

Groupe de travail Réseau
Request for Comments : 5255
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

C. Newman, Sun Microsystems
 A. Gulbrandsen, Oryx Mail Systems GmHH
 A. Melnikov, Isode Limited
 juin 2008

Internalisation du 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

Le protocole d'accès au message Internet (IMAP, *Internet Message Access Protocol*) version 4rev1 a la prise en charge de base des caractères non ASCII dans les noms de boîte aux lettres et les sous chaînes de recherches. Il prend aussi en charge les en-têtes de message et le contenu codé non ASCII comme spécifié par les extensions multi objets de messagerie Internet (MIME, *Multipurpose Internet Mail Extension*). La présente spécification définit une collection d'extensions IMAP qui améliorent la prise en charge internationale y compris la négociation du langage pour le texte international d'erreurs, les traductions des préfixes d'espace de noms, et la négociation de comparateur pour les recherches, le tri, et l'exécution de tâches individuelles.

Table des matières

1. Introduction.....	2
2. Conventions utilisées dans ce document.....	2
3. Extension LANGUAGE.....	2
3.1 Exigences de l'extension LANGUAGE.....	2
3.2 Commande LANGUAGE.....	3
3.3 Réponses à LANGUAGE.....	4
3.4 Extension TRANSLATION à la réponse NAMESPACE.....	4
3.5 Syntaxe formelle.....	4
4. Extensions I18NLEVEL=1 et I18NLEVEL=2.....	5
4.1 Introduction et généralités.....	5
4.2 Exigences communes à I18NLEVEL=1 et I18NLEVEL=2.....	5
4.3 Exigences de l'extension I18NLEVEL=1.....	6
4.4 Exigences de l'extension I18NLEVEL=2.....	6
4.5 Notes sur la compatibilité.....	6
4.6 Comparateurs et codages de caractères.....	6
4.7 Commande COMPARATOR.....	7
4.8 Réponse à COMPARATOR.....	8
4.9 Code de réponse BADCOMPARATOR.....	8
4.10 Syntaxe formelle.....	8
5. Autres problèmes d'internationalisation de IMAP.....	9
5.1 Identifiants d'utilisateur et mots de passe Unicode.....	9
5.2 Noms de boîte aux lettres UTF-8.....	9
5.3 Domaines, adresses, et en-têtes de messagerie UTF-8.....	9
6. Considérations relatives à l'IANA.....	9
7. Considérations pour la sécurité.....	9
8. Remerciements.....	10
9. Sources pertinentes de documents pour IMAP internationalisé.....	10
10. Références normatives.....	10
11. Références pour information.....	11
Adresse des auteurs.....	11
Déclaration complète de droits de reproduction.....	11

1. Introduction

La présente spécification définit deux extensions à IMAP4rev1 [RFC3501] pour améliorer la prise en charge internationale. Ces extensions peuvent être annoncées et mises en œuvre séparément.

L'extension LANGUAGE permet au client de demander un langage convenable pour les messages d'erreur de protocole et combinée avec l'extension NAMESPACE [RFC2342] elle permet des traductions d'espaces de noms.

L'extension I18NLEVEL=2 permet au client de demander un collationnement convenable qui va modifier le comportement de la commande SEARCH de la spécification de base ainsi que des extensions SORT et THREAD [RFC5256]. Cela renforce le registre de collation de la [RFC4790]. L'extension I18NLEVEL=1 met à jour SEARCH/SORT/THREAD pour utiliser le comparateur ;unicode-casemap, comme défini dans la [RFC5051]. I18NLEVEL=1 est une version plus simple de I18NLEVEL=2 sans capacité de choisir une différente collation.

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].

La syntaxe formelle utilise la forme de notation Backus-Naur augmenté (ABNF) [RFC5234] incluant le cœur des règles définies à l'Appendice A.

Les productions en rapport avec l'UTF-8 sont définies dans la [RFC3629].

Dans les exemples, "C:" et "S:" indiquent les lignes envoyées respectivement 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 la facilité de lecture et ne font pas partie de l'échange de protocole réel.

3. Extension LANGUAGE

IMAP permet que les réponses du serveur incluent du texte lisible par l'homme qui dans de nombreux cas doit être présenté à l'utilisateur. Mais ce texte est limité à l'US-ASCII par la spécification IMAP [RFC3501] afin de préserver la rétro compatibilité avec les mises en œuvre déployées de IMAP. Cette section spécifie un moyen pour qu'un client IMAP négocie quel langage le serveur devrait utiliser quand il envoie du texte lisible par l'homme.

L'extension LANGUAGE fournit seulement un mécanisme pour altérer des chaînes de serveur fixées comme du texte de réponse et des noms de dossier NAMESPACE. Allouer des alias de langage localisé à des boîtes aux lettres partagées serait fait avec un mécanisme distinct comme l'extension proposée METADATA (voir la [RFC5464]).

