

Network Working Group
Internet-Draft
Expires: December 19, 2006

A. Melnikov
Isode Ltd
D. Cridland
Inventure Systems Ltd
C. Wilson
Nokia
June 17, 2006

IMAP4 Extensions for Quick Mailbox Resynchronization
draft-ietf-lemonade-reconnect-client-00.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on December 19, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document defines an IMAP4 extension, which gives an IMAP client the ability to quickly resynchronize any previously opened mailbox. [[anchor1: Need more text here, we already have a quick reconnect extension called CONDSTORE.]]

Table of Contents

1.	Requirements notation	3
2.	Introduction and Overview	3
3.	Quick resynchronization extension	3
3.1.	QRESYNC parameter to SELECT/EXAMINE	4
4.	Formal Syntax	6
5.	Security Considerations	6
6.	IANA Considerations	6
7.	Normative References	7
	Authors' Addresses	8
	Intellectual Property and Copyright Statements	9

1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

2. Introduction and Overview

IMAP [\[CONDSTORE\]](#) extension provides a way to quickly resynchronize changes to IMAP flags since a known moment. This can be done using FETCH modifier once a mailbox is opened.

The [\[EXPUNGED\]](#) IMAP extension allows to quickly discover expunged messages.

The IMAP extension described in this document combines functionality provided by both [\[CONDSTORE\]](#) and [\[EXPUNGED\]](#) extensions, while reducing one more round-trip.

In particular this extension can be useful for mobile clients that can experience frequent disconnects caused by environmental factors (battery life, signal strength, etc.). Such clients would need a way to quickly reconnect to the IMAP server, without forcing the user to experience long delay and pay big bills for the amount of traffic generated by resynchronization.

[[anchor4: Note to RFC editor: This document is compliant with "transitional IMAP capabilities" document [\[TRANS-CAPA\]](#). Please change the capability name below to "RECONNECT"]]

The quick resync IMAP extension is present if an IMAP4 server returns "X-DRAFT-W00-QRESYNC" as one of the supported capabilities to the CAPABILITY command. Note, that this extension REQUIRES support for the [\[CONDSTORE\]](#) and the [\[EXPUNGED\]](#) IMAP extensions, so they MUST be announced in the CAPABILITY response as well.

3. Quick resynchronization extension

3.1. QRESYNC parameter to SELECT/EXAMINE

The Quick Resynchronization parameter to SELECT/EXAMINE commands takes three arguments: the mailbox UIDVALIDITY, the last known modification sequence and the optional list of known UIDs.

Before opening the specified mailbox the server verifies all arguments for syntactic validity. If any parameter is not syntactically valid, the server return the tagged BAD response, and the mailbox remains unselected. Once the check is done the server opens the mailbox as if no SELECT/EXAMINE parameters are specified (this is subject to processing of other parameters as defined in other extensions). In particular this means that server MUST send all untagged responses as specified in [Section 6.3.1/6.3.2](#) of [\[RFC3501\]](#).

After that the server checks the UIDVALIDITY value provided by the client. If the provided UIDVALIDITY doesn't match the UIDVALIDITY for the mailbox being opened, then the server MUST ignore the remaining parameters and behave as if each message in the mailbox had some flag changed and the client specified "1:*" as the UID list parameter. (See below.) [[anchor7: Should we give the client a possibility not to synchronize the mailbox instead?]]

Example: Attempting to resynchronize INBOX, but the provided UIDVALIDITY parameter doesn't match the current UIDVALIDITY value.

```
C: A02 SELECT INBOX (QRESYNC (67890007 20050715194045000
41,43:211,214:541))
S: * 464 EXISTS
S: * 3 RECENT
S: * OK [UIDVALIDITY 3857529045] UIDVALIDITY
S: * OK [UIDNEXT 550] Predicted next UID
S: * OK [HIGHESTMODSEQ 20060128194045007]
S: * OK [UNSEEN 12] Message 12 is first unseen
S: * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
S: * OK [PERMANENTFLAGS (\Answered \Flagged \Draft
\Deleted \Seen \*)] Permanent flags
S: * 1 FETCH (UID 4 FLAGS (\Seen \Answered $MDNSent))
...
S: * 464 FETCH (UID 601 FLAGS (\Draft))
S: A02 OK [READ-WRITE] Sorry, UIDVALIDITY mismatch
```

Modification Sequence and UID Parameters:

If the provided UIDVALIDITY matches that of the selected mailbox, the server then checks the last known modification sequence.

