

The IMAP VIEW extension

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 in August, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

The VIEW extension allows an [[IMAP](#)] client to create and manipulate mailbox views (persistent searches).

All [[IMAP](#)] clients can access a view, as if it were an ordinary mailbox. Clients which implement the VIEW extension can create views. Clients which implement both VIEW and [[LISTEXT](#)] can find and modify existing views.

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](#)].

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

Introduction and Overview

An IMAP server that supports this extension announces "VIEW" as one of its capabilities.

The goal of VIEW is to provide efficient and useful access to large mailboxes (in particular archives) and require small changes to clients. The extension should also be practically implementable for current-day servers (even though it may cost some CPU).

The extension adds one new command (VIEW CREATE) and one new mailbox flag (\View). It adds no new responses. If a server also supports the [[LISTEXT](#)] extension, VIEW adds one new selection option and one new return option.

Mailbox Views

A view, as defined by this document, is a mailbox which does not contain any messages of its own, but instead provides access to a selected subset of another mailbox's messages.

Only a VIEW-supporting IMAP client can create a view, however any IMAP client can use the view.

The view specification uses regular IMAP search syntax, except that session-dependent search keys MUST NOT be used. At the time of writing, the session-dependent keys are MSN sequence sets, NEW, OLD and RECENT. (IMAP extensions may add more in the future.)

When a view specification uses dynamic attributes of a message (primarily flags), a message may be included in a view, later excluded and later still included again. When this happens, the server MUST issue a new UID every time. This implies that the order of messages in a view need not match the order in the underlying mailbox.

Views and mailboxes exist in the same name space: creating a view fails if there already is a mailbox of that name, and creating a mailbox fails if there already is a view of that name.

Permanent flags are an attribute of the message, not of the mailbox or session; hence, it does not matter whether the message is accessed via a mailbox or view. The effect on permanent flags of a STORE or FETCH command is exactly the same in both cases.

The VIEW extension makes a message appear in multiple "mailboxes" at a time; one actual mailbox and zero or more views. Messages can also disappear and reappear in views. This complicates the semantics of the \Recent pseudoflag considerably. To simplify implementation, the server MAY omit computing any \Recent pseudoflag for view mailboxes. In that case, a message is only \Recent when viewed in the underlying mailbox. If it does compute \Recent, it should present the view exactly as an ordinary mailbox.

The VIEW CREATE Command

Arguments: Name of underlying mailbox
 Name of view to be created
 Search expression

Responses: None

Result: OK The server has created the view.
 NO The client lacks permission to create the view or
 access the mailbox, the view exists, or the mailbox
 does not exist.
 BAD Command unknown or invalid argument.

The server creates the target view, unless a view or mailbox already exists with that name, and stores the search expression. When a client later accesses the view with SELECT or EXAMINE, the server performs a search on the mailbox and shows the client only the resulting messages.

The VIEW CREATE command is valid in Authenticated and Selected states.

Base IMAP Commands

Most commands (SELECT, EXAMINE, SUBSCRIBE, etc) can be used to select, examine, subscribe, etc. views as well as mailboxes.

The CREATE Command ([RFC 3501 section 6.3.3](#)) fails if a view already exists with the desired name.

The DELETE Command ([RFC 3501 section 6.3.4](#)) offers a special problem if a mailbox is deleted while there are views onto that mailbox. In order to simplify implementation, no particular behaviour is mandated in that case. However, servers MUST NOT show messages in deleted mailboxes to clients.

The RENAME Command ([RFC 3501 section 6.3.5](#)) has a similar problem: If a mailbox is renamed, what happens to views onto that mailbox? Again, no particular behaviour is mandated.

The LIST Command ([RFC 3501 section 6.3.8](#)) MUST tag views with the new \View mailbox flag. (LIST is also described below.)

The APPEND Command ([RFC 3501 section 6.3.11](#)) and the COPY Command ([RFC 3501 section 6.4.7](#)) cannot be used to append/copy messages to views.

The EXPUNGE Command ([RFC 3501 section 6.4.3](#)) causes the underlying mailbox to be expunged when a view is expunged. Servers MAY expunge only the messages visible in the view, or MAY expunge the entire mailbox. The former is more desirable, if possible.