3.1 Exigences de l'extension LANGUAGE

Les serveurs IMAP qui prennent en charge cette extension DOIVENT mentionner le mot clé LANGUAGE dans leur réponse CAPABILITY ainsi que dans les données CAPABILITY d'accueil.

Un serveur qui annonce cette extension DOIT utiliser le langage "i-default" comme décrit dans la [RFC2277] comme son langage par défaut jusqu'à ce qu'un autre langage pris en charge soit négocié par le client. Un serveur DOIT inclure "i-default" comme un de ses langages pris en charge. Les serveurs IMAP NE DEVRAIENT PAS annoncer l'extension LANGUAGE si il découvrent qu'ils prennent seulement en charge "i-default".

Les clients et serveurs qui prennent en charge cette extension DOIVENT aussi prendre en charge l'extension NAMESPACE [RFC2342].

La commande LANGUAGE est valide dans tous les états. Les clients DEVRAIENT produire LANGUAGE avant l'authentification, car certains serveurs envoient des informations d'utilisateur précieuses au titre de l'authentification (par exemple, "mot de passe correct, mais expiré"). Si une couche de sécurité (comme SASL ou TLS) est ensuite négociée par le client, il DOIT produire à nouveau la commande LANGUAGE afin de s'assurer qu'aucune attaque active antérieure (si il

en est) sur la négociation de LANGUAGE n'a d'effet sur les messages d'erreur suivants. (Voir à la Section 7 une explication plus détaillée de cette attaque.)

3.2 Commande LANGUAGE

Arguments : arguments facultatifs de gamme de langages.

Réponse : une réponse LANGUAGE possible (voir le paragraphe 3.3). Une réponse NAMESPACE possible (voir le paragraphe 3.4).

Résultat : OK – Commande exécutée
 NO – la commande n'a pas pu être exécutée
 BAD - Arguments invalides

La commande LANGUAGE demande qu'un texte lisible par l'homme émis par le serveur soit localisé dans un langage correspondant à un des arguments de gamme de langage comme décrit à la Section 2 de la [RFC4647].

Si la commande réussit, le serveur va retourner des réponses lisibles par l'homme dans le premier langage pris en charge spécifié. Ces réponses vont être en UTF-8 [RFC3629]. Le serveur DOIT envoyer une réponse LANGUAGE spécifiant le langage utilisé, et le changement prend effet immédiatement après la réponse LANGUAGE.

Si la commande échoue, le serveur continue de retourner des réponses lisibles par l'homme dans le langage qu'il utilisait précédemment.

L'argument spécial de gamme de langage "défaut" indique la demande d'utiliser un langage désigné comme préféré par l'administrateur du serveur. Le langage préféré PEUT varier selon l'utilisateur actuellement actif.

Si une gamme de langages ne correspond pas exactement à une étiquette de langage connue mais correspond à un langage selon les règles de la [RFC4647], le serveur DOIT envoyer une réponse LANGUAGE non étiquetée indiquant le langage choisi.

Si il n'y a aucun argument, le serveur DEVRAIT envoyer une réponse LANGUAGE non étiquetée faisant la liste des langages qu'il supporte. Si le serveur est incapable de faire la liste des langages qu'il supporte il PEUT retourner une réponse NO étiquetée à la demande d'énumération. Si, après avoir reçu une demande LANGUAGE, le serveur découvre qu'il ne prend en charge aucun langage autre que i-default, il DOIT retourner une réponse NO étiquetée à la demande d'énumération.

<Le serveur utilise par défaut des réponses i-default en anglais jusqu'à ce que l'utilisateur change explicitement le langage.>

C: A001 LOGIN KAREN PASSWORD
 S: A001 OK LOGIN completed

< Le client a demandé le langage MUL, qu'aucun serveur ne supporte. >

C: A002 LANGUAGE MUL
 S: A002 NO Langage MUL non pris en charge

< Une commande LANGUAGE sans argument est une demande de faire la liste des langages que le serveur supporte. >

C: A003 LANGUAGE
 S: * LANGUAGE (EN DE IT i-default)
 S: A003 OK Les langages supportés ont été énumérés

C: B001 LANGUAGE
 S: B001 NO Le serveur est incapable d'énumérer les langages pris en charge.

< Une fois que le client a changé le langage, toutes les réponses vont être dans ce langage à partir de la réponse LANGUAGE. Noter que cela inclut la réponse NAMESPACE. Parce que les RFC sont en US-ASCII, le présent document utilise une transcription de l'ASCII plutôt que du texte UTF-8, par exemple, "ue" dans le mot "ausgefuehrt" >

```
C: C001 LANGUAGE DE
S: * LANGUAGE (DE)
S: * NAMESPACE ((" " "/") (("Other Users/" " " "TRANSLATION" ("Andere Ben&APw-tzer/"))) (("Public Folders/" " "
"TRANSLATION" ("Gemeinsame Postf&AM8-cher/")))
S: C001 OK Sprachwechsel durch LANGUAGE-Befehl ausgefuehrt
```

