

Groupe de travail Réseau  
**Request for Comments : 4469**  
RFC mises à jour : 3501, 3502  
Catégorie : En cours de normalisation

P. Resnick  
QUALCOMM Incorporated  
avril 2006  
Traduction Claude Brière de L'Isle

# Protocole d'accès aux messages Internet (IMAP) Extension CATENATE

## Statut du présent mémoire

Le présent document spécifie un protocole de normalisation Internet pour la communauté Internet, et appelle à discussion et suggestions en vue de son amélioration. Prière de se rapporter à l'édition en cours des "Internet Official Protocol Standards" (normes officielles du protocole Internet) (STD 1) pour connaître l'état de la normalisation et le statut du présent protocole. La distribution du présent mémo n'est pas soumise à restrictions.

## Déclaration de copyright

Copyright (C) The Internet Society (2006).

## Résumé

L'extension CATENATE au protocole d'accès aux messages Internet (IMAP, *Internet Message Access Protocol*) étend la commande APPEND pour permettre aux clients de créer sur le serveur IMAP des messages qui peuvent contenir une combinaison de nouvelles données avec des parties de messages (ou des messages entiers) déjà sur le serveur. En utilisant cette extension, le client peut enchaîner des parties d'un message déjà existant à un nouveau message sans avoir à télécharger d'abord les données puis à les recharger sur le serveur.

## 1. Introduction

L'extension CATENATE au protocole d'accès aux messages Internet (IMAP) [1] permet au client de créer sur le serveur un message qui puisse inclure le texte de messages (ou parties de messages) qui existent déjà sur le serveur sans avoir à les extraire par la commande FETCH et à les rajouter ensuite sur le serveur par la commande APPEND. L'extension CATENATE étend la commande APPEND de telle sorte que, au lieu du texte d'un seul message, la commande peut prendre comme arguments toute combinaison des textes de messages (comme décrit dans IMAP [1]) et d'URL de messages (comme décrit dans la spécification de schéma d'URL IMAP [2]). Le serveur prend tous les morceaux et les enchaîne dans le message de sortie. L'extension CATENATE peut aussi coexister avec l'extension MULTIAPPEND [3] pour ajouter plusieurs messages en une seule commande.

Certains usages de l'extension CATENATE sont évidents. Le cas d'utilisation qui a motivé sa création était de fournir à un client ayant des contraintes de ressources le moyen de composer un message à soumettre ultérieurement, qui contienne des données déjà existantes dans la mémoire IMAP de ce client. Parce que le client n'a pas à télécharger et recharger des parties de messages qui peuvent être volumineuses, les limitations de bande passante et de traitement n'ont pas autant d'impact. De plus, comme le client peut créer un message dans sa propre mémoire IMAP, la commande répond aussi au désir du client d'archiver une copie d'un message envoyé sans avoir à télécharger deux fois le message. (Les mécanismes d'envoi du message sortent du domaine d'application du présent document.)

La commande APPEND étendue peut aussi être utilisée pour copier des parties d'un message dans une autre boîte aux lettres pour des besoins d'archivage tout en se débarrassant de parties non désirées. Dans des environnements où la mémoire du serveur est limitée, un client pourrait se débarrasser de parties de grands messages en ne recopiant que la partie nécessaire et en supprimant ensuite le message original. Le mécanisme pourrait aussi être utilisé pour ajouter des données à un message (comme d'ajouter des champs d'en-tête au message) ou d'inclure d'autres données en faisant une copie de l'original et d'y enchaîner les nouvelles données.

## 2. Capacité CATENATE

Un serveur qui accepte cette extension retourne "CATENATE" comme une des réponses à la commande CAPABILITY.

