

Groupe de travail Réseau
Request for Comments : 5267
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

D. Cridland, Isode Limited
 C. King, Isode Limited
 juillet 2008

Contextes pour IMAP4

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 IMAP4rev1 a des facilités de recherche puissantes au titre du cœur de protocole, mais manque de la capacité de créer des résultats vivants et à jour qui peuvent être facilement traités. Le présent mémoire fournit une extension pour cela, et montre comment elle peut être utilisée pour fournir une facilité similaire à celle des boîtes aux lettres virtuelles.

Table des matières

1. Introduction.....	1
2. Conventions utilisées dans ce document.....	2
3. Syntaxe étendue de SORT.....	2
3.1 Extension ESORT.....	2
3.2 Gammes de résultats SORT étendus.....	2
3.3 Exemple de SORT étendu.....	3
4. Contextes.....	3
4.1 Généralités.....	3
4.2 Indication de contexte.....	3
4.3 Notifications de changements.....	3
4.4 Résultats partiels.....	6
4.5 Mise en antémémoire des résultats.....	7
5. Syntaxe formelle.....	7
6. Considérations pour la sécurité.....	8
7. Considérations relatives à l'IANA.....	8
8. Remerciements.....	8
9. Références.....	9
9.1 Références normatives.....	9
9.2 Références pour information.....	9
Appendice A. Livre de recettes.....	9
A.1 Boîtes aux lettres virtuelles.....	9
A.2 Boîtes aux lettres de rebut.....	9
A.3 Notifications EXPUNGE immédiates.....	9
A.4 Comptes de surveillance.....	10
A.5 Contextes de resynchronisation.....	10
Appendice B. Notes de mise en œuvre de serveur.....	10
Adresse des auteurs.....	10
Déclaration complète de droits de reproduction.....	11

1. Introduction

Bien que la commande SEARCH de base définie dans [RFC3501], et telle qu'améliorée par la [RFC4731], soit de représentation relativement compacte, cette réduction n'économise qu'une certaine quantité de données, et d'énormes boîtes aux lettres peuvent dépasser la mémorisation disponible pour les résultats même sur des machines haut de gamme.

La commande SORT définie dans la [RFC5256] fournit des caractéristiques utiles, mais est difficile à utiliser efficacement sur des boîtes aux lettres qui passent sur des connexions à faible bande passante.

Le présent mémoire emprunte des concepts de la [RFC2244], comme de donner une vue par fenêtre sur les résultats de recherche ou de tri, et de faire des mises à jour efficaces en termes de bande passante et nombre d'allers-retours. Cela est fourni par les deux extensions "ESORT" et "CONTEXT".

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 les lignes envoyées respectivement par le client d'agent d'utilisateur de messagerie et le serveur IMAP4rev1 ([RFC3501]). "/" indique des commentaires dans la ligne qui ne font pas partie de l'échange de protocole. Les coupures de ligne sont insérées pour faciliter la lecture. Les exemples sont destinés à être lus dans l'ordre, afin que l'état reste d'un exemple au suivant.

Bien que les exemples montrent un serveur qui prend en charge la [RFC4731], ce n'est pas une exigence stricte de la présente spécification.

Les autres mots en majuscule sont normalement les noms des extensions ou commandes IMAP – ils sont en majuscules pour la clarté de la lecture, et sont insensibles à la casse.

3. Syntaxe étendue de SORT

Les serveurs qui mettent en œuvre l'extension SORT fournissent une suite d'extensions aux commandes SORT et UID SORT définies dans la [RFC5256]. Cela permet des options de retour, comme utilisées avec SEARCH et spécifiées dans la [RFC4466], pour être utilisées de la même manière avec SORT.

Les commandes SORT et UID SORT sont étendues par l'ajout d'une liste facultative d'options de retour qui suivent un RETURN atomique immédiatement après la commande. Si cela manque, le serveur va retourner les résultats comme spécifié dans la [RFC5256].

