

Groupe de travail Réseau
Request for Comments : 5147
 RFC mise à jour : 2046
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

E. Wilde, UC Berkeley
 M. Dürst, Aoyama Gakuin University
 avril 2008

Identifiants de fragment d'URI pour le type de support text/plain

Statut du présent mémoire

Le présent document spécifie un protocole Internet sur la voie de la normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

Le présent mémoire définit des identifiants de fragment d'URI pour les entités text/plain MIME. Ces identifiants de fragment rendent possible de se référer à des parties d'une entité text/plain MIME, identifiées par position ou gamme de caractères, ou par position ou gamme de lignes. Les identifiants de fragment peuvent aussi contenir des informations pour des vérifications d'intégrité afin de les rendre plus robustes.

Table des Matières

| | |
|---|---|
| 1. Introduction..... | 1 |
| 1.1 Qu'est-ce que text/plain ?..... | 2 |
| 1.2 Qu'est un identifiant de fragment d'URI ?..... | 2 |
| 1.3 Pourquoi des identifiants de fragment text/plain ?..... | 2 |
| 1.4 Déploiement incrémentaire..... | 3 |
| 1.5 Notation utilisée dans ce mémoire..... | 3 |
| 2. Méthodes d'identification de fragment..... | 3 |
| 2.1 Principes d'identification de fragment..... | 3 |
| 2.2 Combinaison des principes..... | 4 |
| 2.3 Robustesse d'identifiant de fragment..... | 4 |
| 3. Syntaxe d'identification de fragment..... | 5 |
| 3.1 Vérifications d'intégrité..... | 5 |
| 4. Traitement d'identifiant de fragment..... | 6 |
| 4.1 Traitement des fins de lignes dans les entités text/plain MIME..... | 6 |
| 4.2 Traitement des valeurs de position..... | 6 |
| 4.3 Vérifications d'intégrité de fragment..... | 6 |
| 4.4 Erreurs de syntaxe dans les identifiants de fragment..... | 6 |
| 5. Exemples..... | 6 |
| 6. Considérations relatives à l'IANA..... | 7 |
| 7. Considérations sur la sécurité..... | 7 |
| 8. Références..... | 8 |
| 8.1 Références normatives..... | 8 |
| 8.2 Références pour information..... | 8 |
| Appendice A. Remerciements..... | 9 |
| Adresse des auteurs..... | 9 |
| Déclaration complète de droits de reproduction..... | 9 |

1. Introduction

Le présent mémoire met à jour le type de support text/plain défini dans la [RFC2046] en définissant les identifiants de fragment d'URI pour les entités text/plain de MIME. Cela rend possible de se référer à des parties d'une entité MIME text/plain. De telles parties peuvent être identifiées par position ou gamme de caractères, ou par position de ligne ou gamme de lignes. Des informations de vérification d'intégrité peuvent être ajoutées à un identifiant de fragment pour le rendre plus robuste, et permettre aux applications de détecter des changements de l'entité.

Cette section fait une introduction aux concepts généraux des entités text/ plain MIME et des identifiants de fragment d'URI, et elle discute du besoin des identifiants de fragment pour text/plain et des questions de déploiement. La Section 2

discute des principes et méthodes sur lesquels se fonde le présent mémoire. La Section 3 définit la syntaxe, et la Section 4 discute du traitement des identifiants de fragment text/plain. La Section 5 donne quelques exemples.

1.1 Qu'est-ce que text/plain ?

Les types de supports de l'Internet (souvent appelés des "types MIME") comme définis dans les [RFC2045] et [RFC2046], sont utilisés pour identifier différents types et sous-types de supports. Les [RFC2046] et [RFC3676] spécifient le type de support text/plain, qui est utilisé pour du texte simple, non formaté. Pour citer la [RFC2046] : " text/plain n'assure ni ne permet de commandes de formatage, de spécifications d'attributs de fonte, d'instructions de traitement, de directives d'interprétation, ou de balisage de contenu. text/plain est vu simplement comme une séquence linéaire de caractères, éventuellement interrompue par des coupures de ligne ou des sauts de page".