< Si un serveur ne prend pas en charge le langage principal demandé, les réponses vont continuer d'être retournées dans le langage que le serveur utilise actuellement. >

```
C: D001 LANGUAGE FR
S: D001 NO Diese Sprache ist nicht unterstuetzt
C: D002 LANGUAGE DE-IT
S: * LANGUAGE (DE-IT)
S: * NAMESPACE ((" " "/") (("Other Users/" " " "TRANSLATION" ("Andere Ben&APw-tzer/"))) (("Public Folders/" " "
"TRANSLATION" ("Gemeinsame Postf&AM8-cher/")))
S: D002 OK Sprachwechsel durch LANGUAGE-Befehl ausgefuehrt
C: D003 LANGUAGE "default"
S: * LANGUAGE (DE)
S: D003 OK Sprachwechsel durch LANGUAGE-Befehl ausgefuehrt
```

< Le serveur ne parle pas français, mais parle anglais. L'utilisateur parle canadien français et canadien anglais. >

```
C: E001 LANGUAGE FR-CA EN-CA
S: * LANGUAGE (EN)
S: E001 OK Now speaking English
```

3.3 Réponses à LANGUAGE

Contenu : une liste de une ou plusieurs étiquettes de langage.

La réponse LANGUAGE se produit par suite d'une commande LANGUAGE. Une réponse LANGUAGE avec une liste contenant une seule étiquette de langage indique que le serveur utilise maintenant ce langage. Une réponse LANGUAGE avec une liste contenant plusieurs étiquettes de langage indique que le serveur communique une liste de langages disponibles pour le client, et qu'aucun changement de langage actif n'a été fait.

3.4 Extension TRANSLATION à la réponse NAMESPACE

Si des représentations localisées des préfixes d'espace de noms sont disponibles dans le langage choisi, le serveur DEVRAIT les inclure dans l'extension TRANSLATION à la réponse NAMESPACE.

L'extension TRANSLATION à la réponse NAMESPACE retournera une seule chaîne, contenant la traduction codée en UTF-7 modifié [RFC3501] du préfixe d'espace de noms. Il est de la responsabilité du client de faire la conversion entre le préfixe d'espace de noms et la traduction du préfixe d'espace de noms quand il présente les noms de boîtes aux lettres à l'utilisateur.

Dans cet exemple, un serveur prend en charge la commande IMAP4 NAMESPACE. Il n'utilise pas de préfixe pour l'espace de nom personnel de l'utilisateur, un préfixe de "Other Users" pour son espace de nom d'autres utilisateurs, et un préfixe de "Public Folders" pour son espace de noms seulement partagé. Comme un client va souvent afficher ces préfixes à l'utilisateur, le serveur inclut leur traduction qui peut être présentée à l'utilisateur.

```
C: A001 LANGUAGE DE-IT
S: * NAMESPACE ((" " "/") (("Other Users/" " " "TRANSLATION" ("Andere Ben&APw-tzer/"))) (("Public Folders/" " "
"TRANSLATION" ("Gemeinsame Postf&AM8-cher/")))
S: A001 OK LANGUAGE-Befehl ausgefuehrt
```

3.5 Syntaxe formelle

La spécification de syntaxe suivante hérite des règles ABNF [RFC5234] de IMAP4rev1 [RFC3501], espace de noms IMAP4 [RFC2342], étiquettes pour l'identification des langues [RFC4646], UTF-8 [RFC3629], et récapitulation des extensions à l'ABNF IMAP4 [RFC4466].

command-any =/ language-cmd ; la commande LANGUAGE est valide dans tous les états

language-cmd = "LANGUAGE" *(SP lang-range-quoted)

response-payload =/ language-data

language-data = "LANGUAGE" SP "(" lang-tag-quoted *(SP lang-tag-quoted) ")"

namespace-trans = SP DQUOTE "TRANSLATION" DQUOTE SP "(" string ")"

; la chaîne est codée en UTF-7 modifié. C'est un sous ensemble de la syntaxe permise par la règle Namespace-Response-Extension de la [RFC4466] ;

lang-range-quoted = astring

; une fois supprimés tous les enveloppeurs ou guillemets littéraux, cela suit la règle language-range de la [RFC4647] ;

lang-tag-quoted = astring

; une fois supprimés tous les enveloppeurs ou guillemets littéraux, cela suit la règle Language-Tag de la [RFC4646] ;

resp-text = ["[" resp-text-code "]" SP] UTF8-TEXT-CHAR *(UTF8-TEXT-CHAR / "["

; après que le serveur a changé pour un langage autre que i-default, cette règle resp-text remplace la règle resp-text de la [RFC3501]. ;

UTF8-TEXT-CHAR = %x20-5A / %x5C-7E / UTF8-2 / UTF8-3 / UTF8-4

; UTF-8 excluant les caractères de contrôle de 7 bits et "[" ;

4. Extensions I18NLEVEL=1 et I18NLEVEL=2

4.1 Introduction et généralités

IMAP4rev1 [RFC3501] inclut la commande SEARCH qui peut être utilisée pour localiser les messages qui correspondent à des critères incluant du texte lisible par l'homme. L'extension SORT [RFC5256] à IMAP permet au client de demander au serveur de déterminer l'ordre des messages sur la base de critères incluant du texte lisible par l'homme. Ces mécanismes exigent la capacité de prendre en charge des fonctions de recherche et de tri non en anglais.

