

Groupe de travail Réseau
Request for Comments : 5259
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

A. Melnikov, éditeur, Isode Limited
 P. Coates, éditeur, Sun Microsystems

juillet 2008

Extension CONVERT au protocole d'accès au message Internet

Statut du présent mémoire

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

Résumé

CONVERT définit des extensions à IMAP qui permettent aux clients de demander une adaptation et/ou transcodage des pièces jointes. Les clients peuvent spécifier les détails de conversion ou permettre aux serveurs de décider sur la base de la connaissance des capacités du client, des préférences de l'utilisateur ou administrateur, ou des réglages du serveur.

Table des matières

1. Introduction.....	1
2. Conventions utilisées dans ce document.....	2
3. Relation avec les autres spécifications IMAP.....	2
3.1 Réponse CAPABILITY.....	2
4. Portée des conversions.....	2
5. Découverte des conversions disponibles.....	3
5.1 Commande CONVERSIONS.....	3
5.2 Réponse à CONVERSION.....	3
6. Commandes CONVERT et UID CONVERT.....	4
7. Paramètres de conversion CONVERT.....	7
7.1 Conversions et paramètres de conversion de mise en œuvre obligatoire.....	7
7.2 Caractéristiques supplémentaires pour l'usage mobile.....	8
8. Éléments de données de demande/réponse aux commandes CONVERT/UID CONVERT.....	8
8.1 Réponse CONVERTED non étiquetée.....	8
8.2 Élément de demande et réponse BODYPARTSTRUCTURE CONVERT.....	8
8.3 Élément de demande et réponse BINARY.SIZE CONVERT.....	9
8.4 Élément de demande et réponse AVAILABLECONVERSIONS CONVERT.....	9
8.5 Considérations de mise en œuvre.....	10
9. Extensions de réponses d'état et de code de réponse.....	10
10 Syntaxe formelle.....	12
11. Considérations sur la gestion.....	14
12. Considérations relatives à l'IANA.....	14
12.1 Enregistrement de paramètre de type de support unknown-character-replacement.....	14
13. Considérations pour la sécurité.....	15
14. Remerciements.....	15
15. Références.....	15
15.1 Références normatives.....	15
15.2 Références pour information.....	16
Adresse des auteurs.....	16
Déclaration complète de droits de reproduction.....	17

1. Introduction

Le présent document définit l'extension CONVERT à IMAP4 [RFC3501]. CONVERT fournit l'adaptation et le transcodage de parties de corps nécessaires au client. La conversion (adaptation, transcodage) peut être demandée par le client et effectuée par le serveur au mieux ou, quand c'est demandé par le client, décidée par le serveur sur la base de la

connaissance par le serveur des capacités du client, des préférences de l'utilisateur ou de l'administrateur, ou des réglages du serveur.

Cette extension est principalement destinées à être utile aux clients mobiles. Elle satisfait les exigences spécifiées dans [OMA-ME-RD].

Un serveur qui prend en charge CONVERT peut convertir des parties de corps dans d'autres formats pour être vus (par exemple) sur un appareil mobile. Le client peut demander explicitement une conversion particulière ou demander au serveur de choisir la meilleure conversion disponible. Quand c'est permis par le client, le serveur détermine comment convertir sur la base de sa propre stratégie (par exemple, sur la base de la connaissance du client comme expliqué ci après). Si le serveur connaît les caractéristiques de l'appareil (ce qui sort du domaine d'application de CONVERT) ou peut les déterminer (par exemple, en utilisant un paramètre de conversion contenant le type d'appareil) les parties de corps converties peuvent aussi être optimisées pour les capacités de l'appareil (par exemple, le facteur de forme pour les images). Le client est capable de contrôler les conversions en utilisant des paramètres de conversion facultatifs (aussi appelé un "transcodage" dans ce document).

Le présent document s'appuie sur le registre des paramètres de conversion établi par la [RFC2506]. Le registre peut être utilisé pour découvrir les valeurs légales sous-jacentes que peuvent prendre ces paramètres. Des paramètres de conversion supplémentaires, comme ceux définis par [OMA-STI], sont supposés être enregistrés à l'avenir.

2. Conventions utilisées dans ce document

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

Dans les exemples, "C:" et "S:" indiquent respectivement les lignes envoyées par le client et le serveur. Si une seule étiquette "C:" ou "S:" s'applique à plusieurs lignes, les sauts à la ligne entre ces lignes sont seulement pour faciliter la lecture et ne font pas partie de l'échange de protocole réel. Les cinq caractères [...] signifient que quelque chose a été coupé.

Pour la description de la syntaxe générale, certaines définitions sont omises parce que définies dans la [RFC3501]. En particulier, le terme "session" est utilisé dans le présent document comme défini au paragraphe 1.2 de la [RFC3501].

3. Relation avec les autres spécifications IMAP

La conversion de pièces jointes dans des flux directs sort du domaine d'application de l'extension CONVERT et est décrite dans un autre document du groupe de travail Lemonade [RFC5616].

Un serveur qui revendique la conformité à la présente spécification DOIT prendre en charge la spécification de IMAP binaire [RFC3516].

3.1 Réponse CAPABILITY

Un serveur qui prend en charge l'extension CONVERT DOIT retourner "CONVERT" et "BINARY" dans la réponse ou le code de réponse CAPABILITY. (On rappelle aux auteurs de client et de serveur que l'ordre des jetons retournés dans une réponse ou code de réponse CAPABILITY est arbitraire.)

Exemple d'un serveur qui met en œuvre CONVERT :

```
C: a000 CAPABILITY
S: * CAPABILITY IMAP4rev1 CONVERT BINARY [...]
S: a000 OK CAPABILITY achevé
```

4. Portée des conversions

Les conversions affectent seulement ce qui est envoyé au client ; les données originales dans le magasin de messages NE DOIVENT PAS être altérées. Le présent document ne spécifie pas comment le serveur effectue les conversions.

Note : l'exigence que les données originales soient non altérées permet que de telles données restent accessibles aux autres clients, permet des réponses ou des retransmissions des documents d'origine, permet la vérification des signatures (les parties de corps converties ne vont probablement pas contenir de signatures) et préserve les BODYSTRUCTURE et informations qui s'y rapportent.

5. Découverte des conversions disponibles

5.1 Commande CONVERSIONS

Arguments : type MIME de source ; type MIME cible

Réponses : réponses non étiquetées : CONVERSION

Résultat : OK – commande CONVERSIONS achevée

BAD – syntaxe non reconnue d'un argument, argument supplémentaire inattendu, argument manquant, etc.

La commande CONVERSIONS est permise dans les états IMAP Authentifié et Choisi.

Le premier paramètre à la commande CONVERSIONS est un type MIME de source, le second paramètre est le type MIME cible. Les deux paramètres sont partiellement (par exemple, "text/*") ou complètement ("*") remplaçables par un caractère générique.

Les conversions qui correspondent à la paire source/cible et leurs paramètres de conversion associés sont retournées dans des réponses CONVERSION non étiquetées. Si la source/cible ne correspond à aucune conversion prise en charge par le serveur, aucune réponse CONVERSION n'est retournée.

