

Groupe de travail Réseau  
**Request for Comments : 5256**  
 Catégorie : Sur la voie de la normalisation  
 Traduction Claude Brière de L'Isle

M. Crispin, Panda Programming  
 K. Murchison, Carnegie Mellon University  
 juin 2008

# Extensions SORT et THREAD au protocole d'accès au message Internet

## Statut du présent mémoire

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

## Résumé

Le présent document décrit les extensions de tri et de tramage au protocole IMAP de base fondées sur le serveur. Ces extensions apportent des améliorations de performances substantielles aux clients IMAP qui offrent des vues triées et tramées.

## Table des matières

1. Introduction.....	1
2. Terminologie.....	2
2.1 Sujet de base.....	2
2.2 Date d'envoi.....	2
3. Commandes supplémentaires.....	3
4. Réponses supplémentaires.....	8
5. Syntaxe formelle des commandes et réponses SORT et THREAD.....	8
6. Considérations pour la sécurité.....	10
7. Considérations d'internationalisation.....	10
8. Considérations relatives à l'IANA.....	10
9. Références normatives.....	10
10. Références pour information.....	11
Adresse des auteurs.....	11
Déclaration complète de droits de reproduction.....	11

## 1. Introduction

Les extensions SORT et THREAD au protocole de la [RFC3501] fournissent un moyen de tri et de tramage des messages fondé sur le serveur, sans exiger que le client télécharge les données nécessaires pour le faire lui-même. Ceci est particulièrement utile pour les clients en ligne, comme décrit dans la [RFC1733].

Un serveur qui prend en charge le niveau de base de l'extension SORT l'indique avec un nom de capacité qui commence par "SORT". De futures extensions compatibles avec l'extension SORT vont commencer par "SORT", indiquant la prise en charge de ce niveau de base.

Un serveur qui prend en charge l'extension THREAD l'indique avec un ou plusieurs noms de capacité consistant en "THREAD=" suivi par un nom d'algorithme de prise en charge du tramage comme décrit dans le présent document. Cela assure de futures extensions compatibles.

Un serveur qui met en œuvre les extensions SORT et/ou THREAD DOIT collationner les chaînes en accord avec les exigences de I18NLEVEL=1, comme décrit dans la [RFC5255], et DEVRAIT mettre en œuvre et annoncer l'extension I18NLEVEL=1. Autrement, un serveur PEUT mettre en œuvre I18NLEVEL=2 (ou plus) et se conformer aux règles de ce niveau.

Discussion : les extensions SORT et THREAD précèdent la [RFC5255] de plusieurs années. Au moment de la rédaction du présent document, toutes les mises en œuvre de serveur connues de SORT et THREAD se conforment aux règles de

I18NLEVEL=1, mais ne l'annoncent pas nécessairement. Comme discuté au paragraphe 4.5 de la [RFC5255], toutes les mises en œuvre de serveur devraient éventuellement être mises à jour pour se conformer à l'extension I18NLEVEL=2.

Note historique : l'algorithme de tramage REFERENCES se fonde sur l'algorithme [THREADING] écrit et utilisé dans les versions 2.0 à 3.0 de "Messagerie et nouvelles Netscape".

## 2. Terminologie

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].

Le mot "peut" est utilisé pour se référer à une circonstance ou situation possible, par opposition à une facilité facultative du protocole.

"Utilisateur" est utilisé pour se référer à un utilisateur humain, tandis que "client" se réfère au logiciel de l'utilisateur.

Dans les exemples, "C:" et "S:" indiquent les lignes envoyées respectivement par le client et le serveur.

### 2.1 Sujet de base

Le tri et le tramage de sujet utilisent le "sujet de base", qui a les artefacts spécifiques de sujet supprimés. Du fait de la complexité de ces artefacts, la syntaxe formelle pour les règles d'extraction du sujet est ambiguë. La procédure suivante est respectée pour déterminer le "sujet de base", en utilisant les règles de syntaxe formelle de la [RFC5234] décrites à la Section 5 :

- (1) Convertir tout mot codé de la RFC 2047 en le sujet de la [RFC3629] comme décrit dans les "Considérations d'internationalisation". Convertir toutes les tabulations et continuations en espaces. Convertir toutes les espaces multiples en une seule espace.
- (2) Supprimer tout le texte en queue du sujet qui correspond à l'ABNF subj-trailer ; répéter jusqu'à ce qu'aucune correspondance ne soit possible.
- (3) Supprimer tout texte de préfixe du sujet qui correspond à l'ABNF subj-leader.
- (4) Si il y a du texte de préfixe du sujet qui correspond à l'ABNF subj-blob, et si la suppression de ce préfixe laisse une subj-base non vide, supprimer alors le texte de préfixe.
- (5) Répéter (3) et (4) jusqu'à ce qu'il ne reste plus de correspondance.

