Groupe de travail Réseau
Request for Comments: 3987
Catégorie : Standards Track

M. Duerst
W3C
M. Suignard
Microsoft Corporation

janvier 2005

Identifiants de ressource internationalisée (IRI)

Etat du présent Mémo

Le présent document spécifie un protocole en Internet standard track pour la communauté Internet, et appelle à discussion et suggestions d'amélioration. Prière de se référer à l'édition actuelle de l'"Internet Official Protocol Standards" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution de ce mémo n'est pas soumise à restriction.

Copyright

Copyright (C) The Internet Society (2005).

Résumé

Le présent document définit un nouvel élément de protocole, l'identifiant de ressource internationalisée (IRI, *Internationalized Resource Identifier*), comme complément à l'identifiant de ressource uniforme (URI, *Uniform Resource Identifier*). Un IRI est une séquence de caractères de l'ensemble de caractères universel (*Universal Character Set*) (Unicode/ISO 10646). Il définit une transposition de l'IRI à l'URI, ce qui signifie que les IRI peuvent être utilisés à la place des URI, aux endroits appropriés, pour identifier des ressources.

La définition d'un nouveau protocole a été préférée à l'extension ou au changement de la définition des URI, afin de permettre une distinction claire et d'éviter les incompatibilités entre les logiciels existants. Des lignes directrices sont données pour l'utilisation et le développement des IRI dans divers protocoles, formats, et composants de logiciels qui sont actuellement en rapport avec les URI.

Table des matières

1. Introduction....3
2. Syntaxe d'IRI....5
3. Relations entre les IRI et les URI....9
4. IRI bidirectionnels pour langages de droite à gauche....13
5. Normalisation et comparaison....16
6. Utilisation des IRI....21
7. Guide de traitement URI/IRI (pour information)....23
8. Considérations sur la sécurité....27
9. Remerciements....28
10. Références....28

Appendice A. Autres conceptions envisagées....31

1. Introduction

1.1. Généralités et motivations

Un identifiant de ressource uniforme (URI, *Uniform Resource Identifier*) est défini dans [RFC3986] comme une séquence de caractères choisis dans un sous-ensemble limité du répertoire des caractères US-ASCII [ASCII].

Les caractères dans les URI sont fréquemment utilisés pour représenter des mots des langages naturels. Cette utilisation a de nombreux avantages : de tels URI sont plus faciles à mémoriser, plus faciles à interpréter, plus faciles à transcrire, plus faciles à créer, et plus faciles deviner. Cependant, pour tous les langages autres que l'anglais, l'écriture naturelle utilise des caractères autres que A - Z. Pour de nombreuses personnes, le traitement des caractères latins est aussi difficile que le traitement des caractères des autres écritures pour ceux qui n'utilisent que l'alphabet latin. De nombreux langages qui s'écrivent avec une écriture non latine sont transcris avec des lettres latines. Ces transcriptions sont maintenant souvent utilisées dans les URI, mais elles introduisent des ambiguïtés supplémentaires.

L'infrastructure pour le traitement approprié des caractères provenant d'écritures locales est maintenant largement déployée dans des versions locales de systèmes d'exploitation et de logiciels d'application. Un logiciel pouvant traiter en même temps une large variété d'écritures et de langages est de plus en plus courant. Et aussi, un nombre croissant de protocoles et de formats supportent une large gamme de caractères.

Le présent document définit un nouvel élément de protocole appelé Identifiant de ressource internationalisée (IRI, *Internationalized Resource Identifier*) en étendant la syntaxe des URI à un répertoire de caractères beaucoup plus large. Il définit aussi des versions "internationalisées" correspondant aux autres produits de [RFC3986], comme les références d'URI. La syntaxe des IRI est définie à la Section 2, et les relations entre IRI et URI à la Section 3.

L'utilisation de caractères hors de la gamme A - Z dans les IRI cause quelques difficultés. La Section 4 discute du cas particulier des IRI bidirectionnels, la Section 5 de diverses formes d'équivalence entre les IRI, et la Section 6 de l'utilisation des IRI dans différentes situations. La Section 7 donne des lignes directrices informatives supplémentaires, et la Section 8 traite des considérations de sécurité.

1.2. Applicabilité

Les IRI sont destinés à être compatibles avec les recommandations sur les nouveaux schémas d'URI de [RFC2718]. La compatibilité est établie en spécifiant une transposition bien définie et déterministe entre la séquence de caractère de l'IRI et la séquence de caractères d'URI fonctionnellement équivalente. L'utilisation pratique des IRI (ou des références d'IRI) à la place des URI (ou des références d'URI) dépend de la satisfaction des conditions suivantes :

a. Un élément de protocole ou de format devrait être explicitement conçu pour pouvoir porter les IRI. L'intention n'est pas d'introduire les IRI dans des contextes qui ne sont pas définis pour les accepter. Par exemple, le schéma XML [XMLSchema] a un type explicite "anyURI" qui inclut les IRI et les références d'IRI. Donc, les IRI et les références d'IRI peuvent être dans des attributs et éléments de type "anyURI". D'un autre côté, dans le protocole HTTP [RFC2616], la Demande d'URI est définie comme un URI, ce qui signifie que l'utilisation directe des IRI n'est pas permise dans les demandes HTTP.

b. Le protocole ou format portant les IRI devrait avoir un mécanisme pour représenter la large gamme de caractères utilisés dans les IRI, de façon native ou par un mécanisme d'échappement spécifique du protocole ou du format (par exemple, des références de caractères numériques dans [XML1]).

c. L'URI correspondant à l'IRI en question doit coder les caractères originaux en octets en utilisant UTF-8. Pour les nouveaux schémas d'URI, ceci est recommandé dans [RFC2718]. Cela peut s'appliquer à tout un schéma (par exemple, les URL IMAP [RFC2192] et les URL POP [RFC2384], ou la syntaxe d'URN [RFC2141]). Cela peut s'appliquer à une partie spécifique d'un URI, comme l'identifiant de fragment (par exemple, [XPointer]). Cela peut s'appliquer à un URI spécifique d'une ou plusieurs de ses parties. Voir les détails au paragraphe 6.4.

1.3. Définitions

Les définitions sont utilisées dans le présent document, suivant les termes de [RFC2130], [RFC2277], et [ISO10646].

caractère : membre d'un ensemble d'éléments utilisé pour l'organisation, le contrôle, ou la représentation de données. Par exemple, "LETTRE MAJUSCULE LATINE A" nomme un caractère.

octet : séquence ordonnée de huit bits considérée comme une unité.

répertoire de caractères : ensemble (au sens mathématique) de caractères.

séquence de caractères : caractères en séquence (un après l'autre).

séquence d'octets : octets en séquence (un après l'autre).

codage de caractère : méthode de représentation d'une séquence de caractères comme une séquence d'octets (éventuellement avec des variantes). Et aussi, méthode de conversion (non ambiguë) d'une séquence d'octets en séquence de caractères.

charset : nom d'un paramètre ou attribut utilisé pour identifier un codage de caractère.

UCS : ensemble de caractères universel (*Universal Character Set*). Ensemble de caractères codés défini par la norme ISO/CEI 10646 [ISO10646] et la norme Unicode [UNIV4].

référence d'IRI : note l'utilisation commune d'un identifiant de ressource internationalisée. Une référence d'IRI peut être absolue ou relative. Cependant, l'"IRI" qui résulte d'une telle référence n'inclut que des IRI absolus ; toute référence d'IRI relative est résolue dans sa forme absolue. Noter que dans [RFC2396] les URI n'incluent pas les identifiants de fragment, mais dans [RFC3986] les identifiants de fragment font partie des URI.

texte courant : texte humain (paragraphes, propositions, phrases) avec une syntaxe conforme aux conventions orthographiques d'un langage naturel, par opposition à la syntaxe définie pour faciliter le traitement par les machines (par exemple, langages de balisage ou de programmation).

élément de protocole : toute portion d'un message qui affecte le traitement de ce message par le protocole en question.

élément de présentation : forme de présentation correspondant à un élément de protocole ; par exemple, utilisant une plus large gamme de caractères.

créer (un URI ou IRI) **:** en ce qui concerne les URI et IRI, le terme est utilisé pour la création initiale. Cela peut être la création initiale d'une ressource avec un certain identifiant, ou l'exposition initiale d'une ressource sous un identifiant particulier.

générer (un URI ou IRI) **:** en ce qui concerne les URI et IRI, le terme est utilisé lorsque l'IRI est généré par déduction à partir d'une autre information.

1.4. Notation

Les RFC et les Projets Internet n'admettent actuellement aucun caractère en dehors du répertoire US-ASCII. Le présent document utilise donc diverses notations spéciales pour noter de tels caractères dans les exemples.

Dans le texte, les caractères en-dehors du répertoire US-ASCII sont parfois référencés en utilisant le préfixe 'U+', suivi de quatre à six chiffres hexadécimaux.

Pour représenter les caractères non US-ASCII dans les exemples, le présent document utilise deux notations : 'Notation XML' et ' Notation Bidi'.

La notation XML utilise un préfixe '&#x', un postfixe ';', et le nombre hexadécimal du caractère en UCS entre les deux. Par exemple, я représente la LETTRE CYRILIQUE MAJUSCULE YA. Dans cette notation, un '&' réel est noté '&'.

La notation Bidi est utilisée pour des exemples bidirectionnels : les lettres minuscules représentent les lettres latines ou d'autres lettres qui sont écrites de gauche à droite, alors que les lettres majuscules représentent les lettres arabes ou hébreux qui sont écrites de droite à gauche.

Pour noter les octets réels dans les exemples (par opposition aux octets codés en pourcentage), les deux chiffres hexadécimaux qui notent l'octet sont compris entre "<" et ">". Par exemple, l'octet souvent noté 0xc9 est noté ici <c9>.

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" doivent être interprétés comme décrit par la [RFC2119].

2. Syntaxe d'IRI

La présente section définit la syntaxe des Identifiants de Ressource Internationalisés (IRI).

Comme avec les URI, un IRI est défini comme une séquence de caractères, et non comme une séquence d'octets. Cette définition incorpore le fait que les IRI peuvent être écrits sur du papier ou lus à la radio aussi bien que mémorisés ou transmis numériquement. Le même IRI peut être représenté comme différentes séquences d'octets dans différent protocoles ou documents si ces protocoles ou documents utilisent différents codages de caractère (et/ou codages de transfert). Utiliser le même codage de caractère que le protocole ou document contenant garantit que les caractères en IRI peuvent être traités (par exemple, recherchés, convertis, affichés) de la même façon que le reste du protocole ou document.

2.1. Résumé de la syntaxe d'IRI

Les IRI sont définis de la même façon que les URI dans [RFC3986], mais la classe des caractères non réservés est étendue par l'ajout des caractères de l'UCS (Universal Character Set, [ISO10646]) (*ensemble universel de caractères*) au-delà de U+007F, du fait des limitations apportées par les règles syntaxiques ci-dessous et du paragraphe 6.1.

Autrement, la syntaxe et l'utilisation des composants et des caractères réservés sont les mêmes que dans [RFC3986]. Toutes les opérations définies dans [RFC3986], telles que la résolution des références croisées, peuvent s'appliquer aux IRI par un logiciel de traitement d'IRI exactement de la même façon que le sont les URI par les logiciels de traitement d'URI.

Les caractères en-dehors du répertoire US-ASCII ne sont pas réservés et par conséquent NE DOIVENT PAS être utilisés pour les besoins de la syntaxe, tels que pour délimiter les composants dans les schéma nouvellement définis. Par exemple, U+00A2, SIGNE CENT, n'est pas admis comme délimiteur dans les IRI, parce qu'il est dans la catégorie 'non réservée'. Ceci est semblable au fait qu'il n'est pas possible d'utiliser '-' comme délimiteur dans les URI, parce qu'il est dans la catégorie 'non réservée'.

2.2. ABNF pour les références d'IRI et les IRI

Bien qu'il soit possible de définir les références d'IRI et les IRI en les transformant simplement en références d'URI et en URI, ils peuvent aussi être acceptés et traités directement. Donc, on donne ici une définition ABNF pour les références d'IRI (qui sont le concept le plus général et le début de la grammaire) et les IRI. La syntaxe de cet ABNF est décrite dans [RFC2234]. Les numéros des caractères sont tirés de l'UCS, sans que cela implique aucun codage binaire réel. Les produits terminaux en ABNF sont des caractères, pas des octets.

La grammaire suivante suit de près la grammaire d'URI de [RFC3986], excepté que la gamme des caractères non réservés est étendue pour inclure les caractères UCS, avec cette restriction que les caractères UCS à usage privé ne peuvent survenir que dans les parties d'interrogation. La grammaire est

divisée en deux parties : les règles qui diffèrent de celles de [RFC3986] à cause de l’extension mentionnée ci-dessus, et les règles qui sont les mêmes que celles de [RFC3986]. Pour les règles qui sont différentes de celles de [RFC3986], les noms des non terminaux ont été changés comme suit. Si le non terminal contient 'URI', cela a été changé en 'IRI'. Autrement, un 'i' a été mis en préfixe.