Exemples :

Pour des informations de conversion du format d'image GIF en JPEG (aucune réponse CONVERSION non étiquetée ne va être retournée si aucune conversion n'est possible) :

```
C: a CONVERSIONS "image/gif" "image/jpeg"
S: * CONVERSION "image/gif" "image/jpeg" ("pix-y" "pix-x" "image-interleave")
S: a OK CONVERSIONS achevée
```

Pour des informations de conversion du format d'image en n'importe quoi :

```
C: b CONVERSIONS "image/gif" "*"
S: * CONVERSION "image/gif" "image/jpeg" ("pix-y" "pix-x" "image-interleave")
S: * CONVERSION "image/gif" "image/png" ([...])
[...]
S: b OK CONVERSIONS achevée
```

Pour la conversion de n'importe quoi en JPEG :

```
C: c CONVERSIONS "*" "image/jpeg"
S: * CONVERSION "image/gif" "image/jpeg" ("pix-y" "pix-x" "image-interleave")
S: * CONVERSION "image/png" "image/jpeg" (...)
[...]
S: c OK CONVERSIONS achevée
```

Pour les conversions de tous formats d'image en tous formats de texte, le client peut produire la commande suivante :

```
C: d CONVERSIONS "image/*" "text/*"
```

5.2 Réponse à CONVERSION

Contenu : type MIME de source ; type MIME cible ; liste facultative des paramètres de conversion pris en charge

Par suite de l'exécution d'une commande CONVERSIONS, le serveur peut retourner une ou plusieurs réponses CONVERSION. Chaque réponse CONVERSION spécifie quel type MIME de source peut être converti en le type MIME cible, et fait aussi la liste des paramètres de conversion pris en charge.

6. Commandes CONVERT et UID CONVERT

Arguments : ensemble de séquences ; paramètres de conversion ; noms d'éléments de données CONVERT

Réponses : réponses non étiquetées CONVERTED

Résultat : OK – conversion achevée

NO – erreur de conversion : ne peut pas aller chercher et/ou convertir ces données

BAD – syntaxe non reconnue d'un argument, argument supplémentaire inattendu, argument manquant, etc.

L'extension CONVERT définit les commandes CONVERT et UID CONVERT qui sont utilisées pour transcoder le type de support d'une partie MIME en un autre type de support, et/ou le même type de support avec différents paramètres de codage. Ces commandes sont structurées et se comportent de façon similaire aux commandes FETCH/UID FETCH telles que étendues par la [RFC3516] :

- o Une commande CONVERT/UID CONVERT réussie résulte en une ou plusieurs réponses CONVERTED non étiquetées (une par message). Elles sont similaires aux réponses FETCH non étiquetées. Note qu'une seule commande CONVERT/UID CONVERT peut seulement effectuer un seul type de conversion comme défini par les paramètres de conversion. Un client qui a besoin d'effectuer plusieurs conversions différentes doit produire plusieurs commandes CONVERT/UID CONVERT. Un tel client PEUT les traiter en parallèles.
- o Les éléments de données BINARY[...] demandent la conversion d'une partie de corps ou de tout le message selon les paramètres de conversion et demandent que le message/partie de corps converti soit retourné en binaire.
- o L'élément de données BINARY.SIZE est similaire à RFC822.SIZE, mais il demande la taille d'une partie de corps/message converti.
- o L'élément de données BODYPARTSTRUCTURE est similaire à l'élément de données BODYSTRUCTURE FETCH, mais il retourne la structure MIME de la partie de corps convertie.
- o Les mots codés BODY[...HEADER] dans les en-têtes demandés sont convertis dans le jeu de caractères spécifié. Le paramètre CHARSET est EXIGÉ pour cette conversion.
- o Les mots codés BODY[...MIME] dans les en-têtes demandés sont convertis dans le jeu de caractères spécifié. Le paramètre CHARSET est EXIGÉ pour cette conversion.
- o L'élément de données AVAILABLECONVERSIONS demande la liste des types MIME cibles dans lesquels la partie de corps spécifiée (ou du message complet) peut être convertie.

L'extension CONVERT ajoute aussi un nouveau code de réponse. Voir les détails à la Section 9.

Les clients vont normalement demander la conversion des parties de corps terminales. En plus de prendre en charge la conversion des parties de corps terminales, les serveurs PEUVENT offrir la conversion de parties de corps non terminales (par exemple, la conversion de multipart/related).

Au lieu de spécifier le type exact de support MIME cible en lequel le client veut convertir, le client PEUT utiliser un marqueur NIL spécial (aussi appelé une "conversion par défaut") pour demander au serveur de prendre un type de support cible convenable. Le présent document ne décrit pas exactement comment le serveur fait un tel choix ; cependant, ce paragraphe donne quelques lignes directrices. Si le serveur connaît les caractéristiques de l'appareil en utilisant un mécanisme dans la bande (comme un type d'appareil spécifié dans un paramètre de conversion) ou hors bande, il devrait alors convertir le corps de la demande en un type de support que l'appareil a des chances de prendre en charge et d'afficher/exécuter avec succès. Sauf spécifiquement outrepassé par un paramètre de conversion, le serveur PEUT aussi

supprimer tout détail non indispensable qui excéderait les capacités de l'appareil (par exemple, adapter les images pour qu'elles tiennent juste sur l'écran de l'appareil). En l'absence de tout mécanisme dans la bande ou hors bande pour déterminer les caractéristiques de l'appareil, le serveur devrait convertir la partie de corps de la demande en le type de support le plus standard ou largement déployé disponible dans cette catégorie de supports, par exemple, de convertir en text/plain, image/jpeg. Dans ce cas, le serveur devrait minimiser la perte de qualité. Il est EXIGÉ des serveurs qu'ils prennent en charge les demandes de "conversion par défaut". Les mises en œuvre de serveur qui prennent en charge les conversions en plusieurs types MIME cible DEVRAIENT rendre la conversion par défaut configurable. Les clients DEVRAIENT éviter d'utiliser la conversion par défaut sauf si ils fournissent un moyen (dans la bande ou hors bande) pour signaler leurs capacités au serveur, car il n'est pas garanti que le serveur puisse deviner correctement leurs capacités. Les mises en œuvre de client devraient envisager d'utiliser l'élément de données AVAILABLECONVERSIONS CONVERT ou la commande CONVERSIONS plutôt que la conversion par défaut.

La syntaxe de la commande CONVERT est modélisée d'après la syntaxe de la commande FETCH dans la [RFC3501], et étendue par la [RFC3516]. Les éléments de données CONVERT sont généralement structurés comme :

```
BINARY[section-part]<partial>
BINARY.SIZE[section-part]
BODYPARTSTRUCTURE[section-part]
BODY[HEADER]
BODY[section-part.HEADER]
BODY[section-part.MIME]
AVAILABLECONVERSIONS[section-part]
```

La sémantique d'une commande partielle CONVERT BINARY[...] est la même que pour une commande partielle FETCH BODY[...], à l'exception que les arguments <partial> se réfèrent aux données de section TRANSCODED et DECODED.

Noter qu'à la différence de la commande FETCH, la commande CONVERT n'établit jamais le fanion \Seen sur les messages convertis. Un client qui souhaite marquer un message avec le fanion \Seen devrait pour le faire produire une commande STORE (éventuellement traitée en parallèle avec la demande CONVERT).

La commande UID CONVERT est différente de la commande CONVERT de la même façon que la commande UID FETCH est différente de la commande FETCH :

- o UID CONVERT prend comme paramètre une séquence d'UID au lieu d'une séquence de numéros de message.
- o La commande UID CONVERT DOIT résulter en l'élément de données UID dans une réponse CONVERTED correspondante.
- o Une réponse EXPUNGE NE DOIT PAS être envoyée lors d'une réponse à une commande CONVERT. Cette règle est nécessaire pour empêcher une perte de synchronisation des numéros de séquence des messages entre client et serveur. Noter qu'une réponse EXPUNGE PEUT être envoyée durant une commande UID CONVERT.

