

lemonade  
Internet-Draft  
Expires: May 6, 2009

R. Gellens  
QUALCOMM Incorporated  
C. Newman  
Sun Microsystems  
November 2, 2008

**Internet Message Store Events**  
**draft-ietf-lemonade-msgevent-07.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 May 6, 2009.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

One of the missing features in the existing Internet mail and messaging standards is a facility for server-to-server and server-to-client event notifications related to message store events. As the scope of Internet mail expands to support more diverse media (such as voice mail), devices (such as cell phones) and to provide rich interactions with other services (such as web portals and legal compliance systems), the need for an interoperable notification

system increases. This document attempts to enumerate the types of events which interest real-world consumers of such a system.

This document describes events and event parameters which are useful for several cases, including notification to administrative systems and end users. This is not intended as a replacement for a message access facility such as IMAP.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">3</a>
<a href="#">1.1.</a>	<a href="#">Conventions Used in this Document . . . . .</a>	<a href="#">4</a>
<a href="#">1.2.</a>	<a href="#">Change History . . . . .</a>	<a href="#">4</a>
<a href="#">1.2.1.</a>	<a href="#">Changes from -06 to -07 . . . . .</a>	<a href="#">4</a>
<a href="#">1.2.2.</a>	<a href="#">Changes from -05 to -06 . . . . .</a>	<a href="#">5</a>
<a href="#">1.2.3.</a>	<a href="#">Changes from -04 to -05 . . . . .</a>	<a href="#">5</a>
<a href="#">1.2.4.</a>	<a href="#">Changes from -03 to -04 . . . . .</a>	<a href="#">6</a>
<a href="#">1.2.5.</a>	<a href="#">Changes from -02 to -03 . . . . .</a>	<a href="#">6</a>
<a href="#">1.2.6.</a>	<a href="#">Changes from -01 to -02 . . . . .</a>	<a href="#">6</a>
<a href="#">1.2.7.</a>	<a href="#">Changes from -00 to -01 . . . . .</a>	<a href="#">7</a>
<a href="#">1.2.8.</a>	<a href="#">Changes from <a href="#">draft-newman-lemonade-msgevent-00.txt</a> to <a href="#">draft-ietf-lemonade-msgevent-00.txt</a> . . . . .</a>	<a href="#">7</a>
<a href="#">2.</a>	<a href="#">Terminology . . . . .</a>	<a href="#">7</a>
<a href="#">3.</a>	<a href="#">Event Model . . . . .</a>	<a href="#">8</a>
<a href="#">4.</a>	<a href="#">Event Types . . . . .</a>	<a href="#">9</a>
<a href="#">4.1.</a>	<a href="#">Message Addition and Deletion . . . . .</a>	<a href="#">9</a>
<a href="#">4.2.</a>	<a href="#">Message Flags . . . . .</a>	<a href="#">11</a>
<a href="#">4.3.</a>	<a href="#">Access Accounting . . . . .</a>	<a href="#">11</a>
<a href="#">4.4.</a>	<a href="#">Mailbox Management . . . . .</a>	<a href="#">12</a>
<a href="#">5.</a>	<a href="#">Event Parameters . . . . .</a>	<a href="#">13</a>
<a href="#">6.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">18</a>
<a href="#">7.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">18</a>
<a href="#">8.</a>	<a href="#">Acknowledgments . . . . .</a>	<a href="#">19</a>
<a href="#">9.</a>	<a href="#">References . . . . .</a>	<a href="#">19</a>
<a href="#">9.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">19</a>
<a href="#">9.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">19</a>
<a href="#">Appendix A.</a>	<a href="#">Future Extensions . . . . .</a>	<a href="#">20</a>
	<a href="#">Authors' Addresses . . . . .</a>	<a href="#">21</a>
	<a href="#">Intellectual Property and Copyright Statements . . . . .</a>	<a href="#">22</a>



## 1. Introduction

A message store is used to organize Internet Messages [[RFC2822](#)] into one or more mailboxes, possibly hierarchical, annotate them in various ways and provide access to these messages and associated meta-data. Three different standards-based protocols have been widely deployed to remotely access a message store. Post Office Protocol (POP) [[RFC1939](#)] provides simple download-and-delete access to a single mail drop (which is a subset of the functionality typically associated with a message store). Internet Message Access Protocol (IMAP) [[RFC3501](#)] provides an extensible feature-rich model for online, offline and disconnected access to a message store with minimal constraints on any associated "fat-client" user interface. Finally, mail access applications built on top of Hypertext Transfer Protocol (HTTP) [[RFC2616](#)] which run in standards-based web browsers provide a third standards-based access mechanism for online-only access.

