IMAP Extensions Working Group
Internet Draft: IMAP VIEW Extension

Document: draft-daboo-imapext-view-02.txt

C. Daboo
M. Pustilnik
M. Crispin

March 2000

IMAP VIEW Extension

Status of this Memo

Copyright Notice

Table of Contents

1 Abstract

The VIEW extension to the Internet Message Access Protocol [IMAP4] permits a subset of messages in the mailbox to be processed separately from the entire set of messages in the mailbox. This allows a client to restrict its view to only the messages appearing in this set. The subset of messages also need not be returned in order of their sequence numbers, allowing clients to access messages in a particular sort order.

The subsetting and sorting of messages is handled by existing [IMAP4] commands that return a set of messages as their result. This extension takes the set returned from such a command and applies the VIEW methodology to it.

2 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 using ABNF [ABNF].

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

3 Introduction and Overview

The VIEW extension is present in any IMAP4 implementation which returns "VIEW" as one of the supported capabilities in the CAPABILITY command.

The VIEW extension specifies one new command and introduces one new untagged response, modifies two existing [IMAP4] commands and two existing [IMAP4] untagged responses, as well as putting requirements on standard IMAP4 protocol responses when the command is in effect.

This document also specifies how the VIEW extension works when the IMAP4 IDLE [IDLE] extension is present on a server. However, the VIEW extension does not require the IDLE extension to be present on the server.

Each of the features in VIEW are optimizations; clients can provide the same functionality, albeit more slowly, by using existing commands.

The design goal of the VIEW extension is to allow clients to reference messages only by their "view position numbers" as opposed to sequence numbers or UIDs, if so desired, and thus remove the need for the client to maintain its own mapping between sequence number, UID and view position.

This extension makes the following changes to the IMAP4 protocol:

(a) Adds a new VIEW command which takes one of two forms:

(i) The first form, VIEW SET, takes an optional sub-command and is used to initiate or cancel the view mechanism on the server.

(ii) In the second form, the VIEW command is used with a FETCH, COPY, STORE, SEARCH, NOOP [IMAP4] or IDLE [IDLE] command, and instructs the server to interpret the sequence numbers provided by the client in the command as VIEW positions (as described below), or to return results using view positions rather than sequence numbers, or to do a complete update of the view. The VIEW COPY command also requires that messages be copied in view order into the destination mailbox. The VIEW IDLE command is only available when the server also supports the IDLE [IDLE] extension.

(b) Allows the arguments of the VIEW SET command to appear as arguments of the SELECT or EXAMINE commands [IMAP4] to allow selecting a mailbox and setting a view in a single round-trip.

When the VIEW SET command is used by the client, the following changes also take place:

(c) New FETCH message data response item:

VIEW

Indicates the position of messages in the view, which may be different from message sequence number.

(d) New untagged response:

VIEW

Indicates new message arrival, a change in position of a message within the current view or a message moving in or out of the current view. The sequence number, UID and new view position number of the message are supplied, and optionally the old position when the message is not new. Can also be used to return SEARCH results using view position numbers rather than sequence numbers or UIDs.

(e) Changes to untagged responses and response codes:

EXISTS

The standard EXISTS response is enhanced to include information about the total number of messages in the current view.

EXPUNGE

The standard EXPUNGE response is enhanced to include information about message UID and view position.

The rest of this document describes these changes more rigorously. The document will use the IMAP4 SEARCH command as an example of a sub-command to VIEW SET. However, other commands can be used with VIEW SET as well.

4 Commands

4.1 VIEW Command

The VIEW command takes one of two forms.

4.1.1 VIEW SET

Arguments: OPTIONAL sub-command and associated arguments: this draft currently only allows the SEARCH command as the sub-command, however, anticipated SORT [SORT] and THREAD [THREAD] commands are expected to be added to this.

Responses: REQUIRED untagged responses: modified EXISTS if sub-command present

Result: OK - view completed, now in view state

NO - view failure
BAD - command unknown or arguments invalid

The VIEW SET command is only available when the server is in 'selected state' [IMAP4]. Thus a successful SELECT or EXAMINE command MUST have been issued before VIEW SET can be used. If a client attempts to use VIEW SET while the server is not in 'selected state', then the server MUST respond with a BAD response. Note that a view can also be initiated during a SELECT or EXAMINE command, as described in Section 4.2.

When the VIEW SET command is in effect (either by an explicit VIEW SET command or by a modified SELECT or EXAMINE command), it results in the following changes to the IMAP4 protocol:

(a) The server creates a 'view' that is a subset of messages in the mailbox that match the results of the sub-command. The position of a message in the view, its view position number, is then given by the VIEW fetch item response. Messages outside of the view have a view position of 0 (zero).