Exemple : le client va chercher la section 3 de la partie de corps dans le message avec le numéro de séquence de message 2 et demande que la pièce jointe soit convertie en format pdf.

```
C: a001 CONVERT 2 ("APPLICATION/PDF") BINARY[3]
S: * 2 CONVERTED (TAG "a001") (BINARY[3] {2135} <the document in .pdf format> )
S: a001 OK CONVERT COMPLETED
```

Exemple : le client demande la conversion d'une partie de corps text/html en text/plain et demande un jeu de caractères de us-ascii. Le serveur ne peut pas satisfaire la demande de conversion de jeu de caractères parce que il y a des caractères non us-ascii dans la partie de corps text/html, donc il fait échouer la demande en retournant une phrase ERROR à la place des données converties (voir la Section 9).

```
C: b001 CONVERT 2 ("text/plain" ("charset" "us-ascii")) BINARY[3]
S: * 2 CONVERTED (tag "b001") (BINARY[3]
  (ERROR "Le texte source a du non us-ascii" BADPARAMETERS
    "text/html" "text/plain" ("charset" "us-ascii")))
S: b001 NON Toutes les conversions ont échoué
```

Si le client avait aussi spécifié le paramètre de conversion "remplacement des caractères inconnus" (voir le paragraphe 12.1) le même exemple pourrait ressembler à :

```
C: b001 CONVERT 2 ("text/plain" ("charset" "us-ascii" "remplacement des caractères inconnus" "?")) BINARY[3]
S: * 2 CONVERTED (TAG "b001") (BINARY[3] {2135} <le document en format text/plain avec le jeu de caractères us-ascii> )
S: b001 OK CONVERT COMPLETED
```

Le serveur a remplacé les caractères non us-ascii par un caractère us-ascii comme "?".

Exemple : Le client demande d'abord la taille convertie d'une partie de corps text/html convertie en text/plain :

```
C: c000 CONVERT 2 ("TEXT/PLAIN" ("CHARSET" "us-ascii")) BINARY.SIZE[4]
S: * 2 CONVERTED (TAG "c000") (BINARY.SIZE[4] 3135)
S: c000 OK CONVERT COMPLETED
```

Plus tard, le client demande 1000 octets provenant de la partie de corps convertie, en commençant à l'octet 2001 :

```
C: c001 CONVERT 2 ("TEXT/PLAIN" ("CHARSET" "us-ascii")) BINARY[4]<2001.1000>
S: * 2 CONVERTED (TAG "c001") (BINARY[4]<2001> {135} <octets 2001 - 2135 du document en format text/plain>)
S: c001 OK CONVERT COMPLETED
```

Le serveur **DOIT** respecter le type MIME cible et les paramètres de conversion spécifiés par le client dans la demande de transcodage. Noter que certains paramètres de conversion peuvent restreindre la sorte de conversion possible, tandis que d'autres peuvent supprimer certaines restrictions.

Il est légal qu'un client demande la conversion d'une partie de corps non terminale, par exemple, pour demander la conversion d'un multipart/* en document PDF. Cependant, il n'est pas exigé des serveurs qui mettent en œuvre cette extension qu'ils prennent en charge de telles conversions. Les serveurs qui prennent en charge ces conversions **DOIVENT** retourner une ou plusieurs réponses **CONVERSION** en réponse à une commande **"CONVERSIONS" multipart/* "*****"**. Voir les détails au paragraphe 5.1.

Le client peut demander des conversions d'en-tête en utilisant la demande **BODY[...HEADER] CONVERT**, par exemple :

```
C: D001 FETCH 2 BODY[HEADER]
S: * 2 FETCH (BODY[HEADER] {158}
S: Date: Mon, 20 Apr 2007 20:05:43 +0200
S: From: Peter <peter@siroe.exemple.com>
S: To: Alexey <alexey@siroe.exemple.com>
S: Subject: =?KOI8-R?Q?pourquoi coder cela ?=
S:
S: )
S: D001 OK
C: D002 CONVERT 2 (NIL ("CHARSET" "utf-8")) BODY[HEADER]
S: * 2 CONVERTED (TAG "d002") (BODY[HEADER] {157}
S: Date: Mon, 20 Apr 2007 20:05:43 +0200
S: From: Peter <peter@siroe.exemple.com>
S: To: Alexey <alexey@siroe.exemple.com>
S: Subject: =?UTF-8?Q?pourquoi coder cela ?=
S:
S: )
S: D002 OK
```

Une telle demande **DOIT** inclure le paramètre **CHARSET**. À réception de la demande, le serveur **DOIT** décoder tous les mots codés (comme décrit dans la [RFC2047]) dans les en-têtes et les retourner re-codés dans le jeu de caractères spécifié. (Noter que les mots codés pourraient n'être pas nécessaires si le résultat peut être représenté entièrement en US-ASCII, de sorte que le serveur **PUISSE** remplacer les mots codés résultants par leur pure représentation US-ASCII.) Si le serveur ne peut pas décoder un mot codé particulier, par exemple, si le jeu de caractères ou le codage n'est pas reconnu, il **DOIT** le laisser comme il est. Les serveurs **DEVRAIENT** aussi prendre en charge le décodage de tous les paramètres comme décrit

dans la [RFC2231]. La prise en charge des paramètres de la RFC 2231 pourrait exiger de reformater les champs d'en-tête durant la conversion. Considérons ce qui suit :

```
C: D011 FETCH 3 BODY[1.MIME]
S: * 3 FETCH (BODY[1.MIME] {118}
S: Content-Type: text/plain; charset=utf-8;
S: foo*0*=utf-8'fr'tr%c0;
S: foo*1*(very)=%03s_m%c0;
S: foo*2*=(nasty)%09chant
S:
S: D011 OK
```

Le serveur devrait autant que possible préserver les en-têtes durant la conversion. Dans le cas où les caractères sont séparés (légalement !) entre des fragments d'un paramètre codé, le serveur DOIT consolider les fragments de paramètre, et les convertir, émettre, et refragmenter comme nécessaire afin de garder la longueur de ligne à moins de 78. Les commentaires incorporés comme cela DEVRAIENT être préservés durant la conversion, mais les clients DOIVENT traiter en douceur la situation où les commentaires sont entièrement supprimés. Si les commentaires sont préservés, ils PEUVENT être déplacés après le paramètre. Par exemple (en continuant l'exemple précédent) :

```
C: D012 CONVERT 3 (NIL) BODY[1.MIME]
S: * 3 CONVERTED (TAG "D012") (BODY[1.MIME] {109}
S: Content-Type: text/plain; charset=utf-8;
S: foo*0*=utf-8'fr'tr%c0%03s_ ;
S: foo*1*=%m%c0%09chant (très)(méchant)
S:
S: D012 OK
```

Aucun type de destination MIME ne DOIT être spécifié avec BODY[HEADER], BODY[section.HEADER], ou BODY[section.MIME]. C'est-à-dire, BODY[HEADER], BODY[section.HEADER], ou BODY[section.MIME] peuvent seulement être utilisés avec la "conversion par défaut". Quand il effectue ces conversions, le serveur DEVRAIT laisser les mots codés comme des mots codés. Manquer à le faire peut altérer la sémantique des en-têtes structurés.

7. Paramètres de conversion CONVERT

Le registre établi par la [RFC2506] définit les noms des paramètres de conversion qui peuvent être utilisés dans la commande CONVERT. La prise en charge de certains paramètres de conversion est obligatoire, comme décrit au paragraphe 7.1.