La Section 4 définit deux extensions à IMAP pour l'internationalisation de SEARCH, SORT, et THREAD [RFC5256] IMAP en utilisant le comparateur cadre [RFC4790].

L'extension I18NLEVEL=1 met à jour SEARCH/SORT/THREAD pour utiliser le comparateur i;unicode-casemap, comme défini dans la [RFC5051]. Voir les détails aux paragraphes 4.2 et 4.3.

L'extension I18NLEVEL=2 est un sur-ensemble de l'extension I18NLEVEL=1. Elle ajoute à l'extension I18NLEVEL=1 la capacité de déterminer le comparateur actif (voir la définition ci-dessous) et de négocier l'utilisation de comparateurs qui se servent de la commande COMPARATOR. Elle ajoute aussi la réponse COMPARATOR qui indique le comparateur actif et les autres comparateurs éventuellement disponibles. Voir les détails aux paragraphes 4.2 et 4.4.

4.2 Exigences communes à I18NLEVEL=1 et I18NLEVEL=2

Le terme de "comparateur par défaut" se réfère au comparateur utilisé par SEARCH et SORT en l'absence de toute négociation utilisant la commande COMPARATOR (voir le paragraphe 4.7). Le terme de "comparateur actif" se réfère au comparateur qui va être utilisé au sein d'une session, par exemple, par SEARCH et SORT. La commande COMPARATOR est utilisée pour changer le comparateur actif.

Le comparateur actif s'applique aux clés SEARCH suivantes : "BCC", "BODY", "CC", "FROM", "SUBJECT", "TEXT", "TO", et "HEADER". Si le serveur annonce aussi l'extension "SORT", alors le comparateur actif s'applique aux clés SORT suivantes : "CC", "FROM", "SUBJECT", et "TO". Si le serveur annonce THREAD=ORDEREDSUBJECT, alors le

comparateur actif s'applique à l'algorithme de tramage ORDEREDSUBJECT. Si le serveur annonce THREAD=REFERENCES, alors le comparateur actif s'applique aux comparaisons de champ "subject" faites par l'algorithme de tramage REFERENCES. De futures extensions pourront choisir d'appliquer le comparateur actif à leurs clés SEARCH.

Pour SORT et THREAD, le pré-traitement nécessaire pour extraire le texte du sujet de base d'un en-tête Subject survient avant l'application d'un comparateur.

Un serveur qui annonce l'extension I18NLEVEL=1 ou I18NLEVEL=2 DOIT mettre en œuvre le comparateur i;unicode-casemap, comme défini dans la [RFC5051].

Un serveur qui annonce l'extension I18NLEVEL=1 ou I18NLEVEL=2 DOIT prendre en charge UTF-8 comme jeu de caractères de SEARCH.

4.3 Exigences de l'extension I18NLEVEL=1

Un serveur IMAP qui satisfait toutes les exigences spécifiées aux paragraphes 4.2 et 4.6 (et qui ne prend pas en charge ni n'annonce d'autre extension I18NLEVEL=<n>, où n > 1) DOIT mentionner le mot-clé I18NLEVEL=1 dans ses données CAPABILITY une fois que IMAP entre dans l'état authentifié, et PEUT mentionner ce mot-clé dans d'autres états.

4.4 Exigences de l'extension I18NLEVEL=2

Un serveur IMAP qui satisfait toutes les exigences spécifiées aux paragraphes 4.2, 4.4, et 4.6 à 4.10 (et qui ne prend pas en charge ni n'annonce d'autre extension I18NLEVEL=<n>, où n > 2) DOIT mentionner le mot-clé I18NLEVEL=2 dans ses données CAPABILITY une fois que IMAP entre dans l'état authentifié, et PEUT mentionner ce mot-clé dans d'autres états.

Un serveur qui annonce cette extension DOIT mettre en œuvre le comparateur i;unicode-casemap, comme défini dans la [RFC5051]. Il PEUT mettre en œuvre d'autres comparateurs provenant du registre IANA établi par la [RFC4790]. Voir aussi le paragraphe 4.5 du présent document.

Un serveur qui annonce cette extension DEVRAIT utiliser i;unicode-casemap comme comparateur par défaut. (Noter que i;unicode-casemap est le comparateur par défaut pour I18NLEVEL=1, mais pas nécessairement pour I18NLEVEL=2.) Le choix du comparateur par défaut PEUT être ajustable par l'administrateur du serveur, et PEUT être sensible à l'utilisateur actuel. Une fois que la connexion IMAP est entrée dans l'état authentifié, le comparateur par défaut DOIT rester statique pour le reste de cette connexion.