While simple and/or ad-hoc mechanisms for notifications have sufficed to some degree in the past (e.g., Simple New Mail Notification [[RFC4146](#)], IMAP4 IDLE command [[RFC2177](#)]), as the scope and importance of message stores expands, the demand for a more complete store notification system increases. Some of the driving forces behind this demand include:

- o Mobile devices with intermittent network connectivity that have "new mail" or "message count" indicators
- o Unified messaging systems which include both Internet and voice mail require support for a message waiting indicator on phones
- o Interaction with systems for event-based or utility-computing billing
- o Simplify the process of passing message store events to non-Internet notification systems
- o A calendar system may wish to subscribe to MessageNew notifications in order to support iMIP [[RFC2447](#)].
- o Some jurisdictions have laws or regulations for information protection and auditing which require interoperable protocols between message stores built by messaging experts and compliance auditing systems built by compliance experts.

Vendors who have deployed proprietary notification systems for their Internet message stores have seen significant demand to provide notifications for more and more events. As a first step towards



building a notification system, this document attempts to enumerate the core events that real-world customers demand.

This document includes those events which can be generated by the use of IMAP4Rev1 [[RFC3501](#)] and some existing extensions. As new IMAP extensions are defined, or additional event types or parameters need to be added, the set specified here can be extended by means of an IANA registry with update requirements, as specified in [Section 6](#).

### **1.1. Conventions Used in this Document**

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 [RFC 2119](#) [[RFC2119](#)]. When these words appear in lower-case or with initial-capital letters, they are not [RFC 2119](#) key words.

### **1.2. Change History**

This section to be removed if/when this draft is published as an RFC.

#### **1.2.1. Changes from -06 to -07**

- o Corrected occurrences of "MUST not" to be "MUST NOT".
- o Clarified that an "IP address" can be IPv4 or IPv6.
- o Updated [RFC 2434](#) to [RFC 5226](#).
- o Clarified "subscription list" in MailboxSubscribe description.
- o Changed specification of the timestamp parameter to be when the event which triggered the notification occurred, not when the notification was generated.
- o Clarified that the timestamp parameter is expressed in local time and offset from UTC, and is normally in [RFC 3339](#) format.
- o Clarified the description of the flagNames parameter so that space-separation is not normative.
- o Clarified the description of the tags parameter so that comma-separation is not normative.
- o Clarified the description of the user parameter to not rely on SASL.



- o Added new paragraph to Abstract explaining that the notifications described here are not intended as a substitute for a message access facility such as IMAP.
- o Clarified that the timestamp parameter MAY be an approximate value.

#### **1.2.2. Changes from -05 to -06**

- o Added discussion of message integrity to Security Considerations ([Section 7](#)).
- o Reviewed use of "may", "must" and "should" for conversion to "MAY", "MUST" and "SHOULD".
- o Reworded MessageNew and MessageExpire text.
- o Additional clarifying text added to Future Extensions (Appendix A).
- o Added explicit prohibition on \Recent in FlagsSet to match the one in FlagsClear (\Recent was also explicitly excluded from flagNames).
- o Clarified that Logout disconnection types are suggestions only.
- o Added serverDomain and serverPort parameters, and clarified clientPort.
- o Clarified that a kilobyte is 1024 octets.
- o Clarified that FlagSet and FlagClear include one or more flags or keywords.
- o Changed "cover about 95% of the known use cases" to "an overwhelming majority of known use cases" in Event Types ([Section 4](#)).

#### **1.2.3. Changes from -04 to -05**

- o Reworded statement on optional versus mandatory and removed Anchor 11.
- o Included mailbox admin events in list of exceptions in [Appendix A](#), and deleted Anchor 23.





#### **1.2.4. Changes from -03 to -04**

- o Added QuotaChange event.