Selon la [RFC2506], les noms des paramètres de conversion sont insensibles à la casse.

L'exemple suivant illustre comment les dimensions de l'image cible peuvent être spécifiées dans une demande CONVERT en utilisant les paramètres PIX-X et PIX-Y définis dans la [RFC2534].

```
C: e001 UID CONVERT 100 ("IMAGE/JPEG" ("PIX-X" "128" "PIX-Y" "96")) BINARY[2]
S: * 2 CONVERTED (TAG "e001") (UID 100 BINARY[2] ~{4182}
  <cette partie d'un document est une image adaptée en format JPEG avec largeur=128, hauteur=96.> )
S: e001 OK UID CONVERT COMPLETED
```

7.1 Conversions et paramètres de conversion de mise en œuvre obligatoire

Un serveur qui met en œuvre CONVERT DOIT prendre en charge les conversions de jeu de caractère pour le type MIME text/plain, et DOIT prendre en charge les conversions de jeu de caractère de iso-8859-1, iso-8859-2, iso-8859-3, iso-8859-4, iso-8859-5, iso-8859-6, iso-8859-7, iso-8859-8, et iso-8859-15 en utf-8.

Le serveur DOIT mentionner "text/plain" comme une conversion de destination permise du type MIME "text/plain" (voir le paragraphe 5.1). Une commande "CONVERSIONS "text/plain" "text/plain"" DOIT aussi retourner "charset" et "unknown-character-replacement" (voir le paragraphe 12.1) comme paramètres de conversion pris en charge dans la réponse CONVERSION correspondante.

Les serveurs IMAP qui mettent en œuvre l'extension CONVERT DOIVENT prendre en charge la reconnaissance du paramètre "charset" [RFC2987] pour les types MIME text/plain, text/html, text/css, text/csv, text/enriched, et text/xml. Noter qu'une mise en œuvre de serveur n'est pas obligée de prendre en charge une conversion d'un des sous types MIME text/ spécifiés ci-dessus, sauf pour la conversion de mise en œuvre obligatoire décrite ci-dessus. C'est-à-dire qu'une mise en œuvre de serveur DOIT ne prendre en charge le paramètre "charset" pour text/csv que si elle prend en charge toute conversion à partir de text/csv.

Le serveur DOIT prendre en charge le décodage des en-têtes de la [RFC2047] et leur conversion en UTF-8 pour autant que les mots codés soient dans un des jeux de caractères pris en charge.

Les serveurs DEVRAIENT offrir des conversions de codage de caractères supplémentaires lorsque cela a un sens, car des bibliothèques de conversion de caractères sont généralement disponibles sur de nombreuses plate-formes.

Si le serveur ne peut pas effectuer la conversion de jeu de caractères tout en préservant tous les caractères (c'est-à-dire, un caractère de source ne peut pas être représenté dans le jeu de caractères cible) et le paramètre de conversion "unknown-character-replacement" n'est pas spécifié, alors le serveur DOIT faire échouer la conversion et DOIT retourner la réponse ERROR BADPARAMETERS non étiquetée (voir la Section 9). Si la valeur spécifiée dans la conversion "unknown-character-replacement" elle-même ne peut pas être représentée dans le jeu de caractères cible, alors le serveur DOIT aussi faire échouer la conversion et DOIT retourner la réponse ERROR BADPARAMETERS non étiquetée.

7.2 Caractéristiques supplémentaires pour l'usage mobile

Ce paragraphe est pour information.

Sur la base de l'usage attendu de CONVERT dans les environnements mobiles, les mises en œuvre de serveur devraient envisager la prise en charge des conversions suivantes :

- o Conversion des documents HTML et XHTML en text/plain d'une façon qui préserve au minimum la structure et les tableaux du document.
- o Les conversions d'image parmi les types image/gif, image/jpeg, et image/png pour au moins les paramètres suivants :
 - * limite de taille (c'est-à-dire, réduire la qualité)
 - * largeur (paramètre "pix-x")
 - * hauteur (paramètre "pix-y")
 - * directive redimensionner (grain, étirement, ratio d'aspect)

La prise en charge de "profondeur" (*depth*) peut aussi être intéressante.

La conversion audio est aussi intéressante mais les formats pertinents dépendent significativement du contexte d'utilisation.

8. Éléments de données de demande/réponse aux commandes CONVERT/UID CONVERT

8.1 Réponse CONVERTED non étiquetée

Contenu : convert correlator ; CONVERTED return éléments de données

La réponse CONVERTED peut être envoyée par suite d'une commande CONVERT ou UID CONVERT réussie, partiellement réussie, ou échouée, spécifiée à la Section 6.

La réponse CONVERTED commence par un numéro de message, suivi par l'étiquette "CONVERTED". L'étiquette est suivie par un corrélateur de conversion, qui contient l'étiquette de la commande causant le retour de la réponse. Cela peut être utilisé par un client pour confronter une réponse CONVERTED à une commande CONVERT/UID CONVERT correspondante.

Le corrélateur de conversion est suivi d'une liste d'un ou plusieurs éléments de données de retour CONVERT. Si l'élément de données UID est retourné, il DOIT être retourné comme premier élément de données dans la réponse CONVERTED. Cette exigence est destinée à simplifier les mises en œuvre de client. Voir les détails à la Section 10 et dans la suite de la Section 8.

8.2 Élément de demande et réponse BODYPARTSTRUCTURE CONVERT

BODYPARTSTRUCTURE[section-part]

L'extension CONVERT définit l'élément de données BODYPARTSTRUCTURE CONVERT. Les données contenues dans BODYPARTSTRUCTURE retournent l'élément de données suivant la syntaxe exacte spécifiée dans l'élément de données BODYSTRUCTURE de la [RFC3501], mais contiennent seulement les informations pour la partie convertie. Toutes les informations contenues dans BODYPARTSTRUCTURE relèvent de l'état de la partie après sa conversion, comme le type, sous type, taille, ou jeu de caractères MIME converti. Noter que le client peut s'attendre à ce que le type MIME retourné corresponde à celui qu'il a demandé (car le serveur est obligé de respecter le type MIME demandé) et peut traiter une discordance comme une erreur.

Les données de BODYPARTSTRUCTURE retournées DOIVENT correspondre aux données BINARY retournées pour exactement la même conversion dans la même "session" IMAP. Cette exigence permet à un client de demander les données de BODYPARTSTRUCTURE et BINARY dans des commandes séparées dans la même session IMAP.

Si le client fait la liste des éléments de données BODYPARTSTRUCTURE pour une partie de section avant un élément de données BINARY pour la même partie de section, alors, dans la réponse CONVERTED, le serveur DOIT retourner les données BODYPARTSTRUCTURE avant les données BINARY correspondantes. Aussi, tout élément de données BODYSTRUCTURE DOIT être après l'élément de données UID si l'élément de données UID est présent. Ces deux exigences sont pour simplifier le traitement des données converties chez les clients.