Les règles suivantes sont différentes de celles de [RFC3986] :

```
   IRI           = scheme ":" ihier-part [ "?" iquery ]

                       [ "#" ifragment ]

ihier-part     = "//" iauthority ipath-abempty

               / ipath-absolute

               / ipath-rootless

               / ipath-empty

IRI-reference  = IRI / irelative-ref

absolute-IRI   = scheme ":" ihier-part [ "?" iquery ]

irelative-ref  = irelative-part [ "?" iquery ] [ "#" ifragment ]

irelative-part = "//" iauthority ipath-abempty

                 / ipath-absolute

               / ipath-noscheme

               / ipath-empty

iauthority     = [ iuserinfo "@" ] ihost [ ":" port ]

iuserinfo      = *( iunreserved / pct-encoded / sub-delims / ":" )

ihost          = IP-literal / IPv4address / ireg-name

ireg-name      = *( iunreserved / pct-encoded / sub-delims )

ipath          = ipath-abempty   ; commence par "/" ou est vide

               / ipath-absolute  ; commence par "/" mais pas par "//"

               / ipath-noscheme  ; commence par un segment qui n’est pas deux points

               / ipath-rootless  ; commence par un segment

               / ipath-empty     ; caractères zéro

ipath-abempty  = *( "/" isegment )

ipath-absolute = "/" [ isegment-nz *( "/" isegment ) ]

ipath-noscheme = isegment-nz-nc *( "/" isegment )

ipath-rootless = isegment-nz *( "/" isegment )
```

```
ipath-empty    = 0<ipchar>

isegment       = *ipchar

isegment-nz    = 1*ipchar

isegment-nz-nc = 1*( iunreserved / pct-encoded / sub-delims

                     / "@" )

               ; segment de longueur différente de zéro sans deux points ":"

  ipchar         = iunreserved / pct-encoded / sub-delims / ":"

               / "@"

  iquery         = *( ipchar / iprivate / "/" / "?" )

  ifragment      = *( ipchar / "/" / "?" )

  iunreserved    = ALPHA / DIGIT / "-" / "." / "_" / "~" / ucschar

  ucschar        = %xA0-D7FF / %xF900-FDCF / %xFDF0-FFEF

               / %x10000-1FFFD / %x20000-2FFFD / %x30000-3FFFD

               / %x40000-4FFFD / %x50000-5FFFD / %x60000-6FFFD

               / %x70000-7FFFD / %x80000-8FFFD / %x90000-9FFFD

               / %xA0000-AFFFD / %xB0000-BFFFD / %xC0000-CFFFD

               / %xD0000-DFFFD / %xE1000-EFFFD

  iprivate       = %xE000-F8FF / %xF0000-FFFFD / %x100000-10FFFD
```

Certaines productions sont ambiguës. L'algorithme "le premier qui correspond l'emporte" (dit aussi "glouton") s'applique. Pour plus de détails, voir la [RFC3986].

Les règles suivantes sont les mêmes que celles de [RFC3986]:

```
scheme         = ALPHA *( ALPHA / DIGIT / "+" / "-" / "." )

port           = *DIGIT

IP-literal     = "[" ( IPv6address / IPvFuture  ) "]"

IPvFuture      = "v" 1*HEXDIG "." 1*( unreserved / sub-delims / ":" )

IPv6address    =                            6( h16 ":" ) ls32

               /                       "::" 5( h16 ":" ) ls32

               / [               h16 ] "::" 4( h16 ":" ) ls32

               / [ *1( h16 ":" ) h16 ] "::" 3( h16 ":" ) ls32

               / [ *2( h16 ":" ) h16 ] "::" 2( h16 ":" ) ls32
```

```
            / [ *3( h16 ":" ) h16 ] "::"    h16 ":"   ls32
            / [ *4( h16 ":" ) h16 ] "::"              ls32
            / [ *5( h16 ":" ) h16 ] "::"              h16
            / [ *6( h16 ":" ) h16 ] "::"
h16           = 1*4HEXDIG
ls32          = ( h16 ":" h16 ) / IPv4address
IPv4address    = dec-octet "." dec-octet "." dec-octet "." dec-octet
dec-octet      = DIGIT                ; 0-9
            / %x31-39 DIGIT         ; 10-99
            / "1" 2DIGIT            ; 100-199
            / "2" %x30-34 DIGIT     ; 200-249
            / "25" %x30-35          ; 250-255
pct-encoded    = "%" HEXDIG HEXDIG
unreserved     = ALPHA / DIGIT / "-" / "." / "_" / "~"
reserved       = gen-delims / sub-delims
gen-delims     = ":" / "/" / "?" / "#" / "[" / "]" / "@"
sub-delims     = "!" / "$" / "&" / "'" / "(" / ")"
            / "*" / "+" / "," / ";" / "="
```

Cette syntaxe ne prend pas en charge les identifiants de zone d’adressage orientés IPv6.

3. Relations entre les IRI et les URI

Les IRI sont conçus pour le remplacement des URI comme identifiants de ressources pour les protocoles, formats, et composants de logiciels qui utilisent un répertoire de caractères fondé sur UCS. Ces protocoles et composants peuvent n'avoir jamais besoin d'utiliser directement des URI, particulièrement lorsque l'identifiant de ressource n'est utilisé que pour la simple identification. Cependant, lorsque l'identifiant de ressource est utilisé pour récupérer des ressources, il est nécessaire dans de nombreux cas de déterminer l'URI associé, parce que la plupart des mécanismes de récupération actuels ne sont définis que pour les URI. Dans ce cas, les IRI peuvent servir comme éléments de présentation pour les éléments de protocole d'URI. Par exemple, une barre d'adresse dans un agent d'utilisateur Web. (Des justifications supplémentaires figurent au paragraphe 3.1.)

3.1. Transposition des IRI en URI

Ce paragraphe définit comment transposer un IRI en un URI. Tout ce paragraphe s'applique aussi aux références d'IRI et aux références d'URI, ainsi qu'à tous leurs composants (par exemple, les identifiants de fragment).

Cette transposition a deux objets.

Syntaxique. De nombreux schémas et composants d'URI définissent des restrictions syntaxiques supplémentaires qui ne sont pas traitées au paragraphe 2.2. Les restrictions spécifiques d'un schéma s'appliquent aux IRI en convertissant les IRI en URI et en vérifiant les URI à l'égard des restrictions spécifiques du schéma.

D'interprétation. Les URI identifient les ressources de différentes façons. Les IRI identifient aussi des ressources. Lorsque l'IRI est seulement utilisé pour des besoins d'identification, il n'est pas nécessaire de transposer l'IRI en URI (voir la section 5). Cependant, lorsqu'un IRI est utilisé pour restituer des ressources, la ressource que l'IRI localise est la même que celle localisée par l' URI obtenu après conversion de l'IRI conformément à la procédure définie ici. Cela signifie qu'il n'est pas besoin de définir une résolution particulière au niveau de l'IRI.

Les applications DOIVENT transposer les IRI en URI en utilisant les étapes suivantes.

étape 1. Générer une séquence de caractères UCS à partir du format IRI d'origine. Cette étape possède les trois variantes suivantes, selon la forme de l'entrée :

a. Si l'IRI écrit sur papier, lu à voix haute, ou autrement représenté comme une séquence de caractères indépendamment de tout codage de caractère, représenter l'IRI comme une séquence de caractères tirés de l'UCS normalisés conformément à la Forme C de normalisation (NFC, [UTR15]).

b. Si l'IRI est en représentation numérique (par exemple, un flux d'octets) dans un codage de caractère non-Unicode connu, convertir l'IRI en une séquence de caractères tirés de l'UCS normalisés conformément à NFC.

c. Si l'IRI est un codage de caractères fondé sur Unicode (par exemple, UTF-8 ou UTF-16), ne pas normaliser (voir les détails au paragraphe 5.3.2.2). Appliquer directement l'étape 2 à la séquence de caractères Unicode codée.

étape 2. Pour chaque caractère en 'ucschar' ou 'iprivate', appliquer les étapes 2.1 à 2.3 ci-dessous.

2.1. Convertir le caractère en une séquence de un ou plusieurs octets en utilisant UTF-8 [RFC3629].

2.2. Convertir chaque octet en %HH, où HH est la notation hexadécimale de la valeur de l'octet. Noter que ceci est identique au mécanisme de codage en pourcentage du paragraphe de la [RFC3986]. Pour réduire la variance, la notation hexadécimale DEVRAIT utiliser des lettres majuscules.

2.3. Remplacer le caractère d'origine par la séquence de caractères résultante (c'est-à-dire une séquence de triplets %HH).

La transposition ci-dessus d'IRI en URI produit des URI parfaitement conformes à la [RFC3986]. La transposition est aussi une transformation d'identité pour les URI et est idempotente ; appliquer la transposition une seconde fois n'y changera rien. Chaque URI est par définition un IRI.

Les systèmes qui acceptent les IRI PEUVENT convertir le composant ireg-name d'un IRI comme suit (avant l'étape 2 ci-dessus) pour les schémas connus pour utiliser les noms de domaine en ireg-name, si la définition de schéma ne permet pas le codage en pourcentage pour ireg-name :

Remplacer la partie ireg-name de l'IRI par la partie convertie en utilisant l'opération ToASCII spécifiée au paragraphe 4.1 de la [RFC3490] sur chaque étiquette séparée par un point, et en utilisant U+002E (POINT) comme étiquette de séparation, avec le fanion UseSTD3ASCIIRules mis à VRAI, et le fanion AllowUnassigned mis à FAUX pour créer des IRI et mis à VRAI autrement.

L'opération ToASCII peut échouer, mais cela voudrait dire que l'IRI ne peut être résout. Cette conversion DEVRAIT être utilisée lorsque le but est de maximiser l'interopérabilité avec les solveurs d'URI habituels. Par exemple, l'IRI http://résumé.example.org peut être converti en "http://xn--rsum-bpad.example.org" au lieu de "http://r%C3%A9sum%C3%A9.example.org".

Un IRI avec un schéma connu pour utiliser les noms de domaine en ireg-name, mais dont la définition de schéma ne permet pas le codage en pourcentage pour ireg-name, satisfait aux restrictions spécifiques du schéma si la conversion directe ou la conversion utilisant l'opération ToASCII sur l'ireg-name résulte en un URI qui satisfait aux restrictions spécifiques du schéma.

Un tel IRI se résout en l'URI obtenu après conversion de l'IRI et utilise l'opération ToASCII sur le ireg-name. Les mises en oeuvre n'ont pas à faire cette conversion du moment qu'elles produisent le même résultat.

Note : La différence entre les variantes b et c de l'étape 1 (en utilisant la normalisation avec NFC, l'autre n'utilisant aucune normalisation) tient au fait que dans de nombreux codages de caractères non-Unicode, certain texte ne peut pas être représenté directement. Par exemple, le mot "Vietnam" est dans son pays écrit "Việt Nam" (contenant une LETTRE LATINE MINUSCULE E AVEC ACCENT CIRCONFLEXE ET POINT EN-DESSOUS) en NFC, mais un transcodage direct à partir du codage de caractère windows-1258 conduit à "Việt Nam" (contenant une LETTRE LATINE MINUSCULE E AVEC ACCENT CIRCONFLEXE suivie par un POINT EN COMBINAISON EN DESSOUS). Le transcodage direct d'autres codages à 8 bits de vietnamien peut conduire à d'autres représentations.

Note : Le traitement uniforme de tout l'IRI dans l'étape 2 est important pou r rendre le traitement indépendant du schéma d'URI. Voir [Gettys] pour une discussion en profondeur.

Note : En pratique, que la transposition générale (étapes 1 et 2) ou l'opération ToASCII de la [RFC3490] soit utilisée pour l'ireg-name ne sera pas remarqué si la transposition d'IRI en URI et la résolution sont étroitement intégrées (par exemple, portées dans le même agent d'utilisateur). Mais la conversion en utilisant [RFC3490] peut apporter un meilleur traitement pour ce qui concerne les questions de compatibilité amont dans le cas où transposition et résolution sont séparées, comme dans le cas d'utilisation d'un mandataire HTTP.

Note : Les noms de domaine internationalisés peuvent être contenus dans des parties d'un IRI autres que la partie ireg-name. Il est du ressort des mises en œuvre spécifiques de schéma (si le nom de domaine Internationalisé fait partie de la syntaxe du schéma) ou si les mises en œuvre du côté serveur (si le nom de domaine internationalisé fait partie de 'iquery') d'appliquer les conversions nécessaires au point approprié. Exemple : essayer de valider la page Web à http://résumé.example.org conduirait à un IRI de http://validator.w3.org/check?uri=http%3A%2F%2Frésumé.example.org, qui se convertirait en un URI de http://validator.w3.org/check?uri=http%3A%2F%2Fr%C3%A9sum%C3%A9.example.org. La mise en oeuvre côté serveur aurait à effectuer les conversions nécessaires pour restaurer la page Web.

Les systèmes qui acceptent les IRI PEUVENT aussi s'accommoder des caractères imprimables en US-ASCII qui ne sont pas admis dans les URI, à savoir "<", ">", "", espace, "{", "}", "|", "\", "^", et "`", dans l'étape 2 ci-dessus. Si ces caractères sont rencontrés mais ne sont pas convertis, la conversion DEVRAIT alors échouer. Prière de noter que les caractères dièse ("#"),pourcent ("%"), et les crochets ("[", "]") ne font pas partie de la liste ci-dessus et NE DOIVENT PAS être convertis. Les protocoles et formats qui ont utilisé des définitions plus anciennes d'IRI qui incluent ces caractères PEUVENT avoir besoin des codages en

pourcentage de ces caractères comme étape de prétraitement pour extraire l'IRI réel à partir d'un champ donné. Ce prétraitement PEUT aussi être utilisé par des applications qui permettent à l'utilisateur d'entrer dans un IRI.