La commande SORT étendue retourne toujours les résultats dans l'ordre de tri demandé, mais est par ailleurs identique dans son comportement à la commande SEARCH étendue définie dans la [RFC4466], telle qu'étendue par la [RFC4731]. En particulier, la commande SORT étendue retourne les résultats dans une réponse ESEARCH.

3.1 Extension ESORT

Les serveurs qui annoncent la capacité "ESORT" prennent en charge les options de retour spécifiées dans la [RFC4731] dans la commande SORT. Ces options de retour sont adaptées comme suit :

MIN : Retourne le numéro/UID de message du plus bas message trié qui satisfait les critères de recherche.

MAX : Retourne le numéro/UID de message du plus haut message trié qui satisfait les critères de recherche.

ALL : Retourne tous les numéros/UID de message qui correspondent aux critères de recherche, dans l'ordre de tri demandé, en utilisant un sequence-set. Noter l'utilisation des gammes décrites au paragraphe 3.2.

COUNT : comme dans la [RFC4731].

3.2 Gammes de résultats SORT étendus

Toutes les gammes données par le serveur, incluant celles données au titre du sequence-set, dans une réponse ESEARCH résultant d'une commande SORT ou UID SORT étendue, DOIVENT être ordonnées en rang numérique croissant après expansion, comme d'après les règles usuelles de la [RFC3501].

En particulier cela signifie que 10:12 est équivalent à 12:10, et 10,11,12. Pour éviter la confusion, les serveurs DEVRAIENT présenter les gammes seulement quand le premier numéro de séquence est inférieur au second ; c'est-à-dire, la forme 10:12 ou la forme 10,11,12 est acceptable, mais 12:10 DEVRAIT être évité.

3.3 Exemple de SORT étendu

Si la liste des options de retour est présente mais vide, alors le serveur fournit TOUS les éléments de données de retour dans une réponse ESEARCH. Cela est fonctionnellement équivalent à une commande UID SORT non étendue, mais peut utiliser une représentation plus courte :

```
C: E01 UID SORT RETURN () (REVERSE DATE) UTF-8 UNDELETED UNKEYWORD $Junk
S: * ESEARCH (TAG "E01") UID ALL 23765,23764,23763,23761,[...]
S: E01 OK Tri achevé
```

Noter que les trois résultats initiaux ne sont pas représentés dans la gamme 23765:23763 comme exigé au paragraphe 3.2.

4. Contextes

4.1 Généralités

L'extension CONTEXT est présente dans tout serveur IMAP4rev1 qui inclut la chaîne "CONTEXT=SEARCH", et/ou "CONTEXT=SORT", dans l'annonce de ses capacités.

Dans le cas de CONTEXT=SEARCH, le serveur prend en charge la syntaxe de la commande SEARCH étendue décrite dans la [RFC4466], et accepte trois options de retour supplémentaires.

Les serveurs qui annoncent CONTEXT=SORT annoncent aussi la capacité SORT, comme décrit dans la [RFC5256], la prise en charge de la syntaxe de la commande SORT étendue décrite à la Section 3, et acceptent trois options de retour supplémentaires pour ce SORT étendu.

Ces options de retour supplémentaires permettent la notification de changements aux résultats des commandes SEARCH ou SORT, et permettent aussi l'accès à des résultats partiels.

Un serveur qui annonce l'extension CONTEXT=SEARCH va ordonner tous les résultats de SEARCH, qu'ils soient d'une commande UID SEARCH ou SEARCH, dans l'ordre de la boîte aux lettres -- c'est-à-dire, par numéro de message et UID. Donc, les commandes UID SEARCH, SEARCH, UID SORT, ou SORT utilisées -- appelées collectivement une commande de recherche -- va toujours avoir un ordre, l'ordre demandé, qui va être l'ordre de la boîte aux lettres pour les commandes UID SEARCH et SEARCH.