Exemple :

```
C: e002 CONVERT 2 (NIL ("PIX-X" "128" "PIX-Y" "96")) (BINARY[2] BODYPARTSTRUCTURE[2])
S: * 2 CONVERTED (TAG "e002") (BODYPARTSTRUCTURE[2] ("IMAGE"
    "JPEG" () NIL NIL "8bit" 4182 NIL NIL NIL) BINARY[2] ~{4182}
    <cette partie d'un document est une image redimensionnée en format JPEG avec largeur=128, hauteur=96.>
    )
S: e002 OK CONVERT COMPLETED
```

8.3 Élément de demande et réponse BINARY.SIZE CONVERT

BINARY.SIZE[section-part]

Cet élément demande la taille convertie de la section (c'est-à-dire, la taille à attendre dans une réponse à la demande CONVERT BINARY correspondante). La valeur retournée DOIT être exacte et NE DOIT PAS changer durant la "session" IMAP, sauf si le message est omis dans une autre session (voir ci-dessous). Cela permet à un client de télécharger une partie convertie par tronçons (en utilisant "<partial>"). Cette exigence signifie que dans la plupart des cas le serveur a besoin d'effectuer la conversion de la partie de corps demandée avant de retourner sa taille.

Si le message est omis dans une autre session, alors le serveur PEUT retourner la valeur 0 en réponse à l'élément demande BINARY.SIZE plus tard dans la même session.

Afin de permettre la mise à niveau des composants de transcodage du serveur, les clients NE DOIVENT PAS supposer que répéter une conversion de partie de corps particulière dans une autre "session" IMAP donnerait le même résultat qu'une conversion précédente de la même partie de corps -- une caractéristique de la partie de corps convertie pourrait être différente (format, taille, etc.). En particulier, les clients NE DOIVENT PAS mettre en antémémoire les tailles des messages/parties de corps converties après une "session" IMAP, ou utiliser les tailles obtenues dans une connexion dans une autre connexion IMAP au même serveur.

Note historique : l'expérience passée des serveurs IMAP qui retournaient des valeurs estimées RFC822.SIZE montre que cela a causé des problèmes d'interopérabilité. Si le serveur retourne une valeur plus petite que la taille réelle, il va en résulter la troncature des données si un téléchargement <partial> est utilisé. Si le serveur retourne une valeur supérieure à la taille réelle, cela peut conduire un client à croire qu'il n'a pas assez de mémorisation pour télécharger une partie de corps.

Note pour les mises en œuvre de client : les auteurs de client sont avertis que ceci peut être une opération coûteuse pour certaines mises en œuvre de serveur. Demander BINARY.SIZE pour un grand nombre de parties de corps converties ou pour plusieurs conversions de la même partie de corps peut résulter en un ralentissement des performances et/ou une charge excessive du serveur et est déconseillé. Les mises en œuvre de client devraient envisager des approches de mise en œuvre qui limitent cette demande aux seuls cas les plus nécessaires et sont encouragées à vérifier l'impact sur les performances de BINARY.SIZE avec plusieurs mises en œuvre de serveur.

8.4 Élément de demande et réponse AVAILABLECONVERSIONS CONVERT

AVAILABLECONVERSIONS[section-part] permet au client de demander la liste des types MIME cibles dans lesquels la partie de corps spécifiée d'un message ou le message entier peut être converti. Cet élément de données n'est utile que quand la conversion par défaut (voir la Section 6) est demandée.

Cet élément de données DOIT retourner une liste des types MIME cibles qui est un sous ensemble de la liste retournée par la commande CONVERSIONS pour la même paire de types MIME de source et cible. Si une conversion spécifique est demandée, elle DOIT retourner le type MIME cible comme demandé dans la commande CONVERT, ou ERROR.

Pour les deux demandes de conversion spécifique ou par défaut, si les paramètres de conversion sont spécifiés, alors le serveur doit les prendre en compte quand il génère la liste des types MIME cibles. Par exemple, si un ou plusieurs des paramètres de conversion ne s'appliquent pas à un type MIME cible potentiel, alors ce type MIME DOIT être omis de la liste résultante. Si le serveur a seulement un type MIME cible candidat et qu'il a été éliminé à cause de la liste des paramètres de conversion, alors le serveur DEVRAIT retourner une ERROR plutôt que la liste vide des types MIME cibles.

La demande AVAILABLECONVERSIONS DEVRAIT être traitée rapidement si elle est spécifiée par elle-même. Noter que si un type MIME est retourné en réponse à AVAILABLECONVERSIONS, il n'est pas garanti que la demande BINARY/BINARY.SIZE/BODYPARTSTRUCTURE CONVERT correspondante ne va pas échouer.

Exemple :

```
C: f001 CONVERT 2 (NIL) (AVAILABLECONVERSIONS[2])
S: * 2 CONVERTED (TAG "f001") (AVAILABLECONVERSIONS[2] ("IMAGE/JPEG" "application/PostScript"))
S: f001 OK CONVERT COMPLETED
```

8.5 Considérations de mise en œuvre

Noter que ce paragraphe est normatif.

Les serveurs PEUVENT refuser d'exécuter des demandes de conversion qui convertissent plusieurs messages et/ou parties de corps en une seule fois, par exemple, une demande de conversion qui spécifie plusieurs numéros de message/UID. Si le serveur refuse une conversion parce que la demande comporte trop de messages, le serveur DOIT retourner le code de réponse MAXCONVERTMESSAGES (voir la Section 9). Par exemple :

```
C: g001 CONVERT 1:* ("text/plain" ("charset" "us-ascii")) BINARY[3]
S: g001 NO [MAXCONVERTMESSAGES 1]
```

Si le serveur refuse une conversion parce que la demande comporte trop de parties de corps, le serveur DOIT retourner le code de réponse MAXCONVERTPARTS (voir la Section 9). Par exemple :

```
C: h001 CONVERT 1 ("text/plain" ("charset" "us-ascii")) (BINARY[1] BINARY[2])
S: g001 NO [MAXCONVERTPARTS 1] Vous ne pouvez demander qu'une partie de corps à la fois.
```

Note pour les mises en œuvre de serveur : afin d'améliorer les performances, les mises en œuvre DEVRAIENT mettre en antémémoire les parties de corps converties. Par exemple, le serveur peut effectuer une conversion de partie de corps quand il reçoit la première demande BINARY.SIZE[...], BODYPARTSTRUCTURE[...], ou BINARY[...] et la mettre en antémémoire jusqu'à ce que le client demande une conversion/téléchargement d'une autre partie de corps, une conversion différente de la même partie de corps, ou jusqu'à ce que la boîte aux lettres soit close. Afin d'atténuer les attaques de déni de service de la part de clients au mauvais comportement ou mal conçus, un serveur DEVRAIT limiter le nombre de parties de corps converties qu'il peut mettre en antémémoire. Les serveurs DEVRAIENT être capables de mettre en antémémoire au moins deux conversions à tout moment.

9. Extensions de réponses d'état et de code de réponse

Un type de support MIME syntaxiquement invalide DEVRAIT générer une réponse BAD étiquetée de la part du serveur. Un type de support MIME non reconnu génère une réponse NO étiquetée.

Certains transcodages peuvent exiger des paramètres. Si une demande de transcodage sans paramètre est envoyée pour un format qui exige des paramètres, le serveur va retourner une phrase ERROR MISSINGPARAMETERS à la place des

données associées aux éléments de données demandés. Ceci est analogue à la réponse NIL dans FETCH, mais avec des données structurées associées à la défaillance.

Si le serveur est incapable d'effectuer la conversion demandée parce que une ressource est temporairement indisponible (par exemple, manque d'espace de disque, d'une erreur temporaire interne, de la défaillance du service de transcodage) le serveur DOIT alors retourner une réponse NO étiquetée qui DEVRAIT contenir le code de réponse TEMPFAIL (voir ci-dessous) ou une phrase ERROR TEMPFAIL.

