

Groupe de travail Réseau  
**Request for Comments : 5182**  
RFC mise à jour : 3501  
Catégorie : En cours de normalisation

A. Melnikov, Isode Ltd.  
mars 2008  
Traduction Claude Brière de L'Isle

## Extension IMAP pour référencer le dernier résultat SEARCH

### Statut du présent mémoire

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

### Résumé

De nombreux clients IMAP utilisent le résultat d'une commande SEARCH comme entrée pour effectuer un autre opération, par exemple, aller chercher les messages trouvés, les supprimer, ou les copier dans d'autres boîtes aux lettres.

On peut y arriver en utilisant les opérations IMAP standard décrites dans la RFC 3501 ; cependant, cela ne serait pas optimal. Le serveur va envoyer au client la liste des messages trouvés, après quoi, le client devra analyser la liste, la reformater, et la renvoyer au serveur. Le client ne peut faire un traitement en parallèle de la commande SEARCH avec la commande suivante, et par suite, le serveur peut n'être pas capable d'effectuer certaines optimisations.

Le présent document propose une extension IMAP qui permet au client de dire au serveur d'utiliser le résultat d'une commande SEARCH (ou UID SEARCH, *recherche d'identifiant unique*) comme entrée de toute commande ultérieure.

## 1. Introduction

De nombreux clients IMAP utilisent le résultat d'une commande SEARCH comme entrée pour effectuer un autre opération, par exemple, aller chercher les messages trouvés, les supprimer, ou les copier dans d'autres boîtes aux lettres.

Le présent document propose une extension IMAP qui permet à un client de dire à un serveur d'utiliser le résultat d'une commande SEARCH (ou UID SEARCH) comme entrée à toute commande suivante.

L'extension de référence du résultat de SEARCH définit une nouvelle option de résultat SEARCH [IMAPABNF] "SAVE" qui dit au serveur de se rappeler du résultat de la commande SEARCH ou UID SEARCH (ainsi que de toute commande fondée sur SEARCH, par exemple, SORT et THREAD [SORT]) et de le mémoriser dans une variable interne que nous allons appeler "variable de résultat de recherche". Le client peut utiliser le marqueur "\$" pour référencer le contenu de cette variable interne. Le marqueur "\$" peut être utilisé à la place de la séquence de message ou de la séquence d'UID afin d'indiquer que le serveur devrait lui substituer la liste des messages provenant de la variable de résultat de recherche. Et donc, le client peut utiliser le résultat de la dernière commande SEARCH mémorisée comme paramètre vers une autre commande. Le marqueur de résultat de recherche a plusieurs avantages :

- \* il évite le gaspillage de bande passante et les délais qui y sont associés ;
- \* il permet au client d'effectuer un traitement en parallèle d'une commande SEARCH [IMAP4] avec une commande FETCH/STORE/COPY/SEARCH [IMAP4] ou UID EXPUNGE [UIDPLUS] suivante ;
- \* le client n'a pas besoin de passer du temps à reformater le résultat d'une commande SEARCH dans un ensemble de message utilisé dans la commande suivante ;
- \* il permet au serveur de faire des optimisations. Par exemple, si le serveur peut exécuter plusieurs commandes en parallèle (ou dans le désordre), la présence du marqueur de résultat de recherche peut permettre au serveur de décider quelles commandes peuvent être exécutées en désordre ou non.

En l'absence de toute autre option de résultat de SEARCH, l'option de résultat de SAVE supprime aussi toute réponse SEARCH qui aurait autrement été retournée par la commande SEARCH.

## 1.1 Conventions utilisées dans le document

Dans les exemples, "C:" indique les lignes envoyées par un client qui est connecté à un serveur. "S:" indique les lignes envoyées du serveur au client.

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

Les commentaires explicatifs dans les exemples commencent par // et ne font pas partie du protocole.

## 2. Généralités

### 2.1. Description normative de l'extension SEARCHRES

L'extension de référence du résultat de SEARCH décrite dans le présent document est présente dans toute mise en œuvre de serveur IMAP4 qui retourne "SEARCHRES" dans les capacités prises en charge de la réponse à la commande CAPABILITY. Un tel serveur DOIT aussi mettre en œuvre l'extension [ESEARCH].

Après achèvement réussi d'une commande SELECT ou EXAMINE (après la réponse étiquetée OK), la variable de résultat de recherche en cours est remise à la séquence vide.