Note : Il est possible de différer l'étape (2) jusqu'à l'étape (6), mais cela exige de vérifier si il y a un subj-trailer dans l'étape (4).

- (6) Si le texte résultant commence par l'ABNF subj-fwd-hdr et se termine par l'ABNF subj-fwd-trl, supprimer le subj-fwd-hdr et le subj-fwd-trl et répéter à partir de l'étape (2).
- (7) Le texte résultant est le "sujet de base" utilisé dans SORT.

Tous les serveurs et les clients déconnectés (comme décrit dans la [RFC1733]) DOIVENT utiliser exactement cet algorithme pour déterminer le "sujet de base". Autrement, il y a une possibilité qu'un utilisateur obtienne des résultats incohérents selon qu'il fonctionne en mode connecté ou déconnecté.

### 2.2 Date d'envoi

Comme utilisé dans ce document, le terme "date d'envoi" se réfère à la date et heure provenant de l'en-tête Date:, ajustée par la zone horaire pour être normalisée en UTC. Par exemple, "31 dec 2000 16:01:33 -0800" est équivalent à la date et heure UTC de "1 jan 2001 00:01:33 +0000".

Si la zone horaire est invalide, la date et heure DEVRAIT être traitée comme UTC. Si l'heure est aussi invalide, l'heure DEVRAIT être traitée comme 00:00:00. Si il n'y a pas de date ou heure valide, la date et l'heure DEVRAIENT être traitées comme 00:00:00 à la date possible au plus tôt.

Cela diffère des critères relatifs à la date dans la commande SEARCH (décrite au paragraphe 6.4.4 de la [RFC3501]) qui utilise juste la date et non l'heure, et ne sont pas ajustées par la zone horaire.

Si la date d'envoi ne peut pas être déterminée (un en-tête Date: manque ou ne peut pas être analysé) la INTERNALDATE pour ce message est utilisée comme date d'envoi.

Quand on compare deux dates d'envoi qui correspondent exactement, l'ordre dans lequel les deux messages apparaissent dans la boîte aux lettres (c'est-à-dire, par numéro de séquence) est utilisé comme départage pour déterminer l'ordre.

### 3. Commandes supplémentaires

Ces commandes sont des extensions au protocole de base [RFC3501].

Les titres de paragraphes sont destinés à correspondre à leur localisation dans le document principal si ils faisaient partie de la spécification de base.

Commande SORT BASE.6.4.SORT.

Arguments : sort program (*programme de tri*) ; charset specification (*spécification de jeu de caractères*) ; critères de recherche (un ou plusieurs)

Données : réponses non étiquetées : SORT

Résultat : OK – tri achevé

NO – erreur de tri : ne peut pas trier ce jeu de caractères ou critère

BAD – commande inconnue ou arguments invalides

La commande SORT est une variante de SEARCH avec une sémantique de tri pour le résultat. Il y a deux arguments avant l'argument critères de recherche : une liste entre parenthèses des critères de tri, et le jeu de caractères de recherche.

L'argument "charset" est obligatoire (à la différence de SEARCH) et indique le jeu de caractères [RFC2978] des chaînes qui apparaissent dans les critères de recherche. Les jeux de caractères US-ASCII et UTF-8 [RFC3629] DOIVENT être mis en œuvre. Tous les autres jeux de caractères sont facultatifs.

Il y a aussi une commande UID SORT qui retourne des identifiants univoques au lieu des numéros de séquence de message. Noter qu'il y a des critères de recherche séparés pour les numéros de séquence de message et les UID ; donc, les arguments à UID SORT sont interprétés de la même manière que dans SORT. Ceci est analogue au comportement de UID SEARCH, par rapport à UID COPY, UID FETCH, ou UID STORE.

La commande SORT recherche d'abord dans la boîte aux lettres les messages qui correspondent aux critères de recherche donnés en utilisant l'argument "charset" pour l'interprétation des chaînes dans les critères de recherche. Elle retourne alors les messages qui correspondent dans une réponse SORT non étiquetée, triée en accord avec un ou plusieurs critères de tri.