Si la conversion demandée ne peut pas être effectuée à cause d'une erreur permanente, par exemple, si un format de document propriétaire n'a pas de mise en œuvre de transcodage existante, le serveur DOIT retourner une réponse CONVERTED contenant une phrase ERROR BADPARAMETERS ou ERROR MISSINGPARAMETERS.

Le serveur PEUT choisir de retourner une ERROR pour une seule conversion si plusieurs éléments de données en rapport sont demandés. Par exemple :

```
C: b002 CONVERT 2 ("text/plain" ("charset" "us-ascii")) (BINARY[3] BODYPARTSTRUCTURE[3])
S: * 2 CONVERTED (tag "b002") (BODYPARTSTRUCTURE[3]
  (ERROR "Le texte source a des caractères non us-ascii" BADPARAMETERS
    "text/html" "text/plain" ("charset" "us-ascii")))
S: b002 NO Toutes les conversions ont échoué
```

Si au moins one conversion réussit, le serveur DOIT retourner une réponse OK. Si toutes les conversions échouent, le serveur PEUT retourner OK ou NO. Par exemple :

```
C: b002 CONVERT 2 ("text/plain" ("charset" "us-ascii"))
  (BINARY[3] BODYPARTSTRUCTURE[3] BINARY[4] BODYPARTSTRUCTURE[4])
S: * 2 CONVERTED (tag "b002") (BODYPARTSTRUCTURE[3]
  (ERROR "Le texte source est non us-ascii" BADPARAMETERS
    "text/html" "text/plain" ("charset" "us-ascii"))
  BODYSTRUCTURE[4] ("TEXT" "PLAIN" (CHARSET US-ASCII)
    NIL NIL "8bit" 4182 NIL NIL NIL) BINARY[4] {4182}
  <corps en text plain>
  )
S: b002 OK Certaines conversions ont échoué
```

En général, le client peut dire à partir de la réponse BODYPARTSTRUCTURE si sa demande a ou non été honorée exactement, mais ne peut pas en savoir la raison.

Le présent document définit les codes de réponse suivants qui peuvent être retournés dans le code de réponse NO étiqueté.

TEMPFAIL : la demande de transcodage a échoué temporairement. Elle pourrait réussir plus tard, donc le client PEUT réessayer.

MAXCONVERTMESSAGES <number> : le serveur est incapable ou ne veut pas convertir plus de <number> messages dans une demande CONVERT/UID CONVERT.

MAXCONVERTPARTS <number> : le serveur est incapable ou ne veut pas convertir plus de <number> parties de corps dans un message à la fois dans une demande CONVERT/UID CONVERT.

Le mot ERROR est toujours suivi par un texte descriptif informel lisible par l'homme, qui est suivi par le code d'erreur de conversion. Le code d'erreur de conversion DOIT être un des suivants :

TEMPFAIL mm : la demande de transcodage a échoué temporairement. Elle pourrait réussir plus tard, donc le client PEUT réessayer. Le client DEVRAIT attendre au moins mm minutes avant de réessayer.

BADPARAMETERS from-concrete-mime-type to-mime-type "(" transcoding-params ")" : la liste des paramètres n'a pas été comprise, n'est pas valide pour la paire de types MIME source/destination, a des valeurs invalides ou pourrait n'être pas honorée pour une autre raison notée dans le texte lisible par l'homme qui a été spécifié après l'étiquette ERROR. Le transcoding-params peut être omis, et dans ce cas, cela signifie que la conversion de from-concrete-mime-type en to-mime-type n'est pas possible. Si le from-concrete-mime-type est NIL, cela signifie que la partie de corps spécifiée

n'existe pas. Tous les paramètres non reconnus ou non pertinents DOIVENT être mentionnés dans le transcoding-params. Ce n'est pas un comportement légal d'ignorer les paramètres non pertinents.

Noter que si le client a demandé la "conversion par défaut" (voir la Section 6) le to-mime-type contient le type de destination MIME choisi par le serveur.

MISSINGPARAMETERS from-concrete-mime-type to-mime-type "(" transcoding-params ")": la liste des paramètres est exigée pour la conversion du type MIME de source spécifié en type MIME de destination, mais ne figurait pas dans la demande. Noter que si le client a demandé la "conversion par défaut" (voir la Section 6) le to-mime-type contient le type MIME de destination choisi par le serveur.

Exemples :

C: b002 CONVERT 2 ("APPLICATION/PDF") BINARY[3]
S: b002 NO [TEMPFAIL] Toutes les conversions ont échoué

C: b003 CONVERT 2 ("TEXT/PLAIN") BINARY[3]
S: * 2 CONVERTED (tag "b003") (BINARY[3]
(ERROR "CHARSET doit être spécifié pour les conversions de texte"
MISSINGPARAMETERS (CHARSET)))
S: b003 NO Toutes les conversions ont échoué

C: b005 CONVERT 2 ("TEXT/PLAIN" (CHARSET "US-ASCII"
UNKNOWN-CHARACTER-REPLACEMENT "<badchar>")) BINARY[3]
S: * 2 CONVERTED (tag "b005") (BINARY[3]
(ERROR "UNKNOWN-CHARACTER-REPLACEMENT limité à 4 octets"
BADPARAMETERS (UNKNOWN-CHARACTER-REPLACEMENT "<badchar>")))
S: b005 NO Toutes les conversions ont échoué

10 Syntaxe formelle

La spécification de syntaxe suivante utilise la notation en format Backus-Naur augmenté (ABNF) telle qu'utilisée dans la [RFC5234], et incorpore par référence les règles définies dans le présent document.

Cette syntaxe augmente la grammaire spécifiée dans les [RFC3501] et [RFC3516]. Les non terminaux non définis dans le présent document se trouvent dans les [RFC3501], [RFC3516], [RFC4466], [RFC4288], et [RFC2506].

command-select =/ convert

uid =/ "UID" SP convert ; identifiants univoques utilisés à la place des numéros de séquence de message

convert = "CONVERT" SP sequence-set SP convert-params SP (convert-att / "(" convert-att *(SP convert-att) ")")

convert-att = "UID" / "BODYPARTSTRUCTURE" section-convert /
"BINARY" section-convert [partial] / "BINARY.SIZE" section-convert /
"BODY[HEADER]" / "BODY[" section-part ".HEADER]" /
"BODY[" section-part ".MIME]" / "AVAILABLECONVERSIONS" section-convert
; <partial> est défini dans la [RFC3516]. <section-part> est défini dans la [RFC3501].

convert-params = "(" (quoted-to-mime-type / default-conversion) [SP "(" transcoding-params ")"] ")"

quoted-to-mime-type = DQUOTE to-mime-type DQUOTE

transcoding-params = transcoding-param *(SP transcoding-param)

transcoding-param-names = transcoding-param-name *(SP transcoding-param-name)

transcoding-param = transcoding-param-name SP transcoding-param-value

transcoding-param-name = astring
; <transcod-param-name-nq> représenté comme quoted, literal ou atom. Noter que <transcod-param-name-nq> permet un "%", qui n'est pas permis dans atom. Ces valeurs doivent être représentées comme quoted ou literal. ;

transcod-param-name-nq = Feature-tag ; <Feature-tag> est défini dans la [RFC2506].

transcoding-param-value = astring

default-conversion = "NIL"

message-data = / nz-number SP "CONVERTED" SP convert-correlator SP convert-msg-attrs

convert-correlator = "(" "TAG" SP tag-string ")"

tag-string = string ; étiquette de la commande qui cause la réponse CONVERTED, envoyée comme chaîne.