Note : Dans ce processus (à l'étape 2.3), les caractères permis dans les références d'URI et les séquences existantes codées en pourcentage ne sont pas autrement codés. (Cette transposition est similaire, tout en restant différente, du codage appliqué lorsqu'un contenu arbitraire est inclus dans une partie d'un URI.) Par exemple, un IRI de "http://www.example.org/red%09rosé#red" (en notation XML) est converti en "http://www.example.org/red%09ros%C3%A9#red", et non pas en quelque chose comme "http%3A%2F%2Fwww.example.org%2Fred%2509ros%C3%A9%23red".

Note : Certains anciens logiciels de transcodage en UTF-8 peuvent produire des résultats illégaux pour certaines entrées, en particulier pour les caractères qui sont en dehors du BMP (*Basic Multilingual Plane*, plan de base multilingue). Par exemple, comme IRI avec des caractères non-BMP (en notation XML) : "http://example.com/𐌀𐌁&#x10302"; qui contient les trois premières lettres de l'ancien alphabet italique, la conversion correcte en URI est "http://example.com/%F0%90%8C%80%F0%90%8C%81%F0%90%8C%82"

3.2. Conversion des URI en IRI

Dans certaines situations, il peut être souhaitable de convertir un URI en un IRI équivalent. Une procédure de conversion est donnée dans ce paragraphe. La conversion décrite va toujours résulter en un IRI qui correspond à l'URI utilisé en entrée pour la conversion (excepté des différences potentielles pour des cas de codages en pourcentage et de caractères non réservés qui peuvent être codés en pourcentage). Cependant, l'IRI résultant de cette conversion peut n'être pas exactement le même que l'IRI d'origine (s'il y en a jamais eu un).

La conversion d'URI en IRI supprime les codages en pourcentage, mais tous les codages en pourcentage ne peuvent être éliminés. Il y a à cela plusieurs raisons :

1. Certains codages en pourcentage sont nécessaires pour distinguer les utilisations codées en pourcentage de caractères réservés de celles qui ne le sont pas.

2. Certains codages en pourcentage ne peuvent pas être interprétés comme des séquences d'octets en UTF-8.

(Note : Le schéma d'octet de l'UTF-8 est très régulier. Il y a donc une très forte probabilité, mais pas de certitude, que les codages en pourcentage puissent être interprétés comme des séquences d'octets en UTF-8 effectivement originaires de l'UTF-8. Pour un examen détaillé, voir [Duerst97].)

3. La conversion peut avoir pour résultat qu'un caractère ne soit pas approprié dans un IRI. Voir les paragraphes 2.2, 4.1, et 6.1 pour des détails complémentaires.

La conversion d'un URI en IRI est faite selon les étapes suivantes (ou tout autre algorithme qui produit le même résultat) :

1. Représenter l'URI comme une séquence d'octets en US-ASCII.

2. Convertir tous les codages en pourcentage ("%" suivi de deux chiffres hexadécimaux) en octets correspondants, excepté ceux correspondant à "%", les caractères en "réservé", et les caractères en US-ASCII non admis dans les URI.

3. Remettre en codage en pourcentage tout octet produit à l'étape 2 qui ne fait pas partie d'une séquence d'octet UTF-8 strictement légale.

4. Remettre en codage en pourcentage tous les octets produits à l'étape 3 qui représentent en UTF-8 des caractères qui ne sont pas appropriés conformément aux paragraphes 2.2, 4.1, et 6.1.

5. Interpréter la séquence d'octets résultante comme une séquence de caractères codés en UTF-8.

Cette procédure va convertir autant de caractères codés en pourcentage que possible en caractères dans un IRI. Comme il y a plusieurs choix lors de l'application de l'étape 4 (voir au paragraphe 6.1), les résultats peuvent varier.

Les conversions d'URI en IRI NE DOIVENT PAS utiliser de codage de caractères autre que l' UTF-8 dans les étapes 3 et 4, même si il est possible de déduire du contexte qu'un autre codage de caractères que l'UTF-8 a été utilisé dans l'URI. Par exemple, l'URI "http://www.example.org/r%E9sum%E9.html" peut avec quelques hypothèses être interprété comme contenant deux caractères e accent aigu codés selon iso-8859-1. Il ne doivent pas être convertis en un IRI contenant ces caractères e accent aigu. Sinon, le futur IRI sera transposé en "http://www.example.org/r%C3%A9sum%C3%A9.html", qui est un URI différent de "http://www.example.org/r%E9sum%E9.html".

3.2.1. Exemples

Plusieurs exemples de conversion d'URI en IRI sont donnés dans ce paragraphe. Chaque exemple montre le résultat après l'application de chacune des étapes 1 à 5. La notation XML est utilisée pour le résultat final. Les octets sont notés par "<" suivi par deux chiffres hexadécimaux, suivis de ">".

L'exemple suivant contient la séquence "%C3%BC", qui est une séquence UTF-8 strictement légale, et qui est convertie qui est le caractère réel U+00FC, LETTRE LATINE MINUSCULE U TREMA (aussi appelé u-umlaut).

1. http://www.example.org/D%C3%BCrst

2. http://www.example.org/D<c3><bc>rst

3. http://www.example.org/D<c3><bc>rst

4. http://www.example.org/D<c3><bc>rst

5. http://www.example.org/Dürst

L'exemple suivant contient la séquence "%FC", qui peut représenter U+00FC, LETTRE LATINE MINUSCULE U TREMA, dans le codage de caractère iso-8859-1. (Elle peut représenter d'autres caractères dans d'autres codages de caractère. Par exemple, l'octet <fc> en iso-8859-5 représente U+045C, LETTRE CYRILLIQUE MINUSCULE KJE.) Comme <fc> ne fait pas partie d'une séquence UTF-8 strictement légale, il est recodé en pourcentage à l'étape 3.

1. http://www.example.org/D%FCrst

2. http://www.example.org/D<fc>rst

3. http://www.example.org/D%FCrst

4. http://www.example.org/D%FCrst

5. http://www.example.org/D%FCrst

L'exemple suivant contient "%e2%80%ae", qui est le codage de caractère UTF-8 codé en pourcentage de U+202E, LECTURE DE DROITE A GAUCHE. Le paragraphe 4.1 interdit l'utilisation directe de ce caractère dans un IRI. Donc, les octets correspondants sont recodés en pourcentage à l'étape 4. Cet exemple montre que la casse (majuscules ou minuscules) des lettres utilisées dans les codages en pourcentage peuvent n'être pas préservés. L'exemple contient aussi une étiquette de nom de domaine codée en punycode (xn--99zt52a), qui n'est pas convertie.

1. http://xn--99zt52a.example.org/%e2%80%ae

2. http://xn--99zt52a.example.org/<e2><80><ae>

3. http://xn--99zt52a.example.org/<e2><80><ae>

4. http://xn--99zt52a.example.org/%E2%80%AE

5. http://xn--99zt52a.example.org/%E2%80%AE

Les mises en œuvre qui ont une connaissance spécifique du schéma PEUVENT convertir les étiquettes de nom de domaine codées en punycode dans les caractères correspondants en utilisant la procédure ToUnicode. Et donc, pour l'exemple ci-dessus, l'étiquette "xn--99zt52a" peut être convertie en U+7D0D U+8C46 (Natto japonais), ce qui donne pour l'ensemble de l'IRI : "http://納豆.example.org/%E2%80%AE".

4. IRI bidirectionnels pour langages de droite à gauche

Certains caractères UCS, tels que ceux utilisés dans les écritures arabe et hébreu, ont un sens d'écriture inhérent de droite à gauche (rtl). Les IRI qui contiennent ces caractères (qu'on appelle IRI bidirectionnels IRI ou IRI Bidi) nécessitent une attention particulière à cause des rapports non triviaux entre la représentation logique (utilisée pour la représentation numérique et pour l'écriture/lecture) et la représentation visuelle (utilisée pour l'affichage/impression).

A cause de l' interaction complexe entre représentation logique, représentation visuelle , et syntaxe d'un IRI Bidi, il est nécessaire de faire l'équilibre entre les différentes exigences. Les principales sont :

1. Une conversion prévisible par l'utilisateur entre les représentations logique et visuelle ;

2. La capacité à inclure une large gamme de caractères dans les diverses parties de l'IRI ; et

3. Des changements mineurs ou pas de changement ou des restrictions pour les mises en oeuvre.

4.1. Mémorisation logique et présentation visuelle

Lorsqu'ils sont mémorisés ou transmis en représentation numérique, les IRI bidirectionnels DOIVENT être en ordre logique et DOIVENT se conformer aux règles de la syntaxe d'IRI (qui incluent les règles pertinentes pour leur schéma). Ceci garantit que les IRI bidirectionnels peuvent être traités de la même façon que les autres IRI.

Les IRI bidirectionnels DOIVENT être rendus en utilisant l'algorithme Unicode bidirectionnel [UNIV4], [UNI9]. Les IRI bidirectionnels DOIVENT être rendus de la même façon que s'ils étaient dans une disposition de gauche à droite ; c'est-à-dire, comme si ils étaient précédés par U+202A, LEFT-TO-RIGHT EMBEDDING (LRE), et suivis de U+202C, POP DIRECTIONAL FORMATTING (PDF). On peut aussi régler la direction de disposition au moyen d'un protocole de plus haut niveau (par exemple, l'attribut dir='ltr' en HTML).

Il n'est pas exigé d'utiliser la disposition mentionnée ci-dessus si l'affichage reste le même sans cette disposition. Par exemple, un IRI bidirectionnel dans un texte dont la direction de base est de gauche à droite (comme utilisée par l'anglais ou le cyrillique) qui est précédé et suivi par un espace blanc et des caractères de gauche à droite forts n'a pas besoin d'être enchâssé. De même une référence croisée d'IRI qui ne contient que des caractères forts de droite à gauche et des caractères faibles et qui commence et se termine par un caractère fort de droite à gauche et apparaît dans un texte dont la direction de base est de droite à gauche (comme par exemple en arabe ou en hébreu) et est précédé et suivi d'un espace blanc et de caractères forts, n'a pas besoin d'être enchâssé. Dans certains autres cas, l'utilisation de U+200E, MARQUE DE GAUCHE A DROITE (LRM), peut être suffisant pour forcer le comportement d'affichage correct. Cependant, les détails de l'algorithme bidirectionnel Unicode ne sont pas toujours faciles à comprendre. Il est fortement conseillé aux auteurs de mises en œuvre de rester extrêmement prudents et d'utiliser l'enchâssement dans tous les cas où il ne sont pas complètement sûrs que le comportement d'affichage sera inchangé sans enchâssement.

Le paragraphe 4.3 de l'Algorithme bidirectionnel Unicode ([UNI9] permet à des protocole de niveau

supérieur d'influencer l'effet bidirectionnel. De tels changements par des protocoles de niveau supérieur NE DOIVENT PAS être utilisés si ils changent la façon de rendre les IRI.

Les caractères de formatage bidirectionnel qui peuvent être utilisés avant ou après l'IRI pour assurer l'affichage correct ne font pas eux-mêmes partie de l'IRI. Les IRI NE DOIVENT PAS contenir de caractères de formatage bidirectionnel (LRM, RLM, LRE, RLE, LRO, RLO, et PDF). Ils affectent le rendu visuel de l'IRI mais n'apparaissent pas eux-mêmes. Il ne serait donc pas possible d'entrer correctement un IRI avec de tels caractères.

4.2. Structure d'IRI en Bidi

L'algorithme bidirectionnel Unicode a été conçu principalement pour traiter du texte. Pour s'assurer qu'il n'affecte pas trop le rendu des IRI bidirectionnels, quelques restrictions sont nécessaires sur les IRI bidirectionnels. Ces restrictions sont données par des délimiteurs (caractères structurels, principalement de ponctuation tels que "@", ".", ":", et "/") et composants (consistant habituellement en lettres et chiffres).

Les règles de syntaxe suivantes tirées du paragraphe 2.2 correspondent aux composants pour les besoins du comportement Bidi: iuserinfo, ireg-name, isegment, isegment-nz, isegment-nz-nc, ireg-name, iquery, et ifragment.

Les spécifications qui définissent la syntaxe de tous les composants ci-dessus PEUVENT les subdiviser et définir de plus petites parties comme étant des composants conformément au présent document. Par exemple, les restrictions de [RFC3490] sur les noms de domaine bidirectionnels équivalent à traiter chaque étiquette d'un nom de domaine comme un composant, pour les schémas qui ont ireg-name comme nom de domaine. Même lorsque les composants ne sont pas définis formellement, il peut être utile de considérer la syntaxe en termes de composants et d'appliquer les restrictions pertinentes. Par exemple, pour la syntaxe habituelle nom/valeur dans les parties d'interrogation, il est pratique de traiter chaque nom et chaque valeur comme un composant. Comme autre exemple, les extensions dans un nom de ressource peuvent être traitées comme des composants séparés.

Pour chaque composant, les restrictions suivantes s'appliquent :

1. Un composant NE DEVRAIT PAS utiliser à la fois des caractères de droite à gauche et des caractères de gauche à droite.

2. Un composant qui utilise des caractères de droite à gauche DEVRAIT commencer et se terminer par des caractères de droite à gauche.

Les restrictions ci-dessus sont données comme des conseil plus que des obligations.

Pour les IRI qui ne sont jamais présentés visuellement, elles ne sont pas pertinentes. Cependant, pour les IRI en général, elles sont très importantes pour assurer une conversion cohérente entre la présentation visuelle et la représentation logique, dans les deux sens.