The server sends the client any pending flag changes (using FETCH responses that MUST contain UIDs) and expunges that have occurred in this mailbox since the provided modification sequence.

If the list of known UIDs was also provided, the server should only report flag changes and expunges for the provided messages. If the client doesn't provide the list of UIDs, the server acts as if the client has specified "1:*".

Thus, the client can process just these pending events and need not perform a full resynchronization. The result of this step is semantically equivalent to the client issuing:
tag1 UID FETCH "known-uids" (FLAGS) (CHANGEDSINCE "mod-sequence-value" REPORTEXPUNGES)

Example:

```
C: A02 SELECT INBOX (QRESYNC (67890007
    20060115194045000 41,43:211,214:541))

S: * 314 EXISTS

S: 15 RECENT

S: * OK [UIDVALIDITY 67890007] UIDVALIDITY

S: * OK [UIDNEXT 567] Predicted next UID

S: * OK [HIGHESTMODSEQ 20060115205545359]

S: * OK [UNSEEN 7] There are some unseen messages in the mailbox

S: * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)

S: * OK [PERMANENTFLAGS (\Answered \Flagged \Draft
    \Deleted \Seen *)] Permanent flags

S: * 49 FETCH (UID 117 FLAGS (\Seen \Answered))

S: * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent))

S: ...

S: * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded))
```


S: * EXPUNGED 41,43:116,118,120:211,214:540

S: A02 OK [READ-WRITE] mailbox selected

4. Formal Syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) notation as specified in [ABNF].

Non-terminals referenced but not defined below are as defined by [RFC3501], [CONDSTORE], [EXPUNGED] or [IMAPABNF].

Except as noted otherwise, all alphabetic characters are case-insensitive. The use of upper or lower case characters to define token strings is for editorial clarity only. Implementations MUST accept these strings in a case-insensitive fashion.

select-param = "QRESYNC" SP "(" uidvalidity SP
 mod-sequence-value [SP known-uids] ")"
 ;; conforms to the generic select-param
 ;; syntax defined in [IMAPABNF]

uidvalidity = nz-number

known-uids = sequence-set
 ;; sequence of UIDs, "*" is not allowed

5. Security Considerations

Security considerations relevant to [CONDSTORE] and [EXPUNGED] are also relevant to this extension.

[[anchor10: TBD]]

6. IANA Considerations

IMAP4 capabilities are registered by publishing a standards track or IESG approved experimental RFC. The registry is currently located at:

<http://www.iana.org/assignments/imap4-capabilities>

This document defines the X-DRAFT-W00-QRESYNC [[anchor12: Fix capability name upon publication]] IMAP capability. IANA is requested to add this capability to the registry.

7. Normative References

- [ABNF] Crocker, D., Ed. and P. Overell, Ed., "Augmented BNF for Syntax Specifications: ABNF", [RFC 4234](#), October 2005.
- [CONDSTORE] Melnikov, A. and S. Hole, "IMAP Extension for Conditional STORE Operation or Quick Flag Changes Resynchronization", [RFC 4551](#), June 2006.
- [EXPUNGED] Melnikov, A., "IMAP4 extension for reporting expunged messages", June 2006.
- [IMAPABNF] Melnikov, A. and C. Daboo, "Collected Extensions to IMAP4 ABNF", [RFC 4466](#), April 2006.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", [RFC 3501](#), March 2003.

Authors' Addresses

Alexey Melnikov
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

Email: Alexey.Melnikov@isode.com

Dave Cridland
Inventure Systems Ltd
21, Heol Bronwydd
Caerfyrddin, Cymru SA31 2AJ
GB

Email: dave.cridland@invsys.co.uk

Corby Wilson
Nokia
5 Wayside Rd.
Burlington, MA 01803
USA

Email: corby@computer.org

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