Les spécificateurs de retour n'ont pas d'interaction soit les uns avec les autres, soit avec les spécificateurs de retour définis dans la [RFC4731] ou au paragraphe 3.1 ; cependant, on estime que les mises en œuvre qui prennent en charge CONTEXT vont aussi prendre en charge ESEARCH et ESORT.

4.2 Indication de contexte

L'option de retour CONTEXT DEVRAIT être utilisée par un client pour indiquer que l'utilisation suivante des critères de recherche est probable. Les serveurs PEUVENT ignorer cette option de retour ou l'utiliser comme un conseil de tenir une antémémoire de tous les résultats, ou un index.

Un client pourrait choisir d'obtenir un compte des messages correspondants avant d'obtenir les résultats réels. Donc, le client signale son intention d'aller chercher les résultats eux-mêmes :

```
C: A01 SEARCH RETURN (CONTEXT COUNT) UNDELETED UNKEYWORD $Junk
S: * ESEARCH (TAG "A01") COUNT 23765
S: A01 OK recherche terminée.
```

4.3 Notifications de changements

L'option de retour de recherche UPDATE, si elle est utilisée par un client, cause la production par le serveur de notifications non sollicitées contenant des mises à jour des résultats qui auraient été retournés par une commande de recherche non modifiée. Ces ensembles de mise à jour sont portés dans les éléments de données ADDTO et REMOVEFROM dans les réponses ESEARCH.

Ces réponses ESEARCH portent un corrélateur de recherche de la commande de recherche, donc les clients NE DOIVENT PAS réutiliser les étiquettes, comme déjà spécifié au paragraphe 2.2.1 de la [RFC3501]. Une tentative d'utiliser UPDATE lorsque une étiquette est utilisée avec une précédente commande de recherche qui elle-même utilisait UPDATE DEVRA résulter en le rejet par le serveur de la commande de recherche avec une réponse BAD.

Les éléments de données ADDTO et REMOVEFROM DEVRAIENT être livrés aux clients en temps utile, quand les résultats changent, que ce soit par de nouveaux messages qui arrivent dans la boîte aux lettres, par des métadonnées comme des fanions changés, ou des messages omis.

Normalement, cela va se produire au même moment que les réponses FETCH, EXISTS, ou EXPUNGE portant la source du changement.

Les mises à jour vont cesser quand la boîte aux lettres n'est plus sélectionnée, ou quand la commande CANCELUPDATE, définie au paragraphe 4.3.5, est produite par le client, selon le premier produit.

À la différence de la [RFC2244], il n'est pas exigé qu'un contexte soit créé avec CONTEXT pour utiliser UPDATE, et de plus, l'absence d'un UPDATE avec un CONTEXT n'affecte pas les résultats causés par des commandes de recherche ultérieures – il n'y a pas de facilité de "photographie".

Il n'y a pas d'interaction entre UPDATE et toute autre option de retour ; donc, l'utilisation de RETURN (UPDATE MIN) par exemple, ne notifie pas l'UID ou numéro de séquence minimum, mais notifie plutôt tous les changements à l'ensemble de messages correspondants. En particulier, cela signifie qu'un client qui utilise UPDATE et PARTIAL sur le même programme de recherche pourrait recevoir des notifications sur des messages qui ne l'intéressent pas actuellement.

Finalement, comme spécifié dans les errata à la [RFC3501], tout numéro de séquence de message utilisé dans le programme de recherche est évalué au moment de la réception de la commande ; donc, si les messages auxquels ces numéros de séquence se réfèrent changent, aucune notification ne va être émise.