Note : dans certains composants, les restrictions ci-dessus peut réellement être misses en application de façon stricte. Par exemple, la [RFC3490] demande qu'on applique ces restrictions aux étiquettes d'un nom d'hôte pour les schémas dans lesquels ireg-name est un nom d'hôte. Dans certains autres composants (par exemple, les composants de chemin) suivre ces restrictions peut n'être pas trop difficile. Pour d'autres composants, comme les parties de l'interrogation, il peut être très difficile d'appliquer les restrictions parce que les valeurs des paramètres d'interrogation peuvent être des séquences arbitraires de caractères.

Si les restrictions ci-dessus ne peuvent être satisfaites autrement, le composant affecté peut toujours être transposé en notation d'URI comme décrit au paragraphe 3.1. Noter que c'est tout le composant qui doit être transposé (voir aussi l'Exemple 9 ci-dessous).

4.3. Entrées d’IRI en Bidi

Les méthodes d’entrée de Bidi DOIVENT générer des IRI en Bidi en ordre logique tout en les rendant conformément au paragraphe 4.1. Pendant l’entrée, le rendu DEVRAIT être mis à jour après l’entrée de chaque nouveau caractère pour éviter toute confusion chez l’utilisateur final.

4.4. Exemples

Ce paragraphe donne des exemples d’IRI bidirectionnels, en notation Bidi. Il montre des IRI légaux avec les relations entre les représentations logique et visuelle et explique comment certains phénomènes de ces relations peuvent paraître étranges à quelqu’un qui n’est pas familier du comportement bidirectionnel, mais normales pour les utilisateurs de l’arabe et de l’hébreu. Il montre aussi ce qui arrive si les restrictions données au paragraphe 4.2 ne sont pas suivies. Les exemples ci-dessous peuvent être examinés à [BidiEx], dans les variantes arabe, hébreu, et notation Bidi.
Pour lire le texte bidi dans les exemples, lire la représentation visuelle de gauche à droite jusqu’à rencontrer un bloc de texte rtl. Lire le bloc rtl (y compris les barres obliques et autres caractères spéciaux) de droite à gauche, puis continuer au prochain caractère ltr non lu.

Exemple 1 : Un seul composant à caractères rtl est inversé :

Représentation logique : "http://ab.CDEFGH.ij/kl/mn/op.html"

Représentation visuelle : "http://ab.HGFEDC.ij/kl/mn/op.html"

Les composants peuvent être lus un par un, et chaque composant peut être lu dans sa direction naturelle.

Exemple 2 : Plus d’un composant consécutif avec des caractères rtl est inversé comme un tout :

Représentation logique : "http://ab.CDE.FGH/ij/kl/mn/op.html"

Représentation visuelle : "http://ab.HGF.EDC/ij/kl/mn/op.html"

Une séquence de composants rtl est lue rtl, de la même façon qu’une séquence de mots rtl est lue rtl dans un texte bidi.

Exemple 3 : Tous les composants d’un IRI (excepté le schéma) sont rtl. Tous les composants rtl sont inversés globalement :

Représentation logique : "http://AB.CD.EF/GH/IJ/KL?MN=OP;QR=ST#UV"

Représentation visuelle : "http://VU#TS=RQ;PO=NM?LK/JI/HG/FE.DC.BA"

Tout l’IRI (excepté le schéma) est lu rtl. Les délimiteurs entre les composants rtl restent entre les composants respectifs ; les délimiteurs entre composants ltr et rtl ne bougent pas.

Exemple 4 : Chacune des différentes séquences des composants rtl est inversée séparément :

Représentation logique : "http://AB.CD.ef/gh/IJ/KL.html"

Représentation visuelle : "http://DC.BA.ef/gh/LK/JI.html"

Chaque séquence de composants rtl est lue rtl, de la même façon que chaque séquence de mots rtl dans un texte ltr est lue rtl.

Exemple 5 : C’est l’exemple 2, appliqué aux composants de différentes sortes :

Représentation logique : "http://ab.cd.EF/GH/ij/kl.html"

Représentation visuelle : "http://ab.cd.HG/FE/ij/kl.html"

L’inversion de l’étiquette de nom de domaine et du composant de chemin peut être inattendue, mais elle est cohérente avec le comportement bidi. Pour s’assurer que le composant de domaine est réellement "ab.cd.EF", il peut être utile de lire à voix haute la représentation visuelle qui suit l’algorithme bidi. Après "http://ab.cd." on lit le bloc rtl "E-F-slash-G-H", qui correspond à la représentation logique.

Exemple 6 : Comme l’exemple 5, avec plus de composants rtl :

Représentation logique : "http://ab.CD.EF/GH/IJ/kl.html"

Représentation visuelle : "http://ab.JI/HG/FE.DC/kl.html"

L’inversion des étiquettes de nom de domaine et des composants de chemin peut être plus facile à identifier parce que les délimiteurs bougent aussi.

Exemple 7 : Un seul composant rtl inclut des chiffres:

Représentation logique : "http://ab.CDE123FGH.ij/kl/mn/op.html"

Représentation visuelle : "http://ab.HGF123EDC.ij/kl/mn/op.html"

Les nombres sont écrits ltr dans tous les cas mais sont traités comme un enchâssement supplémentaire au sein d’une volée de caractères rtl. Ceci est parfaitement cohérent avec le texte bidirectionnel habituel.

Exemple 8 (non permis) : Nombres qui sont au début ou à la fin d’un composant rtl :

Représentation logique : "http://ab.cd.ef/GH1/2IJ/KL.html"

Représentation visuelle : "http://ab.cd.ef/LK/JI1/2HG.html"

La séquence "1/2" est interprétée par l’algorithme bidi comme une fraction, qui fragmente les composants et amène la confusion. Il y a d’autres caractères qui sont interprétés d’une façon particulière à proximité des nombres; en particulier, "+", "-", "#", "$", "%", ",", ".", et ":".

Exemple 9 (non permis) : Les nombres de l’exemple précédent sont codés en pourcentage :
Représentation logique : "http://ab.cd.ef/GH%31/%32IJ/KL.html",
Représentation visuelle (hébreu) : "http://ab.cd.ef/%31HG/LK/JI%32.html"
Représentation visuelle (arabe) : "http://ab.cd.ef/31%HG/%LK/JI32.html"
Selon que les lettres majuscules représentent l’arabe ou l’hébreu, la représentation visuelle est différente.

Exemple 10 (admis mais pas recommandé) :
Représentation logique : "http://ab.CDEFGH.123/kl/mn/op.html"
Représentation visuelle : "http://ab.123.HGFEDC/kl/mn/op.html"
Les composants consistant seulement en nombres sont admis (il serait assez difficile des les interdire), mais ils peuvent interagir avec des composants rtl adjacents de façon difficilement prévisible.

5. Normalisation et comparaison

Note : La structure et la plus grande partie du contenu de la présente section sont tirés de la section 6 de la [RFC3986] ; les différences proviennent des spécificités des IRI.

Une des opérations les plus courantes sur les IRI est la simple comparaison : déterminer si deux IRI sont équivalents sans utiliser les IRI ou l’URI transposé pour accéder à leur ressources respectives. Une comparaison est effectuée chaque fois qu’on accède à une mémoire cache de réponse, un navigateur vérifie son historique pour retenir une liaison, ou un analyseur XML traite les étiquettes au sein d’un espace de nom. Une normalisation intense peut être faite par les serveurs et moteurs de recherche avant de comparer les IRI pour élaguer l’espace de recherche ou réduire les duplications d’actions de demande et de stockage de réponses.

La comparaison d’IRI est effectuée dans des buts précis. Les protocoles ou mises en œuvre qui comparent les IRI pour différents objets font souvent l’objet de divergences de conception quant à l’évaluation de la

quantité d'efforts qui doivent être déployés pour réduire les noms d'emprunt d'identifiants. La présente section décrit diverses méthodes qui peuvent être utilisées pour comparer les IRI, les échanges entre eux, et les types d'applications qui pourraient les utiliser.

5.1. Equivalence

Comme les IRI existent pour identifier des ressources, on peut penser qu'ils devraient être considérés comme équivalents lorsqu'ils identifient la même ressource. Cependant, cette définition d'équivalence n'est pas d'une grande utilité pratique, car une mise en œuvre n'a aucun moyen de comparer deux ressources si elle n'a pas une connaissance et un contrôle complet sur elles. Pour cette raison, la détermination de l'équivalence ou de la différence des IRI se fonde sur la comparaison des chaînes, qui peut être améliorée par la référence à des règles supplémentaires fournies par les définitions de schéma d'URI. Les termes "différent" et "équivalent" sont utilisés pour décrire les résultats possibles de telles comparaisons, mais il y a de nombreuses versions dépendantes de l'application de la notion d'équivalence.

Même s'il est possible de déterminer que deux IRI sont équivalents, la comparaison d'URI n'est pas suffisante pour déterminer si deux IRI identifient des ressources différentes. Par exemple, le propriétaire de deux noms de domaine différents peut décider de desservir la même ressource à partir des deux, ce qui donnera deux IRI différents. Donc, les méthodes de comparaison sont conçues pour minimiser les faux négatifs tout en évitant strictement les faux positifs.

Dans les essais d'équivalence, les applications ne devraient pas comparer directement les références croisées ; les références devraient être converties en leur IRI cible respectif avant la comparaison. Lorsque les IRI sont comparés pour choisir (ou éviter) une action de réseau, comme la restitution d'une représentation, les composants de fragment (s'il en est) devraient être exclus de la comparaison.

Les applications qui utilisent les IRI comme des jetons d'identité sans relation avec un protocole DOIVENT utiliser la Comparaison de chaîne simple (voir au paragraphe section 5.3.1). Toutes les autres applications DOIVENT choisir une des pratiques de comparaison de l'échelle de comparaison (voir au paragraphe 5.3 ou, après la conversion d'IRI en URI, choisir une des pratiques de comparaison d'après l'échelle de comparaison d'URI de [RFC3986], paragraphe 6.2)

5.2. Séparation pour comparaison

Chaque type de comparaison d'IRI EXIGE que tous les échappements ou codages dans le protocole ou format qui portent un IRI soient résolus. Cela est fait habituellement lorsque le protocole ou format est analysé grammaticalement. Des exemples de tels échappements ou codages sont des entités et des références de caractère numérique en [HTML4] et [XML1]. Par exemple, "http://example.org/rosé" (en HTML), "http://example.org/ros&#233"; (en HTML ou XML), et "http://example.org/ros&#xE9"; (en HTML ou XML) sont tous résolus en ce qui est noté dans le présent document (voir au paragraphe 1.4) sous la forme "http://example.org/ros&#xE9"; (le "é" signifiant ici le caractère e accent aigu, pour compenser le fait que ce document ne peut pas contenir de caractères non-ASCII).

Des considérations similaires s'appliquent aux codages tels que les codages de transfert en HTTP (voir [RFC2616]) et les codages de transfert de contenu en MIME ([RFC2045]), bien que dans ces cas, le codage soit fondé non sur les caractères mais sur les octets, et il faut bien faire attention à comparer les caractères et pas simplement des octets arbitraires (voir au paragraphe 5.3.1).

5.3. Echelle de comparaison

Diverses méthodes sont utilisées en pratique pour vérifier l'équivalence d'URI. Ces méthodes se classent en distinguant selon la quantité de traitement nécessaire et le degré de réduction de la probabilité de faux négatif. Comme noté ci-dessus, les faux négatifs ne peuvent être éliminés. En pratique, leur probabilité peut être réduite, mais cette réduction exige plus de traitement et n'est pas rentable pour toutes les applications.

Si cette gamme de pratiques de comparaison est considérée comme une échelle, l'examen suivant va remonter l'échelle, en commençant par les pratiques qui sont peu coûteuses mais ont une chance relativement grande de produire des faux négatifs, et en remontant jusqu'à celles qui ont un coût de

traitement plus élevé et un moindre risque de faux négatifs.

5.3.1. Comparaison de chaîne simple

Si deux URI, considérés comme chaînes de caractères, sont identiques, on peut conclure en toute sécurité qu'ils sont équivalents. Ce type d'essai d'équivalence a un très faible coût de traitement et est d'un large usage dans des applications très diverses, en particulier dans le domaine de l'analyse grammaticale. Il est aussi utilisé lorsqu'on a besoin d'une réponse définitive à la question de l'équivalence d'IRI indépendamment du schéma utilisé qu'on puisse calculer vite et sans accéder à un réseau. L'exemple d'un tel cas est celui des espaces de nom XML ([XMLNamespace]).

Les chaînes d'essai d'équivalence requièrent quelques précautions de base. Cette procédure est souvent désignée sous le nom de comparaison "bit à bit" ou "octet par octet", ce qui peut induire en erreur. L'essai des chaînes pour vérifier leur égalité se fonde normalement sur la comparaison paire par paire des caractères qui composent les chaînes, en commençant par la première et en continuant jusqu'à épuisement des deux chaînes et que tous les caractères aient été trouvés égaux, jusqu'à ce qu'une paire de caractères se révèle inégale ou qu'une des chaînes arrive à épuisement avant l'autre.

Cette comparaison de caractères exige que chaque paire de caractères puisse être mise dans une forme comparable. Par exemple, si on a un IRI stocké sur une matrice binaire en codage UTF-8 et le second dans un codage UTF-16, les comparaisons bit à bit appliquées de façon ingénue produiront des erreurs. Il vaut mieux parler d'égalité caractère par caractère plutôt que bit à bit ou octet par octet. Pour parler de façon concrète, les comparaisons caractère par caractère devraient être faites point de code par point de code après conversion à un codage de caractères commun. Dans une comparaison caractère par caractère, la fonction de comparaison NE DOIT PAS transposer les IRI en URI, parce qu'une telle transposition créerait des équivalences parasites supplémentaires. Il en découle qu'un IRI NE DEVRAIT PAS être modifié lors d'un transport s'il y a la moindre chance qu'il soit utilisé comme identifiant.