Une commande SEARCH réussie avec l'option de résultat SAVE règle la valeur de la variable de résultat de recherche à la liste des messages trouvés dans la commande SEARCH. Par exemple, si aucun message n'a été trouvé, la variable de résultat de recherche contiendra la liste vide.

Les commandes SEARCH suivantes NE DOIVENT PAS changer la variable de résultat de recherche :

- o une commande SEARCH qui a amené le serveur à retourner la réponse étiquetée BAD,
- o une commande SEARCH sans option de résultat SAVE qui a amené le serveur à retourner la réponse étiquetée NO,
- o une commande SEARCH réussie sans option de résultat SAVE.

Une commande SEARCH avec l'option de résultat SAVE qui a amené le serveur à retourner la réponse étiquetée NO règle la valeur de la variable de résultat de recherche à la séquence vide.

Lorsque un message qui figure dans la liste des variables de résultat de recherche est EXPUNGE (*supprimé*), il est automatiquement retiré de la liste. On rappelle aux mises en œuvre que si le serveur mémorise la liste comme une liste de numéros de message, il DOIT automatiquement les ajuster lorsqu'il notifie les messages supprimés au client, comme décrit au paragraphe 7.4.1 de [IMAP4].

Si le serveur décide d'envoyer une nouvelle valeur UIDVALIDITY alors que la boîte aux lettres est ouverte, cela cause le recalage de la variable de recherche sur la liste vide.

Noter que même si le marqueur "\$" contient la liste des messages vide, il doit être traité comme valide par toutes les commandes qui acceptent les messages établis comme paramètres, mais ne correspondant pas à la liste des messages. Par exemple, la commande "FETCH \$" retournerait une réponse étiquetée OK et pas de réponse FETCH. Voir aussi l'exemple 5 ci-dessous.

Noter que même si le marqueur "\$" contient la liste de messages vide, il doit être traité comme liste de messages valide mais non correspondante, par toutes les commandes qui acceptent les messages établis comme paramètres.

Note de mise en œuvre : les mises en œuvre de serveurs devraient noter que "\$" peut faire référence à des séquences de message IMAP ou à des séquences d'UID, selon le contexte d'utilisation. Par exemple, le marqueur "\$" peut être mis par suite d'une commande SEARCH (SAVE) et utilisé comme paramètre pour une commande UID FETCH (qui accepte une séquence d'UID, mais pas une séquence de message), ou le marqueur "\$" peut être établi par suite d'une commande UID SEARCH (SAVE) et utilisé comme paramètre pour une commande FETCH (qui accepte une séquence de message, mais pas une séquence d'UID).

## 2.2 Exemples

- 1) L'exemple suivant démontre comment le client peut utiliser le résultat d'une commande SEARCH pour aller chercher (*FETCH*) les en-têtes des messages intéressants :

Exemple 1 :

```
C: A282 SEARCH RETURN (SAVE) FLAGGED SINCE 1-Feb-1994 NOT FROM "Smith"
S: A282 OK SEARCH terminé, résultat sauvegardé
C: A283 FETCH $ (UID INTERNALDATE FLAGS RFC822.HEADER)
S: * 2 FETCH (UID 14 ...
S: * 84 FETCH (UID 100 ...
S: * 882 FETCH (UID 1115 ...
S: A283 OK terminé
```

Le client peut aussi traiter les deux commandes en parallèle :

Exemple 2 :

```
C: A282 SEARCH RETURN (SAVE) FLAGGED SINCE 1-Feb-1994 NOT FROM "Smith"
C: A283 FETCH $ (UID INTERNALDATE FLAGS RFC822.HEADER)
S: A282 OK SEARCH terminé
S: * 2 FETCH (UID 14 ...
S: * 84 FETCH (UID 100 ...
S: * 882 FETCH (UID 1115 ...
S: A283 OK terminé
```

- 2) L'exemple suivant démontre que le résultat d'une commande SEARCH peut être utilisé comme entrée d'une autre commande SEARCH :

Exemple 3 :

```
C: A300 SEARCH RETURN (SAVE) SINCE 1-Jan-2004 NOT FROM "Smith"
S: A300 OK SEARCH terminé
C: A301 UID SEARCH UID $ SMALLER 4096
S: * SEARCH 17 900 901
S: A301 OK terminé
```

Noter que la seconde commande de l'exemple 3 peut être remplacée par : C: A301 UID SEARCH \$ SMALLER 4096 et le résultat de la commande serait le même.