convert-msg-attrs = "(" convert-msg-att *(SP convert-msg-att) ")"
; "UID" DOIT être le premier élément de données, si il est présent.

convert-msg-att = msg-att-semistat / msg-att-conv-static

msg-att-conv-static = "UID" SP uniqueid ; NE DOIT PAS changer pour un message

msg-att-semistat = ("BINARY" section-convert ["<" number ">"] SP (nstring / literal8 / converterror-phrase)) /
("BINARY.SIZE" section-convert SP (number / converterror-phrase)) /
("BODYPARTSTRUCTURE" section-convert SP (body / converterror-phrase)) /
("AVAILABLECONVERSIONS" section-convert SP (mimetype-list / converterror-phrase))
; NE DOIT PAS changer durant une "session" IMAP, mais pas nécessairement statique à long terme. ;

section-convert = section-binary
; <section-binary> est défini dans la [RFC3516]. Noter qu'à la différence de la [RFC3516], la conversion d'un multipart/* de niveau supérieur est permise. ;

resp-text-code = / "TEMPFAIL" / "MAXCONVERTMESSAGES" SP nz-number /
"MAXCONVERTPARTS" SP nz-number
; <resp-text-code> est défini dans la [RFC3501]. ;

mimetype-and-params = quoted-to-mime-type [SP "(" transcoding-params ")"]
; inclut toujours un type MIME spécifique ;

mimetype-list = "(" "(" [quoted-to-mime-type *(SP quoted-to-mime-type)] ")" ")"
; liste non ordonnée de types MIME. Elle peut être vide. Deux niveaux de parenthèses sont nécessaires pour distinguer ces données de <converterror-phrase>. ;

converterror-phrase = "(" "ERROR" SP convert-err-descript SP convert-error-code ")"

convert-error-code = "TEMPFAIL" [SP nz-number] / bad-params / missing-params

convert-err-descript = string
; texte lisible par l'homme expliquant l'erreur de conversion. Le jeu de caractères par défaut est US-ASCII, sauf si la commande LANGUAGE [RFC5255] est invoquée, quand le jeu de caractères change pour l'UTF-8. ;

quoted-from-mime-type = DQUOTE from-concrete-mime-type DQUOTE

bad-params = "BADPARAMETERS" 1*(SP (quoted-from-mime-type / nil) SP mimetype-et-params)
; nil n'est retourné que quand la partie de corps n'existe pas. ;

missing-params = "MISSINGPARAMETERS" 1*(SP quoted-from-mime-type SP mimetype-and-missing-params)

mimetype-and-missing-params = quoted-to-mime-type "(" transcoding-param-names ")"

; inclut toujours un type MIME spécifique ;

concrete-mime-type = type-name "/" subtype-name

; c'est-à-dire, "type/subtype". type-name et subtype-name sont définis dans la [RFC4288]. ;

from-concrete-mime-type = concrete-mime-type

to-mime-type = concrete-mime-type

command-auth =/ conversions-cmd

conversions-cmd = "CONVERSIONS" SP from-mime-type-req SP to-mime-type-req

from-mime-type-req = astring

; "mime-type-req" représenté comme <atom>, <quoted> ou <literal> IMAP ;

to-mime-type-req = astring

; <mime-type-req> représenté comme <atom>, <quoted> ou <literal> IMAP. Noter que <mime-type-req> permet un "*", qui n'est pas permis dans <atom>. De telles valeurs doivent être représentées comme <quoted> ou <literal>. ;

any-mime-type = "*"

mime-type-req = any-mime-type / (type-name "/" any-mime-type) / concrete-mime-type

; '*', 'type/*' ou 'type/subtype'. type-name sont définis dans la [RFC4288]. ;

response-payload =/ conversion-data

conversion-data = "CONVERSION" SP quoted-from-mime-type SP quoted-to-mime-type [SP "(" transcoding-param-name *(SP transcoding-param-name) ")"]

11. Considérations sur la gestion

La surveillance de l'opération CONVERT est similaire à celle de l'opération IMAP FETCH.

Au moment de la rédaction du présent document, aucune norme de MIB IMAP n'est définie. De même, une norme de MIB pour la surveillance des opérations CONVERT et leurs défaillances n'existe pas. Cependant, les auteurs pensent qu'en l'absence d'une telle MIB, les mises en œuvre de serveur DEVRAIENT fournir aux opérateurs des outils pour rapporter les informations suivantes :

- o quelles conversions (types MIME source et cible et éventuellement paramètres de conversion utilisés) sont impliquées le plus fréquemment et combien de temps elles durent,
- o des informations sur les erreurs de conversion et quelle condition d'erreur les a causées (voir la Section 9),
- o des informations sur les utilisateurs qui invoquent des opérations de conversion.

Ces informations peuvent aider les opérateurs à détecter les abus de cette extension par les clients et les problèmes d'adaptabilité qui pourraient résulter de son utilisation.

La normalisation de ces outils peut être l'objet d'un travail futur.

12. Considérations relatives à l'IANA

Les capacités IMAP4 sont enregistrées par la publication d'une RFC sur la voie de la normalisation ou d'une RFC expérimentale approuvée par l'IESG. Le présent document définit la capacité IMAP CONVERT. L'IANA a ajouté cette extension au registre Capacités IMAP.

L'IANA a effectué les enregistrements comme défini dans le paragraphe qui suit.

12.1 Enregistrement de paramètre de type de prise en charge unknown-character-replacement

L'IANA a ajouté l'enregistrement suivant au registre établi par la RFC 2506.

Pour : "Liste de diffusion Étiquettes de caractéristique de supports" <media-feature-tags@apps.ietf.org>

Sujet : Enregistrement de l'étiquette de caractéristique de support unknown-character-replacement

Nom de l'étiquette de caractéristique de support : unknown-character-replacement

Identifiant ASN.1 associé à l'étiquette de caractéristique : 1.3.6.1.8.1.33

Résumé de la caractéristique de support indiquée par cette étiquette de caractéristique : permet aux serveurs qui peuvent effectuer la conversion de jeu de caractère pour les types MIME text/plain text/html, text/css, text/csv, text/enriched, et text/xml de remplacer les caractères non pris en charge par le jeu de caractères cible par une chaîne fixe, comme "?". Cette étiquette de caractéristique est aussi applicable aux autres conversions en texte, par exemple, la conversion d'images usant la reconnaissance de caractère optique (OCR, *optical character recognition*).

Valeurs appropriées à utiliser avec cette étiquette de caractéristique : l'étiquette de caractéristique contient une chaîne UTF-8 utilisée pour remplacer tout caractère du type de support source qui ne peut pas être représenté dans le type de support cible. L'étiquette de caractéristique est principalement destinée à être utilisée dans les applications, protocoles, services, ou mécanismes de négociation suivants : extension IMAP CONVERT [RFC5259]

Exemples d'utilisation typiques : C: b001 CONVERT 2 BINARY[3 ("text/plain" ("charset" "us-ascii" "unknown-character-replacement" "?"))]

Normes ou documents en rapport : [RFC5259] [RFC2987]

Considérations particulières d'utilisation dans des applications, protocoles, services, ou mécanismes de négociation individuels : aucune

Considérations d'interopérabilité : aucune

Considérations de sécurité : aucune

Informations supplémentaires : cette caractéristique de support n'a de sens que pour les types MIME qui prennent aussi en charge le paramètre de type de support "charset" [RFC2987].

