicite des retours de la part de la communauté concernée. Il appartient au réviseur appointé de déterminer si la demande d'éclaircissements est satisfaite. Il est de la responsabilité du réviseur de vérifier que toutes les demandes raisonnables d'éclaircissements sont satisfaites avant que le gabarit soit soumis pour inclusion dans le registre de l'IANA.

Lorsque le réviseur a déterminé que la soumission du gabarit est prête, il va le soumettre à l'IANA pour qu'il soit inclus dans un registre. Les participants à la liste de diffusion fournissent des réactions au processus mais ne prennent pas la décision d'accepter un gabarit de service.

Si une dispute survient quand aux décisions prises par le réviseur, la question peut venir en appel selon la procédure normale de l'IETF décrite pour le processus de normalisation.

L'IANA conservera un nom de boîte aux lettres de messagerie pour le travail de cette liste de diffusion, afin que "svrloc-list@iana.org" pointe sur un serveur de messagerie fourni par une organisation volontaire.

La soumission de gabarit de service DOIT être de la forme :

Nom du soumettant :

Langage du gabarit de service :

Considérations de sécurité

Texte du gabarit :

-----le gabarit commence ici-----

...

-----le gabarit se termine ici-----

Le fichier de gabarit de service a une convention de désignation :

<type-de-service> "." <numéro-de-version> "." <étiquette-de-langue>

Chacun de ces champs est défini à la Section 2. Ils correspondent aux valeurs des champs de gabarit "type", "template-version". Les fichiers des gabarits données en exemple dans ce document (à l'Appendice A) sont appelés :

"foo.0.0.en",

"Net-Transducer.0.0.en",

"Net-Transducer:Thermometer.0.0.de", et

"Net-Transducer:Thermomoter.0.0.en".

Le réviseur s'assurera que le gabarit soumis à l'IANA a la forme correcte et les champs requis.

Aucun gabarit de type de service ne sera accepté pour inclusion dans le registre des gabarits de service si il ne comporte une section de considérations de sécurité et d'informations de contact sur l'auteur du document du gabarit.

L'IANA tiendra un registre contenant à la fois les gabarits de type de service, et le document de description des gabarits contenant le gabarit de type de service, incluant toutes les versions précédentes. L'IANA recevra une notice pour inclure un gabarit de service dans le registre par un message électronique du réviseur. Ce message inclura le gabarit de service lui-même, qui est à enregistrer.

Ni le réviseur ni l'IANA ne prendront position sur les revendications de droits de reproduction ou les marques commerciales produites par rapport aux gabarits.

6. Considérations d'internationalisation

L'URL service: doit être codé en utilisant les règles établies dans [5]. Le codage du jeu de caractères est limité aux gammes spécifiques au sein du jeu de caractères US-ASCII [4].

Le gabarit est codé en caractères UTF-8.

6.1 Identification de langage et traduction

Le langage utilisé dans les chaînes d'attribut devrait être identifié en fournissant une étiquette de langue [3] dans la soumission de gabarit de service (voir la Section 5).

Un programme peut traduire un enregistrement de service d'une langue à une autre pourvu qu'il ait à la fois le gabarit de la langue pour l'enregistrement et le gabarit de la langue cible désirée. Tous les attributs normalisés sont dans le même ordre dans les deux gabarits. Toutes les chaînes non arbitraires, y compris le texte d'aide descriptif, sont directement traduisibles d'une langue à l'autre. Les définitions d'attribut non littérales, les identifiants d'attribut, les noms de type d'attribut, les fanions d'attribut, et la constante booléenne "vrai" et "faux" ne sont jamais traduites. La traduction des identifiants d'attributs est interdite parce que, comme avec les noms de domaines, ils peuvent faire partie d'un URL service: et donc, leur jeu de caractères est restreint. De plus, comme avec les identifiants variables dans les langages de programmation, ils pourraient être incorporés dans le code de programme.

Toutes les chaînes utilisées dans les valeurs d'attribut sont supposées traduisibles sauf définition explicite qu'elles sont littérales, de sorte qu'une traduction au mieux (voir ci-dessous) ne modifie pas les chaînes qui sont destinées à être interprétées par un programme, pas par une personne.

Un exemple de gabarit de service traduit figure à l'Appendice A.

Il y a deux façons de faire la traduction : normalisation et au mieux.

Lorsque le type de service est normalisé, plus d'un document peut être soumis à la révision. Une description de type de service est approuvée comme maître, de sorte que lorsque un gabarit de type de service est mis à jour dans une langue, toutes les traductions (au moins éventuellement) reflètent la même sémantique.

Si aucun document n'existe pour décrire la traduction standard du type de service, il devrait être fait une traduction 'au mieux' des chaînes.