#### **1.2.5. Changes from -02 to -03**

- o Fix typo in Login event
- o Remove UIDVALIDITY from MailboxRename event.
- o Made event names hierarchical: Changed AppendMessage to MessageAppend, ExpireMessage to MessageExpire, ExpungeMessage to MessageExpunge, NewMessage to MessageNew, OverQuota to QuotaExceed, UnderQuota to QuotaWithin, ReadMessages to MessageRead, TrashMessages to MessageTrash, SetFlags to FlagsSet, and ClearFlags to FlagsClear; deleted Editor's Note asking if this should be done.
- o Made ACL information a future extension in MailboxCreate.

#### **1.2.6. Changes from -01 to -02**

- o Add text indicating that mailboxes may contain Internet messages and/or child mailboxes.
- o Remove word "folder" from definition of "mailbox."
- o Add editor's note regarding optionality in this document.
- o Add editor's note regarding optional vs. mandatory events.
- o Add editor's note regarding event names.
- o Remove U.S.-centric wording regarding laws.
- o Review uses of "will" and change as appropriate.
- o Clarification of server address in login event.
- o Add MailboxCreate, MailboxDelete, MailboxRename, and MailboxSubscribe events.
- o Add mailboxName and oldMailboxName parameters.
- o Move [RFC2822](#) from normative to informative.
- o Add IANA Considerations and reference to [RFC 2434](#).



- o Minor grammatical improvements.
- o Incorporate edits from Alexey Melnikov.
- o Add editor's note regarding deployment of mailbox admin events.
- o Add Acknowledgments section.
- o Fix formatting to add blank line between paragraphs in event and parameter lists.
- o Add reference to [RFC 2119](#) and "Conventions" section.
- o Update [RFC 2222](#) to [RFC 4422](#).

#### **1.2.7. Changes from -00 to -01**

- o Add modseq event parameter.
- o Add tags event parameter.

#### **1.2.8. Changes from [draft-newman-lemonade-msgevent-00.txt](#) to [draft-ietf-lemonade-msgevent-00.txt](#)**

- o Rename draft title
- o Add Change History section.
- o Update reference to URLAUTH.
- o Add FlagsSet, FlagsClear events and flagNames parameter. Update future events section to reflect this change.
- o Removed unnecessary normative reference to NAMESPACE.

## **2. Terminology**

The following terminology is used in this document:

### **mailbox**

A container for Internet messages and/or child mailboxes. A mailbox may or may not permit delivery of new messages via a mail delivery agent.



#### mailbox identifier

A mailbox identifier provides sufficient information to identify a specific mailbox on a specific server instance. An IMAP URL can be a mailbox identifier.

#### message access protocols

Protocols which provide clients (e.g., a mail user agent or web browser) with access to the message store including but not limited to IMAP, POP and HTTP.

#### message context

As defined in [[RFC3458](#)].

#### UIDVALIDITY

As defined in IMAP4rev1 [[RFC3501](#)]. UIDVALIDITY is critical to the correct operation of a caching mail client. When it changes, the client MUST flush its cache. It's particularly important to include UIDVALIDITY with event notifications related to message addition or removal in order to keep the message data correctly synchronized.

### **3. Event Model**

The events that are generated by a message store depend to some degree on the model used to represent a message store. The model the IETF has for a message store is implicit from IMAP4rev1 and extensions, so that model is assumed by this document.

A message store event typically has an associated mailbox name and usually has an associated user name (or authorization identity if using the terminology from SASL [[RFC4422](#)]). Events referring to a specific message can use an IMAP URL [[RFC5092](#)] to do so. Events referring to a set of messages can use an IMAP URL to the mailbox plus an IMAP UID set.

Each notification has a type and parameters. The type determines the type of event, while the parameters supply information about the context of the event that may be used to adjust subscription preferences or may simply supply data associated with the event. The types and parameter names in this document are restricted to US-ASCII printable characters so these events can be easily mapped to an arbitrary notification system. However, this document assumes arbitrary parameter values (including large and multi-line values) can be encoded with the notification system. Systems which lack that feature could only implement a subset of these events.

This document does not indicate which event parameters are mandatory



or optional. That is done in documents which specify specific message formats or bindings to a notification system.