Le tri est en ordre ascendant. Les dates antérieures sont triées avant les dates postérieures, les plus petites tailles sont triées avant les plus grandes tailles, et les chaînes sont triées selon les valeurs ascendantes établies par leur algorithme de collationnement (voir les "Considérations d'internationalisation").

Si deux messages ou plus correspondent exactement selon les critères de tri, ces messages sont triés selon l'ordre dans lequel ils apparaissent dans la boîte aux lettres. En d'autres termes, il y a un critère de tri implicite de "numéro de séquence".

Quand plusieurs critères de tri sont spécifiés, le résultat est trié dans l'ordre de priorité dans lequel les critères apparaissent. Par exemple, (SUBJECT DATE) va trier les messages dans l'ordre de leur texte de sujet de base, et pour les messages avec le même texte de sujet de base, va trier par leur date d'envoi.

Les réponses EXPUNGE non étiquetées ne sont pas permises lorsque le serveur répond à une commande SORT, mais sont permises durant une commande UID SORT.

Les critères de tri définis sont les suivants. Se référer à la section de syntaxe formelle pour les définitions syntaxiques précises des arguments. Si l'en-tête associé de la RFC-822 est absent pour un critère particulier, il est traité comme la chaîne vide. La chaîne vide est toujours collationnée avant les chaînes non vides.

ARRIVAL : date et heure interne du message. C'est différent du critère ON dans SEARCH, qui utilise juste la date interne.

CC : addr-mailbox [RFC3501] de la première adresse "cc".

DATE : date et heure d'envoi, comme décrit au paragraphe 2.2.

FROM : addr-mailbox [RFC3501] de la première adresse "From".

REVERSE : suivi par un autre critère de tri, il a l'effet de ce critère mais en ordre inverse (descendant). Note : REVERSE inverse seulement un seul critère, et n'affecte pas le critère de tri implicite "numéro de séquence" si tous les autres critères sont identiques. Par conséquent, un tri de REVERSE SUBJECT n'est pas le même que l'ordre inverse d'un tri SUBJECT. Cela peut être évité par l'utilisation de critères supplémentaires, par exemple, SUBJECT DATE vs. REVERSE SUBJECT REVERSE DATE. En général, cependant, il vaut mieux (et est plus rapide, si le client a une commande "inverser l'ordre actuel") pour inverser le résultat chez le client plutôt que de produire un nouveau SORT.

SIZE : taille du message en octets.

SUBJECT : texte du sujet de base.

TO : addr-mailbox [RFC3501] de la première adresse "To".

Exemple :

C: A282 SORT (SUBJECT) UTF-8 SINCE 1-Feb-1994

S: \* SORT 2 84 882

S: A282 OK SORT achevé

C: A283 SORT (SUBJECT REVERSE DATE) UTF-8 ALL

S: \* SORT 5 3 4 1 2

S: A283 OK SORT achevé

C: A284 SORT (SUBJECT) US-ASCII TEXT "pas dans la boîte aux lettres"

S: \* SORT

S: A284 OK SORT achevé

Commande THREAD BASE.6.4.THREAD.

Arguments : algorithme de tramage ; spécification du jeu de caractères ; critères de recherche (un ou plusieurs)

Données : réponses non étiquetées : THREAD

Résultat : OK – tramage achevé

NO – erreur de tramage : ne peut pas tramer ce jeu de caractères ou critère

BAD – commande inconnue ou arguments invalid

La commande THREAD est une variante de SEARCH avec une sémantique de tramage pour les résultats. Thread a deux arguments avant l'argument de critère de recherche : un algorithme de tramage et le jeu de caractères de la recherche.

L'argument "charset" est obligatoire (à la différence de SEARCH) et indique le jeu de caractères des chaînes qui apparaissent dans les critères de recherche. Les jeux de caractères US-ASCII et UTF-8 DOIVENT être mis en œuvre. Tous les autres jeux de caractères sont facultatifs.

Il y a aussi une commande UID THREAD qui retourne des identifiants univoques à la place des numéros de séquence des messages. Noter qu'il y a des critères de recherche séparés pour les numéros de séquence de message et les UID ; donc les arguments à UID THREAD sont interprétés de la même façon que dans THREAD. Ceci est analogue au comportement de UID SEARCH, par opposition à UID COPY, UID FETCH, ou UID STORE.