### 3. Commande APPEND

Arguments : nom de boîte aux lettres

(Ce qui suit peut être répété en présence de l'extension MULTIAPPEND [3])

Liste entre parenthèse de fanions FACULTATIVE

Chaîne date/heure FACULTATIVE

un seul texte de message ou une ou plusieurs parties de message à enchaîner, spécifié par :

le texte du message, ou

l'URL du message (ou de la partie de message)

Réponses : réponses NON FACULTATIVES : BADURL, TOOBIG

Résultat : OK – ajout achevé

NO – erreur d'ajout : impossible d'ajouter à la boîte aux lettres, erreur dans les fanions ou dans la date/heure ou dans le texte du message, ou ne peut pas extraire ces données

BAD – commande inconnue ou arguments invalides

La commande APPEND enchaîne toutes les parties du message et les ajoute comme un nouveau message à la fin de la boîte aux lettres spécifiée. La liste des fanions entre parenthèses et la chaîne date/heure établissent respectivement les fanions et la date interne, comme décrit dans IMAP [1]. Les paramètres de commande suivants spécifient les parties du message qui sont ajoutées successivement au message de sortie.

Si la forme originale de APPEND est utilisée, un texte de message suit la liste des fanions facultatifs et la chaîne date/heure, et il est ajouté comme décrit dans IMAP [1]. Si la forme étendue est utilisée, "CATENATE" et une liste entre parenthèses de textes de message et d'URL de message suit, chacun étant ajouté au nouveau message. Si un texte de message est spécifié (indiqué par "TEXT"), les octets sont ajoutés suivant le compte. Si un URL de message est spécifié (indiqué par "URL"), les octets de la partie de corps pointée par cet URL sont ajoutés, comme si le texte retourné dans une réponse FETCH BODY était mis à la place du spécificateur de partie de message. La commande APPEND ne cause pas l'établissement du fanion \Seen pour une des parties de corps de message enchaînées. La commande APPEND ne change pas les boîtes aux lettres choisies.

Dans la commande APPEND étendue, la chaîne qui suit "URL" est un URL IMAP [2] et est interprétée conformément aux règles de [2]. Le présent document ne décrit que le comportement de la commande utilisant les URL IMAP qui se réfèrent à des messages ou parties de message spécifiques sur le serveur IMAP en cours à partir de la session IMAP actuellement authentifiée. À cause de cela, seuls les URL relatifs au message ou partie de message IMAP (c'est-à-dire, ceux qui n'ont pas de schéma ou de <iserver>) sont utilisés. L'URL de base pour évaluer l'URL relatif est considéré comme "imap://user@server/", où "user" est le nom d'utilisateur de l'utilisateur actuellement authentifié et "server" est le nom de domaine du serveur actuel. Lorsqu'il est dans l'état sélectionné, l'URL de base est considéré comme "imap://user@server/mailbox", où "mailbox" est le nom codé de la boîte aux lettres actuellement choisie. De plus, comme la commande APPEND est valide dans l'état authentifié d'une session IMAP, aucune autre commande LOGIN ou AUTHENTICATE n'est effectuée pour les URL spécifiés dans la commande APPEND étendue.

Note : L'utilisation d'un URL IMAP absolu ou de tout URL qui se réfère à quelque chose d'autre qu'un message ou partie de message provenant de la session IMAP authentifiée en cours sort du domaine d'application du présent document et exigerait une extension de cette spécification, et un serveur qui ne met en œuvre que la présente spécification retournerait NON à une telle demande.

Le client est chargé de s'assurer que le message enchaîné est dans le format de message Internet de la RFC 2822 [4] ou dans le format de message des extensions de messagerie Internet multi objets (MIME, *Multipurpose Internet Mail Extension*) [5]. En particulier, lorsque un URL est enchaîné, le serveur copie les octets, inchangés, depuis le message ou la partie de message indiquée du message enchaîné. Il ne fait pas de conversion de données (par exemple, de codages de transfert MIME) ni aucune vérification que les données sont appropriées pour la partie MIME du message dans lequel elles sont insérées. Le client est aussi chargé d'insérer les limites MIME appropriées entre les parties de corps de message, et d'écrire les lignes de Content-Type et Content-Transfer-Encoding de MIME en tant que de besoin aux endroits appropriés.