Noter que comme SEARCH utilise l'opération de sous chaîne, les serveurs IMAP peuvent seulement mettre en œuvre des collations qui offrent l'opération sous chaîne (voir le paragraphe 4.2.3 de la [RFC4790]). Comme SORT utilise l'opération d'ordre (qui à son tour utilise l'opération égalité) les serveurs IMAP qui annoncent l'extension SORT peuvent seulement mettre en œuvre des collations qui offrent les trois opérations (voir les paragraphes 4.2.2 à 4.2.4 de la [RFC4790]).

Si la collation active ne fournit pas les opérations nécessaires pour une commande IMAP, le serveur DOIT répondre avec un BAD étiqueté.

4.5 Notes sur la compatibilité

Plusieurs mises en œuvre de serveur déployées avant la publication de cette spécification se conforment à I18NLEVEL=1 (voir le paragraphe 4.3) mais ne l'annoncent pas. D'autres serveurs traditionnels utilisent le comparateur i;ascii-casemap (voir la [RFC4790]).

Il n'y a pas de bonne façon pour qu'un client sache quel comparateur utilise un serveur traditionnel. Si le client doit supposer le pire, il peut finir par effectuer des opérations locales coûteuses pour obtenir des comparaisons de i;unicode-casemap même quand le serveur le met en œuvre.

Les mises en œuvre traditionnelles de serveur qui se conforment à I18NLEVEL=1 devraient être mises à jour pour annoncer I18NLEVEL=1. Toutes les mises en œuvre de serveur devraient finalement être mises à jour pour se conformer à l'extension I18NLEVEL=2.

4.6 Comparateurs et codages de caractères

La paragraphe 6.4.4 de la RFC 3501 dit : "Dans toutes les clés de recherche qui utilisent des chaînes, un message correspond à la clé si la chaîne est une sous chaîne du champ. La confrontation est insensible à la casse."

Quand il effectue l'opération SEARCH, le comparateur actif est appliqué à la place de la confrontation insensible à la casse spécifiée ci-dessus.

Un serveur IMAP qui effectue des opérations de collationnement (par exemple, au titre de commandes comme SEARCH, SORT, et THREAD) le fait en respectant la procédure suivante :

- (a) Le codage MIME (par exemple, voir la [RFC2047] pour les en-têtes et la [RFC2045] pour les parties de corps) DOIT être supprimé dans les textes collationnés. Si la suppression du codage MIME échoue pour un message (par exemple, une partie de corps du message a un codage de transfert de contenu non pris en charge, utilise des caractères non permis par le codage de transfert de contenu, etc.) le collationnement de ce message est indéfini par la présente spécification, et est traité d'une manière qui dépend de la mise en œuvre.
- (b) Le texte décodé de (a) DOIT être converti en le jeu de caractères attendu par le comparateur actif.
- (c) Pour l'opération de sous chaîne : si l'étape (b) a échoué (par exemple, le texte est dans un jeu de caractères inconnu, contient une séquence qui n'est pas valide selon ce jeu de caractères, etc.) le texte original décodé de (a) (c'est-à-dire, avant la tentative de conversion du jeu de caractères) est collationné en utilisant le comparateur i;octet (voir la [RFC4790]). Si l'étape (b) a réussi, le texte converti de (b) est collationné en accord avec le comparateur actif.

Pour l'opération d'ordre : toutes les chaînes qui ont été bien converties par l'étape (b) sont séparées de toutes les chaînes qui ont échoué à l'étape b). Les chaînes de chaque groupe sont collationnées indépendamment. Toutes les chaînes bien converties par l'étape (b) sont alors validées par le comparateur actif. Les chaînes qui réussissent la validation sont collationnées en utilisant le comparateur actif. Toutes les chaînes qui ont échoué à l'étape (b) ou échoué à l'opération de validité de collationnement actif sont collationnées (après l'application de l'étape (a)) en utilisant le comparateur i;octet (voir la [RFC4790]). La liste triée résultante est produite en ajoutant toutes les chaînes collationnées "échouées" après toutes les chaînes collationnées en utilisant le comparateur actif.

Exemple : l'exemple suivant montre l'ordre de 4 chaînes différentes en utilisant le comparateur i;unicode-casemap [RFC5051]. Les chaînes sont représentées en notation hexadécimale utilisée par l'ABNF [RFC5234].