For scalability reasons, some degree of filtering at event generation is necessary. At the very least, the ability to turn on and off groups of related events and to suppress inclusion of large parameters (such as `messageContent`) is needed. A sophisticated publish/subscribe notification system may be able to propagate cumulative subscription information to the publisher.

Some of these events might be logically collapsed into a single event type with a required parameter to distinguish between the cases (e.g., `QuotaExceed` and `QuotaWithin`). However until such time that an event subscription model is formulated, it's not practical to make such decisions. We thus note only the fact that some of these events may be viewed as a single event type.

## **4. Event Types**

This section discusses the different types of events useful in a message store event notification system. The intention is to document the events sufficient to cover an overwhelming majority of known use cases while leaving less common event types for the future. This section mentions parameters which are important or specific to the events described here. Event parameters likely to be included in most or all notifications are discussed in the next section.

### **4.1. Message Addition and Deletion**

This section includes events related to message addition and deletion.

#### **MessageAppend**

A message was appended or concatenated to a mailbox by a message access client. For the most part, this is identical to the `MessageNew` event type, except that the SMTP envelope information is not included as a parameter, but information about which protocol triggered the event MAY be included. See the `MessageNew` event for more information.

#### **MessageExpire**

One or more messages were expired from a mailbox due to server expiration policy and are no longer accessible by the end-user.

The parameters include a mailbox identifier which **MUST** include `UIDVALIDITY`, and a UID set which describes the messages.





Information about which server expiration policy was applied may be included in the future.

#### MessageExpunge

One or more messages were expunged from a mailbox by an IMAP CLOSE/EXPUNGE, POP3 DELE+QUIT, HTTP or equivalent client action and are no longer accessible by the end-user.

The parameters include a mailbox identifier which MUST include UIDVALIDITY, a UID set, and MAY also indicate which access protocol triggered the event.

#### MessageNew

A new message was received into a mailbox via a message delivery agent.

The parameters include a message identifier which for IMAP-accessible message stores MUST include UIDVALIDITY and UID. The parameters MAY also include SMTP envelope and other arbitrary message and mailbox meta-data. In some cases, the entire new message itself may be included. The set of parameters SHOULD be adjustable to the client's preference with limits set by server policy. An interesting policy, for example, would be to include messages up to 2K in size with the notification, but for larger messages to include a URLAUTH [[RFC4467](#)] reference.

#### QuotaExceed

An operation failed (typically MessageNew) because the user's mailbox exceeded one of the quotas (e.g., disk quota, message quota, quota by message context, etc). The parameters SHOULD include at least the relevant user and quota, and optionally the mailbox. Quota usage SHOULD be included if possible. Parameters needed to extend this to support quota by context are not presently described in this document, but could be added in the future.

#### QuotaWithin

An operation occurred (typically MessageExpunges or MessageExpires) which reduced the user's quota usage under their limit.

#### QuotaChange

The user's quota was changed.



## **4.2. Message Flags**

This section includes events related to changes in message flags.

### **MessageRead**

One or more messages in the mailbox were marked as read or seen by a user. Note that POP has no concept of read or seen messages, so these events are only generated by IMAP or HTTP clients (or equivalent).

The parameters include a mailbox identifier and a set of message UIDs.

### **MessageTrash**

One or more messages were marked for future deletion by the user but are still accessible over protocol (the user's client may or may not make these messages accessible through its user interface).

The parameters include a mailbox identifier and a set of message UIDs.

### **FlagsSet**

One or more messages in the mailbox had one or more IMAP flags or keyword sets.

The parameters include a list of IMAP flag or keyword names that were set, a mailbox identifier and a set of message UIDs that were impacted. The flagNames MUST NOT include \Recent. For compatibility with simpler clients, it SHOULD be configurable whether setting the \Seen or \Deleted flags results in this event or the simpler MessageRead/MessageTrash events. By default, the simpler message forms SHOULD be used for MessageRead and MessageTrash.

### **FlagsClear**

One or more messages in the mailbox had one or more IMAP flags or keywords cleared.

The parameters include a list of IMAP flag or keyword names that were cleared, a mailbox identifier and a set of message UIDs that were impacted. The flagNames parameter MUST NOT include \Recent.

## **4.3. Access Accounting**

This section lists events related to message store access accounting.



### Login

A user has logged in to the system via IMAP, HTTP, POP or some other mechanism.