Les réponses se comportent juste comme la commande APPEND originale décrite dans IMAP [1]. Si le serveur met en œuvre l'extension IMAP UIDPLUS [6], il va aussi retourner un code de réponse APPENDUID dans la réponse OK étiquetée. Deux codes de réponse sont données à la Section 4, qui peuvent être utilisés dans la réponse NO étiquetée si la commande APPEND échoue

## 4. Codes de réponse

Lorsqu'une commande APPEND échoue, elle peut retourner un code de réponse qui décrit la raison de l'échec.

### 4.1 Réponse BADURL

Le code de réponse BADURL est retourné si la commande APPEND échoue à traiter un des URL spécifiés. Les causes possible en sont une mauvaise syntaxe d'URL, un schéma d'URL non reconnu, un UID de message invalide, ou une partie de corps de message invalide. Le code de réponse BADURL contient le premier URL spécifié comme paramètre de la commande APPEND qui a causé l'échec de l'opération.

### 4.2 Réponse TOOBIG

Le code de réponse TOOBIG est retourné si le message résultant excéderait la limite de message IMAP de 4-GB. Cela peut arriver, par exemple, si le client spécifie 3 URL pour 2-GB messages. Noter que même si le serveur ne retourne pas TOOBIG, il doit quand même être sur la défensive à l'égard de clients qui se comportent mal ou nuisibles qui essayent de construire un message au-delà de la limite de 4-GB. Le serveur peut aussi souhaiter retourner le code de réponse TOOBIG si le message résultant excède une limite de taille de message spécifique du serveur.

## 5. Syntaxe formelle

La spécification de syntaxe suivante utilise la notation de forme Backus-Naur augmentée (ABNF) [7]. Les éléments non définis ici pourront être trouvés dans les spécifications de la syntaxe formelle de l'ABNF [7], d'IMAP [1], et des extensions ABNF d'IMAP [8]. Noter que capability et resp-text-code sont étendus à partir de la spécification IMAP [1] et que append-data est étendu à partir de la spécification des extensions ABNF d'IMAP [8].

```

append-data =/ "CATENATE" SP "(" cat-part *(SP cat-part) ")"
cat-part = text-literal / url
text-literal = "TEXT" SP literal
url = "URL" SP astring
resp-text-code =/ toobig-response-code / badurl-response-code
toobig-response-code = "TOOBIG"
badurl-response-code = "BADURL" SP url-resp-text
url-resp-text = 1*(%x01-09 /
    %x0B-0C /
    %x0E-5B /
    %x5D-FE) ; Any TEXT-CHAR except "]"
capability =/ "CATENATE"

```

Le astring dans la définition d'url et le url-resp-text dans la définition de badurl-response-code contiennent chacun un imapurl comme défini par [2].

## 6. Remerciements

Merci aux membres du groupe de travail LEMONADE pour leurs contributions. Des remerciements particuliers à Alexey Melnikov pour les exemples.

## 7. Considérations sur la sécurité

L'extension CATENATE ne soulève aucun problème de sécurité qui ne soit déjà présent pour le protocole de base ou dans l'utilisation des URL IMAP, et ces problèmes sont exposés dans les documents IMAP [1] et URL IMAP [2].

## 8. Considérations relatives à l'IANA

Les capacités d'IMAP4 sont enregistrées par la publication de RFC en cours de normalisation ou de RFC expérimentales approuvées par l'IESG. Le registre est actuellement localisé à <<http://www.iana.org/assignments/imap4-capabilities>>. Le présent document définit la capacité IMAP CATENATE. L'IANA a ajouté cette capacité au registre.

### Appendice A. Exemples

Les lignes qui ne commencent pas par "C: " ou "S: " sont les continuations des lignes précédentes.