7. Considérations pour la sécurité

Les gabarits de type de service fournissent des informations qui sont utilisées pour interpréter les informations obtenues par le protocole de localisation de service. Si ces gabarits sont modifiés ou si de faux gabarits sont distribués, les services peuvent ne pas s'enregistrer correctement, ou les clients peuvent n'être pas capables d'interpréter les informations de service.

Les URL service: eux-mêmes spécifient le point d'accès au service et le protocole pour un type de service particulier. Ces URL service: pourraient être distribués et indiquer une localisation d'un service autre que celui qu'on veut normalement utiliser. Le protocole de localisation de service [10] distribue les URL service: et a un mécanisme d'authentification qui permet aux URL service: de services enregistrés d'être signés et aux signatures d'être vérifiées par les clients.

Chaque gabarit de service devra comporter une section de considérations de sécurité qui décrira les questions de sécurité posées par l'utilisation du schéma de service pour le type de service spécifié.

Appendice A. Exemples de gabarit de service

Le texte des paragraphes de l'exemple de gabarit est à considérer comme étant un seul fichier. Il est complètement fictif (c'est-à-dire que les exemples ne représentent pas des services réels).

L'exemple FOO montre comment utiliser les gabarits de service dans une application qui a très peu d'attributs. Les clients demandent au serveur FOO où sont situées leurs données d'utilisateur en incluant leur nom d'utilisateur comme valeur de l'attribut d'utilisateur.

L'exemple Net-Transducer montre comment sont définis les types de service abstraits et comment une instance concrète correspondante est définie. Un système peut prendre en charge n'importe lequel des services NetTransducer. On ne donne ici qu'une seule instance concrète du type abstrait.

Il n'est pas nécessaire d'enregistrer les gabarits concrets pour un type de service abstrait si le gabarit de type de service abstrait est parfaitement clair quant aux valeurs qu'il est possibles d'utiliser comme type concret, et à leur interprétation.

A.1 FOO

Voici l'exemple de la soumission du gabarit de service FOO :

Nom du soumettant : "Erik Guttman" <Erik.Guttman@sun.com>

Langage du gabarit de service : fr

Considérations de sécurité :

Si les attributs USAGER et GROUPES sont inclus, il existe une possibilité que la liste des identités des usagers ou des groupes puisse être découverte. Ces informations seraient autrement difficiles à découvrir.

Texte du gabarit

-----Le gabarit commence ici-----

type-de-gabarit=FOO

version-de-gabarit=0.0

description-du-gabarit=L'URL service FOO fournit la localisation d'un service FOO.

syntaxe-de-gabarit-d'url=

chemin-d'url= ; Il n'y a pas de chemin d'URL de défini pour un URL FOO.

usagers= chaîne M L O

La liste de tous les usagers que prend en charge le serveur FOO.

groupes= chaîne M L O

La liste de tous les groupes que prend en charge le serveur FOO.

-----Le gabarit se termine ici-----

Ce gabarit pourrait être internationalisé en enregistrant une autre version, disons en allemand.

Name of submitter: "Erik Guttman" <Erik.Guttman@sun.com>

Language of service template: de

Security Considerations:

Wenn die USER und GROUPS Eigenschaften inbegriffen sind, besteht die Moeglichkeit, dass die Liste der Identitaeten von Benutzern oder Gruppen entdeckt werden kann. Diese Information wurde unter anderen Umstaenden schwierig zu entdecken sein.

Template Text:

-----template begins here-----

template-type=FOO

template-version=0.0

template-description=

Der FOO Service URL zeigt die Stelle von einem Foo Service an.

template-url-syntax=

url-path= ; Es gibt keinen fuer den FOO URL definierten Pfad.

users= string M L O

Die Liste aller Users, die der FOO Server unterstuetzt.

groups= string M L O

Die Liste aller Gruppen, die der FOO Server unterstuetzt.

-----template ends here-----

Noter que les étiquettes d'attribut ne sont pas traduites. Si on désire une traduction, la convention suggérée pour ce faire est de définir un attribut séparé appelé <étiquette>-de-localisation pour chaque étiquette d'attribut qui est à localiser. Cela va aider à afficher les étiquettes d'attribut sur une interface humaine.