The parameters include the domain name and port used to access the server and the user's authorization identity. Additional possible parameters include the client's IP address and port, the authentication identity (if different from the authorization identity), the service name, the authentication mechanism, information about any negotiated security layers, a timestamp and other information.

### Logout

A user has logged out or otherwise been disconnected from the message store via IMAP, HTTP, POP or some other mechanism.

The parameters include the server domain name and the user's authorization identity. Additional parameters MAY include any of the information from the "Login" event as well as information about the type of disconnect (suggested values include graceful, abort, timeout, and security layer error), the duration of the connection or session and other information.

## **4.4. Mailbox Management**

This section lists events related to the management of mailboxes.

### MailboxCreate

A mailbox has been created, or an access control changed on an existing mailbox so that it is now accessible by the user. If the mailbox creation caused the creation of new mailboxes earlier in the hierarchy, separate MailboxCreate events are not generated for those as their creation is implied.

The parameters include the created mailbox identifier, its UIDVALIDITY for IMAP-accessible message stores, and MAY also indicate which access protocol triggered the event. Access/permissions information (such as ACL [[RFC4314](#)] settings) require a standardized format to be included, and so are left for future extension.

### MailboxDelete

A mailbox has been deleted, or an access control changed on an existing mailbox so that it is no longer accessible by the user. Note that if the mailbox has child mailboxes, only the specified mailbox has been deleted, not the children. The mailbox becomes \NOSELECT and the hierarchy remains unchanged, as per the description of the DELETE command in RFC 3501IMAP4rev1 [[RFC3501](#)].



The parameters include the deleted mailbox identifier, and MAY also indicate which access protocol triggered the event.

#### MailboxRename

A mailbox has been renamed. Note that, per the description of the RENAME command in RFC 3501IMAP4rev1 [[RFC3501](#)], special semantics regarding the mailbox hierarchy apply when INBOX is renamed (child mailboxes are usually included in the rename, but are excluded when INBOX is renamed). When a mailbox other than INBOX is renamed and its child mailboxes are also renamed as a result, separate MailboxRename events are not generated for the child mailboxes, as their renaming is implied. If the rename caused the creation of new mailboxes earlier in the hierarchy, separate MailboxCreate events are not generated for those as their creation is implied. When INBOX is renamed, a new INBOX is created. A MailboxCreate event is not generated for the new INBOX, since it is implied.

The parameters include the old mailbox identifier, the new mailbox identifier, and MAY also indicate which access protocol triggered the event.

#### MailboxSubscribe

A mailbox has been added to the server-stored subscription list, such as the one managed by the IMAP SUBSCRIBE and UNSUBSCRIBE commands.

The parameters include the user whose subscription list has been affected and the mailbox identifier, and MAY also indicate which access protocol triggered the event.

#### MailboxUnSubscribe

A mailbox has been removed from the subscription list.

The parameters include the user whose subscription list has been affected and the mailbox identifier, and MAY also indicate which access protocol triggered the event.

## 5. Event Parameters

This section lists parameters included with these events.

#### admin

Included with all events generated by message access protocols.

The authentication identity associated with this event, as





distinct from the authorization identity (see "user"). This is not included when it is the same as the value of the user parameter.

#### bodyStructure

May be included with MessageAppend and MessageNew.

The IMAP BODYSTRUCTURE of the message.

#### clientIP

Included with all events generated by message access protocols.

The IPv4 or IPv6 address of the message store access client which performed the action which triggered the notification.

#### clientPort

Included with all events generated by message access protocols.

The port number of the message store access client which performed an action which triggered the notification (the port from which the connection occurred).

#### diskQuota

Included with QuotaExceed, QuotaWithin, and QuotaChange notifications relating to a user or mailbox disk quota. May be included with other notifications.

Disk quota limit in kilobytes (1024 octets).

#### diskUsed

Included with QuotaExceed and QuotaWithin notifications relating to a user or mailbox disk quota. May be included with other notifications.

Disk space used in kilobytes (1024 octets). Only disk space which counts against the quota is included.

#### envelope

May be included with the MessageNew notification.

The message transfer envelope associated with final delivery of the message for the MessageNew notification. This includes the



MAIL FROM and relevant RCPT TO line(s) used for final delivery with CRLF delimiters and any ESMTP parameters.