(b) The VIEW command can be used by the client to request information about messages in the current view set, without requiring the client to determine the view positions of every message in the mailbox (see Section 4.1.2).

(c) The server MUST return a VIEW fetch item in every FETCH response, including unsolicted FETCH responses. This ensures the client does not need to maintain its own mapping from sequence numbers to view positions.

(d) A new VIEW untagged response MUST be sent by the server when changes are made to the view by the arrival of new messages, the removal of messages, and the change in status of messages. The rules governing when these responses are issued are described in more detail in Section 5.1.

(e) The server MUST modify the EXISTS untagged response (see Section 5.3) to include additional information to indicate the current number of messages in the view set.

(f) The server MUST modify the EXPUNGE untagged response (see Section 5.4) to include additional information to indicate the expunged message's UID and view position number.

(g) The VIEW SET command with no sub-command can be used to turn off the view facility and return the server to standard IMAP4 behaviour.

(h) A VIEW SET command with a sub-command (see Section 4.1.2) can be used while another VIEW SET is in effect. This results in the view being 'reset' to use the new VIEW SET parameters.

(i) Any SELECT, EXAMINE or CLOSE command issued while a VIEW SET is in effect has the behaviour of cancelling the current view behaviour and returns the server to standard [IMAP4] behaviour, unless the SELECT or EXAMINE commands include a new VIEW SET arguments as described in Section 4.2.

Examples:

```
    C: A999 VIEW SET SEARCH FROM "Smith"
    S: * 23 EXISTS 5
    S: A999 OK VIEW SET completed
```

4.1.2 VIEW <sub-command>

```
Arguments:  command name
            command arguments

Responses:  untagged responses: FETCH, VIEW SEARCH, VIEW
                                responses appropriate for
                                NOOP and IDLE commands

Result:     OK - VIEW command completed
            NO - VIEW command error
            BAD - command unknown or arguments invalid
```

This version of the VIEW command is only available when the VIEW SET command is in effect. Clients MUST NOT issue this version of the VIEW command when a VIEW SET is not in effect, and servers MUST respond with a tagged BAD response if it receives a VIEW command of this type when VIEW SET is not in effect. This version of the VIEW command has three forms.

In the first form, it takes as its arguments a COPY, FETCH, or STORE command [IMAP4] with arguments appropriate for the associated command. However, the numbers in the message set argument are view position numbers instead of message sequence numbers.

When the VIEW COPY form is used, the server MUST copy messages into the destination mailbox in the order they appear in the current view. This may be different from the ordering of the view position numbers provided as the argument to the VIEW COPY command.

In the second form, the VIEW command takes a SEARCH command with SEARCH command arguments [IMAP4]. The interpretation of the arguments is the same as with SEARCH [IMAP4]; however, the numbers returned in a VIEW SEARCH response (see Section 5.2.3) for a VIEW SEARCH command are view position numbers instead of message sequence numbers. In addition, the search is carried out only on those messages in the current view set, so a view position of 0 (zero) MUST NOT be returned in the search results.

In the third form, the VIEW command takes a NOOP [IMAP4] or an IDLE [IDLE] command as its argument. The use of these commands, as opposed to their unmodified alternatives of NOOP and IDLE, allows the server to inform the client of changes to the view that it is normally not allowed to do. This is described in more detail in Section 5.1.

When a FETCH response [IMAP4] is returned by the server when the VIEW SET command is in effect, the number after the "*" in an untagged FETCH response is always a message sequence number, not a view position number, even for a response to any of the above VIEW commands. However, server implementations MUST implicitly include the VIEW fetch item as part of any FETCH response caused by a VIEW command, regardless of whether a VIEW was specified as a message data item to any FETCH, as described in Section 4.1.1 (c).

Examples:

```
    C: A999 VIEW FETCH 1:3 FLAGS
    S: * 23 FETCH (FLAGS (\Seen) VIEW 2)
    S: * 24 FETCH (FLAGS (\Seen) VIEW 3)
    S: * 25 FETCH (FLAGS (\Seen) VIEW 1)
```

```
        S: A999 OK VIEW FETCH completed

        C: A999 VIEW SEARCH UNSEEN
        S: * VIEW SEARCH 1 2 4 6
        S: A999 OK VIEW SEARCH completed
```

4.2 Modified SELECT and EXAMINE commands

This extension modifies the syntax of the existing IMAP4 SELECT and EXAMINE commands as follows.

```
Arguments:  mailbox name
            OPTIONAL view set arguments

Response:   same as [IMAP4] except for modified EXISTS

Result:     same as [IMAP4]
```

The IMAP4 SELECT and EXAMINE commands are modified to include optional VIEW SET command arguments after the mailbox name. This allows a client to select a mailbox and set a view in a single command. The syntax for the optional arguments are the same as for the VIEW SET command arguments.

