Workgroup: Network Working Group
Internet-Draft: draft-ietf-extra-quota-10
Obsoletes: 2087 (if approved)
Published: 18 November 2021
Intended Status: Standards Track
Expires: 22 May 2022
Authors: A. Melnikov
Isode

IMAP QUOTA Extension

Abstract

This document defines a QUOTA extension of the Internet Message Access Protocol (RFC 3501/RFC 9051) that permits administrative limits on resource usage (quotas) to be manipulated through the IMAP protocol.

This document obsoletes RFC 2087, but attempts to remain backwards compatible whenever possible.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on 22 May 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Document Conventions
2. Introduction and Overview
3. Terms
  3.1. Resource
    3.1.1. Name
    3.1.2. Definition
  3.2. Quota Root
4. Definitions
  4.1. Commands
    4.1.1. GETQUOTA
    4.1.2. GETQUOTAROOT
    4.1.3. SETQUOTA
    4.1.4. New STATUS attributes
  4.2. Responses
    4.2.1. QUOTA
    4.2.2. QUOTAROOT
  4.3. Response Codes
    4.3.1. OVERQUOTA
5. Resource Type Definitions
  5.1. STORAGE
  5.2. MESSAGE
  5.3. MAILBOX
  5.4. ANNOTATION-STORAGE
6. Interaction with IMAP ACL extension (RFC 4314)
7. Formal syntax
8. Security Considerations
9. IANA Considerations
  9.1. Changes/additions to the IMAP4 capabilities registry
  9.2. IMAP quota resource type registry
  9.3. Registrations of IMAP Quota Resource Types
10. Contributors
11. Acknowledgments

12. Changes since RFC 2087
13. References
  13.1. Normative References
  13.2. Informative References
Author's Address

1. Document Conventions

In protocol examples, this document uses a prefix of "C: " to denote lines sent by the client to the server, and "S: " for lines sent by the server to the client. Lines prefixed with "// " are comments explaining the previous protocol line. These prefixes and comments are not part of the protocol. Lines without any of these prefixes are continuations of the previous line, and no line break is present in the protocol before such lines unless specifically mentioned.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Other capitalised words are IMAP keywords [RFC3501][RFC9051] or keywords from this document.

2. Introduction and Overview

This document defines a couple of extensions to the Internet Message Access Protocol [RFC3501] for querying and manipulating administrative limits on resource usage (quotas). This extension is compatible with both IMAP4rev1 [RFC3501] and IMAP4rev2 [RFC9051].

The capability "QUOTA", denotes a RFC2087 [RFC2087] compliant server. Some responses and response codes defined in this document are not present in such servers (see Section 12 for more details), and clients MUST NOT rely on their presence in the absence of any capability beginning with "QUOTA=".

Any server compliant with this document MUST also return at least one capability starting with "QUOTA=RES-" prefix, as described in Section 3.1.

Any server compliant with this document that implements the SETQUOTA command (see Section 4.1.3) MUST also return the "QUOTASET" capability.

This document also reserves all other capabilities starting with "QUOTA=" prefix for future IETF stream standard track, informational or experimental extensions to this document.

Quotas can be used to restrict clients for administrative reasons, but the QUOTA extension can also be used to indicate system limits and current usage levels to clients.

Although RFC2087 [RFC2087] specified an IMAP4 QUOTA extension, and this has seen deployment in servers, it has seen little deployment in clients. Since the meaning of the resources was left implementation-dependent, it was impossible for a client implementation to determine which resources were supported, and impossible to determine which mailboxes were in a given quota root (see Section 3.2), without a priori knowledge of the implementation.

3. Terms

3.1. Resource

A resource has a name, a formal definition.

3.1.1. Name

The resource name is an atom, as defined in IMAP4rev1 [RFC3501]. These MUST be registered with IANA.

Supported resource names MUST be advertised as a capability, by prepending the resource name with "QUOTA=RES-". A server compliant with this specification is not required to support all reported resource types on all quota roots.

3.1.2. Definition

The resource definition or document containing it, while not visible through the protocol, SHOULD be registered with IANA.