Le type de support text/plain ne fait pas de restriction sur le codage de caractères ; tout codage de caractères peut être utilisé. En l'absence d'une déclaration explicite de codage de caractères, l'US-ASCII [ANSI] est supposé comme codage de caractères par défaut. Cette variabilité du codage de caractères rend impossible de compter les caractères dans une entité MIME text/plain sans tenir compte du codage de caractères, parce que de nombreux codages de caractères utilisent plus d'un octet par caractère.

Le plus gros avantage des entités text/plain MIME est leur facilité d'utilisation et leur portabilité parmi différentes plateformes. Tant qu'elles utilisent des codages de caractères populaires (comme US-ASCII ou UTF-8 [RFC3629]) elles peuvent être affichées et traitées sur virtuellement tous les systèmes informatiques. Le seul problème d'interopérabilité restant est la représentation des terminaisons de ligne, qui est discutée au paragraphe 4.1.

1.2 Qu'est un identifiant de fragment d'URI ?

Les URI sont le mécanisme d'identification des ressources sur la Toile. La syntaxe d'URI spécifiée dans la [RFC3986] inclut facultativement un "identifiant de fragment", séparé par un signe dièse ("#") pour indiquer un numéro. L'identifiant de fragment consiste en informations de référence supplémentaires à interpréter par l'agent d'utilisateur après l'achèvement réussi de l'action de restitution. La sémantique d'un identifiant de fragment est une propriété des données résultant d'une action de restitution, sans considération du type d'URI utilisé dans la référence. Donc, le format et l'interprétation des identifiants de fragment dépend du type de support du résultat de la restitution.

L'identifiant de fragment le plus populaire est défini pour text/html (défini dans la [RFC2854]) et rend possible de se référer à un élément spécifique (identifié par la valeur d'un attribut "name" ou "id") d'un document HTML. Cela rend possible de faire référence à une partie spécifique d'une page de la Toile, plutôt qu'à la page entière.

1.3 Pourquoi des identifiants de fragment text/plain ?

Se référer à des parties spécifiques d'une ressource peut être très utile parce que cela permet aux utilisateurs et applications de créer des références plus spécifiques. Les utilisateurs peuvent créer des références à la partie qui les intéresse réellement ou dont ils veulent parler, plutôt que de toujours pointer sur une ressource complète. Bien qu'il soit suggéré que des méthodes d'identification de fragments soient spécifiées dans un enregistrement MIME de type de support (voir la [RFC4288]) de nombreux types de supports n'ont pas de méthode d'identification de fragment associée.

Les identifiants de fragment sont seulement utiles si ils sont pris en charge par le client, parce que ils sont seulement interprétés par le client. Donc, une nouvelle méthode d'identification de fragment va exiger un certain temps pour être adoptée par les clients, et les plus anciens clients ne vont pas la prendre en charge. Cependant, parce que l'URI fonctionne même si l'identifiant de fragment n'est pas pris en charge (la ressource est restituée, mais l'identifiant de fragment n'est pas interprété) une adoption rapide n'est pas très critique pour assurer le succès d'une nouvelle méthode d'identification de fragment

Les identifiants de fragment pour text/plain, comme définis dans le présent mémoire, rendent possible de se référer à des parties spécifiques d'une entité MIME text/plain, en utilisant les concepts de positions et de gammes, qui peuvent être appliqués aux caractères et lignes. Donc, les identifiants de fragment text/plain permettent aux utilisateurs d'échanger des informations de façon plus spécifique, réduisant par là le temps et les efforts nécessaires pour chercher manuellement la partie pertinente d'une entité MIME text/plain.

Le format text/plain ne prend pas en charge l'incorporation de liens, de sorte que dans la plupart des environnements, les ressources text/plain peuvent seulement servir de cibles aux liens, et pas de source. Cependant, quand on combine les identifiants de fragment text/plain spécifiés dans le présent mémoire avec des mécanismes de liaison hors ligne tels que XLink [XML], il devient possible de "lier" des ressources de lien à des ressources text/plain et ainsi d'incorporer des liens

dans des ressources text/plain. Donc, les identifiants de fragment text/plain spécifiés dans le présent mémoire ouvrent un chemin pour que les fichiers text/plain deviennent des ressources à navigation bidirectionnelle dans les systèmes hypermédia comme la Toile.

