Network Working Group
Request for Comments: DRAFT
draft-gulbrandsen-imap-nostore-00.txt

Arnt Gulbrandsen
Oryx Mail Systems GmbH

February 2006

The IMAP NOSTORE Extension

Status of this Memo

Copyright Notice

Abstract

The NOSTORE extension allows an IMAP server to send EXPUNGE and EXISTS responses to a support client at any time, while preserving the advantages of message-number arithmentic.

The extension requires that UID STORE be used instead of STORE.

Conventions Used in This Document

The key words "REQUIRED", "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [KEYWORDS]. Formal syntax is defined by [ABNF] as modified by [IMAP] and [IMAPABNF].

In the example, "C:" and "S:" indicate lines sent by the client and server respectively.

Introduction

An [IMAP] server that supports this extension announces "NOSTORE" as one of its capabilities. This extension adds one new select parameter, no commands and no responses.

While the extension is active, the server notifies the client at once about external changes to the mailbox.

Client Requirements

To enable NOSTORE for a mailbox session, the client uses the NOSTORE select-param on the SELECT command.

In such a session, the client MUST NOT use the STORE command. It should always use the UID STORE command instead.

The client MUST tolerate that EXPUNGE, EXISTS and other untagged responses can arrive at any time, even when no command is being executed.

It follows that commands with MSN arguments are best not used. This includes FETCH, COPY, often SEARCH and sometimes UID SEARCH. The commands UID STORE, UID FETCH, UID COPY and UID SEARCH can do the same job.

Server Requirements

When NOSTORE has been enabled for a mailbox session, the server changes its behaviour as follows:

The server MUST reject the STORE command with a BAD response.

If a command is found to contains MSNs outside the currently valid range, the server MUST silently ignore these MSNs. (This is similar to how UIDs are handled.)

The server SHOULD send EXISTS and EXPUNGED responses promptly when messages are added to the mailbox or expunged from the mailbox. The server MAY send flag updates and other unsolicited responses at any time.

Examples

In examples, some lines have been wrapped for editorial clarity.

In this example, a client selects a mailbox and updates its UID and flag cache.

```
C: a SELECT INBOX (NOSTORE)
S: * FLAGS (\Deleted \Answered \Flagged \Draft \Seen)
S: * 999 EXISTS
S: * 0 RECENT
S: * OK [UIDNEXT 10001] next uid
S: * OK [UIDVALIDITY 1] uid validity
S: * OK [PERMANENTFLAGS (\Deleted \Answered \Flagged \Draft
   \Seen \*)] permanent flags
S: a OK [READ-WRITE] Mailbox selected with NOSTORE active
C: b UID FETCH 1:* (FLAGS)
S: * 1 FETCH (UID 10 FLAGS (\seen))
S: * 2 FETCH (UID 20 FLAGS (\seen))
[996 similar responses elided]
S: * 999 FETCH (UID 10000 FLAGS ())
S: c OK That was fun
```

In the next example, an advanced client uses MSN arithmetic to do the same job much more efficiently. Before connecting, the client has 1000 messages cached, with UIDs 10, 20 and so on to 10000. When connecting, it sees that the server has just 999 messages, and so knows that at least one message has been deleted.

```
C: a SELECT INBOX (NOSTORE)
S: * FLAGS (\Deleted \Answered \Flagged \Draft \Seen)
S: * 999 EXISTS
S: * 0 RECENT
S: * OK [UIDNEXT 10001] next uid
S: * OK [UIDVALIDITY 1] uid validity
S: * OK [PERMANENTFLAGS (\Deleted \Answered \Flagged \Draft
   \Seen \*)] permanent flags
S: a OK [READ-WRITE] Mailbox selected with NOSTORE active
```

To find out which message was deleted without using too much bandwidth, this very smart client starts with a search command using MSNs:

```
C: d UID SEARCH 1,200,400,600,800,999,* UID
S: * SEARCH 10,2000,4000,6000,8000,10000
S: d OK Search completed
```

At this time, the client knows the approximate UID of the deleted messages. The server has 198 messages with UIDs between 10 and 2000, 199 between 2000 and 4000, etc. The client determines that since the server has 198 messages between UID 8000 and 10000 and its own cache contains 199 messages, the deleted message must be in this range. Accordingly it sends another search:

```
C: e UID SEARCH 801:998
S: * SEARCH 8010,8030,8040,8040,[196 more UIDs elided]
S: e OK Search completed
```

The second search reveals that the cached message with UID 8020 has been expunged.

Note that this command sequence works perfectly even if the server expunges messages at the same time.

If the server supports [CONDSTORE], the client can now go on to update its flag cache.

```
C: f UID FETCH 1:* FLAGS (CHANGEDSINCE 42)
S: * 997 FETCH (UID 9980 FLAGS (\seen) MODSEQ 42)
S: f OK That was much more fun!
```

Formal Syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) notation as specified in [ABNF]. [IMAPABNF] defines the non-terminals search-param.

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.

```
search-param =/ "NOSTORE"
```

Security considerations

There are no known security issues with this extension.

IANA considerations

The IANA is requested to add NOSTORE to the list of IMAP extensions.

Credits

(Your name here :)

Normative References

[ABNF] Crocker, Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium, Demon Internet Ltd, November 1997.

[KEYWORDS] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Harvard University, March 1997.

[IMAP] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 3501, University of Washington, June 2003.

[IMAPABNF] Melnikov, A. and Daboo, C., "Collected Extensions to IMAP4 ABNF", RFC 4466, Isode Ltd., April 2006.

[CONDSTORE] Melnikov, A. and Hole, S.., "IMAP Extension for Conditional STORE Operation or Quick Flag Changes Resynchronization", RFC 4551, Isode Ltd., June 2006.

Author's Address

Arnt Gulbrandsen
Oryx Mail Systems GmbH
Schweppermannstr. 8
D-81671 Muenchen
Germany

Fax: +49 89 4502 9758

Email: arnt@oryx.com

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.

Full Copyright Statement

Acknowledgment

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