The usage of a resource MUST be represented as a 63 bit unsigned integer. 0 indicates that the resource is exhausted. Usage integers don't necessarily represent proportional use, so clients MUST NOT compare available resource between two separate quota roots on the same or different servers.

Limits will be specified as, and MUST be represented as, an integer. 0 indicates that any usage is prohibited.

Limits may be hard or soft - that is, an implementation MAY choose, or be configured, to disallow any command if the limit on a resource is or would be exceeded.

All resources which the server handles MUST be advertised in a CAPABILITY response/response code consisting of the resource name prefixed by "QUOTA=RES-".

The resources STORAGE (Section 5.1), MESSAGE (Section 5.2), MAILBOX (Section 5.3) and ANNOTATION-STORAGE (Section 5.4) are defined in this document.

3.2. Quota Root

This document introduces a concept of a "quota root", as resource limits can apply across multiple IMAP mailboxes.

Each mailbox has zero or more implementation-defined named "quota roots". Each quota root has zero or more resource limits (quotas). All mailboxes that share the same named quota root share the resource limits of the quota root.

Quota root names need not be mailbox names, nor is there any relationship defined by this document between a quota root name and a mailbox name. A quota root name is an astring, as defined in IMAP4 [RFC3501]. It SHOULD be treated as an opaque string by any clients.

Quota roots are used since not all implementations may be able to calculate usage, or apply quotas, on arbitrary mailboxes or mailbox hierarchies.

Not all resources may be limitable or calculable for all quota roots. Further, not all resources may support all limits - some limits may be present in the underlying system. A server implementation of this memo SHOULD advise the client of such inherent limits, by generating QUOTA (Section 4.2.1) responses, and SHOULD advise the client of which resources are limitable for a particular quota root. A SETQUOTA (Section 4.1.3) command MAY also round a quota limit in an implementation-dependent way, if the granularity of the underlying system demands it. A client MUST be prepared for a SETQUOTA (Section 4.1.3) command to fail if a limit cannot be set.

Implementation Notes: This means that, for example under UNIX, a quota root may have a MESSAGE (Section 5.2) quota always set due to the number of inodes available on the filesystem, and similarly STORAGE (Section 5.1) may be rounded to the nearest block and limited by free filesystem space.

4. Definitions

4.1. Commands

The following commands exist for manipulation and querying quotas.

4.1.1. GETQUOTA

Arguments: quota root

Responses:
REQUIRED untagged responses: QUOTA

Result: OK - getquota completed

NO - getquota error: no such quota root, permission denied

BAD - command unknown or arguments invalid

The GETQUOTA command takes the name of a quota root and returns the quota root's resource usage and limits in an untagged QUOTA response. (Names of quota roots applicable to a particular mailbox can be discovered by issuing the GETQUOTAROOT command, see Section 4.1.2.) Note that the server is not required to support any specific resource type (as advertised in the CAPABILITY response, i.e. all capability items with the "QUOTA=RES-" prefix) for any particular quota root.

Example:

S: * CAPABILITY [...] QUOTA QUOTA=RES-STORAGE [...]

[...]

C: G0001 GETQUOTA "!partition/sda4"

S: * QUOTA "!partition/sda4" (STORAGE 104 10923847)

S: G0001 OK Getquota complete

4.1.2. GETQUOTAROOT

Arguments: mailbox name

Responses: REQUIRED untagged responses: QUOTAROOT, QUOTA

Result: OK - getquotaroot completed

NO - getquotaroot error: permission denied

BAD - command unknown or arguments invalid

The GETQUOTAROOT command takes a mailbox name and returns the list of quota roots for the mailbox in an untagged QUOTAROOT response. For each listed quota root, it also returns the quota root's resource usage and limits in an untagged QUOTA response.

Note that the mailbox name parameter doesn't have to reference an existing mailbox. This can be handy in order to determine which quotaroot would apply to a mailbox when it gets created.

Example:

S: * CAPABILITY [...] QUOTA QUOTA=RES-STORAGE QUOTA=RES-MESSAGE [...]

[...]