#### flagNames

Included with FlagsSet and FlagsClear events. May be included with MessageAppend and MessageNew to indicate flags which were set initially by the APPEND command or delivery agent respectively.

A list (likely to be space-separated) of IMAP flag or keyword names that were set or cleared. Flag names begin with backslash while keyword names do not. The \Recent flag is explicitly not permitted in the list.

#### mailboxID

Included in events which affect mailboxes. URI describing the mailbox. In the case of MailboxRename, this refers to the new name.

#### maxMessages

Included with QuotaExceed and QuotaWithin notifications relating to a user or mailbox message count quota. May be included with other notifications.

Quota limit on the number of messages in the mailbox, for events referring to a mailbox.

#### messageContent

May be included with MessageAppend and MessageNew.

The entire message itself. Size based suppression of this SHOULD be available.

#### messageSize

May be included with MessageAppend and MessageNew.

Size of the [RFC 2822](#) message itself in octets. This value matches the length of the IMAP literal returned in response to an IMAP FETCH of BODY[] for the referenced message.

#### messages

Included with QuotaExceed and QuotaWithin notifications relating to a user or mailbox message count quota. May be included with



other notifications.

Number of messages in the mailbox. This is typically included with message addition and deletion events.

#### modseq

May be included with any notification referring to one message.

This is the 64-bit integer MODSEQ as defined in [[RFC4551](#)]. No assumptions about MODSEQ can be made if this is omitted.

#### oldMailboxID

URI describing the old name of a renamed or moved mailbox.

#### pid

May be included with any notification.

The process ID of the process which generated the notification.

#### process

May be included with any notification.

The name of the process which generated the notification.

**serverDomain** Included in Login, and optionally in Logout or other events. The domain name or IP address (v4 or v6) used to access the server or mailbox.

**serverPort** Included in Login, and optionally in Logout or other events. The port number used to access the server. This is often a well-known port.

#### serverFQDN

May be included with any notification.

The fully qualified domain name of the server which generated the event. Note that this may be different from the server name used to access the mailbox included in the mailbox identifier.

#### service

May be included with any notification.



The name of the service which triggered the event. Suggested values include "imap", "pop", "http", and "admincli" (for an administrative client).

#### tags

May be included with any notification.

A list of UTF-8 tags (likely to be comma-separated). One or more tags can be set at the time a notification criteria or notification subscription is created. Subscribers can use tags for additional client-side filtering or dispatch of events.

#### timestamp

May be included with any notification.

The time at which the event occurred which triggered the notification (the underlying protocol carrying the notification may contain a timestamp for when the notification was generated). This MAY be an approximate time.

Timestamps are expressed in local time and contain the offset from UTC (this information is used in several places in Internet mail), and would normally be in [[RFC3339](#)] format.

#### uidnext

May be included with any notification referring to a mailbox.

The UID that is projected to be assigned next in the mailbox. This is typically included with message addition and deletion events. This is equivalent to the UIDNEXT status item in the IMAP STATUS command.

#### uidset

Included with MessageExpires, MessageExpunges, MessageRead, MessageTrash, FlagsSet and FlagsClear.

This includes the set of IMAP UIDs referenced.

#### uri

Included with all notifications and refers to the IMAP server, a mailbox or a message.

Typically an IMAP URL. This can include the name of the server





used to access the mailbox/message, the mailbox name, the UIDVALIDITY of the mailbox, and the UID of a specific message.

user

Included with all events generated by message access protocols.

This is the authorization identifier used when the client connected to the access protocol which triggered the event. Some protocols (for example, many SASL mechanisms) distinguish between authorization and authentication identifiers. For events associated with a mailbox, this may be different from the owner of the mailbox specified in the IMAP URL.

## **6. IANA Considerations**

The IANA is requested to create a new registry for "Internet Message Store Events" containing two sub-registries: event names and event parameters. For both event names and event parameters, entries which do not start with "vnd." are added by the IETF and intended for interoperable use. Entries which start with "vnd." are intended for private use by one or more parties and are allocated to avoid collisions.

The initial values are contained in this document.

Using IANA Considerations [[RFC5226](#)] terminology, entries which do not start with "vnd." are allocated by IETF Consensus, while those starting with "vnd." are allocated First Come First Served.