Cette fois, le client va demander des notifications de mises à jour et choisit d'obtenir un compte :

```
C: B01 UID SEARCH RETURN (UPDATE COUNT) DELETED KEYWORD $Junk
S: * ESEARCH (TAG "B01") COUNT 74
S: B01 OK Recherche terminée, notification à suivre.
// Noter que ce qui suit est rejeté, et n'a pas d'effet :
C: B01 SORT RETURN (UPDATE) FLAGGED
S: B01 BAD Réutilisation d'étiquette
```

4.3.1 Refus de mettre à jour les contextes

Dans certains cas, le serveur PEUT refuser de fournir des mises à jour, comme quand une limite interne au nombre de contextes de mise à jour est atteinte. Dans ce cas, un NO non étiqueté est généré durant le traitement de la commande avec un code de réponse de NOUPDATE. Le code de réponse contient comme argument l'étiquette de la commande de recherche pour laquelle le serveur refuse d'honorer la demande UPDATE.

Les autres options de retour spécifiées DEVRONT quand même être honorées.

Les serveurs DOIVENT fournir au moins un contexte de mise à jour par client, et DEVRAIENT en fournir plus – voir à l'Appendice B les stratégies de réduction de l'impact de contextes de mise à jour supplémentaires. Comme le tri des contextes exige un coût de mise en œuvre supérieur à celui des contextes non triés, refuser de fournir des mises à jour pour une commande SORT n'implique pas que les contextes SEARCH vont aussi être refusés.

Cette fois, le client va demander des notifications de mises à jour, et choisit d'obtenir un compte :

```
C: B02 UID SORT RETURN (UPDATE COUNT) UTF-8 KEYWORD $Junk
```

S: * ESEARCH (TAG "B02") COUNT 74
 S: * NO [NOUPDATE "B02"] Trop de contextes.
 S: B02 OK Recherche terminée, pas de notification.

Le comportement du client pourrait être de réessayer avec une commande UID SEARCH, ou autrement d'annuler un contexte existant ; voir le paragraphe 4.3.5.

4.3.2 Caractéristiques communes de ADDTO et de REMOVEFROM

Le résultat d'ensemble de mise à jour inclus dans l'élément de données de retour est spécifié comme des UID ou des numéros de message, selon la façon dont le UPDATE a été spécifié. Si le UPDATE était présent dans une commande SEARCH ou SORT, le résultat sera des numéros de message ; dans une commande UID SEARCH ou UID SORT, ce sera des UID.

Le client DOIT traiter les éléments de données de retour ADDTO et REMOVEFROM dans l'ordre de leur apparition, les incluant dans une seule réponse ESEARCH. De même, les serveurs DOIVENT générer des réponses ADDTO et REMOVEFROM telles que les résultats soient maintenus dans l'ordre des demandes.

Comme avec toute réponse à part EXPUNGE, les réponses ESEARCH qui portent des éléments de données de retour ADDTO et/ou REMOVEFROM PEUVENT être envoyées à tout moment. En particulier, les serveurs PEUVENT envoyer de telles réponses quand aucune commande n'est en cours, durant le traitement de toute commande, ou quand le client utilise la facilité IDLE décrite dans la [RFC2177]. Il est recommandé de lire dans la [RFC5465] un mécanisme pour que les clients signalent aux serveurs qu'ils veulent traiter les réponses à tout moment, et il est aussi recommandé de lire attentivement le paragraphe 5.3 de la [RFC3501].

Il est prévu que les mises en œuvre normales de serveur vont émettre ADDTO quand elles émettent normalement le FETCH ou EXISTS causal, et de même, émettent REMOVEFROM quand elles émettent normalement le FETCH ou EXPUNGE causal.

4.3.3 Élément de données de retour ADDTO

L'élément de données de retour ADDTO contient, comme charge utile, une liste contenant des paires d'une position de contexte et d'un ensemble de mises à jour de résultats dans l'ordre de la demande pour être insérées dans la position du contexte. Lorsque la commande de recherche est SEARCH ou UID SEARCH, la position du contexte PEUT être zéro. Chaque paire est traitée dans l'ordre de son apparition.