When the optional VIEW SET command arguments are present, the EXISTS response that is normally sent by the server as part of the SELECT or EXAMINE command response is modified to include the current number of messages in the view, as described in Section 4.1.1 (e).

Examples:

```
    C: A142 SELECT INBOX SEARCH FROM "Smith"
    S: * 172 EXISTS 5
    S: * 1 RECENT
    S: * OK [UNSEEN 12] Message 12 is first unseen
    S: * OK [UIDVALIDITY 3857529045] UIDs valid
    S: * OK [UIDNEXT 4392] Predicted next UID
    S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
    S: * OK [PERMANENTFLAGS (\Deleted \Seen \*)] Limited
    S: A142 OK [READ-WRITE] SELECT with VIEW SET completed
```

5 Responses

5.1 Response Conditions

There are a number of conditions that can give rise to changes in the state of a mailbox when a VIEW SET command is in effect:

(a) New messages arriving in the mailbox that also match the current view criteria

(b) New messages arriving in the mailbox that do not match the current view criteria

(c) Messages removed from the mailbox (via EXPUNGE) that also match the current view criteria

(d) Messages removed from the mailbox (via EXPUNGE) that do not match the current view criteria

(e) Messages that do not match the current view criteria, but whose state changes (e.g. a change in the message FLAGS) such that they now match the current view criteria

(f) Messages that match the current view criteria, but whose state changes (e.g. FLAGS) such that they no longer match the current view criteria

(g) Messages that match the current view criteria, but whose state changes (e.g. FLAGS) resulting in a change to their view position, whilst still matching the current view criteria

When VIEW SET is in effect, a server MUST always send untagged responses for conditions (a), (b), (c), (d) and (e). Changes caused by conditions (f) and (g) MUST NOT be sent unless a VIEW NOOP or VIEW IDLE command is in effect. This behaviour ensures that the 'view' presented to the user via their client remains 'static', except for new message arrival and message removal, until a 'refresh' of the view is requested via a VIEW NOOP or VIEW IDLE command, or the view is cancelled or reset. Clients that want to present the user with a 'dynamic' view, one that changes when any of the conditions (a) through (g) occur, should either poll the server using VIEW NOOP (as opposed to just NOOP) or use VIEW IDLE (as opposed to just IDLE [IDLE]).

Summary of response conditions and resulting untagged responses when the VIEW SET command is in effect:

Command in progress	untagged response returned for
any, except NOOP or IDLE	(a), (b), (c), (d), (e)

VIEW NOOP or VIEW IDLE	(a), (b), (c), (d), (e), (f), (g)

Response Condition	untagged response used
(a), (b)	VIEW (Section 5.2.1)
(c), (d)	EXPUNGE (Section 5.4)
(e), (f), (g)	VIEW (Section 5.2.2)

5.2 VIEW Untagged Response

The VIEW untagged response is used to inform the client of changes to the current view. This includes new messages appearing in the view, messages being removed from the view, and messages changing position inside the view.

5.2.1 New message arrival

```
<seq> VIEW <uid> <newpos>
```

indicates that a new message has arrived with the given sequence number, UID and view position number

When using the VIEW SET command, the VIEW untagged response augments the EXISTS untagged response as a means of new mail notification. Specifically it is used to tell the client where new messages appear in the current view. The server MUST send an untagged VIEW response for every new message that arrives in the mailbox, after the corresponding untagged EXISTS response is sent to indicate this, when the VIEW SET command is in effect.

When the VIEW SET command is in effect, and the view position number of a new message is non-zero (i.e. the new message appears in the current view), the untagged VIEW response causes the view position numbers of all subsequent messages in the mailbox's current view to be incremented. This corresponds to response condition (a) described in Section 5.1.

Example:

```
C: NOOP
S: * 173 EXISTS 24
S: * 1 RECENT
S: * 173 VIEW 5274234 14
S: NOOP complete
```

indicates that a new message with sequence number 173 and UID 5274234 has been inserted into the view with view position number 14. Any message that used to have a view position number 14 is now at position 15, etc.

If a new message arrives outside the current view (i.e. not matching the specified search criteria), the view position number of the new message is given as 0. This corresponds to response condition (b) described in Section 5.1.