La commande THREAD cherche d'abord dans la boîte aux lettres les messages qui correspondent aux critères de recherche donnés en utilisant l'argument "charset" pour l'interprétation des chaînes dans les critères de recherche. Elle retourne alors les messages correspondants dans la réponse THREAD non étiquetée, tramée selon l'algorithme de tramage spécifié.

Tout le collationnement est fait dans l'ordre ascendant. Les dates antérieures sont collationnées avant les dates postérieures et les chaînes sont collationnées selon les valeurs ascendantes établies par leur algorithme de collationnement (voir les "Considérations d'internationalisation").

Les réponses EXPUNGE non étiquetées ne sont pas permises lorsque le serveur répond à une commande THREAD, mais sont permises durant une commande UID THREAD.

Les algorithmes de tramage définis sont les suivants :

**ORDEREDSUBJECT** : l'algorithme de tramage ORDEREDSUBJECT est aussi appelé "tramage du pauvre". Les messages cherchés sont triés par sujet de base et ensuite par date d'envoi. Les messages sont alors répartis dans des trames séparées, chaque trame contenant les messages avec le même texte de sujet de base. Finalement, les trames sont triées par date d'envoi du premier message de la trame.

Le niveau supérieur ou "racine" dans le tramage ORDEREDSUBJECT contient le premier message de chaque trame. Tous les messages dans la racine sont apparentés. Le second message d'une trame est le fils du premier message, et les messages suivants de la trame sont apparentés au second message et donc enfants du message à la racine. Donc, il n'y a pas de petits enfants dans un tramage ORDEREDSUBJECT.

Les enfants dans le tramage ORDEREDSUBJECT n'ont pas de descendants. Les mises en œuvre de clients DEVRAIENT traiter les descendants d'un enfant dans une réponse de serveur comme étant des apparentés de cet enfant.

**REFERENCES** : l'algorithme de tramage REFERENCES trame les messages recherchés en les groupant dans des relations parent/enfant sur la base des messages qui sont des réponses aux autres messages. Les relations parent/enfant sont construites en utilisant deux méthodes : reconstruction des ancêtres d'un message en utilisant les références contenues dedans, et vérification que le sujet original (non de base) d'un message est une réponse à (ou la retransmission d'un) autre message.

Note : "Message ID" dans la description qui suit se réfère à une forme normalisée de l'identifiant de message dans la [RFC2822]. Le texte actuel de la RFC 2822 peut utiliser des guillemets, résultant en plusieurs façons d'exprimer le même identifiant de message. Les mises en œuvre de l'algorithme de tramage REFERENCES DOIVENT normaliser tout msg-id afin d'éviter de fausses non correspondances à des différences de méthode de citation.

Par exemple, l'identifiant de message <"01KF8JCEOCBS0045PS"@xxx.yyy.com> et le msg-id <01KF8JCEOCBS0045PS@xxx.yyy.com> DOIVENT être interprétés comme étant le même identifiant de message.

Les références utilisées pour reconstruire les ancêtres d'un message sont trouvées en utilisant les règles suivantes :

Si un message contient une ligne d'en-tête References, utiliser alors les identifiants de message dans la ligne d'en-tête References comme références.

Si un message ne contient pas de ligne d'en-tête References, ou si la ligne d'en-tête References ne contient pas d'identifiant de message valide, utiliser alors le premier (si il en est) identifiant de message valide trouvé dans la ligne d'en-tête In-Reply-To comme la seule référence (parente) pour ce message.

Note : bien que la [RFC2822] permette plusieurs identifiants de message dans l'en-tête In-Reply-To, dans la pratique cela n'a pas été suivi. Par exemple, on a observé des en-têtes In-Reply-To avec des adresses de message après l'identifiant de message, et il n'y a pas de bonne heuristique pour que le logiciel détermine la différence. Ceci n'est cependant pas un problème avec l'en-tête References.

Si un message ne contient pas de ligne d'en-tête In-Reply-To, ou si la ligne d'en-tête In-Reply-To ne contient pas un identifiant de message valide, alors le message n'a aucune référence (NIL).

Un message est considéré être une réponse ou une retransmission si les règles d'extraction du sujet de base, appliquées au sujet original, suppriment tout ce qui suit : un subj-refwd, un "(fwd)" subj-trailer, ou un subj-fwd-hdr et un subj-fwd-trl.