Noter qu'un ADDTO contenant des numéros de séquence de message par suite de la livraison ou de l'ajout de ces messages DOIT être envoyé après la notification EXISTS elle-même, afin que ces numéros de séquence soient valides.

Si la position de contexte n'est pas zéro, la mise à jour résultante est insérée dans la position de contexte donnée, ce qui signifie que le premier résultat dans l'ensemble va occuper la nouvelle position de contexte après l'insertion, et tout résultat existant antérieurement à cette position de contexte va être décalé à une position de contexte postérieure.

Lorsque la position de contexte est zéro, le client PEUT insérer les numéros de message ou UID dans la liste de résultats de telle façon que la liste des résultats soit maintenue dans l'ordre de la boîte aux lettres. Dans ce cas, il est RECOMMANDÉ aux serveurs de ranger la mise à jour de résultats dans l'ordre de la boîte aux lettres pour produire la plus courte représentation dans "set-syntax".

```
[...]
S: * 23762 FETCH (FLAGS (\Deleted \Seen))
S: * 23763 FETCH (FLAGS ($Junk \Seen))
S: * ESEARCH (TAG "B01") UID ADDTO (0 32768:32769)
```

Noter que dans cet exemple, on suppose que les messages 23762 et 23763 avec les UID 32768 et 32769 (respectivement) n'avaient précédemment ni \Deleted ni \$Junk établi. Noter aussi que seul le ADDTO est inclus, et pas le COUNT (maintenant changé).

Si la commande de recherche "C01" avait initialement généré une liste de résultats de 2734:2735, alors les trois réponses suivantes seraient équivalentes, et donneraient une liste de résultats de 2731:2735:

```
[...]
S: * ESEARCH (TAG "C01") UID ADDTO (1 2733 1 2732 1 2731)
S: * ESEARCH (TAG "C01") UID ADDTO (1 2733) ADDTO (1 2731:2732)
S: * ESEARCH (TAG "C01") UID ADDTO (1 2731:2733)
```

La représentation préférée est la dernière.

4.3.4 Élément de données de retour REMOVEFROM

L'élément de données de retour REMOVEFROM contient, comme charge utile, une liste contenant des paires d'une position de contexte et d'un ensemble de mises à jour de résultats dans l'ordre de la demande à supprimer en commençant à la position de contexte. Lorsque la commande de recherche est SEARCH ou UID SEARCH, la position de contexte PEUT être zéro. Chaque paire est traitée dans l'ordre de son apparition.

Si la position de contexte n'est pas zéro, les résultats sont supprimés à la position de contexte donnée, ce qui signifie que le premier résultat dans l'ensemble va occuper la position de contexte donnée avant la suppression, et tout résultat existant antérieurement à cette position de contexte va être décalé à une position de contexte antérieure.

Lorsque la position de contexte est zéro, le client supprime les numéros de message ou UID dans la liste de résultats chaque fois qu'il se produisent, et il est RECOMMANDÉ aux serveur de ranger la liste des résultats dans l'ordre de la boîte aux lettres pour obtenir le meilleur usage de "set-syntax".

Noter qu'un REMOVEFROM contenant des numéros de séquence de message supprimés par suite de l'éviction de ces messages DOIT être envoyé avant la notification de l'éviction elle-même, afin que ces numéros de séquence restent valides.

Ici, un message dans la liste de résultats est évincé. Le REMOVEFROM est montré arrivant sans aucune commande en cours ; voir au paragraphe 4.3.2. Noter que les réponses EXPUNGE n'ont pas cette propriété.

```
[...]
S: * ESEARCH (TAG "B01") UID REMOVEFROM (0 32768)
C: B03 NOOP
S: * 23762 EXPUNGE
S: B03 OK Rien n'est fait.
```

4.3.5 Commande CANCELUPDATE

Quand un client ne souhaite plus recevoir de mises à jour, il peut produire la commande CANCELUPDATE, qui va empêcher toutes les mises à jour dans les contextes désignés dans les arguments d'être transmises par le serveur. La commande prend comme arguments une ou plusieurs étiquettes des commandes utilisées pour demander les mises à jour.