- 3) L'exemple suivant montre que le marqueur "\$" peut être combiné avec d'autres numéros de message utilisant le critère OR SEARCH.

Exemple 4 :

```
C: P282 SEARCH RETURN (SAVE) SINCE 1-Feb-1994 NOT FROM "Smith"
S: P282 OK SEARCH terminé
C: P283 SEARCH CHARSET UTF-8 (OR $ 1,3000:3021) TEXT {8}
C: YYYYYYYYYY
S: * SEARCH 882 1102 3003 3005 3006
S: P283 OK terminé
```

Note : Comme le format de ce document est limité à du texte ASCII à 7 bits, il n'est pas possible de montrer les données réelles en UTF-8. Le "YYYYYYYYYY" est un fourre-tout pour ce qui serait 8 octets de données dans une transaction réelle.

- 4) L'exemple suivant démontre qu'un échec de recherche (*SEARCH*) règle la variable de résultat de recherche à la liste vide.

Exemple 5 :

```
C: B282 SEARCH RETURN (SAVE) SINCE 1-Feb-1994 NOT FROM "Smith"
S: B282 OK SEARCH terminée
C: B283 SEARCH CHARSET KOI8-R (OR $ 1,3000:3021) TEXT {4}
C: XXXX
S: B283 NO [BADCHARSET UTF-8] KOI8-R n'est pas pris en charge.
// Après cette commande, la variable de résultat sauvegardée ne contient aucun message. Un client qui veut refaire la
```

commande B283 SEARCH avec un autre CHARSET devrait ressortir aussi la commande B282. Une solution de rechange possible pour cela est d'inclure le paramètre de CHARSET désiré dans la prochaine commande SEARCH RETURN (SAVE) dans une séquence des commandes SEARCH en rapport. Une meilleure approche pourrait être de toujours utiliser à la place CHARSET UTF-8.

Note : Comme le format du présent document est limité au texte ASCII à 7 bits, il n'est pas possible de montrer les données KOI8-R réelles. Le "XXXX" est un fourre tout pour ce qui devrait être 4 octets de données de 8 bits dans une transaction réelle.

5) L'exemple suivant démontre que ce n'est pas une erreur d'utiliser le marqueur "\$" lorsqu'il ne contient pas de message.

Exemple 6 :

```
C: E282 SEARCH RETURN (SAVE) SINCE 28-Oct-2006 NOT FROM "Eric"  
C: E283 COPY $ "Autres messages" //Le "$" ne contient pas de message  
S: E282 OK SEARCH terminée  
S: E283 OK COPY terminé, aucune copie
```

### 2.3 Plusieurs commandes en cours

L'utilisation de la commande SEARCH RETURN (SAVE) suivie par une commande utilisant le marqueur "\$" crée une dépendance directe entre les deux commandes. Comme indiqué au paragraphe 5.5 de [IMAP4], un serveur DOIT exécuter les deux commandes dans l'ordre où elles ont été reçues. (Un serveur capable d'exécution dans le désordre peut dans certains cas exécuter les deux commandes en parallèle, par exemple, si un SEARCH RETURN (SAVE) est suivi par "SEARCH \$", le critère de recherche de la première commande peut être directement transposé dans la seconde commande.)

Un client qui accepte cette extension PEUT exécuter en parallèle une commande SEARCH RETURN (SAVE) avec une ou plusieurs commande utilisant le marqueur "\$", pour autant que cela ne crée par d'ambiguïté, comme décrit au paragraphe 5.5 de [IMAP4].

Exemple 7 :

```
C: F282 SEARCH RETURN (SAVE) KEYWORD $Junk  
C: F283 COPY $ "Junk"  
C: F284 STORE $ +FLAGS.Silent (\Supprimé)  
S: F282 OK SEARCH terminé  
S: F283 OK COPY terminé  
S: F284 OK STORE terminé
```

Exemple 8 :

```
C: G282 SEARCH RETURN (SAVE) KEYWORD $Junk  
C: G283 SEARCH RETURN (ALL) SINCE 28-Oct-2006 FROM "Eric"  
//Le serveur peut exécuter les deux commandes SEARCH dans un ordre quelconque, car elles sont indépendantes.  
//Noter que la seconde commande utilise l'extension [ESEARCH].  
S: * ESEARCH (TAG "G283") ALL 3:15,27,29:103  
S: G283 OK SEARCH terminé  
S: G282 OK SEARCH terminé
```

L'exemple suivant montre que le résultat du second SEARCH prime toujours le résultat du premier.