L'algorithme REFERENCES est significativement plus complexe que ORDEREDSUBJECT et consiste en six étapes principales. Ces étapes sont décrites en détails ci-dessous.

(1) Pour chaque message recherché :

(A) En utilisant les identifiants de message dans les références de message, lier les messages correspondants (ceux dont la ligne d'en-tête Message-ID contient l'identifiant de message de référence donné) avec son parent/enfant. Faire de la première référence le parent de la seconde (et de la seconde un enfant de la première) de la seconde le parent de la troisième (et de la troisième un enfant de la seconde) etc. Les règles suivantes gouvernent la création de ces liens :

Si un message ne contient pas une ligne d'en-tête Message-ID, ou si l'en-tête Message-ID ne contient pas un identifiant de message valide, allouer alors un identifiant de message univoque à ce message.

Si deux messages ou plus ont le même identifiant de message, utiliser alors seulement cet identifiant de message dans le premier message (de plus bas numéro de séquence) et allouer un identifiant de message univoque à chacun des messages suivants avec un double de cet identifiant de message.

Si aucun message ne peut être trouvé avec un certain identifiant de message, créer un message factice avec cet identifiant. Utiliser ce message factice pour toutes les références suivantes à cet identifiant.

Si un message a déjà un parent, ne pas changer le lien existant. C'est parce que la ligne d'en-tête References peut avoir été tronquée par un agent d'utilisateur de messagerie (MUA, *Mail User Agent*). Par suite, il n'est pas garanti que les messages correspondants aux identifiants de message adjacents dans la ligne d'en-tête References soient parent et enfant.

Ne pas créer un lien parent/enfant si la création de ce lien introduirait une boucle. Par exemple, avant de faire du message A le parent de B, on s'assure que A n'est pas un descendant de B.

Note : les comparaisons des identifiants de message sont sensibles à la casse.

(B) Créer un lien parent/enfant entre la dernière référence (ou NIL si il n'y a pas de référence) et le message en cours. Si le message en cours a déjà un parent, il est probablement le résultat d'une ligne d'en-tête References tronquée, casser donc le lien parent/enfant en cours avant de créer le nouveau lien correct. Comme dans l'étape 1.A, ne pas créer le lien parent/enfant si la création de ce lien introduirait une boucle. Noter que si ce message n'a pas de référence, il va maintenant ne pas avoir de parent.

Note : les liens parent/enfant créés dans les étapes 1.A et 1.B DOIVENT rester cohérents les uns avec les autres à TOUT moment.

(2) Rassembler tous les messages qui n'ont pas de parents et en faire tous des enfants (apparentés les uns aux autres) d'un parent factice (la "racine"). Ces messages constituent le premier message (de tête) des trames créés jusque là.

(3) Élaguer les messages factices de l'arborescence de trame. Traverser chaque trame jusqu'à la racine, et pour chaque message :

Si il y a un message factice SANS enfant, l'éliminer.

Si il y a un message factice avec des enfants, le supprimer, mais promouvoir ses enfants au niveau actuel. En d'autres termes, les relier aux apparentés du factice.

Ne pas promouvoir les enfants si le faire les rendrait des enfants de la racine, sauf si il y a seulement un enfant.

(4) Trier les messages sous la racine (seulement les apparentés de niveau supérieur) par date d'envoi comme décrit au paragraphe 2.2. Dans le cas d'un message factice, trier ses enfants par date d'envoi et ensuite utiliser le premier enfant pour le tri de niveau supérieur.

(5) Rassembler sous la racine les messages qui ont le même texte de sujet de base.

(A) Créer un tableau, appelé le tableau des sujets, pour associer les sujets de base aux messages.

(B) Remplir le tableau des sujets avec un message par sujet de base. Pour chaque enfant de la racine :

(i) Trouver le sujet de cette trame, en utilisant le sujet de base provenant du message en cours ou de son premier enfant si le message en cours est factice. C'est le sujet de la trame.

(ii) Si le sujet de la trame est vide, sauter ce message.

(iii) Chercher le message associé au sujet de la trame dans le tableau des sujets.

(iv) Si il n'y a pas de message dans le tableau des sujets avec le sujet de la trame, ajouter le message courant et le sujet de la trame au tableau des sujets.

Autrement, si le message dans le tableau des sujet s n'est pas factice, ET si l'un des critères suivant est vrai :

- le message courant est factice, OU

- le message dans le tableau des sujets est une réponse ou une retransmission et le message courant ne l'est pas.