Le serveur PEUT libérer toute ressource associée à un contexte ainsi désactivé -- cependant, le client est libre de produire d'autres commandes de recherche avec les mêmes critères et ordre de demande, incluant des demandes PARTIAL.

```
C: B04 CANCELUPDATE "B01"
S: B04 OK Pas d'autre mise à jour.
```

4.4 Résultats partiels

L'option de retour de recherche PARTIAL cause la fourniture par le serveur d'un sous ensemble des résultats dans une réponse ESEARCH notés par la gamme de numéros de séquence donnée comme argument obligatoire. Le premier résultat est 1 ; donc, les 500 premiers résultats vont être obtenus par une option de retour de "PARTIAL 1:500", et les 500 seconds par "PARTIAL 501:1000". Cela reflète intentionnellement les numéros de séquence de messages.

Une seule commande NE DOIT PAS contenir plus d'une option de retour de recherche PARTIAL ou ALL -- c'est-à-dire, soit un PARTIAL, un ALL, soit ni PARTIAL ni ALL n'est permis.

Pour les résultats de SEARCH, la liste entière de résultats DOIT être ordonnée dans l'ordre de la boîte aux lettres, c'est-à-dire, par UID ou ordre de numéro de séquence de message.

Lorsque une option de retour de recherche PARTIAL fait référence à des résultats qui n'existent pas, en utilisant une gamme qui commence ou se termine plus haut que le nombre actuel de résultats, le serveur retourne alors les résultats qui sont dans l'ensemble. Cela donne un élément de données de retour PARTIAL qui a, comme charge utile, la gamme originale et un ensemble de résultats potentiellement manquants qui peut être plus court que l'étendue de la gamme.

Les clients n'ont pas besoin de demander des résultats PARTIAL dans un ordre particulier. Parce que les boîtes aux lettres peuvent changer, les clients vont souvent souhaiter utiliser PARTIAL combiné avec UPDATE, en particulier si l'intention est de parcourir un grand ensemble de résultats ; cependant, ces options de retour n'interagissent pas -- le UPDATE va fournir des notifications pour tous les résultats qui correspondent.

```
// On rappelle de A01 qu'il y a 23 764 résultats.
C: A02 UID SEARCH RETURN (PARTIAL 23500:24000) UNDELETED UNKEYWORD $Junk
C: A03 UID SEARCH RETURN (PARTIAL 1:500) UNDELETED UNKEYWORD $Junk
C: A04 UID SEARCH RETURN (PARTIAL 24000:24500) UNDELETED UNKEYWORD $Junk
S: * ESEARCH (TAG "A02") UID PARTIAL (23500:24000 ...)
// 264 résultats dans l'ensemble de syntaxe supprimés, cela couvre la fin des résultats.
S: A02 OK Terminé.
S: * ESEARCH (TAG "A03") UID PARTIAL (1:500 ...)
// 500 résultats dans l'ensemble de syntaxe supprimé.
S: A03 OK Terminé.
S: * ESEARCH (TAG "A04") UID PARTIAL (24000:24500 NIL)
// Aucun résultat n'est présent, cela dépasse la fin des résultats.
S: A04 OK Terminé.
```

4.5 Mise en antémémoire des résultats

Les mises en œuvre de serveur PEUVENT mettre en antémémoire les résultats d'une commande SEARCH ou SORT, qu'il y ait ou non un conseil de CONTEXT, afin de rendre les recherches suivantes plus efficaces, peut-être en recommençant une recherche PARTIAL après qu'une précédente recherche a échoué. Cependant, les serveurs DOIVENT avoir un comportement identique qu'il y ait ou non une mise en antémémoire interne ; donc, il est exigé que toute antémémoire de ce genre soit mise à jour lorsque se produisent des changements à la boîte aux lettres. Une autre stratégie serait d'éliminer les résultats quand un changement se produit dans la boîte aux lettres.