Exemple 9 :

```
C: H282 SEARCH RETURN (SAVE) KEYWORD $Junk  
C: H283 SEARCH RETURN (SAVE) SINCE 28-Oct-2006 FROM "Eric"  
S: H282 OK SEARCH terminé  
S: H283 OK SEARCH terminé
```

### 2.4 Interaction avec l'extension ESEARCH

les serveurs qui mettent en œuvre l'extension définie dans le présent document DOIVENT mettre en œuvre [ESEARCH] et se conformer aux exigences supplémentaires énumérées dans ce paragraphe.

L'option de résultat SAVE ne change pas le fait que le serveur retourne les éléments correspondant aux options de résultat MIN, MAX, ALL, ou COUNT [ESEARCH].

Lorsque l'option de résultat SAVE est combinée avec l'option de résultat MIN ou MAX [ESEARCH], et qu'aucune des autres options de résultat ESEARCH n'est présente, le MIN/MAX correspondant est retourné (si le résultat de la recherche n'est pas vide), mais le marqueur "\$" va contenir un seul message comme retour dans l'élément de retour MIN/MAX.

Si l'option de résultat SAVE est combinée avec les deux options de résultat MIN et MAX, et qu'aucune des autres options de résultat ESEARCH n'est présent, le marqueur "\$" contiendra un ou deux messages comme retour des éléments de retour MIN/MAX.

Si l'option de résultat SAVE est combinée avec la ou les options de résultat ALL et/ou COUNT, le marqueur "\$" contiendra toujours tous les messages trouvés par la commande SEARCH ou UID SEARCH. (Noter que cette dernière règle peut affecter les mises en œuvre ESEARCH qui optimisent la façon dont le résultat COUNT est construit.)

Le tableau suivant résume les exigences supplémentaires décrites dans ce paragraphe pour les mises en œuvre serveur de ESEARCH.

Combinaison d'option de résultat	Valeur du marqueur "\$"
SAVE MIN	MIN
SAVE MAX	MAX
SAVE MIN MAX	MIN & MAX
SAVE * [m]	tous les messages trouvés

\*[m] signifie "TOUS" et/ou "COMPTE" : '[m]' signifie "MIN" et/ou "MAX" facultatif.

L'exemple suivant montre les différences de comportement pour différentes combinaisons d'options de résultat de ESEARCH. Les commentaires explicatifs commencent par // et ne font pas partie du protocole :

Exemple 10 :

C: C282 SEARCH RETURN (ALL) SINCE 12-Feb-2006 NOT FROM "Smith"

S: \* ESEARCH (TAG "C283") ALL 2,10:15,21 //la valeur de \$ n'a pas changé

S: C282 OK SEARCH terminé

C: C283 SEARCH RETURN (ALL SAVE) SINCE 12-Feb-2006 NOT FROM "Smith"

S: \* ESEARCH (TAG "C283") ALL 2,10:15,21 //la valeur de \$ est 2,10:15,21

S: C283 OK SEARCH terminé

C: C284 SEARCH RETURN (SAVE MIN) SINCE 12-Feb-2006 NOT FROM "Smith"

S: \* ESEARCH (TAG "C284") MIN 2 //la valeur de \$ est 2

S: C284 OK SEARCH terminé

C: C285 SEARCH RETURN (MAX SAVE MIN) SINCE 12-Feb-2006 NOT FROM "Smith"

S: \* ESEARCH (TAG "C285") MIN 2 MAX 21 //la valeur de \$ est 2,21

S: C285 OK SEARCH terminé

C: C286 SEARCH RETURN (MAX SAVE MIN COUNT) SINCE 12-Feb-2006 NOT FROM "Smith"

S: \* ESEARCH (TAG "C286") MIN 2 MAX 21 COUNT 8 //la valeur de \$ est 2,10:15,21

S: C286 OK SEARCH terminé

C: C286 SEARCH RETURN (ALL SAVE MIN) SINCE 12-Feb-2006 NOT FROM "Smith"

S: \* ESEARCH (TAG "C286") MIN 2 ALL 2,10:15,21 //la valeur de \$ est 2,10:15,21

S: C286 OK SEARCH terminé

## 2.5 Refus de sauvegarder les résultats de SEARCH

Dans certains cas, le serveur PEUT refuser de sauvegarder un résultat SEARCH (SAVE), par exemple, si une limite interne sur le nombre de résultats sauvegardés est atteinte.