## **7. Security Considerations**

Notifications can produce a large amount of traffic and expose sensitive information. When notification mechanisms are used to maintain state between a different entities, the ability to corrupt or manipulate notification messages could enable an attacker to modulate the state of these entities. For example, if an attacker were able to modify notifications sent from a message store to an auditing server, he could modify the "user" and "messageContent" parameters in MessageNew notifications to create false audit log entries.

A competent transfer protocol for notifications must consider authentication, authorization, privacy, and message integrity, as well as denial-of-service issues. While the IETF has adequate tools and experience to address these issues for mechanisms which involve



only one TCP connection, notification or publish/subscribe protocols which are more sophisticated than a single end-to-end TCP connection will need to pay extra attention to these issues and carefully balance requirements to successfully deploy a system with security and privacy considerations.

## **8. Acknowledgments**

Alexey Melnikov, Arnt Gulbrandsen, and Zoltan Ordogh have reviewed and offered improvements to this draft. Richard Barnes did a nice review during Last Call.

## **9. References**

### **9.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5092] Melnikov, A. and C. Newman, "IMAP URL Scheme", [RFC 5092](#), November 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", [RFC 3501](#), March 2003.

### **9.2. Informative References**

- [RFC4314] Melnikov, A., "IMAP4 Access Control List (ACL) Extension", [RFC 4314](#), December 2005.
- [RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, [RFC 1939](#), May 1996.
- [RFC2177] Leiba, B., "IMAP4 IDLE command", [RFC 2177](#), June 1997.
- [RFC2447] Dawson, F., Mansour, S., and S. Silverberg, "iCalendar Message-Based Interoperability Protocol (iMIP)", [RFC 2447](#), November 1998.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.



- [RFC2822] Resnick, P., "Internet Message Format", [RFC 2822](#), April 2001.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", [RFC 3339](#), July 2002.
- [RFC3458] Burger, E., Candell, E., Eliot, C., and G. Klyne, "Message Context for Internet Mail", [RFC 3458](#), January 2003.
- [RFC4146] Gellens, R., "Simple New Mail Notification", [RFC 4146](#), August 2005.
- [RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", [RFC 4422](#), June 2006.
- [RFC4467] Crispin, M., "Internet Message Access Protocol (IMAP) - URLAUTH Extension", [RFC 4467](#), May 2006.
- [RFC4551] Melnikov, A. and S. Hole, "IMAP Extension for Conditional STORE Operation or Quick Flag Changes Resynchronization", [RFC 4551](#), June 2006.

## [Appendix A](#). Future Extensions

This document specifies core functionality based on events which are believed to be well understood, have known use cases and are implemented by at least one deployed real-world Internet message store. (A few events are exceptions to the last test only: FlagsSet, FlagsClear, MailboxCreate, MailboxDelete, MailboxRename, MailboxSubscribe, and MailboxUnSubscribe.)

Some events have been suggested, but are postponed to future extensions because they do not meet this criteria. These events include messages which have been moved to archive storage and may require extra time to access, quota by message context, authentication failure, user mail account disabled, annotations, and mailbox ACL or metadata change. The descriptions of several events note additional parameters which are likely candidates for future inclusion. See [Section 6](#) for how the list of events and parameters can be extended.

In order to narrow the scope of this document to something that can be completed, only events generated from the message store (by a message access module, administrative module or message delivery agent) are considered. A complete mail system is normally linked with an identity system which would also publish events of interest to a message store event subscriber. Events of interest include



account created/deleted/disabled and password changed/expired.

Authors' Addresses

Randall Gellens  
QUALCOMM Incorporated  
5775 Morehouse Drive  
San Diego, CA 92651  
US

Email: [rg+ietf@qualcomm.com](mailto:rg+ietf@qualcomm.com)

Chris Newman  
Sun Microsystems  
800 Royal Oaks  
Monrovia, CA 91016-6347  
US

Email: [chris.newman@sun.com](mailto:chris.newman@sun.com)





## Full Copyright Statement

Copyright (C) The IETF Trust (2008).

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, THE IETF TRUST 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.

## Intellectual Property

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](mailto:ietf-ipr@ietf.org).

## Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