5. Syntaxe formelle

Syntaxe formelle collectée. Elle utilise l'ABNF défini dans la [RFC5234]. Elle inclut des définitions provenant des [RFC3501], [RFC4466], et [RFC5256].

capability =/ "CONTEXT=SEARCH" / "CONTEXT=SEARCH" / "ESORT" ;; <capability> de la [RFC3501]

command-select =/ "CANCELUPDATE" 1*(SP quoted) ;; <command-select> de la [RFC3501]

context-position = number
;; la position de contexte peut être 0 pour l'addition de résultats SEARCH. <number> de la [RFC3501]

modifier-context = "CONTEXT"

modifier-partial = "PARTIAL" SP partial-range

partial-range = nz-number ":" nz-number
;; Une gamme de 500:400 est le même que 400:500. C'est similaire à <seq-range> de la [RFC3501], mais ne peut pas contenir "*".

modifier-update = "UPDATE"

search-return-opt =/ modifier-context / modifier-partial / modifier-update
;; Tout conforme à <search-return-opt>, de la [RFC4466]

resp-text-code =/ "NOUPDATE" SP quoted ;; <resp-text-code> de la [RFC3501]

9. Références

9.1 Références normatives

- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par [RFC8174](#))
- [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)
- [RFC4466] A. Melnikov, C. Daboo, "[Récapitulation des extensions à l'ABNF](#) pour IMAP4", avril 2006. (P.S.)
- [RFC4731] A. Melnikov, D. Cridland, "[Extension de commande SEARCH](#) à IMAP4 pour contrôler le type d'information retournée", novembre 2006. (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.)

9.2 Références pour information

- [RFC2177] B. Leiba, "Commande [IDLE dans IMAP4](#)", juin 1997. (P.S.)
- [RFC2244] C. Newman, J. G. Myers, "[ACAP – Protocole d'accès à la configuration d'application](#)", novembre 1997. (P.S.)
- [RFC5465] A. Gulbrandsen, C. King, A. Melnikov, "[Extension IMAP NOTIFY](#)", février 2009. (MàJ [RFC5267](#)) (P.S.)

Appendice A. Livre de recettes

A.1 Boîtes aux lettres virtuelles

Il est possible d'utiliser les facilités décrites dans le présent mémoire pour créer une facilité largement similaire à une boîte aux lettres virtuelle, mais traitée du côté client.

Initialement, le client choisit avec SELECT la boîte aux lettres "de soutien" réelle. Ensuite, il peut passer à une vue filtrée à tout moment en produisant un RETURN (COUNT UPDATE CONTEXT) et en utilisant un RETURN (PARTIAL x:y) lorsque l'utilisateur déroule son menu, alimentant avec les résultats un FETCH comme nécessaire pour remplir les vues sommaires.

Une vue normalement utile est "UID SORT (DATE) RETURN (...) UTF-8 UNSEEN UNDELETED", qui peut être utilisé pour montrer la boîte aux lettres triées dans l'ordre INTERNALDATE, filtrée pour montrer seulement les messages qui e sont pas lus et pas encore supprimés.

A.2 Boîtes aux lettres de rebut

Certains contextes sont particulièrement utiles pour les développeurs de client qui souhaitent présenter quelque chose de similaire à la métaphore courante de boîtes aux lettres de rebut dans des environnements à bande passante limitée. Le simple critère de UNDELETED correspond seulement aux messages non supprimés, et la clé de recherche DELETED correspondante peut être utilisée pour afficher une boîte aux lettres virtuelle de rebut par boîte aux lettres.