Nom & adresse de messagerie de la personne à contacter pour plus d'informations : Alexey Melnikov <alexey.melnikov@isode.com>

Usage prévu : COMMUN

Auteur/contrôleur des changements : IETF

13. Considérations pour la sécurité

Il faut noter que certaines conversions peuvent présenter des menaces pour la sécurité (par exemple, convertir un document en un exécutable dommageable, exploitant un débordement de mémoire tampon dans un codec/analyseur de supports, ou une attaque de déni de service contre un client ou un serveur comme en demandant qu'une image soit redimensionnée dans des dimensions extrêmes). Le serveur DEVRAIT refuser d'exécuter des conversions coûteuses en CPU. Les serveurs devraient éviter les conversions dangereuses, si possible. Chaque fois que possible, les serveurs devraient effectuer la vérification des pièces jointes converties avant de les retourner au client. Les clients devrait être prudents quand ils demandent des conversions ou quand ils traitent des pièces jointes transformées. Les clients DEVRAIENT utiliser l'authentification mutuelle de l'authentification simple et couche de sécurité (SASL, *Simple Authentication and Security Layer*) et la couche d'intégrité SASL/TLS pour s'assurer qu'ils parlent à des serveurs de confiance.

Quand le client demande une conversion côté serveur d'une partie de corps signée (par exemple, une partie à l'intérieur de multipart/signed) le client n'a pas de moyen de vérifier que le contenu converti est authentique. Un client qui ne fait pas confiance au serveur pour effectuer la conversion d'une partie de corps signée peut télécharger l'objet signé, vérifier la signature, et effectuer la conversion lui-même.

Un client peut créer un mauvais message très habilement avec la commande APPEND suivie par la commande CONVERT pour attaquer le serveur. Si la fonction ou bibliothèque de conversion du serveur a un problème de sécurité (comme la vulnérabilité au débordement de mémoire tampon) il pourrait en résulter une escalade de privilège ou un déni de service. Afin d'atténuer de telles attaques, les serveurs DEVRAIENT enregistrer l'identité d'authentification du client sur les opérations APPEND et/ou CONVERT afin de faciliter le traçage des clients abusifs. Les mises en œuvre de serveur DEVRAIENT isoler la fonction ou bibliothèque de conversion de la mémorisation privilégiée de messagerie, peut-être en la faisant fonctionner dans un processus distinct.

Il est recommandé aux déploiements dans lesquels le transcodage réel est fait en dehors du serveur IMAP dans un serveur distinct de garder les serveurs dans le même domaine de confiance (par exemple, sous réseau).

14. Remerciements

Stephane H. Maes et Ray Cromwell de Oracle ont édité plusieurs versions antérieures de ce document. Leur contribution a été hautement appréciée.

Les auteurs tiennent spécifiquement à remercier de leurs excellentes critiques et commentaires Randall Gellens (Qualcomm), Arnt Gulbrandsen (Oryx), Zoltan Ordogh (Nokia), Ben Last (Emccsoft), Dan Karp (Zimbra), Pete Resnick (Qualcomm), Chris Newman (Sun), Ted Hardie (Qualcomm), Larry Masinter (Adobe), Philip Guenther (Sendmail), Greg Vaudreuil (Alcatel-Lucent), David Harrington (Comcast), Dave Cridland (Isode), Pasi Eronen (Nokia), Magnus Westerlund (Ericsson), et Jari Arkko (Ericsson) qui ont amélioré considérablement la qualité de cette spécification.

Les auteurs tiennent aussi à remercier particulièrement Dave Cridland de la proposition de commande MEDIACAPS et Dan Karp pour celle de la commande CONVERSIONS.

Les auteurs remercient aussi tous ceux qui ont contribué par des conseils décisifs, et extensivement relu et discuté les concepts de CONVERT et de son prédécesseur P-IMAP. En particulier, cela inclut les auteurs du document LCONVERT : Rafiul Ahad (Oracle Corporation), Eugene Chiu (Oracle Corporation), Ray Cromwell (Oracle Corporation), Jia-der Day (Oracle Corporation), Vi Ha (Oracle Corporation), Wook-Hyun Jeong (Samsung Electronics Co. LTF), Chang Kuang (Oracle Corporation), Rodrigo Lima (Oracle Corporation), Stephane H. Maes (Oracle Corporation), Gustaf Rosell (Sony Ericsson), Jean Sini (Symbol Technologies), Sung-Mu Son (LG Electronics), Fan Xiaohui (China Mobile Communications Corporation (CMCC)), et Zhao Lijun (China Mobile Communications Corporation (CMCC)).

15. Références

15.1 Références normatives

- [RFC2047] K. Moore, "MIME ([Extensions de messagerie Internet](#) multi-objets) Partie trois : extensions d'en-tête de message pour texte non ASCII", novembre 1996. (MàJ par [RFC2184](#), [RFC2231](#)) (D.S.)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par [RFC8174](#))
- [RFC2231] N. Freed, K. Moore, "Extensions MIME [Valeur de paramètre et Mot codé](#) : jeux de caractères, langages, et continuations", novembre 1997. (P.S.)
- [RFC2506] K. Holtman, A. Mutz, T. Hardie, "Procédure d'[enregistrement d'étiquette de caractéristique](#) de support", mars 1999. ([BCP0031](#))
- [RFC2987] P. Hoffman, "[Enregistrement des étiquettes de caractéristiques](#) de support de jeu de caractères et de langage", novembre 2000. (P.S.)
- [RFC3501] M. Crispin, "Protocole d'[accès au message Internet - version 4rev1](#)", mars 2003. (P.S. ; MàJ par [RFC4466](#), [4469](#), [4551](#), [5032](#), [5182](#), [7817](#), [8314](#), [8437](#), [8474](#) ; remplacée par la RFC9051
- [RFC3516] L. Nerenberg, "[Extension Contenu binaire à IMAP4](#)", avril 2003. (MàJ par [RFC4466](#)) (P.S.)
- [RFC4288] N. Freed et J. Klensin, "Spécifications du [type de support et procédures d'enregistrement](#)", [BCP 13](#), décembre 2005.
- [RFC4466] A. Melnikov, C. Daboo, "[Récapitulation des extensions à l'ABNF](#) pour IMAP4", avril 2006. (P.S.)
- [RFC5234] D. Crocker, P. Overell, "[BNF augmenté pour les spécifications de syntaxe](#) : ABNF", janvier 2008. ([STD0068](#))

15.2 Références pour information

- [OMA-ME-RD] OMA 55.919 3.0.0, "Open Mobile Alliance Mobile Email Requirement Document", décembre 2007.

- [OMA-STI] OMA, "Open Mobile Alliance, Standard Transcoding Interface Specification", OMA OMA-STI-V1_0, décembre 2005.
- [RFC2534] L. Masinter et autres, "[Caractéristiques de support pour l'affichage, l'impression et la télécopie](#)", mars 1999. (P.S.)
- [RFC5255] C. Newman et autres, "[Internationalisation du protocole d'accès](#) au message Internet", juin 2008. (P.S.)
- [RFC5616] N. Cook, "Transformation en flux directs des pièces jointes de la messagerie Internet", août 2009. (Information)

Adresse des auteurs

Peter Coates
Sun Microsystems
185 Falcon Drive
Whitehorse, YT Y1A 6T2
Canada
mél : peter.coates@Sun.COM

Alexey Melnikov
Isode Limited
5 Castle Business Village, 36 Station Road,
Hampton, Middlesex, TW12 2BX,
UK
mél : Alexey.Melnikov@isode.com

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 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'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY, le IETF TRUST et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci-encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

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'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 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.