1.4 Déploiement incrémentaire

Tant que les identifiants de fragment text/plain ne sont pas universellement pris en charge, il est important de considérer les implications d'un déploiement incrémentaire. Les clients (par exemple, les navigateurs de la Toile) qui ne prennent pas en charge l'identifiant de fragment text/plain décrit dans ce mémoire vont fonctionner avec des références d'URI à des entités MIME text/plain, mais ils vont échouer à localiser la sous ressource identifiée par l'identifiant de fragment. C'est un comportement de repli raisonnable, et en général, les utilisateurs devraient tenir compte de la possibilité qu'un programme qui interprète un certain URI échoue à interpréter la partie identifiant de fragment. Comme l'évaluation d'identifiant de fragment est locale pour le client (et se produit après la restitution de l'entité MIME) il n'y a pas de façon fiable pour un serveur de déterminer si un client demandeur utilise un URI contenant un identifiant de fragment.

1.5 Notation utilisée dans ce mémoire

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2. Méthodes d'identification de fragment

L'identification des fragments des entités MIME text/plain peut s'appuyer sur différentes fondations. Comme il n'est pas possible d'insérer des identifiants explicites, invisibles dans une entité MIME text/plain (par exemple, comme utilisé dans les documents HTML, mis en œuvre par des attributs dédiés) l'identification de fragment doit s'appuyer sur certaines propriétés inhérentes de l'entité MIME. Le présent mémoire spécifie l'identification de fragment en utilisant quatre méthodes différentes, qui sont les positions et gammes de caractères, et les positions et gammes de lignes, augmentées d'un mécanisme de vérification d'intégrité pour améliorer la robustesse des identifiants de fragment.

Quand elles interprètent le nombre de caractères ou de lignes, les mises en œuvre DOIVENT prendre en compte le codage de caractère de l'entité MIME, parce que le compte de caractères et le compte d'octets peuvent différer pour le codage de caractère utilisé. Par exemple, une entité MIME qui utilise le codage UTF-16 (spécifié dans la [RFC2781]) utilise deux octets par caractère dans la plupart des cas, et parfois quatre octets par caractère. Elle peut aussi avoir une marque d'ordre d'octets (BOM, *Byte-Order Mark*) en tête, qui ne compte pas comme caractère et donc affecte aussi la transposition d'un simple compte d'octets à un compte de caractères.

2.1 Principes d'identification de fragment

L'identification de fragment peut être faite en combinant deux principes orthogonaux, qui sont les positions et les gammes, et les caractères et les lignes. Ce paragraphe décrit les principes eux-mêmes, tandis que le paragraphe 2.2 décrit la combinaison des principes.

2.1.1 Positions et gammes

Une position n'identifie pas un fragment réel de l'entité MIME, mais une position à l'intérieur de l'entité MIME, qui peut être vue comme un fragment de longueur zéro. L'utilité des positions est de fournir des pointeurs pour les applications qui peuvent les utiliser pour mettre en œuvre des fonctions comme "insérer du texte ici", qui a besoin d'une position plutôt que d'un fragment. Les positions sont comptées à partir de zéro ; la position zéro étant avant le premier caractère ou ligne d'une entité MIME text/plain. Donc, une entité MIME text/plain ayant un caractère a deux positions, une avant le premier caractère (position zéro) et une après le premier caractère (position 1).

Comme les positions sont des fragments de longueur zéro, les applications DEVRAIENT utiliser d'autres méthodes que de les mettre au premier plan pour indiquer les positions, la façon la plus évidente étant le positionnement d'un curseur (si l'application prend en charge le concept de curseur).

Par ailleurs, les gammes identifient les fragments d'une entité MIME d'une longueur qui peut être supérieure à zéro. Le principe général des gammes est qu'elles spécifient à la fois une limite inférieure et une limite supérieure. Le début ou la fin d'une spécification de gamme peut être omis, prenant par défaut la première ou la dernière position de l'entité MIME,

respectivement. La fin d'une gamme doit avoir une valeur supérieure ou égale à celle du début. Une gamme avec un début et une fin identiques est légale et identifie une gamme de longueur zéro, qui est équivalente à une position.