- (1) %xD0 %xC0 %xD0 %xBD %xD0 %xB4 %xD1 %x80 %xD0 %xB5 %xD0 %xB9 (marqué avec charset=UTF-8)
- (2) %xD1 %x81 %xD0 %x95 %xD0 %xA0 %xD0 %x93 %xD0 %x95 %xD0 %x99 (marqué avec charset=UTF-8)
- (3) %xD0 %x92 %xD0 %xB0 %xD1 %x81 %xD0 %xB8 %xD0 %xBB %xD0 %xB8 %xFF %xB9 (marqué avec charset=UTF-8)
- (4) %xE1 %xCC %xC5 %xCB %xD3 %xC5 %xCA (marqué avec charset=KOI8-R)

L'étape (b) va convertir la chaîne (4) en la séquence d'octets suivante (en UTF-8) :

%xD0 %x90 %xD0 %xBB %xD0 %xB5 %xD0 %xBA %xD1 %x81 %xD0 %xB5 %xD0 %xB9

et va rejeter les chaînes (1) et (3), car elles contiennent des octets non permis dans charset=UTF-8.

Après cela, en utilisant la collation i;unicode-casemap, la chaîne (4) va venir avant la chaîne (2). En utilisant la collation i;octet sur les chaînes originales, la chaîne (3) va venir avant la chaîne (1). Donc, l'ordre final est (4) (2) (3) (1).

Si l'opération de sous chaîne (par exemple, IMAP SEARCH) du comparateur actif retourne le résultat "indéfini" (voir au paragraphe 4.2.3 de la [RFC4790]) pour le texte spécifié dans la commande SEARCH ou le texte du message, alors l'opération est répétée sur le résultat de l'étape (a) en utilisant le comparateur i;octet.

L'opération d'ordre (par exemple, IMAP SORT et THREAD) DEVRAIT collationner ce qui suit : les chaînes codées en utilisant les codages de caractères inconnus ou invalides, les chaînes dans des jeux de caractères non reconnus, et les entrées invalides (comme défini par la collation active).

4.7 Commande COMPARATOR

Arguments : arguments facultatifs de comparateur d'ordre.

Réponse : une réponse COMPARATOR possible (voir le paragraphe 4.8).

Résultat : OK - Commande achevée

NO – pas trouvé de comparateur correspondant
 BAD - Arguments invalides

La commande COMPARATOR est valide dans les états authentifiés et choisis.

La commande COMPARATOR est utilisée pour déterminer ou changer le comparateur actif. Quand elle est produite sans argument, elle résulte en une réponse COMPARATOR qui indique le comparateur actuellement actif.

Quand elle est produite avec un ou plusieurs arguments "comparator", elle change le comparateur actif comme indiqué. (Si plus d'un comparateur installé est satisfait par un argument, le premier argument gagne.) La réponse COMPARATOR fait la liste de tous les comparateurs qui correspondent si plus d'un satisfait les schémas spécifiés.

L'argument "default" se réfère au comparateur par défaut du serveur. Autrement, chaque argument est une spécification de collation comme défini dans le registre Internet de comparateur de protocole d'application [RFC4790].

< Le client demande d'activer un comparateur tchèque si possible, ou autrement un comparateur international générique qu'il considère convenable pour le tchèque. Le serveur prend le premier comparateur pris en charge. >

```
C: A001 COMPARATOR "cz,*" i;basic
S: * COMPARATOR i;basic
S: A001 OK utilisera i;basic pour la collation
```

4.8 Réponse à COMPARATOR

Contenu : le comparateur actif. Une liste facultative de comparateurs correspondants disponibles.

La réponse COMPARATOR se produit par suite d'une commande COMPARATOR. Le premier argument dans la réponse de comparateur est le nom du comparateur actif. Le second argument est une liste de comparateurs qui correspondaient à un des arguments de la commande COMPARATOR et n'est présent que si plus d'une correspondance est trouvée.

4.9 Code de réponse BADCOMPARATOR

Ce code de réponse DEVRAIT être retourné suite à un échec de la commande IMAP par le serveur (qui retourne NO) quand le serveur sait qu'aucun des comparateurs spécifiés ne correspond aux comparateurs demandés.

4.10 Syntaxe formelle

La spécification de syntaxe suivante hérite des règles d'ABNF [RFC5234] de IMAP4rev1 [RFC3501] et du registre Internet des comparateurs de protocole d'application [RFC4790].

command-auth =/ comparator-cmd

resp-text-code =/ "BADCOMPARATOR"

comparator-cmd = "COMPARATOR" *(SP comp-order-quoted)

response-payload =/ comparator-data

comparator-data = "COMPARATOR" SP comp-sel-quoted [SP "(" comp-id-quoted *(SP comp-id-quoted) ")"]

comp-id-quoted = astring

; une fois que toute enveloppe littérale ou guillemets sont supprimés, cela suit la règle collation-id de la [RFC4790]

comp-order-quoted = astring

; une fois que toute enveloppe littérale ou guillemets sont supprimés, cela suit la règle collation-order de la [RFC4790]

comp-sel-quoted = astring

; une fois que toute enveloppe littérale ou guillemets sont supprimés, cela suit la règle collation-selected de la [RFC4790]

5. Autres problèmes d'internationalisation de IMAP

Les paragraphes qui suivent donnent une vue générale des divers autres problèmes d'internationalisation de IMAP. Ces problèmes ne sont pas résolus par cette spécification, mais pourraient l'être par d'autres travaux de normalisation, comme ceux entrepris par le groupe de travail EAI (voir la [RFC5738]).