Par exemple, dans le cas ci-dessus, les deux attributs suivants pourraient être définis :

localize-users= string

Benutzer

localize-groups= string

Gruppen

Les attributs (dans le format de liste d'attributs SLPv2) pour un enregistrement de service d'un service FOO fondé sur ce gabarit pourrait être, en allemand :

(users=Hans,Fritz),(groups=Verwaltung,Finanzbuchhaltung), (template-type=FOO),(template-version=0.0),(template-description= Der FOO Service URL zeigt die Stelle von einem Foo Service an.), (template-url-syntax= \OD url-path= ; Es gibt kein fuer den FOO URL definiert Pfad. \OD),(localize-users=Benutzer), (localize-groups=Gruppen)

Tous ceux qui obtiennent ces attributs pourraient afficher "Benutzer=Hans,Fritz" sur une interface humaine utilisant les informations incluses. Noter que les attributs du gabarit ont été inclus dans cet enregistrement. Ceci est FACULTATIF, mais rend possible de découvrir quel gabarit a été utilisé pour enregistrer le service.

A.2 Type de service abstrait : Net-Transducer

Un exemple de soumission de gabarit de type de service abstrait est :

Nom du soumettant : "Erik Guttman" <Erik.Guttman@sun.com>

Langage du gabarit de service : fr

Considérations de sécurité :

Voir les considérations de sécurité des types de service concrets.

Texte du gabarit :

-----Le gabarit commence ici-----

type-de-gabarit=Net-Transducer

version-du-gabarit=0.0

description-du-gabarit=

C'est un type de service abstrait. L'objet du type de service Net-Transducer est d'organise en une seule catégorie tous les traducteurs à capacité réseau qui ont certaines propriétés.

syntaxe-de-gabarit-d'url=

chemin-d'url= ; Dépend du type de service concret. Voir ces gabarits.

unités-d'échantillon= chaîne L

Les unités d'échantillon que le traducteurs fournit, par exemple ° C (degrés Celsius), V (Volts), kg (kilogrammes), etc.

résolution-d'échantillon= chaîne L

Résolution du traducteur. Par exemple, 10⁻³ signifie que le traducteur a une résolution de 0,001 unité.

taux-d'échantillon= entier L

Vitesse à laquelle les échantillons sont obtenus par seconde. Par exemple, 1000 signifie qu'un échantillon est obtenu à chaque milliseconde.

-----Le gabarit se termine ici-----

A.3 Type de service concret : Net-Transducer:Thermometer

Voici un autre exemple de soumission de gabarit de service, qui fournit un type de service concret correspondant au gabarit abstrait ci-dessus.

Nom du soumettant : "Erik Guttman" <Erik.Guttman@sun.com>

Langage du gabarit de service : fr

Considérations de sécurité :

Il n'y a pas d'authentification du résultat du traducteur. Donc, le résultat du thermomètre pourrait faciliter être falsifié.

Texte du gabarit :

-----Le gabarit commence ici-----

type-de-gabarit=service:Net-Transducer:Thermometer

version-du-gabarit=0.0

description-du-gabarit=

Le thermomètre est un traducteur réseau qui est capable de lire la température. Les données sont lues en ouvrant une connexion TCP à un des accès dans l'URL service et en lisant une chaîne ASCII jusqu'à ce qu'on rencontre un caractère NUL. Le client peut continuer de lire les données pas plus vite que le taux d'échantillonnage, ou clore la connexion.

syntaxe-de-gabarit-d'url=

chemin-d'url = "accès=" liste des accès

liste des accès = accès / accès "," accès

accès = 1*CHIFFRE ; Voir la règle de production du <accès> d'URL service. Ce sont les accès sur lesquels les connexions peuvent être faites.

description-de-localisation=chaîne

C'est l'endroit où le thermomètre est situé.

opérateur=chaîne O

C'est l'opérateur à contacter pour mettre le thermomètre en service.

-----Le gabarit se termine ici-----

A.4 URL service: et SLP

Un usager avec une application de calendrier à capacité FOO ne devrait pas se soucier de savoir l'adresse de son serveur FOO. Le programme de client de calendrier utilise SLP pour obtenir l'URL service: FOO automatiquement, disons 'service:foo://server1.nosuch.org', en produisant une demande de service. En cas de défaillance de ce serveur FOO, le client de calendrier peut produire à nouveau la même demande de service pour trouver le serveur FOO de secours, disons 'service:foo://server2.nosuch.org'. Dans les deux cas, l'URL service: se conforme au gabarit de service FOO comme le font les attributs associés (usager et groupe).

Un thermomètre réseau fondé sur le gabarit ci-dessus pourrait être annoncé par la liste d'attributs SLPv2 :

URL = service:net-transducer:thermometer://v33.test/ports=3211

Attributs = (description-de-localisation=Missile bay 32),

(opérateur=Joe Agent), (unités-d'échantillon=C),

(résolution-d'échantillon=10⁻¹),(taux-d'échantillon=10),

(type-de-gabarit=service:net-transducer:thermometer),

(version-de-gabarit=0.0),(description-du-gabarit=

Le thermomètre est un traducteur réseau capable de lire la température.