Les faux négatifs sont causés par la production et l'utilisation de noms d'emprunt d'IRI. Les noms d'emprunt non nécessaires peuvent être réduits, indépendamment de la méthode de comparaison, en fournissant de façon cohérente les références d'IRI sous une forme déjà normalisée (c'est-à-dire, une forme identique à ce qui serait produit après application de la normalisation, comme décrit ci-dessous). Les protocoles et formats de données limitent souvent certaines comparaisons d'URI à une simple comparaison de chaînes, fondée sur la théorie que les gens et les mises en oeuvre vont, dans leur propre intérêt, être cohérents dans la fourniture des références d'URI, ou au minimum, assez cohérents pour annihiler toute efficacité qui pourrait être obtenue d'une normalisation ultérieure.

5.3.2. Normalisation fondée sur la syntaxe

Les mises en œuvre peuvent utiliser une logique fondée sur les définitions données par la présente spécification pour réduire la probabilité de faux négatifs. Ce traitement est d'un coût modérément plus élevé que la comparaison de chaînes caractère par caractère. Par exemple, une application utilisant cette approche pourrait raisonnablement considérer les deux IRI suivants comme équivalents :

example://a/b/c/%7Bfoo%7D/rosé
eXAMPLE://a/./b/../b/%63/%7bfoo%7d/ros%C3%A9

Les agents d'usager du Web, comme les navigateurs, appliquent normalement ce type de normalisation d'URI pour déterminer si une réponse est disponible en mémoire cache. La normalisation fondée sur la syntaxe inclut des techniques comme la normalisation de casse, la normalisation de codage en pourcentage, et le retrait des segments point.

5.3.2.1. Normalisation de la casse

Pour tous les IRI, les chiffres hexadécimaux au sein d'un triplet codé en pourcentage (par exemple, "%3a" contre "%3A") sont insensibles à la casse et devraient donc être normalisés à l'utilisation des majuscules pour les chiffres A à F.

Lorsqu'un IRI utilise des composants de la syntaxe générique, les règles d'équivalence syntaxique du

composant s’appliquent toujours ; à savoir, que le schéma et l’hôte en US-ASCII seul sont insensibles à la casse et donc devraient être normalisés en minuscules. Par exemple, l’URI <HTTP://www.EXAMPLE.com/> est équivalent à <http://www.example.com/>.
L’équivalence de la casse pour les caractères non-ASCII dans les composants d’IRI qui sont des IDN est discutée au paragraphe 5.3.3. Les autres composants de syntaxe générique sont supposés être sensibles à la casse à moins qu’il n’en soit spécifiquement décidé autrement par le schéma.

La création de schémas qui permettent des composants de syntaxe insensibles à la casse contenant des caractères non ASCII devrait être évitée. La normalisation de la casse de caractères non ASCII peut dépendre de la culture et est toujours une opération complexe. La seule exception concerne les noms d’hôte non ASCII pour lesquels la normalisation de caractères inclut une étape de transposition dérivée du développement de la casse.

5.3.2.2. Normalisation des caractères

La norme Unicode [UNIV4] définit diverses équivalences entre les séquences de caractères pour divers objets. L’annexe 15 de la norme Unicode [UTR15] définit plusieurs formes de normalisation pour ces équivalences, en particulier la forme de normalisation C (NFC, décomposition canonique, suivie par la composition canonique) et la forme de normalisation KC (NFKC, décomposition de compatibilité, suivie par la composition canonique).

L’équivalence des IRI DOIT s’appuyer sur l’hypothèse que les IRI sont pré normalisés en caractères de façon appropriée plutôt que d’appliquer la normalisation de caractères lors de la comparaison de deux IRI. Les exceptions sont la conversion partir d’une forme non numérique, et la conversion à partir d’un codage de caractères non fondés sur UCS en codage de caractères fondés sur UCS. Dans ces cas, NFC ou un transcodeur de normalisation utilisant NFC DOIT être utilisé pour assurer l’interopérabilité. Pour éviter les faux négatifs et les problèmes de transcodage, les IRI DEVRAIENT être créés en utilisant NFC. L’utilisation de NFKC peut même éviter encore plus de problèmes ; par exemple, en choisissant des lettres latines de demi largeur au lieu de lettres à pleine largeur, et de pleine largeur au lieu de Katakana en demi largeur.

Par exemple, "http://www.example.org/résumé.html" (en notation XML) est en NFC. D’un autre côté, "http://www.example.org/résumé.html" n’est pas en NFC.
Le premier utilise des caractères e accent aigu pré combinés, et le second utilise les caractères "e" suivis par des accents aigus combinants. Les deux usages sont définis comme canoniquement équivalents dans [UNIV4].

Note : Comme on ne sait pas comment une séquence particulière de caractères va être traitée en ce qui concerne la normalisation de caractères, il serait inapproprié de permettre à des tiers de normaliser un IRI de façon arbitraire. Cela ne contredit pas la recommandation de normaliser autant que possible les caractères d’un IRI lorsqu’une ressource est créée (c’est-à-dire, NFC ou même NFKC). Ceci est semblable aux problèmes de majuscules/minuscules. Certaines parties d’un URI sont insensibles à la casse (les noms de domaine). Pour d’autres, on ne sait pas clairement s’ils sont sensibles à la casse, insensibles à la casse, ou quelque chose entre les deux (par exemple, sensible à la casse, mais avec un choix multiple si la mauvaise casse est utilisée, au lieu d’un résultat négatif direct). La meilleure recette est que le créateur emploie les majuscules de façon raisonnable et que, lors du transfert de l’URI, cela ne soit jamais modifié.

Divers schéma d’IRI peuvent permettre l’usage des noms de domaines internationalisés (IDN) [RFC3490] soit dans la partie ireg-name soit ailleurs. La normalisation de caractères s’applique aussi aux IDN, comme examiné au paragraphe 5.3.3.

5.3.2.3. Normalisation du codage en pourcentage

Le mécanisme du codage en pourcentage (paragraphe 2.1 de [RFC3986]) est une source fréquente de variation entre des IRI identiques par ailleurs. En plus de la question de la normalisation de la casse mentionnée ci-dessus, certains producteurs d’IRI codent en pourcentage des octets qui ne le requièrent pas, provoquant l’équivalence des IRI avec des contreparties non codées. Ces IRI devraient être normalisés en décodant tout octet codé en pourcentage qui correspond à un caractère non réservé, comme décrit au paragraphe 2.3 de [RFC3986].

Pour la résolution réelle, des différences de codage en pourcentage (excepté pour le codage en pourcentage de caractères réservés) DOIVENT toujours déboucher sur la même ressource. Par exemple, "http://example.org/~user", "http://example.org/%7euser", et "http://example.org/%7Euser", doivent aboutir à la même ressource.

Si on doit tester cette sorte d'équivalence, le codage en pourcentage des deux IRI à comparer doit être aligné ; par exemple en convertissant les deux IRI en URI (voir au paragraphe 3.1), en éliminant les différences d'échappement dans les URI résultants, et en s'assurant que la casse des caractères hexadécimaux dans les codages en pourcentage est bien toujours la même (de préférence en majuscules). Si l'IRI doit être passé à une autre application ou utilisé plus tard d'une façon différente, sa forme originelle DOIT être préservée. La conversion décrite ici devrait être effectuée seulement pour des besoins de comparaison locale.

5.3.2.4. Normalisation du segment de chemin

Les segments chemin complets "." et ".." sont destinés à être uniquement utilisés dans des références croisées (paragraphe 4.1 de [RFC3986]) et sont retirés dans le cours du processus de résolution de la référence (paragraphe 5.21 de [RFC3986])). Cependant, certaines mises en œuvre développées de façon incorrecte supposent que la résolution de références n'est pas nécessaire lorsque la référence est déjà un IRI et omettent donc de retirer les segments point lorsqu'ils surviennent dans des chemins non relatifs. Les normaliseurs d'IRI devraient retirer les segments point en appliquant l'algorithme remove_dot_segments au chemin, comme décrit au paragraphe 5.2.4 de [RFC3986]).

5.3.3. Normalisation fondée sur le schéma

La syntaxe et la sémantique des IRI varie d'un schéma à l'autre, comme décrit par la spécification de définition de chaque schéma. Les mises en œuvre peuvent utiliser des règles spécifiques du schéma, pour un coût de traitement supérieur, pour réduire la probabilité de faux négatifs. Par exemple, puisque le schéma "http" utilise un composant authority, a un port par défaut de "80", et définit un chemin vide comme équivalent à "/", les quatre IRI suivants sont équivalents :

> http://example.com
> http://example.com/
> http://example.com:/
> http://example.com:80/

En général, un IRI qui utilise la syntaxe générique pour authority avec un chemin vide devrait être normalisé en un chemin de "/". De même, un ":port" explicite, pour lequel le port est vide ou la valeur par défaut pour le schéma, est équivalent à un IRI où le port et son délimiteur ":" sont élidés et devraient donc être retirés par la normalisation fondée sur le schéma. Par exemple, le second IRI ci-dessus est la forme normale pour le schéma "http".

Un autre cas où la normalisation varie selon le schéma est le traitement d'un composant authority vide ou d'un sous-composant hôte vide. Pour de nombreuses spécifications de schéma, une autorité ou hôte vide est considéré comme une erreur ; pour d'autres, c'est considéré comme équivalent à "localhost" ou à l'hôte de l'utilisateur final. Lorsqu'un schéma définit une valeur par défaut pour l'autorité et qu'une référence d'IRI à cette valeur par défaut est souhaitée, la référence devrait être normalisée à une autorité vide au nom de l'uniformité, de la concision, et de l'internationalisation. Si cependant, les sous-composants userinfo ou port sont non vides, l'hôte devrait être donné de façon explicite même s'il correspond à la valeur par défaut.

La normalisation ne devrait pas retirer les délimiteurs lorsque leur composant associé est vide, à moins que cela ne soit autorisé par la spécification de schéma. Par exemple, l'IRI "http://example.com/?" ne peut être supposé équivalent à aucun des exemples ci-dessus. De même, la présence ou l'absence de délimiteurs dans un sous-composant userinfo est normalement significative pour son interprétation. Le composant fragment n'est soumis à aucune normalisation fondée sur le schéma ; et donc, deux IRI qui diffèrent seulement par le suffixe "#" sont considérés comme différents quel que soit leur schéma.

Certains schémas d'IRI peuvent permettre l'usage des noms de domaine internationalisés (IDN) [RFC3490] soit dans leur partie ireg-name soit quelque part ailleurs. Lorsqu'ils sont utilisés dans les IRI, ces noms

DEVRAIENT être validés en utilisant l'opération ToASCII définie dans [RFC3490], avec les fanions "UseSTD3ASCIIRules" et "AllowUnassigned". Un IRI contenant un IDN invalide ne peut pas être bien résolu. Les composants IDN validés des IRI DEVRAIENT être normalisés en caractères en utilisant le processus Nameprep [RFC3491] ; cependant, pour des questions de lisibilité, ils NE DEVRAIENT PAS être convertis en codage compatible ASCII (ACE, *ASCII Compatible Encoding*).

La normalisation fondée sur le schéma peut aussi considérer les composants IDN et leurs conversions en punycode comme équivalents. Par exemple, "http://résumé.example.org" peut être considéré comme équivalent à "http://xn--rsum-bpad.example.org".

D'autres normalisations spécifiques du schéma sont possibles.

5.3.4. Normalisation fondée sur le protocole

Un effort substantiel de réduction de l'incidence des faux négatifs est souvent rentable pour les accélérateurs de web. Ils mettent donc en œuvre des techniques encore plus agressives pour la comparaison d'IRI. Par exemple, si ils observent qu'un IRI tel que http://example.com/data redirige sur un IRI qui en diffère seulement par la barre oblique de queue http://example.com/data/ ils vont vraisemblablement considérer à l'avenir que les deux sont équivalents. Cette sorte de technique n'est appropriée que lorsque l'équivalence est clairement indiquée à la fois par le résultat de l'accès aux ressources et les conventions communes de l'algorithme de déréférencement de leur schéma (dans ce cas, l'utilisation de la redirection par les serveurs HTTP d'origine pour éviter les problèmes de références croisées).

6. Utilisation des IRI

6.1. Limitations quant aux caractères UCS permis dans les IRI

La présente section discute des limitations sur les caractères et les séquences de caractère utilisables pour les IRI au-delà de ceux donnés au paragraphe 2.2 et au paragraphe 4.1. Les considérations de la présente section sont pertinentes lorsque les IRI sont créés et lorsque des URI sont convertis en IRI.

a. Le répertoire des caractères admis dans chaque composant d'IRI est limité par la définition de ce composant. Par exemple, la définition du composant de schéma ne permet pas de caractères en dehors de l'US-ASCII.

(Note : Conformément à la pratique des URI, le logiciel générique d'IRI ne peut pas et ne devrait pas vérifier de telles limitations.)

b. L'UCS contient de nombreuses zones de caractères pour lesquels il y a de forts faux-semblants visuels. A cause de la vraisemblance des erreurs de transcription, ceux-ci devraient aussi être évités. Cela inclut les équivalents de pleine largeur des caractères latins, les caractères Katakana en demi largeur pour le japonais, et de nombreux autres. Cela inclut aussi de nombreux faux-semblants de caractères "espace", "delims", et "différent de", exclus dans [RFC3491].