Les applications qui prennent en charge un concept comme la mise en valeur DEVRAIENT utiliser ce concept pour indiquer des fragments de longueur supérieure à zéro à l'utilisateur.

Pour les positions et gammes, il est implicitement supposé que si un nombre est supérieur au nombre réel d'éléments de l'entité MIME, elles se réfèrent au dernier élément de l'entité MIME (voir les détails à la Section 4).

2.1.2 Caractères et lignes

Le concept de positions et de gammes peut être appliqué aux caractères ou aux lignes. Dans les deux cas, les positions indiquent des points entre ces entités, tandis que les gammes identifient zéro, une ou plusieurs de ces entités en indiquant des positions.

Les positions de caractères sont numérotées en commençant à zéro (en ignorant les marques initiales de BOM ou les concepts similaires qui ne font pas partie du contenu textuel réel d'une entité MIME text/plain) et en comptant chaque caractère séparément, à l'exception des terminaisons de ligne, qui sont toujours comptées comme un seul caractère (voir les détails au paragraphe 4.1).

Les positions de ligne sont numérotées en commençant à zéro (avec la position de ligne zéro qui est toujours identique avec la position de caractère zéro) ; le paragraphe 4.1 décrit comment les terminaisons de ligne sont identifiées. Les fragments identifiés par des lignes incluent les terminaisons de ligne, de sorte que les applications qui identifient des fragments sur la base de la ligne DOIVENT inclure les terminaisons de ligne dans l'identification de fragment qu'elle utilisent (par exemple, la sélection mise en valeur). Si une entité MIME ne contient pas de terminaison de ligne, elle consiste alors en une seule ligne (la première).

2.2 Combinaison des principes

Dans les paragraphes qui suivent, les principes décrits dans le paragraphe précédant (positions/gammes et caractères/lignes) sont combinés, résultant en quatre cas d'utilisation. Les schémas mentionnés ci-dessous se réfèrent à la syntaxe d'identifiant de fragment, décrite en détails à la Section 3.

2.2.1 Position de caractère

Pour identifier une position de caractère (c'est-à-dire, un fragment de longueur zéro entre deux caractères) le schéma "char" suivi par un seul nombre est utilisé. Cette méthode identifie une position entre deux caractères (ou avant le premier ou après le dernier caractère) plutôt que d'identifier un fragment consistant en un nombre de caractères. Le comptage de position de caractère commence à zéro, de sorte que la position de caractère avant le premier caractère d'une entité MIME text/plain a la position de caractère zéro, et une entité MIME contenant n caractères distincts a n+1 positions de caractère distinctes, la dernière ayant la position de caractère n.

2.2.2 Gamme de caractères

Pour identifier un fragment d'un ou plusieurs caractères (une gamme de caractères) le schéma "char" suivi par une spécification de gamme est utilisé. Une gamme de caractères est une région consécutive de l'entité MIME qui s'étend de la position de caractère de début de la gamme à la position de caractère de fin de la gamme.

2.2.3 Position de ligne

Pour identifier une position de ligne (c'est-à-dire, un fragment de longueur zéro entre deux lignes) le schéma "line" suivi par un seul nombre est utilisé. Cette méthode identifie une position entre deux lignes (ou avant la première ou après la dernière ligne) plutôt que d'identifier un fragment consistant en un nombre de lignes. Le comptage de position de ligne commence à zéro, de sorte que la position de ligne avant la première ligne d'une entité MIME text/plain a la position de ligne zéro, et une entité MIME contenant n lignes distinctes a n+1 positions de ligne distinctes, la dernière ayant la position de ligne n.

2.2.4 Gamme de lignes

Pour identifier un fragment d'une ou plusieurs lignes (une gamme de lignes) le schéma "line" suivi par une spécification de gamme est utilisé. Une gamme de lignes est une région consécutive de l'entité MIME qui s'étend de la position de ligne de début de la gamme à la position de ligne de fin de la gamme.

2.3 Robustesse d'identifiant de fragment

Il peut facilement arriver qu'une modification de la ressource référencée casse un identifiant de fragment. Si les applications veulent créer des identifiants de fragment plus robustes, elles peuvent le faire en ajoutant les informations de vérification d'intégrité aux identifiants de fragment. Ces informations sont utilisées pour détecter les changements dans la ressource. Les applications peuvent alors avertir les utilisateurs de la possibilité qu'un identifiant de fragment ait pu être cassé par une modification de la ressource.