Remplacer alors le message dans le tableau des sujets par le message courant.

(C) Fusionner les trames avec le même sujet de trame. Pour chaque enfant de la racine :

(i) Trouver le sujet de trame du message comme à l'étape 5.B.i ci-dessus.

(ii) Si le sujet de trame est vide, sauter ce message.

(iii) Chercher le message associé à ce sujet de trame dans le tableau des sujets.

(iv) Si le message dans le tableau des sujets est le message en cours, sauter ce message.

Autrement, fusionner le message courant avec celui du tableau des sujets en utilisant les règles suivantes :

Si les deux messages sont factices, ajouter les enfants du message courant aux enfants du message dans le tableau des sujets (les enfants des deux messages deviennent apparentés) et ensuite supprimer le message courant.

Si le message dans le tableau des sujets est factice et si le message courant ne l'est pas, faire du message courant un enfant du message dans le tableau des sujets (un apparenté de ses enfants).

Si le message courant est une réponse ou une retransmission et si le message dans le tableau des sujets ne l'est pas, faire du message courant un enfant du message dans le tableau des sujets (un apparenté de ses enfants).

Autrement, créer un nouveau message factice et faire du message courant et du message dans le tableau des sujets des enfants du message factice. Puis remplacer le message dans le tableau des sujets par le message factice.

Note : les comparaisons de sujets sont insensibles à la casse, comme décrit dans les "Considérations d'internationalisation".

(6) Traverser les messages sous la racine et trier chaque ensemble d'apparentés par date d'envoi, comme décrit au paragraphe 2.2. Traverser les messages de telle façon que le "plus jeune" ensemble d'apparentés soit trié en premier, et le "plus vieil" ensemble d'apparentés soit trié en dernier (les petits enfants sont triés avant les enfants, etc). Dans le cas d'un message factice (qui peut seulement arriver avec des apparentés de niveau supérieur) utiliser son premier enfant pour le tri.

Exemple :

C: A283 THREAD ORDEREDSUBJECT UTF-8 SINCE 5-MAR-2000

S: \* THREAD (166)(167)(168)(169)(172)(170)(171)(173)(174 (175)(176)(178)(181)(180))(179)(177)(183)(182)(188)  
(184)(185)(186)(187)(189))(190)(191)(192)(193)(194 195)(196 (197)(198))(199)(200 202)(201)(203)(204)(205)(206  
207)(208)

S: A283 OK THREAD completed

C: A284 THREAD ORDEREDSUBJECT US-ASCII TEXT "gewp"

S: \* THREAD

S: A284 OK THREAD completed

C: A285 THREAD REFERENCES UTF-8 SINCE 5-MAR-2000

S: \* THREAD (166)(167)(168)(169)(172)((170)(179))(171)(173)((174)(175)(176)(178)(181)(180))((177)(183)(182)(188  
(184)(189))(185 186)(187))(190)(191)(192)(193)((194)(195 196))(197 198)(199)(200 202)(201)(203)(204)(205 206  
207)(208)

S: A285 OK THREAD completed

Note : Les sauts à la ligne dans la première et la troisième réponse du serveur sont pour faciliter la lecture et n'apparaissent pas dans les réponses THREAD réelles.

#### 4. Réponses supplémentaires

Ces réponses sont des extensions au protocole de base de la [RFC3501].

Les têtes de paragraphes de ces réponses sont destinées à correspondre avec l'endroit où elles seraient situées dans le document principal.

Réponse SORT BASE.7.2.SORT.

Données : zéro, un ou plusieurs numéros

La réponse SORT est produite suite à une commande SORT ou UID SORT. Le ou les nombres se réfèrent aux messages qui correspondent aux critères de recherche. Pour SORT, ce sont des numéros de séquence de message ; pour UID SORT, ce sont des identifiants univoques. Chaque numéro est délimité par une espace.

Exemple : S: \* SORT 2 3 6

Réponse THREAD BASE.7.2.THREAD.

Données : zéro, une ou plusieurs trames

La réponse THREAD est produite suite à une commande THREAD ou UID THREAD. Elle contient zéro, une ou plusieurs trames. Une trame consiste en une liste entre parenthèses de membres de la trame.

Les membres d'une trame consistent en zéro, un ou plusieurs numéros de message, délimités par des espaces, indiquant les parents et enfants successifs. Cela continue jusqu'à ce que la trame se partage en plusieurs sous trames, moment auquel la trame se niche dans plusieurs sous trames dont le premier membre de chaque sous trame est apparenté à ce niveau. Il n'y a pas de limite à l'incorporation de trames.