Les données sont lues en ouvrant une connexion TCP sur un des accès dans l'URL service et en lisant une chaîne ASCII jusqu'à ce qu'on rencontre un caractère NUL. Le client peut continuer de lire les données pas plus vite que le taux d'échantillon, ou clore la connexion.),

(syntaxe-de-gabarit-d'url= \OD "accès=" liste-d'accès \OD

liste-d'accès = accès / accès "," accès \OD

accès = 1*CHIFFRE \OD

; Voir la règle de production <accès> de l'URL service. \OD

; Ce sont les accès sur lesquels les connexions peuvent être faites.\OD)

Cela peut être très utile pour un technicien qui veut trouver des thermomètres à Missile bay 32, par exemple.

Noter que les attributs du gabarit sont annoncés. La valeur de syntaxe-de-gabarit-d'url exige des caractères CR explicitement échappés afin que la syntaxe ABNF soit correcte.

Appendice B. Remerciements

Merci à Michael Day et Leland Wallace qui ont aidé pour les portions de syntaxe d'adresse IPX et AppleTalk. Ryan Moats a fourni de précieux retours tout au long de la rédaction du présent document.

Appendice C. Références

- [1] "Protocol et service names", octobre 1994. <ftp://ftp.isi.edu/in-notes/iana/assignments/service-names>
- [2] "Port numbers", juillet 1997. <ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers>
- [3] [RFC1766] H. Alvestrand, "Étiquettes pour l'identification des langues", mars 1995. (*Obsolète, voir RFC3066, RFC3282*) (P.S.)
- [4] ANSI. "Coded Character Set -- 7-bit American Standard code for Information Interchange". X3.4-1986, 1986.
- [5] [RFC2396] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiants de ressource uniformes (URI) : Syntaxe générique", août 1998. (*Obsolète, voir RFC3986*)
- [6] [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [7] Apple Computer. "Inside Macintosh". Addison-Wesley, 1993.
- [8] [RFC2234] D. Crocker et P. Overell, "[BNF augmenté pour les spécifications de syntaxe](#) : ABNF", novembre 1997. (*Obsolète, voir RFC5234*)
- [9] S. Gursharan, R. Andrews, et A. Oppenheimer. "Inside AppleTalk". Addison-Wesley, 1990.
- [10] [RFC2608] E. Guttman et autres, "[Protocole de localisation de service](#), version 2", juin 1999. (*MàJ par RFC3224*) (P.S.)
- [11] [RFC2222] J. Myers, "Authentification simple et couche de sécurité (SASL)", octobre 1997. (*Obsolète, voir RFC4422, RFC4752*) (*MàJ par RFC2444*) (P.S.)
- [12] [RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre, 1998. (*Rendue obsolète par la RFC5226*)
- [13] [RFC2244] C. Newman, J. G. Myers, "[ACAP – Protocole d'accès à la configuration d'application](#)", novembre 1997. (P.S.)
- [14] Inc Novell. "IPX RIP et SAP Router Specification. Part Number 107-000029-001, Version 1.30", mai 1996.
- [15] [RFC2165] J. Veizades et autres, "[Protocole de localisation de service](#)", juin 1997. (*MàJ par RFC2608, RFC2609*) (P.S.)
- [16] [RFC2279] F. Yergeau, "[UTF-8, un format de transformation](#) de la norme ISO 10646", janvier 1998. (*Obsolète, voir RFC3629*) (D.S.)

Appendice D. Adresse des auteurs

Les questions sur le présent mémoire peuvent être adressées à :

Erik Guttman
Sun Microsystems
Bahnstr. 2
74915 Waibstadt
Germany
Téléphone : +49 7263 911484
Fax : +1 650 786 5992
mél : erik.guttman@sun.com

Charles E. Perkins
Sun Microsystems
15 Network Circle
Menlo Park, CA 94303
USA
Téléphone : +1 650 786 6464
Fax : +1 650 786 6445
mél : cperkins@sun.com

James Kempf
Sun Microsystems
15 Network Circle
Menlo Park, CA 94303
USA
téléphone : +1 650 786 5890
Fax : +1 650 786 6445
mail : james.kempf@sun.com

Appendice E. Déclaration complète de droits de reproduction

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

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans restriction d'aucune sorte, pourvu que la déclaration de copyright ci-dessus et le présent et paragraphe soient inclus dans toutes telles copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de droits de reproduction ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour le besoin du développement des normes Internet, auquel cas les procédures de droits de reproduction définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society, ses successeurs ou ayant droits.

Le présent document et les informations y contenues sont fournies 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 ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Remerciement

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