Les identifiants de fragment sont interprétés par les clients, et donc les informations de vérification d'intégrité sont définies sur les entités MIME plutôt que sur la ressource elle-même. Cela signifie que les informations de vérification d'intégrité sont spécifiques d'une certaine entité. Précisément, les codages de contenu et/ou les codages de transfert de contenu doivent être retirés avant d'utiliser les informations de vérification d'intégrité.

Les informations de vérification d'intégrité peuvent spécifier le codage de caractère qui a été utilisé à la création de l'information, et si une telle spécification est présente, les clients DOIVENT vérifier si le codage de caractère spécifié et le codage de caractère de l'entité MIME restituée sont égaux, et les clients NE DOIVENT PAS utiliser les informations de vérification d'intégrité si ces valeurs diffèrent. Cependant, les clients PEUVENT choisir de transcoder l'entité MIME restituée dans le cas de codage de caractères différent, et après l'avoir fait, appliquer les vérifications d'intégrité. Noter que cette méthode n'est pas fiable par nature, parce que certains caractères ou séquences de caractères peuvent avoir été perdues ou normalisées à cause de restrictions dans un des codages de caractères utilisés.

3. Syntaxe d'identification de fragment

La syntaxe des identifiants de fragment text/plain est toute simple. Elle définit quatre schémas, "char", "line", et "integrity-check" (qui peut être "length" ou "md5"). Les schémas "char" et "line" peuvent être utilisés dans deux variantes différentes, soit la variante de position (avec un seul nombre) soit la variante gamme (avec deux nombres séparés par une virgule). Une vérification d'intégrité peut utiliser le schéma "length" ou "md5" pour spécifier une valeur. "length" dans ce cas sert de moyen très faible mais facile pour calculer la vérification d'intégrité.

La définition de syntaxe suivante utilise l'ABNF comme défini dans la [RFC5234], incluant les règles DIGIT et HEXDIG. La règle mime-charset est définie dans la [RFC2978].

Note : dans les descriptions qui suivent, les valeurs de texte spécifiées DOIVENT être utilisées exactement comme elles sont données, utilisant exactement les lettres minuscules indiquées. À cet égard, l'usage de l'ABNF diffère de celui de la [RFC5234].

```
text-fragment = text-scheme 0*( ";" integrity-check )
text-scheme   = ( char-scheme / line-scheme )
char-scheme   = "char=" ( position / range )
line-scheme   = "line=" ( position / range )
integrity-check = ( length-scheme / md5-scheme ) [ ";" mime-charset ]
position      = number
range         = ( position "," [ position ] ) / ( ";" position )
number        = 1*( DIGIT )
length-scheme = "length=" number
md5-scheme    = "md5=" md5-value
md5-value     = 32HEXDIG
```

3.1 Vérifications d'intégrité

Une vérification d'intégrité peut spécifier la longueur d'une entité MIME, ou son empreinte MD5. Dans les deux cas, elle peut facultativement spécifier le codage de caractère qui a été utilisé lors du calcul de la vérification d'intégrité, afin que les clients qui interprètent l'identifiant de fragment puissent vérifier si ils utilisent le même codage de caractère pour leurs calculs. Pour les longueurs, le codage de caractère peut être nécessaire parce que il peut influencer le compte de caractères.

Par exemple, Unicode inclut des caractères précomposés pour écrire le vietnamien, mais dans le codage windows-1258, aussi utilisé pour écrire le vietnamien, certains caractères doivent être codés avec des signes diacritiques séparés, ce qui signifie que deux caractères vont être comptés. En appliquant la terminologie Unicode, cela signifie que la longueur d'une entité MIME text/plain est calculée sur la base de ses "codets". Pour les empreintes MD5, le codage de caractère est nécessaire parce que l'algorithme MD5 fonctionne sur la représentation binaire de la ressource text/plain.