Des informations complémentaires sont disponibles sur [UNIXML]. [UNIXML] a été écrit dans un contexte de texte courant plutôt que dans celui des identifiants. Néanmoins, il traite de beaucoup des catégories de caractères qui ne sont pas appropriés pour les IRI.

6.2. Interfaces logiciels et protocoles

Bien qu'un IRI soit défini comme une séquence de caractères, les interfaces logicielles pour les URI fonctionnent normalement sur des séquences d'octets ou d'autres types d'unités de code. Et donc, les interfaces logicielles et les protocoles DOIVENT définir quel codage de caractères est utilisé.

Les interfaces logicielles intermédiaires entre les composants à capacité d'IRI et les composants pour URI seul DOIVENT transposer les IRI conformément au paragraphe 3.1, lorsqu'ils transfèrent de composants à

capacité d' IRI à composants pour URI seul. Cette transposition DEVRAIT intervenir aussi tard que possible. Elle NE DEVRAIT PAS être appliquée entre composants dont on sait qu'ils sont capables de traiter des IRI.

6.3. Format des URI et IRI dans les documents et protocoles

Les formats de document qui transportent des URI peuvent nécessiter une mise à niveau pour permettre le transport des IRI. Dans les cas où le document entier a un codage de caractères d'origine, les IRI DOIVENT aussi être codés dans ce codage de caractère et convertis en conséquence par un analyseur ou interprète. Les caractères d'IRI qui ne peuvent être exprimés dans le codage de caractères d'origine DEVRAIENT être évacués en utilisant les conventions d'échappement du format de document si de telles conventions sont disponibles. Autrement, ils peuvent être codés en pourcentage conformément au paragraphe 3.1. Par exemple, en HTML ou XML, les références de caractères numériques DEVRAIENT être utilisées. Si un document entier a un codage de caractères d'origine et que ce codage de caractère n'est pas UTF-8, les IRI NE DOIVENT PAS alors être placés dans ce document en codage de caractère UTF-8.

Note : Certains formats s'accommodent déjà des IRI, bien qu'ils utilisent une terminologie différente. HTML 4.0 [HTML4] définit la conversion d'IRI en URI comme un comportement d'évitement d'erreur. XML 1.0 [XML1], XLink [XLink], le schéma XML [XMLSchema], et les spécifications fondées sur eux permettent les IRI. Aussi, on s'attend à ce que tous les nouveaux formats et protocoles W3C pertinents soient tenus de traiter les IRI [CharMod].

6.4. Utilisation d'UTF-8 pour le codage de caractères d'origine

Ce paragraphe traite de points de détails et donne des exemples pour le point c) du paragraphe 1.2. Pour pouvoir utiliser les IRI, l'URI correspondant à l'IRI en question doit coder les caractères originaux en octets en utilisant UTF-8. Ceci peut être spécifié pour tous les URI d'un schéma d'URI ou peut s'appliquer à des URI individuels pour les schémas qui ne spécifient pas comment coder les caractères d'origine. Cela peut s'appliquer à tout l'URI, ou seulement à une partie de celui-ci. Pour des informations de base sur le codage des caractères dans les URI, voir aussi le paragraphe 2.5 de [RFC3986].

Pour les nouveaux schémas d'URI, l'utilisation d'UTF-8 est recommandée par [RFC2718]. Des exemples où UTF-8 est déjà utilisé sont la syntaxe URN [RFC2141], les URL IMAP [RFC2192], et les URL POP [RFC2384]. D'un autre côté, seuls quelques URL http peuvent avoir différents IRI correspondants, parce que le schéma d'URL HTTP ne spécifie pas comment coder les caractères d'origine.

Par exemple, pour un document avec un URI de "http://www.example.org/r%C3%A9sum%C3%A9.html", il est possible de construire un IRI correspondant (en notation XML, voir au paragraphe 1.4) : "http://www.example.org/résumé.html" ("&#xE9"; est mis pour le caractère e accent aigu, et "%C3%A9" est le codage UTF-8 et la représentation codée en pourcentage de ce caractère). D'un autre côté, pour un document avec un URI de "http://www.example.org/r%E9sum%E9.html", les octets de codage en pourcentage ne peuvent pas être convertis en caractères réels d'un IRI, car le codage en pourcentage n'est pas fondé sur UTF-8.

Cela signifie que pour la plupart des schémas d'URI, il n'est pas besoin de mettre à niveau leur définition de schéma pour leur permettre de travailler avec des IRI. Le principal cas où la mise à niveau a un sens est lorsqu'une définition de schéma, ou un composant particulier d'un schéma, est strictement limité à l'utilisation de caractères US-ASCII sans perspective d'introduire des caractères/octets non-ASCII via le codage en pourcentage, ou si une définition de schéma utilise normalement des dispositions très spécifiques du schéma pour le codage des caractères non-ASCII. Un exemple de cela est le schéma mailto: [RFC2368].

La présente spécification ne met à niveau aucune spécification de schéma en aucune façon ; cela doit être fait séparément. Noter aussi qu'il n'y a nulle part de "schéma d'IRI " ; tous les IRI utilisent des schémas d'URI, et tous les schémas d'URI peuvent être utilisés avec les IRI, même si dans certains cas c'est seulement en utilisant directement les URI comme IRI, sans aucune conversion.

Les schémas d'URI peuvent imposer des restrictions sur la syntaxe des URI spécifiques d'un schéma ;

c’est-à-dire que les URI qui sont admissibles dans la syntaxe générique d’URI [RFC3986] peuvent n’être pas admissibles du fait de contraintes syntactiques plus étroites imposées par une spécification de schéma d’URI. Les définitions de schéma d’URI ne peuvent pas élargir les restrictions syntactiques de la syntaxe générique d’URI ; autrement, il serait possible de générer des URI qui satisfont aux contraintes syntaxiques spécifiques de schéma sans satisfaire aux contraintes syntaxiques de la syntaxe générique d’URI. Cependant, des contraintes syntaxiques supplémentaires imposées par les spécifications de schéma d’URI sont applicables aux IRI, comme l’URI correspondante qui résulte de la transposition définie au paragraphe 3.1 DOIVENT être un URI valide selon les restrictions syntaxiques de la syntaxe générique d’URI et toute restriction plus étroite imposée par la spécification de schéma d’URI correspondante.

Les exigences pour l’utilisation de l’ UTF-8 s’appliquent à toutes les parties d’un URI (avec l’exception potentielle de la partie ireg-name; voir au paragraphe 3.1. Cependant, il est possible que la capacité des IRI à représenter directement une large gamme de caractères ne soit utilisée que dans certaines parties de l’IRI (ou de la référence d’IRI). Les autres parties de l’IRI peuvent ne contenir que des caractères US-ASCII, ou ils peuvent n’être pas fondés sur UTF-8. Ils peuvent être fondés sur un autre codage de caractères, ou ils peuvent être codé directement en données binaires brutes (voir aussi [RFC2397]).

Par exemple, il est possible d’avoir une référence d’URI de "http://www.example.org/r%E9sum%E9.xml#r%C3%A9sum%C3%A9", où le nome du document est codé en iso-8859-1 sur la base des réglages du serveur, mais où l’identifiant de fragment est codé en in UTF-8 conformément à [XPointer]. L’IRI correspondant à l’URI ci-dessus serait (en notation XML)"http://www.example.org/r%E9sum%E9.xml#résum&#xE9".

Des considérations similaires s’appliquent aux parties d’interrogation. La fonctionnalité des IRI (à savoir, d’être capables d’inclure des caractères non-ASCII) peut seulement être utilisée si la partie d’interrogation est codée en UTF-8.

6.5. Références croisées d’IRI

Le traitement des références d’IRI croisées par rapport à une base s’effectue directement ; les algorithmes de [RFC3986] peuvent s’appliquer directement, en traitant les caractères supplémentaires admis dans les références d’IRI de la même façon que les caractères non réservés dans les références d’URI.

7. Guide de traitement URI/IRI (pour information)

Cette section informative donne des lignes directrices pour la prise en charge des IRI dans les mêmes composants et opérations logiciels qui traitent habituellement les URI : les interfaces logicielles qui traitent les URI, les logiciels qui permettent aux utilisateurs d’entrer les URI, les logiciels qui créent ou génèrent les URI, les logiciels qui affichent les URI, les formats et protocoles qui transportent les URI, et les logiciels qui interprètent les URI. Toutes et tous peuvent nécessiter des modifications avant de fonctionner correctement avec les IRI. Les considérations de la présente section s’appliquent aussi aux références d’URI et d’ IRI.

7.1. Interfaces logicielles URI/IRI

Les interfaces logicielles qui traitent les URI, telles que les API de traitement d’URI et les protocoles de transfert d’URI, ont besoin d’éléments d’interface et de protocole qui soient conçus pour porter les IRI.

Dans le cas où le traitement courant dans une API ou un protocole est fondé sur US-ASCII, UTF-8 est recommandé comme codage de caractère pour les IRI, car il est compatible avec US-ASCII, cela est conforme aux recommandations de [RFC2277], et la conversion en URI est facile. Dans tous les cas, la définition de l’API ou du protocole doit clairement établir le codage de caractères à utiliser.

Le transfert de composants URI seulement à des composants à capacité d’IRI n’exige pas de transposition, bien que la conversion décrite au paragraphe 3.2 ci-dessus puisse être effectuée. Il est préférable de ne pas effectuer la conversion inverse lorsqu’il y a un risque que cela ne soit pas fait correctement.

7.2. Entrées d’URI/IRI

Certains composants permettent aux utilisateurs d’entrer les URI dans le système en les tapant ou en les dictant, par exemple. Ce logiciel doit être mis à niveau pour permettre d’entrer les IRI.

Une personne regardant une représentation visuelle d’un IRI (comme une séquence de glyphes, dans un certain ordre, sur un certain affichage visuel) ou entendant un IRI, va utiliser une méthode d’entrée pour les caractères dans la langue de l’utilisateur pour entrer l’IRI. Selon l’écriture et la méthode d’entrée utilisée, cela peut être un processus plus ou moins compliqué.

Le processus d’entrée de l’IRI doit garantir, autant que possible, que les restrictions définies au paragraphe 2.2 sont satisfaites. Cela peut être fait en choisissant les méthodes d’entrée appropriées ou les variantes/réglages, en convertissant de façon appropriée les caractères à entrer, en éliminant les caractères qui ne peuvent pas être convertis, et/ou en émettant un avertissement ou un message d’erreur pour l’utilisateur.

Comme exemple de variante de réglage, les éditeurs de méthode d’entrée pour les langages de l’Est de l’ Asie permettent habituellement d’entrée de lettres latines et des caractères qui s’y rapportent en pleine largeur et en version demi largeur. Pour l’entrée d’un IRI, l’éditeur de méthode d’entrée devrait se régler pour produire des lettres latines et la ponctuation en demi largeur et le Katakana en pleine largeur.

Un champ d’entrée principalement ou seulement utilisé pour l’entrée des URI/IRI peut permettre à l’utilisateur de voir un IRI lorsqu’il est transposé en URI. Les endroits où l’entrée d’ IRI est fréquente peuvent donner la possibilité de visualiser un IRI alors qu’il est transposé en URI. Cela aidera les utilisateurs lorsque certains des logiciels qu’ils utilisent n’acceptent pas encore les IRI.

Un composant d’entrée d’IRI qui fait l’interface avec des composants qui traitent des URI, mais pas des IRI, doit transposer l’IRI en URI avant de la passer à ces composants.

Pour l’entrée des IRI avec des caractères de droite à gauche, prière de se reporter au paragraphe 4.3.

7.3. Transfert d’URI/IRI entre applications

De nombreuses applications, particulièrement des agents d’utilisateur de messagerie, essayent de détecter des URI apparaissant en texte clair. Pour cela elles utilisent des méthodes heuristiques fondées sur la syntaxe d’URI. Elles permettent ensuite à l’utilisateur de cliquer sur de tels URI et restaurent la ressource correspondante dans une application appropriée (habituellement dépendante du schéma).

De telles applications doivent être mises à niveau pour utiliser la syntaxe d’IRI comme base heuristique. En particulier, un caractère non-ASCII ne devrait pas être pris comme l’indication de la fin d’un IRI. De telles applications doivent aussi s’assurer qu’elles convertissent correctement l’IRI détecté à partir du codage de caractère du document ou application où apparaît l’IRI en codage de caractère utilisé par le mécanisme d’invocation de l’IRI pour l’ensemble du système, ou en un URI (conformément au paragraphe 3.1) si le mécanisme d’invocation pour l’ensemble du système n’accepte que les URI.

Le bloc note est une autre technique très utilisée pour transférer des URI et IRI d’une application à une autre. Sur la plupart des plates-formes, le bloc note est capable de mémoriser et transférer du texte dans de nombreux langages et écritures. Correctement utilisé, le bloc note transfère des caractères, et non des octets, ce qui ira parfaitement avec les IRI.

7.4. Génération d’URI/IRI

Les systèmes qui offrent des ressources à travers l’Internet, sur lequel ces ressources ont des noms logiques, génèrent parfois automatiquement des URI pour les ressources qu’ils offrent. Par exemple, certains serveurs HTTP peuvent générer une liste pour un répertoire de fichiers en répondre ensuite aux URI générés avec les fichiers.

De nombreux codages de caractères usuels sont utilisés dans des systèmes de fichiers variés. De nombreux systèmes courants ne transforment pas la représentation de caractères locale du système sous-jacent avant de générer les URI.