5.1 Identifiants d'utilisateur et mots de passe Unicode

Actuellement, IMAP4rev1 restreint les champs `userid` et `mot de passe` de la commande `LOGIN` à l'US-ASCII. Les champs `"userid"` et `"mot de passe"` de la commande `IMAP LOGIN` sont restreints à l'US-ASCII seulement jusqu'à ce qu'une future RFC sur la voie de la normalisation en décide autrement. Les serveurs sont encouragés à valider les deux champs pour s'assurer qu'ils se conforment à la syntaxe formelle de l'UTF-8 et à rejeter la commande `LOGIN` si cette syntaxe est violée. Les serveurs PEUVENT rejeter la commande `LOGIN` si le champ `"userid"` ou `"mot de passe"` contient un octet avec le bit de poids fort établi.

Quand `AUTHENTICATE` est utilisé, certains serveurs peuvent prendre en charge des identifiants d'utilisateur et des mots de passe en Unicode [RFC3490] car SASL (voir la [RFC4422]) le permet. Cependant, de tels `"userid"` ne peuvent pas être utilisés au titre des adresses de messagerie.

5.2 Noms de boîte aux lettres UTF-8

La convention UTF-7 modifié de dénomination des boîtes aux lettres décrite au paragraphe 5.1.3 de la RFC 3501 doit être vue comme une transition du statu quo de 1996 quand le UTF-7 modifié a été spécifié. À cette époque, il y avait une large utilisation non officielle de jeux de caractères locaux comme l'ISO-8859-1 et Shift-JIS pour les noms de boîte au lettre non ASCII, avec la non interopérabilité résultante.

Les exigences du paragraphe 5.1 de la RFC 3501 sont très importantes si on veut être capable de déployer des noms de boîte aux lettres en UTF-8. Les serveurs sont encouragés à les appliquer.

5.3 Domaines, adresses, et en-têtes de messagerie UTF-8

Il y a maintenant une norme de l'IETF pour l'internationalisation des noms de domaines dans les applications (IDNA) [RFC3490]. Bien que les clients IMAP soient libres de prendre en charge cette norme, un argument en sa faveur est qu'il serait utile aux simples clients que le serveur IMAP puisse effectuer cette conversion (le même argument s'appliquerait au codage d'en-tête MIME [RFC2047]). Cependant, il ne serait pas raisonnable de poursuivre sur cette voie tant que le travail en cours n'aura pas achevé de définir le format des adresses de messagerie internationales.

6. Considérations relatives à l'IANA

L'IANA a ajouté `LANGUAGE`, `I18NLEVEL=1`, et `I18NLEVEL=2` au registre des capacités IMAP4.

7. Considérations pour la sécurité

L'extension `LANGUAGE` rend une nouvelle commande disponible dans l'état "Non authentifié" dans IMAP. Certaines mises en œuvre IMAP fonctionnent avec le privilège de racine quand le serveur est dans l'état "Non authentifié" et ne révoquent pas ce privilège tant que l'authentification n'est pas achevée. De telles mises en œuvre sont particulièrement vulnérables aux erreurs de sécurité de débordement de mémoire tampon à ce stade et doivent mettre en œuvre l'analyse de cette commande avec des soins supplémentaires.

Une commande `LANGUAGE` produite avant l'activation d'une couche de sécurité est soumise à une attaque active qui supprime ou modifie la négociation, et donc rend les messages `STARTTLS` ou d'erreur d'authentification plus difficiles à interpréter. Ceci n'est pas une nouvelle attaque car les messages d'erreur eux-mêmes sont l'objet d'attaques actives. Les clients DOIVENT produire à nouveau la commande `LANGUAGE` une fois qu'une couche de sécurité est active, afin d'empêcher cette attaque d'impacter les opérations de protocole suivantes.

Les extensions LANGUAGE, I18NLEVEL=1, et I18NLEVEL=2 utilisent le jeu de caractères UTF-8 ; donc, les considérations de sécurité pour UTF-8 [RFC3629] sont pertinentes. Cependant, aucune d'elles n'utilise UTF-8 pour les identifiants, de sorte que le souci le plus sérieux ne s'applique pas.

8. Remerciements

L'extension LANGUAGE se fonde sur un document précédent de Mike Gahrns, dont une portion substantielle du texte de cette section est empruntée. De nombreuses personnes ont participé aux discussions sur l'extension LANGUAGE à IMAP dans les divers forums de l'IETF et les groupes de travail de l'Internet, de sorte que toute liste de contributeurs serait forcément incomplète. Cependant, les auteurs tiennent à remercier Andrew McCown pour ses travaux précoces sur la proposition originale, John Myers pour ses suggestions concernant le problème de l'espace de noms, ainsi que Jutta Degener, Mark Crispin, Mark Pustilnik, Larry Osterman, Cyrus Daboo, Martin Duerst, Timo Sirainen, Ben Campbell, et Magnus Nystrom pour leurs nombreuses suggestions qui ont été incorporées dans ce document.