Pour permettre que de futurs changements à la présente spécification suivent les développements de la cryptographie, les mises en œuvre DOIVENT ignorer les nouveaux types de vérifications d'intégrité, avec des noms autres que "length" et "md5". Si plusieurs vérifications d'intégrité sont présentes, une application peut utiliser toute vérification d'intégrité qu'elle comprend, et parmi elles, les vérifications d'intégrité qui donnent un compromis approprié entre performances et besoin de vérification d'intégrité. Voir plus de détails au paragraphe 4.3.

La longueur d'une entité MIME text/plain est calculée en utilisant les principes définis au paragraphe 2.1.2. L'empreinte MD5 d'une entité MIME text/plain est calculée en utilisant l'algorithme présenté dans la [RFC1321], codant le résultat en 32 chiffres hexadécimaux (utilisant des lettres majuscules ou minuscules) comme représentation des 128 bits qui sont le résultat de l'algorithme MD5. Le calcul des vérifications d'intégrité est fait après avoir supprimé tout codage de contenu ou de transfert de contenu potentiel du mécanisme de transport.

4. Traitement d'identifiant de fragment

Les applications qui mettent en œuvre le mécanisme décrit dans le présent mémoire DOIVENT se comporter comme décrit dans les paragraphes qui suivent.

4.1 Traitement des fins de lignes dans les entités text/plain MIME

Dans les messages Internet, les terminaisons de ligne dans les entités MIME text/plain sont représentées par les séquences de caractères CR+LF (voir les [RFC2046] et [RFC3676]). Cependant, certains protocoles (comme HTTP) permettent de plus d'autres conventions pour les terminaisons de ligne. Aussi, certains systèmes d'exploitation mémorisent les entités text/plain en local avec des terminaisons de ligne différentes (dans la plupart des cas, Unix utilise LF, MacOS utilise traditionnellement CR, et Windows utilise CR+LF).

Indépendamment du nombre d'octets ou de caractères utilisé pour représenter une fin de ligne, chaque fin de ligne DOIT être comptée comme un seul caractère. Les mises en œuvre qui interprètent les identifiants de fragment text/plain DOIVENT prendre en compte les conventions de fin de ligne des protocoles et autres contextes dans lesquels ils travaillent.

Par exemple, une mise en œuvre qui travaille dans le contexte d'un navigateur de la Toile qui prend en charge des URI http: doit prendre en charge les diverses conventions de fin de ligne permises par HTTP. Dans un autre exemple, une mise en œuvre utilisée sur des fichiers locaux (par exemple, avec le schéma d'URI file:) doit prendre en charge les conventions utilisées pour la mémorisation locale. Toutes les mises en œuvre DEVRAIENT prendre en charge la convention de terminaison de ligne CR+LF sur tout l'Internet, et PEUVENT prendre en charge des conventions supplémentaires sans relation avec les protocoles ou systèmes avec lesquels elles travaillent.

Les mises en œuvre devrait être conscientes du fait que les terminaisons de ligne dans les entités text/plain peuvent être représentées par d'autres caractères ou séquences de caractères que CR+LF. À côté des CR et LF susmentionnés, il y a aussi NEL et CR+NEL. En général, le codage des terminaisons de ligne peut aussi dépendre du codage de caractère de l'entité MIME, et les mises en œuvre ont à prendre cela en compte lorsque nécessaire.

4.2 Traitement des valeurs de position

Si une valeur de position (comme position ou comme partie d'une gamme) est supérieure à la longueur de l'entité MIME réelle, alors elle identifie la dernière position de caractère ou position de ligne de l'entité MIME. Si la première valeur de position dans une gamme est absente, la gamme s'étend depuis le début de l'entité MIME. Si la seconde valeur de position dans une gamme est absente, alors la gamme s'étend jusqu'à la fin de l'entité MIME. Si les positions d'un schéma de gamme ne sont pas ordonnées de façon appropriée (c'est-à-dire, si le premier nombre est inférieur au second) l'identifiant de fragment DOIT alors être ignoré.

4.3 Vérifications d'intégrité de fragment

Les clients ne sont pas obligés de mettre en œuvre le traitement des vérifications d'intégrité, et ils PEUVENT donc choisir d'ignorer les informations de vérification d'intégrité. Cependant, si ils mettent en œuvre la vérification d'intégrité, ce qui suit s'applique :

Si un identifiant de fragment contient une ou plusieurs vérifications d'intégrité, et si un client restitue une entité MIME et, en utilisant une ou des vérifications d'intégrité, détecte que l'entité a changé (en observant la spécification de codage de caractère décrite au paragraphe 3.1, si elle est présente) le client NE DEVRAIT alors PAS interpréter l'identifiant de fragment text/plain. Un client PEUT signaler cette situation à l'utilisateur.