Les numéros de messages se réfèrent aux messages qui correspondent aux critères de recherche. Pour THREAD, ce sont les numéros de séquence de message ; pour UID THREAD, ce sont des identifiants univoques.

Exemple : S: \* THREAD (2)(3 6 (4 23)(44 7 96))

La première trame comporte seulement le message 2. La seconde trame consiste en les messages 3 (parent) et 6 (enfant) après quoi elle se sépare en deux sous trames, dont la première contient les messages 4 (fils de 6, apparenté à 44) et 23 (fils de 4) et dont la seconde contient les messages 44 (fils de 6, apparenté à 4) 7 (fils de 44) et 96 (fils de 7). Comme des messages ultérieurs sont parents de messages antérieurs, les messages ont probablement été déplacés d'une autre boîte aux lettres à des moments différents.

```
-- 2
-- 3
  \-- 6
    |-- 4
    |  \-- 23
    |
    \-- 44
        \-- 7
            \-- 96
```

Exemple : S: \* THREAD ((3)(5))

Dans cet exemple, 3 et 5 sont enfants d'un parent qui ne correspond pas aux critères de recherche (et/ou n'existe pas dans la boîte aux lettres) ; cependant ils sont membres de la même trame.

## 5. Syntaxe formelle des commandes et réponses SORT et THREAD

La spécification de syntaxe suivante utilise la notation en format Backus-Naur augmenté (ABNF) comme spécifié dans la [RFC5234]. Elle utilise aussi les règles de la [RFC5234] définies dans la [RFC3501].

```
sort = ["UID" SP] "SORT" SP sort-criteria SP search-criteria
```

```
sort-criteria = "(" sort-criterion *(SP sort-criterion) ")"
```

```
sort-criterion = ["REVERSE" SP] sort-key
```

```
sort-key = "ARRIVAL" / "CC" / "DATE" / "FROM" / "SIZE" / "SUBJECT" / "TO"
```

```
thread = ["UID" SP] "THREAD" SP thread-alg SP search-criteria
```



thread-alg = "ORDEREDSUBJECT" / "REFERENCES" / thread-alg-ext

thread-alg-ext = atom ; les nouveaux algorithmes DOIVENT être enregistrés par l'IANA. ;

search-criteria = charset 1\*(SP search-key)

charset = atom / quoted ; les valeurs de CHARSET DOIVENT être enregistrées par l'IANA. ;

sort-data = "SORT" \*(SP nz-number)

thread-data = "THREAD" [SP 1\*thread-list]

thread-list = "(" (thread-members / thread-nested) ")"

thread-members = nz-number \*(SP nz-number) [SP thread-nested]

thread-nested = 2\*thread-list

La syntaxe suivante décrit les règles d'extraction de sujet de base (2) à (6) :

subject = \*subj-leader [subj-middle] \*subj-trailer

subj-refwd = ("re" / ("fw" ["d"])) \*WSP [subj-blob] ":"

subj-blob = "[" \*BLOBCHAR "]" \*WSP

subj-fwd = subj-fwd-hdr subject subj-fwd-trl

subj-fwd-hdr = "[fwd:"

subj-fwd-trl = "]"

subj-leader = (\*subj-blob subj-refwd) / WSP

subj-middle = \*subj-blob (subj-base / subj-fwd)

; le dernier subj-blob est subj-base si subj-base serait vide autrement. ;

subj-trailer = "(fwd)" / WSP

subj-base = NONWSP \*( \*WSP NONWSP) ; peut être un subj-blob ;

BLOBCHAR = %x01-5a / %x5c / %x5e-ff

; tous CHAR8 sauf '[' et ']'. DEVRAIENT se conformer à la [RFC3629] ;

NONWSP = %x01-08 / %x0a-1f / %x21-ff

; tous CHAR8 autres que WSP. DEVRAIENT se conformer à la [RFC3629] ;

## 6. Considérations pour la sécurité

Les extensions SORT et THREAD ne soulèvent aucune considération de sécurité qui ne soit déjà présente dans le protocole de base [RFC3501], et ces questions sont discutées dans la [RFC3501]. Néanmoins, il est important de se rappeler que les transactions du protocole [RFC3501], y compris les données de message, sont envoyées en clair sur le réseau sauf si la protection contre l'espionnage est négociée, soit par l'utilisation de STARTTLS, la protection de la confidentialité dans AUTHENTICATE, soit par un autre mécanisme de protection.