A.3 Notifications EXPUNGE immédiates

La commande "SEARCH RETURN (UPDATE) ALL" peut être utilisée pour créer un contexte qui notifie immédiatement les messages éliminés, qui n'affectent pas les numéros de séquence de message jusqu'à ce que le message EXPUNGE normal puisse être envoyé. Cela peut être utile pour les clients qui souhaitent afficher ce comportement sans perdre le bénéfice du numéro de séquence.

A.4 Comptes de surveillance

Un client n'a pas besoin de tenir du tout d'antémémoire de résultats, mais il peut plutôt tenir un simple compte des messages correspondants aux critères de recherche. Normalement, il utiliserait la commande SEARCH, plutôt que UID SEARCH, à cause de sa plus petite représentation. Cet usage pourrait se révéler utile pour la surveillance du nombre de messages avec fanion, mais sans réponse, par exemple, avec "SEARCH RETURN (UPDATE COUNT) FLAGGED UNANSWERED".

A.5 Contextes de resynchronisation

La création d'un contexte, et son accès immédiat, peut être réalisée en un seul aller-retour. Donc, bien qu'il soit possible de sauter la resynchronisation si aucun changement ne s'est produit, il est plus simple dans la plupart des cas de resynchroniser en simplement créant à nouveau le contexte.

Appendice B. Notes de mise en œuvre de serveur

Bien qu'un serveur puisse mettre les résultats en antémémoire, cela n'est ni exigé ni obligatoire, en particulier quand le client utilise les commandes SEARCH ou UID SEARCH. Le traitement de UPDATE, par exemple, peut être réalisé efficacement en comparant l'ancien état de fanions (si il en est) et le nouveau, et PARTIAL peut être réalisé en faisant à nouveau la recherche jusqu'à ce que la fenêtre convenable soit exigée. Cela résulte de l'absence d'une facilité de photographie.

Par exemple, sur un nouveau message, le serveur peut simplement vérifier les correspondances de tous les programmes de recherche de contexte UPDATE actuels, et pour chacun qui correspond, envoyer les données de retour ADDTO.

De même, pour un changement de fanion sur un message existant, le serveur peut vérifier si le message correspondait à ses anciens fanions, si il correspond à ses nouveaux fanions, et fournir des données de retour ADDTO ou REMOVEFROM en conséquence si ces résultats diffèrent.

Pour les demandes PARTIAL, le serveur peut effectuer une recherche complète, éliminant les résultats jusqu'à ce que la limite inférieure soit atteinte, et arrêtant la recherche quand des résultats suffisants ont été obtenus. Avec plus d'état, il est possible de redémarrer les recherches PARTIAL, évitant donc d'effectuer la phase initiale d'élimination.

Pour les meilleures performances, cependant, la mise en antémémoire des résultats de recherche complets est nécessaire, ce qui peut permettre des réponses plus rapides au prix de plus de mémoire. Une tactique raisonnable serait d'effectuer ce compromis au démarrage, éliminant les résultats de recherche après un délai convenable, et en les régénérant comme nécessaire.

Cela donne des exigences d'état de mémorisation du programme de recherche pour tous les contextes UPDATE, et facultativement de mémoriser le programme de recherche et les résultats (mis à jour) pour plus de contextes comme nécessaire.

Noter qu'en l'absence d'une antémémoire de résultats côté serveur, il peut être impossible de savoir si un message effacé précédemment correspondait sauf si le message original est encore disponible. Donc, certaines mises en œuvre peuvent être forcées d'utiliser une antémémoire de résultats dans de nombreuses circonstances.

Les contextes UPDATE créés avec SORT ou UID SORT vont très certainement exiger cependant une certaine forme de mise en antémémoire de résultats.

Adresse des auteurs

Dave Cridland
Isode Limited
5 Castle Business Village
36, Station Road
Hampton, Middlesex TW12 2BX
mél : dave.cridland@isode.com

Curtis King
Isode Limited
5 Castle Business Village
36, Station Road
Hampton, Middlesex TW12 2BX
mél : cking@mumbo.ca

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.