Dans ce cas, le serveur DOIT retourner une réponse étiquetée NO, contenant le code de réponse NOTSAVED, et régler la

variable de résultat de recherche à séquence vide, comme décrit au paragraphe 2.1.

### 3. Syntaxe formelle

La syntaxe suivante utilise la notation de forme Backus-Naur augmentée (ABNF) comme spécifié dans [ABNF]. Sauf les terminaisons, ce qui est référencé mais non défini ci-dessous est défini dans [IMAP4] ou [IMAPABNF].

Sauf notation contraire, tous les caractères alphabétiques sont insensibles à la casse. L'utilisation de caractères majuscules ou minuscules pour définir des chaînes de jetons ne répond qu'au souci de clarté de la présentation ; Les mises en œuvre DOIVENT accepter ces chaînes de façon insensible à la casse.

capability	=/ "SEARCHRES"	:: capability est définie dans [IMAP4]
sequence-set	=/ seq-last-command	:: étend sequence-set pour permettre l'indicateur "résultat de la dernière commande".
seq-last-command	= "\$"	
search-return-opt	= "SAVE"	:: conforme à la syntaxe générique search-return-opt définie dans [IMAPABNF]
resp-text-code	=/ "NOTSAVED"	:: <resp-text-code> tiré de [IMAP4]

### 4. Considérations pour la sécurité

La présente extension exige que le serveur tienne un état supplémentaire, qui peut être utilisé pour simplifier des attaques de déni de service. Afin de minimiser les dommages résultants de telles attaques, les mises en œuvre de serveur PEUVENT limiter le nombre de recherches sauvegardées qu'ils permettent sur l'ensemble des connexions à un instant donné et retourner la réponse étiquetée NO contenant le code de réponse NOTSAVED (voir au paragraphe 2.5) à une commande SEARCH RETURN (SAVE) lorsque cette limite est dépassée.

À part cela, il est estimé que cette extension ne soulève pas de problème de sécurité qui s'ajouterait à ceux discutés dans [IMAP4].

### 5. Considérations relatives à l'IANA

Le présent document définit la capacité IMAP "SEARCHRES". L'IANA l'a ajoutée au registre IMAP4 Capabilities, qui est actuellement localisé à : <http://www.iana.org/assignments/imap4-capabilities>

### 6. Remerciements

L'auteur tient à remercier Mark Crispin, Cyrus Daboo et Curtis King de s'être souvenu que ce document devait être rédigé, ainsi que pour leurs commentaires et corrections.

Il remercie aussi Dave Cridland, Mark Crispin, Chris Newman, Dan Karp et Spencer Dawkins de leurs commentaires et corrections.

Des commentaires précieux, en accord et en désaccord, ont été fournis par Arnt Gulbrandsen.

### 7. Références

#### 7.1 Références normatives

[KEYWORDS] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, RFC 2119, mars 1997.

- [ABNF] D. Crocker, éd., et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", STD 68, RFC 5234, janvier 2008.
- [IMAP4] M. Crispin, "Protocole d'accès au message Internet - version 4rev1", RFC 3501, mars 2003.
- [IMAPABNF] A. Melnikov et C. Daboo, "Collection des extensions à IMAP4 en ABNF", RFC 4466, avril 2006.
- [ESEARCH] A. Melnikov et D. Cridland, "Extension IMAP4 à la commande SEARCH pour le contrôle du type d'information retournée", RFC 4731, novembre 2006.

## 7.2 Références informatives

- [UIDPLUS] M. Crispin, "Protocole d'accès au message Internet (IMAP) - extension UIDPLUS", RFC 4315, décembre 2005.
- [SORT] M. Crispin et K. Murchison, "Protocole d'accès au message Internet – extensions SORT et THREAD ", Travail en cours, septembre 2007.

### Adresse de l'auteur

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

## Déclaration de copyright

Copyright (C) The IETF Trust (2008).

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

Le présent document et les informations 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 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 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](mailto:ietf-ipr@ietf.org).

**Remerciement**

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