Bien que ce ne soit pas une considération de sécurité, il est important de reconnaître que le tri par REFERENCES peut conduire à des arborescences de tramage trompeuses. Par exemple, un message avec de fausses données d'en-tête References: va causer l'incorporation d'une trame dans une autre trame.

Le processus d'extraction du sujet de base peut conduire à un collationnement incorrect si les données extraites étaient du texte significatif plutôt qu'un sujet artificiel.

## 7. Considérations d'internationalisation

Comme déclaré dans l'introduction, les règles de I18NLEVEL=1 comme décrit dans la [RFC5255] DOIVENT être suivies ; c'est-à-dire, les extensions SORT et THREAD DOIVENT collationner les chaînes conformément à la collation `i;unicode-casemap` décrite dans la [RFC5051]. Les serveurs DEVRAIENT aussi annoncer l'extension I18NLEVEL=1. Autrement, un serveur PEUT mettre en œuvre I18NLEVEL=2 (ou plus) et se conformer aux règles de ce niveau.

Comme discuté au paragraphe 4.5 de la [RFC5255], toutes les mises en œuvre de serveur devraient finalement être mises à jour pour prendre en charge l'extension I18NLEVEL=2 de la [RFC5255].

Les traductions des jetons "re" ou "fw"/"fwd" ne sont pas spécifiées comme à supprimer dans le processus d'extraction de sujet de base. Une tentative d'ajout de tels jetons traduits résulterait en une tâche géométriquement complexe, et en fin de compte impossible à mettre en œuvre.

On notera plutôt que le paragraphe 3.6.5 de la [RFC2822] recommande que "re:" (du latin "res", signifiant "en matière de") soit utilisé pour identifier une réponse. Bien qu'il soit évident que, d'après les multiples formes de jeton pour identifier un message transmis, il y a des variations considérables dans la réalité, les variations sont (quand même) gérables. Par conséquent, il est suggéré que "re:" et une des variantes des jetons pour une prise en charge de transmission par les règles d'extraction de sujet de base soit adopté pour les messages de la messagerie électronique Internet, car cela en fait une simple tâche d'affichage pour situer le jeton de langage pour l'utilisateur.

## 8. Considérations relatives à l'IANA

Les capacités de la [RFC3501] sont enregistrées par la publication d'une RFC sur la voie de la normalisation ou d'une RFC expérimentale approuvée par l'IESG. Le présent document constitue l'enregistrement des capacités SORT et THREAD dans le registre des capacités de la [RFC3501].

Le présent document crée un nouveau registre des algorithmes de tramage de la [RFC3501] qui enregistre les algorithmes de tramage en publiant une RFC sur la voie de la normalisation ou une RFC expérimentale approuvée par l'IESG. Le présent document constitue l'enregistrement des algorithmes ORDEREDSUBJECT et REFERENCES dans ce registre.

## 9. 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](#))
- [RFC2822] P. Resnick, "[Format de message Internet](#)", avril 2001. (Remplace la [RFC0822](#), STD 11, Remplacée par [RFC5322](#))
- [RFC2978] N. Freed et J. Postel, "Procédures d'enregistrement des jeux de caractère par l'IANA", BCP 19, octobre 2000.
- [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.
- [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](#))
- [RFC5255] C. Newman et autres, "[Internationalisation du protocole d'accès](#) au message Internet", juin 2008. (P.S.)

## 10. Références pour information

[RFC1733] M. Crispin, "Modèles de messagerie électronique répartie dans IMAP4", décembre 1994. (*Information*)

[THREADING] J. Zawinski, "Message Threading", <http://www.jwz.org/doc/threading.html>, 1997-2002.

## Adresse des auteurs

Mark R. Crispin  
Panda Programming  
6158 NE Lariat Loop  
Bainbridge Island, WA 98110-2098  
téléphone : +1 (206) 842-2385  
mél : [IMAP+SORT+THREAD@Lingling.Panda.COM](mailto:IMAP+SORT+THREAD@Lingling.Panda.COM)

Kenneth Murchison  
Carnegie Mellon University  
5000 Forbes Avenue  
Cyert Hall 285  
Pittsburgh, PA 15213  
mél : [murch@andrew.cmu.edu](mailto:murch@andrew.cmu.edu)

## 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 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).