Pour une interopérabilité maximum, les systèmes qui génèrent des identifiants de ressource devraient faire les transformations appropriées. Par exemple, si un système de fichier contient un fichier appelé "résumé.html", un serveur devrait afficher cela comme "r%C3%A9sum%C3%A9.html" dans un URI, ce qui permet l'utilisation de "résumé.html" dans un IRI, même si localement le nom de fichier est conservé dans un codage de caractères autre que UTF-8.

Cette recommandation s'applique particulièrement aux serveurs HTTP. Pour les serveurs FTP, des considérations similaires s'appliquent ; voir [RFC2640].

7.5. Choix URI/IRI

Dans certains cas, les propriétaires et les éditeurs de ressources exercent un contrôle sur les IRI utilisés pour identifier leurs ressources. Ce contrôle est le plus souvent exercé par le contrôle direct des noms de ressource, tels que les noms de fichiers. Dans ces cas, il est recommandé d'éviter de choisir des IRI qui peuvent facilement être confondus. Par exemple, pour l' US-ASCII, la minuscule ell ("l") est facilement confondue avec le chiffre un ("1"), et la majuscule oh ("O") est facilement confondue avec le chiffre zéro ("0"). Les éditeurs devraient éviter d'embrouiller les utilisateurs avec des identifiants "br0ken" ou "1ame".

En-dehors du répertoire US-ASCII, il y a de nombreuses opportunités de confusion ; il serait trop long d'inclure ici un ensemble complet de lignes directrices. Tant que les noms sont limités à des caractères d'une seule écriture, les locuteurs d'origine d'une ou langage écriture donné sauront bien quand peuvent apparaître des ambiguïtés, et comment les éviter. Ce qui peut paraître ambigu à un étranger peut être complètement évident pour le national moyen. D'un autre côté, dans certains cas, l'UCS contient des variantes pour des raisons de compatibilité; par exemple, pour des raisons typographiques. Cela pourrait être évité chaque fois que possible. Bien qu'il puisse y avoir des exceptions, les nouveaux noms de ressource créés devraient généralement être en NFKC [UTR15] (ce qui signifie qu'ils sont aussi en NFC).

Par exemple, l'UCS contient la ligature "fi" à U+FB01 pour des raisons de compatibilité. Chaque fois que possible, les IRI devraient utiliser les deux lettres "f" et "i" plutôt que la ligature "fi". Un exemple d'utilisation de ce dernier cas est la partie interrogation d'un IRI pour une recherche explicite d'un mot écrit contenant la ligature "fi".

Dans certains cas, il y a un risque que des caractères provenant d'écritures différentes aient le même aspect. L'exemple le plus connu est la similitude du latin "A", du grec "Alpha", et du cyrillique "A". Pour éviter de tels cas, les IRI ne devraient être créés qu'avec tous les caractères d'un même composant dans un langage donné. Cela signifie normalement que tous ces caractères seront dans la même écriture, mais il y a des langues qui mêlent des caractères provenant de différentes écritures (comme le japonais). Cela est similaire aux méthodes heuristiques utilisées pour distinguer entre les lettres et les chiffres dans les exemples ci-dessus. Aussi, pour le latin, le grec et le cyrillique, l'utilisation de lettres minuscules réduira plus les ambiguïtés que ne le ferait l'utilisation des majuscules.

7.6. Affichage des URI/IRI

Dans les situations où le logiciel de restitution n'est pas supposé afficher des parties d'IRI non-ASCII en utilisant correctement les ressources de mise en page et de police disponibles, ces parties devraient être codées en pourcentage avant d'être affichées.

Pour l'affichage d'IRI en Bidi, se reporter au paragraphe 4.1.

7.7. Interprétation des URI et IRI

Les logiciels qui interprètent les IRI comme noms de ressources locales devraient accepter le IRI sous de multiples formes et les convertir et les faire correspondre avec les noms de ressources locales appropriées.

D'abord, des représentations multiples incluent à la fois les IRI dans le codage de caractères d'origine du protocole et leurs contreparties en URI.

Ensuite, elles peuvent inclure les URI construits sur la base des codages de caractères autres que l'UTF-8. Ces URI peuvent être produits par des agents d'utilisateur qui ne se conforment pas à la présente spécification et utilisent des codages de caractères habituels pour convertir les caractères non-ASCII en URI. Si cela est nécessaire , et quels codages de caractères doivent être couverts dépend d'un certain nombre de facteurs, tels que les codages de caractères habituels utilisés localement, et la distribution des différentes versions des agents d'utilisateur. Par exemple, un logiciel pour le japonais peut accepter des URI en Shift_JIS et/ou EUC-JP en plus de l'UTF-8.

Troisièmement, elles peuvent inclure des transpositions supplémentaires pour être plus faciles à traiter pour l'usager et plus résistantes aux erreurs de transmission. Cela serait assez semblable à la façon dont certains serveurs traitent habituellement les URI comme non sensibles à la casse ou effectuent des essais de correspondance supplémentaires pour tenir compte des erreurs d'orthographe. Pour les caractères qui sortent du répertoire US-ASCII, cela peut, par exemple, inclure d'ignorer les accents sur les IRI reçus ou sur les noms de ressources. Noter que de telles transpositions, y compris les transpositions de casse, dépendent du langage utilisé.

Il peut être difficile d'identifier une ressource de façon non ambiguë si trop de transpositions sont prises en considération. Cependant, les parties codées en pourcentage et les parties non codées en pourcentage des IRI peuvent toujours être distinguées clairement. Aussi, la régularité de l'UTF-8 (voir [Duerst97]) rend l'éventualité de collisions plus faible qu'il ne semblerait à première vue.

7.8. Stratégie d'amélioration

Lorsque la présente recommandation met de nouvelles contraintes sur les logiciels pour lesquels de nombreuses instances existent déjà, il est important d'introduire avec soin des mises à niveau et d'être conscient des diverses interactions.

Si les IRI ne peuvent être interprétés correctement, ils ne devraient pas être créés, générés, ou transportés. Cela suggère que la mise à niveau du logiciel d'interprétation d'URI pour l'acceptation des IRI devrait recevoir la plus haute priorité.

D'un autre côté, un seul IRI n'est interprété que par un seul ou très peu d'interprètes qui sont connus à l'avance, bien qu'il puisse être entré et transporté de façon très large.
Donc, les IRI tirent le plus grand bénéfice d'une large mise à niveau des logiciels pour qu'ils soient capables d'entrer et transporter les IRI. Cependant, avant qu'un IRI individuel ne soit publié, il faut faire attention à mettre à niveau le logiciel d'interprétation correspondant afin de couvrir les formes qu'on s'attend à recevoir par diverses versions de logiciel d'entrée et de transport.

La mise à niveau d'un logiciel de génération d'IRI au lieu de l'utilisation d'un codage de caractères local ne devrait survenir qu'après la mise à niveau du service pour qu'il accepte les IRI. De même, les IRI ne devraient être générés que lorsque le service accepte les IRI et que l' infrastructure et le protocole qui doivent intervenir sont sûrs de les transporter en toute sécurité.

Le logiciel qui convertit d'URI en IRI pour l'affichage ne devrait être mis à niveau qu'après que le logiciel d'entrée aura été largement diffusé auprès de la population qui verra le résultat affiché.

Lorsqu'il y a un libre choix de codages de caractères, il est souvent possible de réduire les efforts et les sujétions de la mise à niveau en IRI en utilisant UTF-8 plutôt qu'un autre codage. Par exemple, lorsqu'un nouveau serveur Web à base de fichiers est établi, utiliser UTF-8 comme codage de caractères pour les noms rendra la transition aux IRI plus facile. Vraisemblablement, lorsqu'une nouvelle forme de Web est établie en utilisant UTF-8 comme codage de caractères de la page de format, les URI d'interrogation en retour utiliseront UTF-8 comme codage de caractères (à moins que l'utilisateur, pour une raison ou une autre, ne change le codage de caractères) et sera donc compatible avec les IRI.

Ces recommandations, prises ensemble, permettront l'extension des URI aux IRI afin de traiter les caractères autres que US-ASCII tout en minimisant les problèmes d'interopérabilité. Pour les considérations concernant la mise à niveau des définitions de schéma d'URI, voir au paragraphe 6.4.

8. Considérations sur la sécurité

Les considérations de sécurité exposées dans [RFC3986] s'appliquent aussi aux IRI. De plus, les questions suivantes requièrent une attention particulière pour les IRI.

Un codage ou décodage incorrect peut amener des problèmes de sécurité. En particulier, certains décodeurs UTF-8 ne font pas de vérification sur les séquences d'octet très longues. Par exemple, une "/" est codé avec l'octet 0x2F à la fois en UTF-8 et en US-ASCII, mais certains décodeurs UTF-8 interprètent aussi de travers la séquence 0xC0 0xAF comme une "/". Une séquence telle que "%C0%AF.." peut passer certains essais de sécurité et être interprétée comme une "/.." dans un chemin si les décodeurs UTF-8 sont tolérants, si la conversion et la vérification ne sont pas faites dans le bon ordre, et/ou si les caractères réservés et non réservés ne sont pas clairement distingués.

Il y a plusieurs façons de faire des "mystifications" avec des IRI. "Mystification" signifie que quelqu'un peut ajouter un nom de ressource qui semble le même ou similaire à l'utilisateur, mais pointe sur une ressource différente. La ressource ajoutée peut prétendre être la ressource réelle en semblant très similaire mais peut contenir toutes sortes de changements qu'il peut être difficile de cerner et qui peuvent causer toutes sortes de problèmes. La plupart de possibilités de mystification pour les IRI sont sur leurs extensions pour les URI.

Il existe diverses raisons aux mystifications. Tout d'abord, les attentes de normalisation d'un utilisateur ou la normalisation réelle lors de l'entrée d'un IRI ou du transcodage d'un IRI à partir d'un codage de caractère habituel ne correspond pas à la normalisation utilisée du côté du serveur. Conceptuellement, ceci n'est pas différent des problèmes entourant l'utilisation des serveurs Web insensibles à la casse. Par exemple, une page web populaire avec un nom à casse mêlée ("http://big.example.com/PopularPage.html") peut être "mystifié" par quelqu'un qui serait capable de créer "http://big.example.com/popularpage.html". Cependant, l'utilisation de séquences de caractères non normalisées, et de transpositions supplémentaires pour le confort de l'utilisateur, peut augmenter les risques de mystification. Les protocoles et les serveurs qui permettent la création de ressources avec des noms qui ne sont pas normalisés sont particulièrement vulnérables à de telles attaques. Ceci est un problème de sécurité inhérent au protocole, au serveur, ou à la ressource concernée, et n'est pas spécifique des IRI, mais est mentionné ici dans un souci d'exhaustivité.

Une mystification peut intervenir dans divers composants d'IRI, tels que la partie de nom de domaine ou une partie de chemin. Pour des considérations spécifiques de la partie nom de domaine, voir la [RFC3491]. Pour la partie chemin, les administrateurs des sites qui permettent aux utilisateurs indépendants de créer des ressources dans la même sous zone pourraient avoir à être attentifs dans leurs vérifications sur les mystifications.

Une mystification peut survenir parce qu'en UCS de nombreux caractères se ressemblent. Des précisions sont données au paragraphe 7.5. A nouveau, ceci est très semblable aux possibilités de mystification de l'US-ASCII, par exemple, en utilisant des URI "br0ken" ou "1ame".

Une mystification peut survenir lorsqu'on accepte que des URI avec des codages en pourcentage fondés sur divers codages de caractères traitent avec des agents d'utilisateur plus anciens. Dans certains cas, particulièrement pour les noms de ressources fondés sur les caractères latins, ceci est habituellement facile à détecter parce que les noms codés en UTF-8, lorsqu'ils sont interprétés et visionnés comme codages de caractères ordinaires, produisent surtout du charabia. Lorsque des codages de caractères utilisés en concurrence ont une structure similaire mais qu'il n'y a pas de caractères qui aient exactement le même codage, la détection est plus difficile.

Une mystification peut survenir avec des IRI bidirectionnels, si les restrictions du paragraphe 4.2 ne sont pas suivies. La même représentation visuelle peut être interprétée comme une représentation logique différente, et vice versa. Il est aussi très important d'utiliser une mise en oeuvre Unicode bidirectionnelle correcte.

9. Remerciements

Nous souhaitons remercier Larry Masinter pour son travail comme co-auteur de nombreuses versions antérieures du présent document (draft-masinter-url-i18n-xx).

Les discussions sur les questions traitées ici ont débité il y a longtemps. Il y en avait des prolégomènes dans le groupe de travail HTML en août 1995 (sur le thème de "mondialisation des URI") et dans la liste de diffusion www-international de juillet 1996 (sur le thème de "Internationalisation et URL"), et il y a eu des réunions ad hoc aux conférences Unicode en septembre 1995 et septembre 1997.

Tous nos remerciements à Francois Yergeau, Matitiahu Allouche, Roy Fielding, Tim Berners-Lee, Mark Davis, M.T. Carrasco Benitez, James Clark, Tim Bray, Chris Wendt, Yaron Goland, Andrea Vine, Misha Wolf, Leslie Daigle, Ted Hardie, Bill Fenner, Margaret Wasserman, Russ Housley, Makoto MURATA, Steven Atkin, Ryan Stansifer, Tex Texin, Graham Klyne, Bjoern Hoehrmann, Chris Lilley, Ian Jacobs, Adam Costello, Dan Oscarson, Elliotte Rusty Harold, Mike J. Brown, Roy Badami, Jonathan Rosenne, Asmus Freytag, Simon Josefsson, Carlos Viegas Damasio, Chris Haynes, Walter Underwood, et à de nombreux autres qui nous ont aidé à mieux comprendre les problèmes et leurs solutions possibles, et à mettre en place tous les détails.