4.4 Erreurs de syntaxe dans les identifiants de fragment

Si un identifiant de fragment contient une erreur de syntaxe (c'est-à-dire, ne se conforme pas à la syntaxe spécifiée à la Section 3) il DOIT alors être ignoré par les clients. Les clients NE DOIVENT PAS tenter de corriger ou deviner les identifiants de fragment. Les erreurs de syntaxe PEUVENT être rapportées par les clients.

5. Exemples

Les exemples suivants montrent certains usages des identifiants de fragment définis dans ce mémoire.

`http://exemple.com/text.txt#char=100`

Cet URI identifie la position après le 100^{ième} caractère de l'entité MIME text.txt. On devrait noter que sans restituer l'entité MIME on ne sait pas clairement quels octets de l'entité MIME cela va être parce qu'on ne sait pas quel codage de caractère est utilisé (dans le cas de HTTP, cette information va être donnée dans l'en-tête Content-Type de la réponse). Si l'entité MIME a moins de 100 caractères, l'URI identifie la position après le dernier caractère de l'entité MIME.

`http://exemple.com/text.txt#line=10,20`

Cet URI identifie les lignes 11 à 20 de l'entité MIME text.txt. Si l'entité MIME a moins de 11 lignes, il identifie la position après la dernière ligne. Si l'entité MIME a moins de 20 lignes mais au moins 11, il identifie la gamme de la ligne 11 à la dernière ligne de l'entité MIME.

`https://exemple.com/text.txt#line=,1`

Cet URI identifie la première ligne. Noter que le schéma d'URI a été changé en https.

`ftp://exemple.com/text.txt#line=10,20;length=9876,UTF-8`

Comme dans le second exemple, cet URI identifie les lignes 11 à 20 de l'entité MIME text.txt. La longueur supplémentaire de vérification d'intégrité spécifie que l'entité MIME a une longueur de 9876 caractères quand elle est codée en UTF-8. Si le client prend en charge le schéma de longueur, il peut vérifier la longueur de l'entité MIME restituée, mais seulement si l'entité MIME restituée utilise le codage UTF-8 ou a été localement transcodée dans ce codage.

Noter que le protocole FTP, ainsi que certains autres protocoles sous-jacents à certains autres schémas d'URI, ne fournit pas d'informations explicites sur le type de support de la ressource restituée. Utiliser des identifiants de fragment avec de tels schémas d'URI est donc non fiable par nature. Les agents d'utilisateur courants utilisent diverses heuristiques pour déduire le type de support pour la suite du traitement. Le traitement de l'identifiant de fragment selon le présent mémoire n'est approprié que si le type de support déduit est text/plain.

6. Considérations relatives à l'IANA

L'IANA a ajouté une référence à la présente spécification dans le registre des types de support text/plain.

7. Considérations sur la sécurité

Le fait que les logiciels qui mettent en œuvre les identifiants de fragment pour text/plain et ceux qui ne les mettent pas en œuvre aient des comportements différents, et le fait que différents logiciels puissent montrer des documents ou fragments aux utilisateurs de différentes façons, peut conduire à de l'incompréhension de la part des utilisateurs. Cette incompréhension pourrait être exploitée d'une façon similaire à l'usurpation d'identité ou à l'ameçonnage.

En particulier, il faut faire attention si des identifiants de fragment sont utilisés avec un mécanisme qui permet de montrer seulement la partie d'un document identifiée par un fragment. Un scénario peut être l'utilisation d'un identifiant de fragment pour cacher du texte légal en petits caractères. Un autre scénario peut être l'inclusion de matériel comme des clés de site, qui peut donner à l'utilisateur l'impression d'utiliser le site réel plutôt qu'un faux site ; d'autres scénarios sont aussi possibles. Des contre-mesures possibles peuvent inclure mais ne sont pas limitées à afficher le contenu inclus dans des limites clairement visibles et de limiter l'inclusion à du matériel provenant du même domaine de sécurité ou de domaines qui donnent la permission explicite d'être inclus dans un autre domaine.