Le message d'origine dans les exemples 1 et 2 ci-dessous (UID = 20) a la structure suivante :

Message MIME multipart/mélangé avec deux parties de corps :

1. texte/clair
2. application/x-zip-compressé

Exemple 1 : L'exemple suivant montre comment un client CATENATE peut remplacer une pièce jointe dans un projet de message, sans avoir besoin de le télécharger au client et de le recharger.

```
C: A003 APPEND Drafts (\Seen \Draft $MDNSent) CATENATE (URL
"/Drafts;UIDVALIDITY=385759045;/UID=20;/section=HEADER" TEXT {42}
S: + Prêt pour données littérales
C:
C: -----030308070208000400050907
C: URL "/Drafts;UIDVALIDITY=385759045;/UID=20;/section=1.MIME"
URL "/Drafts;UIDVALIDITY=385759045;/UID=20;/section=1" TEXT {42}
S: + Prêt pour données littérales
C:
C: -----030308070208000400050907
C: URL "/Drafts;UIDVALIDITY=385759045;/UID=30" TEXT {44}
S: + Prêt pour données littérales
C:
C: -----030308070208000400050907--
C: )
S: A003 OK ajout de catenate terminé
```

Exemple 2 : L'exemple suivant montre comment l'extension CATENATE peut être utilisée pour remplacer du texte édité dans un projet de message, aussi bien que des champs d'en-tête pour la partie supérieure du message (par exemple, le Sujet a changé). La précédente version du projet est marquée comme \Deleted (*supprimée*). Noter que le serveur prend aussi en charge l'extension UIDPLUS, de sorte que le code de réponse APPENDUID est retourné dans la réponse de succès OK à la commande APPEND.

```
C: A003 APPEND Drafts (\Seen \Draft $MDNSent) CATENATE (TEXT {738}S: + Prêt pour données littérales
C: Return-Path: <bar@example.org>
C: Received: from [127.0.0.2]
C: by rufus.example.org via TCP (internal) with ESMTPA;
C: Thu, 11 Nov 2004 16:57:07 +0000
C: Message-ID: <419399E1.6000505@example.org>
C: Date: Thu, 12 Nov 2004 16:57:05 +0000
C: From: Bob Ar <bar@example.org>
C: X-Accept-Language: en-us, en
C: MIME-Version: 1.0
C: To: foo@example.net
C: Sujet : Notre voyage de vacances
C: Content-Type: multipart/mixed;
C: boundary="-----030308070208000400050907"
C:
C: -----030308070208000400050907
C: Content-Type: texte/clair; charset=us-ascii; format=flowed
```

```

C: Content-Transfer-Encoding: 7bit
C:
C: Notre agence de voyage nous a envoyé le programme mis à jour.
C:
C: Amitiés,
C: Bob
C: -----030308070208000400050907
C: URL "/Drafts;UIDVALIDITY=385759045;/UID=20;/Section=2.MIME"
  URL "/Drafts;UIDVALIDITY=385759045;/UID=20;/Section=2" TEXT {44}
S: + Prêt pour données littérales
C:
C: -----030308070208000400050907--
C: )
S: A003 OK [APPENDUID 385759045 45] ajout Terminé
C: A004 UID STORE 20 +FLAGS.SILENT (\Deleted)
S: A004 OK STORE terminé

```

Exemple 3 : L'exemple suivant montre comment l'extension CATENATE peut être utilisée pour ôter des pièces jointes. Ci-dessous, une pièce jointe PowerPoint a été remplacée par un petit texte expliquant que la pièce jointe a été ôtée.

```

C: A003 APPEND Drafts (\Seen \Draft $MDNSent) CATENATE (URL
"/Drafts;UIDVALIDITY=385759045;/UID=21;/section=HEADER" TEXT {42}
S: + Prêt pour données littérales
C:
C: -----030308070208000400050903
C: URL "/Drafts;UIDVALIDITY=385759045;/UID=21;/section=1.MIME" URL
"/Drafts;UIDVALIDITY=385759045;/UID=21;/section=1" TEXT {255}
S: + Prêt pour données littérales
C:
C: -----030308070208000400050903
C: Content-type: texte/clair; charset="us-ascii"
C: Content-transfer-encoding: 7bit
C:
C: Cette partie du corps de message contenait une présentation en Power Point qui a été supprimée à votre demande.
C: -----030308070208000400050903--
C: )
S: A003 OK ajout terminé