Example:

```
C: NOOP
S: * 174 EXISTS 24
S: * 2 RECENT
S: * 174 VIEW 5274235 0
S: NOOP complete
```

indicates that a message has arrived with sequence number 174 and UID 5274235, however, the VIEW SET command in effect prevents it from being in the current view.

5.2.2 Messages moving in the view

```
<seq> VIEW <uid> <new> <old>
```

indicates that a message with the given sequence number and UID has changed its position within the view

Some sub-commands of the VIEW SET command may change the sort order of messages in the view. In such a case, changes to message flags or keywords can cause the position of messages in sorted order to change. Also, when search criteria are specified, changes to message flags or keywords can cause messages to move into or outside of the view. The server MUST communicate these changes to the client by sending it the VIEW untagged response, according to the rules of Section 5.1.

When VIEW SET search criteria are in effect, changes made to a message can cause it to drop out of view or, on the contrary, become visible. When a message drops out of view, the new view position number is sent as 0. Similarly, when a message appears in view, for example, as a result of another agent changing the message attributes, the old view position number is sent as 0.

When the old view position number in the VIEW untagged response is

0, the effect of VIEW is a message inserted into the view at the position given by <new>. The view position numbers of all subsequent messages in the current view are implicitly incremented. This corresponds to response condition (e) described in Section 5.1.

When the new view position number in the VIEW untagged response is 0, the effect of VIEW is a message removed from the view at the position given by <old>. The view position numbers of all subsequent messages in the current view are implicitly decremented. This corresponds to response condition (f) described in Section 5.1. This untagged response MUST only be sent when a VIEW NOOP or VIEW IDLE command is in effect.

When both the <old> and <new> view positions are non-zero, then the view positions of messages between the <old> and <new> positions need to be implicitly decremented or incremented depending on whether the <old> view position is less than or greater than the <new> view position, respectively. This corresponds to response condition (g) described in Section 5.1. This untagged response MUST only be sent when a VIEW NOOP or VIEW IDLE command is in effect.

It is meaningless to have a VIEW untagged response where both <old> and <new> view position numbers are 0; the server MUST NOT generate VIEW responses of this form.

VIEW untagged responses indicating messages moving in the view MUST NOT be sent as the result of messages being expunged (which is handled by the modified untagged EXPUNGE response), or when no command is in progress, or while responding to a FETCH, SEARCH or STORE command. These rules are necessary to prevent loss of synchronization of message sequence numbers between client and server.

Examples:

```
    S: * 172 VIEW 5274234 2 14
```

indicates that the view position of the message with sequence number 172 and UID equal to 5274234 has changed from 14 to 2. The view position numbers of the messages in the range 2:13 are incremented.

```
    S: * 172 VIEW 5274234 0 14
```

indicates that the message with view position number 14 has dropped out of view. The view position numbers of subsequent messages are decremented. The message continues to exist, however, it is outside of the current view and is therefore no longer accessible to the session via the VIEW command.

```
        S: * 173 VIEW 5274234 2 0
```

indicates that the message with UID 5274234 has appeared in the current view and has been assigned view position number 2. The view position numbers of subsequent messages are incremented.

5.2.3 VIEW SEARCH Response

```
        VIEW SEARCH <zero or more view position numbers>
```

The VIEW SEARCH response occurs as a result of a VIEW SEARCH command. The view position number(s) refer to those messages that match the search criteria. Each number is delimited by a space.

```
    Example:  S: * VIEW SEARCH 2 3 6
```

5.3 EXISTS Untagged Response

```
        <# in mailbox> EXISTS <# in view>
```

indicates the total number of messages in the mailbox and in the current view

The modified EXISTS untagged response MUST only be used when the VIEW SET command is in effect. This MUST be sent in response to a VIEW SET command taking a sub-command argument. Otherwise this response is sent under the same circumstances as the unmodified EXISTS response (i.e. in when there is a change to the total number of messages in the mailbox).

Example:

```
        C: A142 SELECT INBOX
        S: * 172 EXISTS
        S: * 3 RECENT
        S: * OK [UNSEEN 12] Message 12 is first unseen
        S: * OK [UIDVALIDITY 3857429045] UIDs valid
        S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
        S: OK [PERMANENTFLAGS (\Deleted \Seen \*)] Limited
        S: A142 OK [READ-WRITE] SELECT completed
        C: A143 VIEW SET SEARCH FROM "Smith"
        S: * 172 EXISTS 23
        S: A143 OK VIEW SET completed
```

In this example, there are 172 messages in the mailbox and 23 messages have a from address including the text "Smith".