Le présent document est un produit du Groupe de travail Internationalisation (I18N WG) du Consortium World Wide Web (W3C). Merci aux membres du Groupe de travail W3C I18N et du Groupe d'Intérêt pour leurs contributions et leur travail sur [CharMod]. Merci aussi aux membres de nombreux autres groupes de travail du W3C pour avoir adopter les IRI, et aux membres de l'atelier Montréal IAB sur l'Internationalisation et la Localisation pour leur travail de révision.

10. Références

10.1. Références normatives

[ASCII] American National Standards Institute, "Coded Character Set -- 7-bit American Standard Code for Information Interchange" *(Ensemble des caractères codés – Code standart américain à 7 bits pour les échanges d'informations)*, ANSI X3.4, 1986.

[ISO10646] International Organization for Standardization, "ISO/IEC 10646:2003: Information Technology - Universal Multiple-Octet Coded Character Set (UCS)" *(Technologies de l'information - Ensemble des caractères codés multi-octet universel (UCS))*, Norme ISO 10646, décembre 2003.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels" *(Mots-clés à utiliser dans les RFC pour indiquer les niveaux d'exigence)*, BCP 14, RFC 2119, mars 1997.

[RFC2234] Crocker, D. et P. Overell, "Augmented BNF for Syntax Specifications: ABNF" *(BNF augmenté pour les spécifications de syntaxe)*, RFC 2234, novembre 1997.

[RFC3490] Faltstrom, P., Hoffman, P., et A. Costello, "Internationalizing Domain Names in Applications (IDNA)" *(Internationalisation des noms de domaine dans les applications (IDNA))*, RFC 3490, mars 2003.

[RFC3491] Hoffman, P. et M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)" *(Nameprepp : un profil Stringprep pour les noms de domaine internationalisés)*, RFC 3491, mars 2003.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646" *(UTF-8, un format de transformation d'ISO 10646)*, STD 63, RFC 3629, novembre 2003.

[RFC3986] Berners-Lee, T., Fielding, R., et L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax" *(Identifiant de ressource universel (URI) : Syntaxe générique)*, STD 66, RFC 3986, janvier 2005.

[UNI9] Davis, M., "The Bidirectional Algorithm" *(L'algorithme bidirectionnel)*, Annexe 9 de la norme Unicode, mars 2004, <http://www.unicode.org/reports/tr9/tr9-13.html>.

[UNIV4] The Unicode Consortium, "The Unicode Standard, Version 4.0.1" *(Norme Unicode, version 4.0.1)*, définie par : The Unicode Standard, Version 4.0 (Reading, MA, Addison-Wesley, 2003. ISBN 0-321-18578-1), telle qu'amendée par Unicode 4.0.1 (http://www.unicode.org/versions/Unicode4.0.1/)", mars 2004.

[UTR15] Davis, M. et M. Duerst, "Unicode Normalization Forms" *(Formes de normalisation Unicode*, Annexe 15 de la norme Unicode, avril 2003, <http://www.unicode.org/unicode/reports/tr15/tr15-23.html>.

10.2. Références informatives

[BidiEx] "Examples of bidirectional IRIs" *(Exemples d'IRI bidirectionnels*, <http://www.w3.org/International/iri-edit/ BidiExamples>.

[CharMod] Düerst, M., Yergeau, F., Ishida, R., Wolf, M., et T. Texin, "Character Model for the World Wide Web: Resource Identifiers" *(Modèle de caractères pour la Toile mondiale : Identifiants de ressource)*, Projet de Recommandation du World Wide Web Consortium, novembre 2004, <http://www.w3.org/TR/charmod-resid>.

[Duerst97] Düerst, M., "The Properties and Promises of UTF-8" *(Propriétés et promesses de l'UTF-8)*, Proc. 11ème Conférence internationale Unicode, San Jose, septembre 1997, <http://www.ifi.unizh.ch/mml/mduerst/papers/PDF/IUC11-UTF-8.pdf>.

[Gettys] Gettys, J., "URI Model Consequences" *(Conséquences du modèle d'URI)*, <http://www.w3.org/DesignIssues/ModelConsequences>.

[HTML4] Raggett, D., Le Hors, A., et I. Jacobs, "HTML 4.01 Specification" *(Spécification HTML 4.01)*, Recommandation du World Wide Web Consortium, décembre 1999, <http://www.w3.org/TR/html401/appendix/notes.html#h-B.2>.

[RFC2045] Freed, N. et N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies" *(Extensions de messagerie Internet multi-objets (MIME) Partie un : Format des corps de messages Internet)*, RFC 2045, novembre 1996.

[RFC2130] Weider, C., Preston, C., Simonsen, K., Alvestrand, H., Atkinson, R., Crispin, M., et P. Svanberg, "The Report of the IAB Character Set Workshop held 29 February - 1 March, 1996" *(Rapport de l'atelier sur l'ensemble de caractères IAB tenu du 29 février au 1er mars 1996)*, RFC 2130, avril 1997.

[RFC2141] Moats, R., "URN Syntax" *(Syntaxe d'URN)*, RFC 2141, mai 1997.

[RFC2192] Newman, C., "IMAP URL Scheme" *(Schéma d'URL IMAP)*, RFC 2192, septembre 1997.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages" *(Politique de l'IETF en matière d'ensembles de caractères et de langages)*, BCP 18, RFC 2277, janvier 1998.

[RFC2368] Hoffman, P., Masinter, L., et J. Zawinski, "The mailto URL scheme" *(Le schéma d'URL mailto)*, RFC 2368, juillet 1998.

[RFC2384] Gellens, R., "POP URL Scheme" *(Le schéma d'URL POP)*, RFC 2384, août 1998.

[RFC2396] Berners-Lee, T., Fielding, R., et L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax" *(Identifiants de ressource uniformes (URI) : syntaxe générique)*, RFC 2396, août 1998.

[RFC2397] Masinter, L., "The "data" URL scheme" *(Le schéma d'URL "données"),* RFC 2397, août 1998.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., et T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1" *(Protocole de transfert Hypertext – HTTP/1.1)*, RFC 2616, juin

1999.

[RFC2640] Curtin, B., "Internationalization of the File Transfer Protocol" *(Internationalisation du protocole de transfert de fichiers)*, RFC 2640, juillet 1999.

[RFC2718] Masinter, L., Alvestrand, H., Zigmond, D., et R. Petke, "Guidelines for new URL Schemes" *(Lignes directrices pour de nouveaux schémas d'URL)*, RFC 2718, novembre 1999.

[UNIXML] Düerst, M. et A. Freytag, "Unicode in XML and other Markup Languages" *(Unicode en XML et autres langages de balisage)*, Rapport technique Unicode n° 20, Note du World Wide Web Consortium, juin 2003, <http://www.w3.org/TR/unicode-xml/>.

[XLink] Rose, S., Maler, E., et D. Orchard, "XML Linking Language (XLink) Version 1.0" *(Langage de liaison XML (Xlink) version 1.0)*, Recommandation du World Wide Web Consortium, juin 2001, <http://www.w3.org/TR/xlink/#link-locators>.

[XML1] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., et F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)" *(Langage de balisage étendu (XML) 1.0 (troisième édition))*, Recommandation du World Wide Web Consortium, février 2004, <http://www.w3.org/TR/REC-xml#sec-external-ent>.

[XMLNamespace] Bray, T., Hollander, D., et A. Layman, "Namespaces in XML" *(Espaces de nom en XML)*, Recommandation du World Wide Web Consortium, janvier 1999, <http://www.w3.org/TR/REC-xml-names>.

[XMLSchema] Biron, P. et A. Malhotra, "XML Schema Part 2: Datatypes" *(Schéma XML, Partie 2 : Datatypes)*, Recommandation du World Wide Web Consortium, mai 2001, <http://www.w3.org/TR/xmlschema-2/#anyURI>.

[XPointer] Grosso, P., Maler, E., Marsh, J. et N. Walsh, "XPointer Framework" *(Trame pour XPointer)*, Recommandation du World Wide Web Consortium, mars 2003,

Appendice A. Autres conceptions envisagées

La présente section résume brièvement les autres conceptions majeures et les raisons pour lesquelles elles n'ont pas été retenues.

Appendice A.1. Nouveaux schémas

Il a été proposé d'introduire de nouveaux schémas (par exemple, httpi:, ftpi:,...) ou un nouveau méta schéma (par exemple, conduisant à des préfixes d'URI/IRI tels que i:http:, i:ftp:,...) pour rendre la conversion d'IRI en URI dépendante du schéma ou de distinguer entre les codages en pourcentage résultants de la conversion d'IRI en URI et les codages en pourcentage provenant des codages de caractère ordinaires.

De nouveaux schémas ne sont pas nécessaires pour distinguer les URI des vrais IRI (c'est-à-dire des IRI qui contiennent des caractères non-ASCII). Le bénéfice de la capacité à détecter l'origine des codages en pourcentage est marginal, car l'UTF-8 peut être détecté avec une très grande fiabilité. Il est très difficile de déployer de nouveaux schémas, et donc le déploiement des IRI est bien plus facile en n'exigeant pas de nouveaux schémas pour les IRI. Il n'est pas à conseiller de rendre la conversion dépendante du schéma, et cela serait encouragé par des schémas séparés pour les IRIs. Utiliser une convention uniforme pour la conversion des IRI en URI rend la mise en œuvre de l'IRI perpendiculaire à l' introduction de nouveaux schémas réels.

Appendice A.2. Codage de caractères autre que UTF-8

A un stade précoce, l'UTF-7 était considéré comme une solution de remplacement pour l'UTF-8 lors de la conversion des IRI en URI. L'UTF-7 n'aurait pas eu besoin de codages en pourcentage et dans bien des cas aurait été plus court que l'UTF-8 codé en pourcentage.

L'utilisation de l'UTF-8 évite une double couche et la surcharge d'utilisation du caractère "+". L'UTF-8 est pleinement compatible avec l'US-ASCII et a donc été recommandé par l'IETF, étant largement utilisé.

L'UTF-7 n'a jamais été très utilisé et il est maintenant clairement déconseillé. Demander aux mises en œuvre de convertir d' UTF-8 en UTF-7 et vice-versa serait un fardeau supplémentaire pour les applications.

Appendice A.3. Nouvelle convention de codage

Au lieu d'utiliser la convention de codage en pourcentage existante des URI, qui se fonde sur les octets, l'idée était de créer une nouvelle convention de codage ; par exemple, d'utiliser "%u" pour introduire les points de code UCS.

L'utilisation du mécanisme existant de codage en pourcentage fondé su l'octet ne nécessite pas de mettre à niveau la syntaxe d'URI ni des serveurs correspondants.

Appendice A.4. Indication des codages de caractère dans l'URI/IRI

Certaines propositions suggéraient d'indiquer les codages de caractère utilisés dans un URI ou IRI avec de nouvelles conventions syntaxiques dans l'URI lui-même, similaires au paramètre "charset" pour la messagerie électronique et les pages Web. A titre d'exemple, l'étiquette entre crochets dans "http://www.example.org/ros[iso-8859-1]&#xE9"; indique que le "&#xE9"; suivant doit être interprété comme iso-8859-1.

Si l'UTF-8 est utilisé de façon exclusive, une mise à niveau de la syntaxe d'URI n'est pas nécessaire. Il évite d'avoir à copier correctement dans tous les cas des étiquettes qui peuvent être nombreuses, même dans un autobus ou sur une nappe en papier, ce qui pose des problèmes d'utilisation (et qui est prodigieusement ennuyeux). L'utilisation exclusive de l'UTF-8 réduit aussi les erreurs de transcodage et la confusion.

Adresses des auteurs

Martin Düerst (Note : Prière d'écrire "Düerst" avec u-umlaut chaque fois que possible, par exemple comme "Dürst" en XML et HTML.) World Wide Web Consortium
5322 Endo
Fujisawa, Kanagawa 252-8520 Japon

Tél : +81 466 49 1170
Fax : +81 466 49 1171
Email : duerst@w3.org
URI : http://www.w3.org/People/D%C3%BCrst/
(Note : Ceci est la forme codée en pourcentage d'un IRI.)

Michel Suignard
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052 U.S.A.

Tél : +1 425 882-8080
EMail: michelsu@microsoft.com
URI: http://www.suignard.com

Déclaration de Copyright

Copyright (C) The Internet Society (2005).

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

Le présent document et les informations y contenues sont fournies sur une base "EN L'ETAT" et LE CONTRIBUTEUR, L'ORGANISATION QU'IL OU ELLE REPRESENTE OU QUI LE/LA FINANCE (S'IL EN EST), LA INTERNET SOCIETY ET LA INTERNET ENGINEERING TASK FORCE DECLINENT TOUTES GARANTIES, EXPRIMEES OU IMPLICITES, Y COMPRIS MAIS NON LIMITEES A TOUTE GARANTIE QUE L'UTILISATION DES INFORMATIONS CI-ENCLOSES NE VIOLENT AUCUN DROIT OU AUCUNE GARANTIE IMPLICITE DE COMMERCIALISATION OU D'APTITUDE A UN OBJET PARTICULIER.

Propriété intellectuelle

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

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

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

Remerciement

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