ACL Commands

SETACL can be used to set access control lists on views, just like on mailboxes. The i right (COPY/APPEND) cannot be granted on a view.

LISTRIGHTS acts as for mailboxes.

MYRIGHTS computes access as for mailboxes. However, it may or may not consider the underlying mailbox ACL, depending on how a server implements VIEW.

If it considers the underlying mailbox ACL, the ACL on a mailbox controls all access to the messages stored there. From a security perspective, this may be considered an advantage.

If it works independently of the underlying mailbox ACL, views can be used to selectively grant access to a few messages in a mailbox. This can also be viewed as a security advantage, since it allows more finegrained access control.

LIST Extensions

If the server also supports [[LISTTEXT](#)], a client can find existing views, and can read the search expression for an existing view.

The selection option VIEWS instructs the server to return LIST responses only for views.

The return option VIEWS instructs the server to include the view's search and underlying mailbox in a LIST response.

Examples

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

This example creates a view onto INBOX that looks at all IETF-Announce messages that contain either "IMAP" or "SMTP" in the body text:

```
C: a VIEW CREATE INBOX "IMAP-announcements" HEADER "Sender"
    "<ietf-announce-request@ietf.org>" OR BODY imap BODY smtp
S: a OK Created view user.fred.IMAP-announcements
```

This example creates a view onto /archive/support that selects all mail to or from example.com:

```
C: b VIEW CREATE "/archive/support" "/customers/example" OR OR
    FROM "@example.com>" TO "@example.com>" CC "@example.com>"
S: b OK Created view /customers/example
```

This example shows an unextended LIST and two extended LIST commands:

```
C: c LIST "" "%"
S: * LIST (\hasnochildren) "." INBOX
S: * LIST (\hasnochildren \view) "." IMAP-announcements
S: c OK
C: d LIST (views) "" "%"
S: * LIST (\hasnochildren \view) "." IMAP-announcements
C: e LIST (views) "" "%" return (views)
S: * LIST (\hasnochildren \view) "." IMAP-announcements ((view
    (INBOX "HEADER \"Sender\" \"<ietf-announce-
    request@ietf.org>\" OR BODY imap BODY smtp")))
S: e OK
```

Note that commands d and e only work if the server supports both VIEW and [[LISTTEXT](#)].

Implementation Notes

In many servers, views can be implemented as a new form of mailbox format.

Open Issues

The way [[LISTTEXT](#)] returns the search key is not. How should it be done? If it's sent as a quoted string, it's hard to express in the formal grammar. Sending it unquoted, writing the grammar so it fits both search-leu and the mbox-list-extended-item-data in [[LISTTEXT](#)] seems a very tricky proposition.

The return option could do with a bit improvement, all over.

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 [[ABNF](#)] (SP, CRLF) or [[IMAP](#)] (all others).

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.

```
command-auth =/ view
               ; command-auth is in [IMAP]

option-extension =/ "VIEW"
                  ; option-extension is in [LISTTEXT]

view            = "VIEW" SP "CREATE" mailbox SP mailbox 1*(SP
                  search-key)

view-extended-item = "VIEW" SP "(" mailbox SP nstring ")"
                    ; see the open issues section
```

Security considerations

While views do not need more server capacity than the equivalent functionality would if the search were implemented on the client or using SEARCH, the presence of VIEW may tempt users to use larger mailboxes and search more often on them, which may lead to resource

problems.

The interaction between VIEW and [[ACL](#)] is nontrivial. An ACL change on a mailbox may either change the ACLs on many views, or it may be partly ineffective, because some messages are also visible via views, bypassing the new ACL.

IANA considerations

The IANA is requested to add VIEW 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.
- [ACL] Melnikov, A., "IMAP4 Access Control List (ACL) Extension", [RFC 4314](#), Isode Ltd., December 2005.
- [LISTEXT] Leiba, B. and A. Melnikov, "IMAP4 LIST Command Extensions", work in progress, [draft-ietf-imapext-list-extensions-xx.txt](#).

Author's Address

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

Fax: +49 89 4502 9758

Email: arnt@oryx.com

Abhijit Menon-Sen
Oryx Mail Systems GmbH
Schweppermannstr. 8
D-81671 Muenchen
Germany

Email: ams@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

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.

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.

Acknowledgment

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