Noter que les questions ci-dessus s'appliquent au côté client ; les identifiants de fragment ne sont pas utilisés dans la résolution d'un URI pour restituer la représentation d'une ressource, mais sont seulement appliqués au côté client.

Les mises en œuvre et utilisateurs d'identifiants de fragment pour text/plain devraient aussi avoir connaissance des considérations sur la sécurité des [RFC3986] et [RFC3987].

8. Références

8.1 Références normatives

- [RFC1321] R. Rivest, "Algorithme de [résumé de message MD5](#)", avril 1992. (*Information*)
- [RFC2045] N. Freed et N. Borenstein, "[Extensions de messagerie Internet](#) multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. (*D. S.*, *MàJ par* [2184](#), [2231](#), [5335](#).)
- [RFC2046] N. Freed et N. Borenstein, "[Extensions de messagerie Internet](#) multi-objets (MIME) Partie 2 : Types de support", novembre 1996. (*D. S.*, *MàJ par* [2646](#), [3798](#), [5147](#), [6657](#), [8098](#))
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (*MàJ par* [RFC8174](#))
- [RFC2978] N. Freed et J. Postel, "Procédures d'enregistrement des jeux de caractère par l'IANA", BCP 19, octobre 2000.
- [RFC3676] R. Gellens, "[Format Text/Plain et paramètres DelSp](#)", février 2004. (*Remplace* [RFC2646](#)) (*P.S.*)
- [RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "[Identifiant de ressource uniforme](#) (URI) : Syntaxe générique", STD 66, janvier 2005. (*P.S.* ; *MàJ par* [RFC8820](#))
- [RFC3987] M. Dürst et M. Suignard, "[Identifiant de ressource internationalisé](#) (IRI)", janvier 2005.
- [RFC5234] D. Crocker, P. Overell, "[BNF augmenté pour les spécifications de syntaxe](#) : ABNF", janvier 2008. ([STD0068](#))

8.2 Références pour information

- [ANSI] ANSI X3.4-1986, "Coded Character Set - 7-Bit American National Standard Code for Information Interchange".
- [RFC2781] P. Hoffman et F. Yergeau, "[UTF-16](#), un codage de la norme ISO 10646", février 2000.
- [RFC2854] D. Connelly et L. Masinter, "Type de support 'text/html'", juin 2000. (*Information*)
- [RFC3629] F. Yergeau, "[UTF-8, un format de transformation](#) de la norme ISO 10646", STD 63, novembre 2003.
- [RFC4288] N. Freed et J. Klensin, "Spécifications du [type de support et procédures d'enregistrement](#)", [BCP 13](#), décembre 2005.
- [XML] DeRose, S., Maler, E., et D. Orchard, "XML Linking Language (XLink) Version 1.0", World Wide Web Consortium Recommendation, juin 2001, <<http://www.w3.org/TR/xlink/>>.

Appendice A. Remerciements

Merci à Marcel Baschnagel, Stephane Bortzmeyer, Tim Bray, Iain Calder, John Cowan, Spencer Dawkins, Lisa Dusseault, Benja Fallenstein, Ted Hardie, Sam Hartman, Sandro Hawke, Jeffrey Hutzelman, Cullen Jennings, Graham Klyne, Dan Kohn, Henrik Levkowetz, Chris Newman, Mark Nottingham, Conrad Parker, et Tim Polk pour leurs commentaires et suggestions.

Adresse des auteurs

Erik Wilde
UC Berkeley
School of Information, 311 South Hall
Berkeley, CA 94720-4600
U.S.A.
téléphone : +1-510-6432253
mél : dret@berkeley.edu
URI : <http://dret.net/netdret/>

Martin Dürst
Aoyama Gakuin University
5-10-1 Fuchinobe
Sagamihara, Kanagawa 229-8558
Japan
téléphone : +81 42 759 6329
mél : duerst@it.aoyama.ac.jp
URI : <http://www.sw.it.aoyama.ac.jp/D%C3%BCrst/>

Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2008).

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

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

Propriété intellectuelle

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

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