```

Exemple 4 : L'exemple suivant montre l'échec d'une commande APPEND. Le serveur retourne le code de réponse BADURL pour indiquer qu'un des URL fournis est invalide. Cet exemple montre aussi comment l'extension CATENATE peut être utilisée pour construire un résumé de plusieurs messages.

```

C: A003 APPEND Sent (\Seen $MDNSent) CATENATE (TEXT {541}
S: + Prêt pour données littérales
C: Return-Path: <foo@example.org>
C: Received: from [127.0.0.2]
  by rufus.example.org via TCP (interne) avec ESMTPA;
  Thu, 11 Nov 2004 16:57:07 +0000
C: Message-ID: <419399E1.6000505@example.org>
C: Date: Thu, 21 Nov 2004 16:57:05 +0000
C: From: Farren Oo <foo@example.org>
C: X-Accept-Language: en-us, en
C: MIME-Version: 1.0
C: To: bar@example.org
C: Sujet : Résumé de la liste des messages d'aujourd'hui
C: Content-Type: multipart/digest;
  boundary="-----030308070208000400050904"
C:
C: -----030308070208000400050904
C: URL "/INBOX;UIDVALIDITY=785799047;/UID=11467" TEXT {42}
S: + Prêt pour données littérales
C:
C: -----030308070208000400050904

```

C: URL "/INBOX;UIDVALIDITY=785799047;/UID=113330;/section=1.5.9" TEXT {42}  
 S: + Prêt pour données littérales  
 C:  
 C: -----030308070208000400050904  
 C: URL "/INBOX;UIDVALIDITY=785799047;/UID=11916" TEXT {44}  
 S: + Prêt pour données littérales  
 C:  
 C: -----030308070208000400050904--  
 C: )  
 S: A003 NO [BADURL "/INBOX;UIDVALIDITY=785799047;/UID=113330; section=1.5.9"] l'ajout CATENATE a échoué, un message supprimé

Noter que le serveur aurait pu valider les URL lors de leur réception et aurait donc pu retourner la réponse étiquetée NO avec le code de réponse BADURL à la place de la continuation de la requête après la réception de l'URL.

## 9. Références normatives

- [1] M. Crispin, "Protocole d'[accès aux messages Internet](#) - Version 4 rev1", RFC 3501, mars 2003.
- [2] C. Newman, "Schéma d'URL IMAP", RFC 2192, septembre 1997.
- [3] M. Crispin, "Protocole d'[accès aux messages Internet](#) (IMAP) - Extension MULTIAPPEND", RFC 3502, mars 2003.
- [4] P. Resnick, "[Format de message Internet](#)", RFC 2822, avril 2001.
- [5] N. Freed et N. Borenstein, "[Extensions multi-usage de messagerie Internet](#) (MIME) Partie une : Format des corps de message Internet", RFC 2045, novembre 1996.
- [6] M. Crispin, "Protocole d'accès aux messages Internet (IMAP) – Extension UIDPLUS", RFC 4315, décembre 2005.
- [7] D. Crocker et P. Overell, "[BNF augmenté](#) pour les spécifications de syntaxe : ABNF", RFC 4234, octobre 2005.
- [8] A. Melnikov et C. Daboo, "Extensions collectées à l'ABNF IMAP4", RFC 4466, avril 2006.

### Adresse de l'auteur

Peter W. Resnick  
 QUALCOMM Incorporated  
 5775 Morehouse Drive San Diego, CA 92121-1714  
 US  
 Téléphone : +1 858 651 4478  
 mèl : presnick@qualcomm.com  
 URI : <http://www.qualcomm.com/~presnick/>

### Déclaration de copyright

Copyright (C) The Internet Society (2006).

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 l'activité de soutien administratif (IASA) de l'IETF.