La discussion initiale de l'extension I18NLEVEL=2 impliquait des apports de Mark Crispin et d'autres participants au groupe de travail Extensions IMAP.

9. Sources pertinentes de documents pour IMAP internationalisé

Voici une liste non normative de sources à considérer pour la mise en œuvre d'un logiciel IMAP à capacité i18n :

- o Les extensions LANGUAGE et I18NLEVEL=2 à IMAP (la présente spécification).
- o Les règles de 8 bits pour la désignation des boîtes aux lettres au paragraphe 5.1 de la RFC 3501.
- o La convention de désignation internationale de boîte aux lettres au paragraphe 5.1.3 de la RFC 3501.
- o MIME [RFC2045] pour les corps de message.
- o Le codage d'en-tête MIME [RFC2047] pour les en-têtes de message.
- o Le groupe de travail EAI de l'IETF.
- o Les extensions MIME Valeur de paramètre et Mot codé [RFC2231] pour les noms de fichier. Les mises en œuvre de serveur IMAP de qualité vont automatiquement combiner les paramètres de multiparties quand ils génèrent la BODYSTRUCTURE. Il y a aussi certaines utilisations non standard déployées de codage d'en-tête MIME entre des guillemets pour des noms de fichier.
- o IDNA [RFC3490] et punycode [RFC3492] pour les noms de domaines (actuellement seulement pertinent pour les clients IMAP).
- o Le jeu de caractères UTF-8 [RFC3629].
- o La politique de l'IETF sur les jeux de caractères et les langages [RFC2277].

10. Références normatives

- [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](#).)
- [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](#))
- [RFC2277] H. Alvestrand, "Politique de l'IETF en matière de [jeux de caractères et de langages](#)", BCP 18, janvier 1998.
- [RFC2342] M. Gahrns, C. Newman, "Espace de noms IMAP4", mai 1998. (*MàJ par* [RFC4466](#)) (*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](#))
- [RFC3629] F. Yergeau, "[UTF-8, un format de transformation](#) de la norme ISO 10646", STD 63, novembre 2003.
- [RFC4422] A. Melnikov et K. Zeilenga, éd, "[Authentification simple et couche de sécurité](#) (SASL)", juin 2006. (*P.S.*)

- [RFC4466] A. Melnikov, C. Daboo, "[Récapitulation des extensions à l'ABNF](#) pour IMAP4", avril 2006. (P.S.)
- [RFC4646] A. Phillips, M. Davis, "[Étiquettes d'identification des langues](#)", [BCP0047](#) septembre 2006. (Remplacée par [RFC5646](#))
- [RFC4667] W. Luo, "[Extensions de couche 2 de réseau virtuel privé](#) (L2VPN) pour le protocole de tunnelage de couche 2 (L2TP)", septembre 2006. (P.S.)
- [RFC4790] C. Newman et autres, "[Registre de collation des protocoles](#) d'application de l'Internet", mars 2007. (P.S.)
- [RFC5051] M. Crispin, "[i:unicode-casemap](#) : un algorithme simple d'interclassement pour chaînes Unicode", octobre 2007. (P.S.)
- [RFC5234] D. Crocker, P. Overell, "[BNF augmenté pour les spécifications de syntaxe](#) : ABNF", janvier 2008. ([STD0068](#))
- [RFC5256] M. Crispin, K. Murchison, "[Protocole d'accès au message Internet](#) - extensions SORT et THREAD", juin 2008. (MàJ par [RFC5957](#)) (P.S.)

11. Références pour information

- [RFC2231] N. Freed, K. Moore, "Extensions MIME [Valeur de paramètre et Mot codé](#) : jeux de caractères, langages, et continuations", novembre 1997. (P.S.)
- [RFC3490] P. Faltstrom et autres, "Internationalisation des noms de domaine dans les applications (IDNA)", mars 2003. (Remplacée par les RFC [5890](#) et [5891](#), P.S.)
- [RFC3492] A. Costello, "[Punycode : Codage Bootstring d'Unicode](#) pour les noms de domaine internationalisés dans les applications (IDNA)", mars 2003. (P.S.)
- [RFC5464] C. Daboo, "Extension IMAP METADATA", février 2009. (P.S.)
- [RFC5738] P. Resnick, C. Newman, "Prise en charge d'UTF-8 par IMAP", mars 2010. (MàJ [RFC3501](#)). (Expérimentale)

Adresse des auteurs

Chris Newman
Sun Microsystems
3401 Centrelake Dr., Suite 410
Ontario, CA 91761
US
mél : chris.newman@sun.com

Arnt Gulbrandsen
Oryx Mail Systems GmbH
Schweppermannstr. 8
D-81671 Muenchen
Germany
mél : arnt@oryx.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 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.