5.4 EXPUNGE Untagged Response

```
        <seq> EXPUNGE <uid> <pos>
```

indicates that a message with given sequence number, UID and view position number has been permanently removed from the mailbox

When VIEW SET is in effect, the EXPUNGE untagged response [IMAP4] is modified to include the UID and view position number of the message being expunged. The server MUST use this modified form of the EXPUNGE response whenever a VIEW SET command is in effect.

When the view position number of the expunged message is non-zero, the modified EXPUNGE untagged response causes the view position numbers of all subsequent messages in the mailbox's current view to be decremented. This corresponds to response condition (c) described in Section 5.1.

When search criteria are specified in the VIEW SET command, and a message outside the current view is expunged, the view position number of the expunged message is given as 0. This corresponds to response condition (d) described in Section 5.1.

Examples:

```
        S: * 172 EXPUNGE 5274234 14
```

indicates that the message with sequence number 172, UID 5274234 and view position number 14 has been expunged from its mailbox. Any message previously at position 15 is now at position 14, etc

```
        S: * 173 EXPUNGE 5274234 0
```

indicates that a message with sequence number 173, UID 5274234 was expunged, however, the message was outside the current view.

A modified EXPUNGE untagged response MUST NOT be sent when no command is in progress; nor while responding to a FETCH, SEARCH or STORE command. This rule is necessary to prevent loss of synchronization of message sequence numbers between client and server.

6 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 [IMAP4].

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.

```
examine             = "EXAMINE" SP mailbox [SP view-set-arg]
                        ; redefines [IMAP4] EXAMINE command

exists_resp         = number "EXISTS" SP number
                        ; replaces standard [IMAP4] EXISTS response
                        ; when VIEW SET is in effect

expunge_resp        = nz-number "EXPUNGE" SP uniqueid SP position
                        ; replaces standard IMAP4 EXPUNGE response
                        ; when VIEW SET is in effect

msg-att-dynamic     =/ "VIEW" SP position
                        ; adds new message data item to IMAP4
                        ; FETCH response

position            = number ["." number]
                        ; position number of messages in the view
                        ; may be hierarchical if sub-command defines
                        ; hierarchic ordering of messages
                        ; 0 means outside of the current view

select              = "SELECT" SP mailbox [SP view-set-arg]
                        ; redefines [IMAP4] SELECT command

view                = "VIEW" SP (view-set / view-reference)

view-mailbox-data   = "VIEW" SP (("SEARCH" *(SP position)) / OTHER)
                        ; OTHER reserved for future expansion

view-msg-data       = nz_number SP "VIEW" SP uniqueid
                      SP position [SP position]

view-reference      = copy / fetch / store / search / "NOOP" / idle
                        ; copy, fetch, store, search defined
                        ; in [IMAP4], idle defined in [IDLE]
```

```
    view-response      = view-msg-data / view-mailbox-data

    view-set           = "SET" [SP view-set-arg]

    view-set-arg       = search / sort / thread / OTHER
                             ; search is defined in [IMAP4]
                             ; sort is defined in [SORT]
                             ; thread is defined in [THREAD]
                             ; OTHER reserved for future definition
```

7 Security Considerations

There are no known security issues with this extension.

8 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.

[IDLE] Leiba, B., "IMAP4 IDLE command", RFC 2177, IBM T.J. Watson Research Center, June 1997.

[IMAP4] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 2060, University of Washington, December 1996.

[MIME-IMB] Freed, N., and Borenstein, N.., "MIME (Multipurpose Internet Mail Extensions) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC-822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC822, August 1982.

[SORT] Crispin, M., "Internet Message Access Protocol - SORT Extension", draft-crispin-imapext-sort-01.txt (Work in progress)

[THREAD] Crispin, M., "Internet Message Access Protocol - THREAD Extension", draft-crispin-imapext-thread-01.txt (Work in progress)

9 Acknowledgments

Steve Hole, Chris Newman and Larry Osterman have made significant contributions to the creation and refinement of the ideas expressed in this document.

10 Authors' Addresses

Cyrus Daboo
Cyrusoft International, Inc.
Suite 780, 5001 Baum Blvd.
Pittsburgh, PA 15213

Phone: (412) 605-0499
Email: daboo@cyrusoft.com

Mark Pustilnik
Microsoft
1 Microsoft Way

Redmond, WA 98052

Phone: (425) 703-5758
Email: markpu@microsoft.com

Mark R. Crispin
Networks and Distributed Computing
University of Washington
4545 15th Aveneue NE
Seattle, WA 98105-4527

Phone: (206) 543-5762
EMail: MRC@CAC.Washington.EDU