C: G0002 GETQUOTAROOT INBOX

S: * QUOTAROOT INBOX "#user/alice" "!partition/sda4"

S: * QUOTA "#user/alice" (MESSAGE 42 1000)

S: * QUOTA "!partition/sda4" (STORAGE 104 10923847)

S: G0002 OK Getquotaroot complete

4.1.3. SETQUOTA

Arguments: quota root

list of resource limits

Responses: untagged responses: QUOTA

Result: OK - setquota completed

NO - setquota error: can't set that data

BAD - command unknown or arguments invalid

Note that unlike other command/responses/response codes defined in this document, support for SETQUOTA command requires the server to advertise "QUOTASET" capability.

The SETQUOTA command takes the name of a mailbox quota root and a list of resource limits. The resource limits for the named quota root are changed to be the specified limits. Any previous resource limits for the named quota root are discarded, even resource limits not explicitly listed in the SETQUOTA command. (For example, if the quota root had both STORAGE and MESSAGE limits assigned to the quota root before the SETQUOTA is called and the SETQUOTA only includes the STORAGE limit, then the MESSAGE limit is removed from the quota root.)

If the named quota root did not previously exist, an implementation may optionally create it and change the quota roots for any number of existing mailboxes in an implementation-defined manner.

If the implementation chooses to change the quota roots for some existing mailboxes such changes SHOULD be announced with untagged QUOTA responses.

Example:

```
S: * CAPABILITY [...] QUOTA QUOTASET QUOTA=RES-STORAGE QUOTA=RES-
MESSAGE [...]

[...]

C: S0000 GETQUOTA "#user/alice"

S: * QUOTA "#user/alice" (STORAGE 54 111 MESSAGE 42 1000)

S: S0000 OK Getquota completed

C: S0001 SETQUOTA "#user/alice" (STORAGE 510)

S: * QUOTA "#user/alice" (STORAGE 58 512)

// The server has rounded the STORAGE quota limit requested to
the nearest 512 blocks of 1024 octects, or else another client
has performed a near simultaneous SETQUOTA, using a limit of 512.

S: S0001 OK Rounded quota

C: S0002 SETQUOTA "!partition/sda4" (STORAGE 99999999)

S: * QUOTA "!partition/sda4" (STORAGE 104 10923847)

// The server has not changed the quota, since this is a
filesystem limit, and cannot be changed. The QUOTA response here
is entirely optional.

S: S0002 NO Cannot change system limit
```

4.1.4. New STATUS attributes

DELETED and DELETED-STORAGE status data items allow to estimate the amount of resource freed by an EXPUNGE on a mailbox.

The DELETED status data item requests the server to return the number of messages with \Deleted flag set. The DELETED status data item is only required to be implemented when the server advertises QUOTA=RES-MESSAGE capability.

The DELETED-STORAGE status data item requests the server to return the amount of storage space that can be reclaimed by performing EXPUNGE on the mailbox. The server SHOULD return the exact value, however it is recognized that the server may have to do non-trivial amount of work to calculate it. If the calculation of the exact value would take a long time, the server MAY instead return the sum of RFC822.SIZEs of messages with the \Deleted flag set. The DELETED-STORAGE status data item is only required to be implemented when the server advertises QUOTA=RES-STORAGE capability.

Example:

S: * CAPABILITY [...] QUOTA QUOTA=RES-STORAGE QUOTA=RES-MESSAGE [...]

[...]

C: S0003 STATUS INBOX (MESSAGES DELETED DELETED-STORAGE)

S: * STATUS INBOX (MESSAGES 12 DELETED 4 DELETED-STORAGE 8)

// 12 messages, 4 of which would be deleted when an EXPUNGE happens.

S: S0003 OK Status complete.

4.2. Responses

The following responses may be sent by the server.

4.2.1. QUOTA

Data: quota root name

list of resource names, usages, and limits

This response occurs as a result of a GETQUOTA, a GETQUOTAROOT or a SETQUOTA command. The first string is the name of the quota root for which this quota applies.

The name is followed by a S-expression format list of the resource usage and limits of the quota root. The list contains zero or more triplets. Each triplet contains a resource name, the current usage of the resource, and the resource limit.

Resources not named in the list are not limited in the quota root. Thus, an empty list means there are no administrative resource limits in the quota root.

Example: S: * QUOTA "" (STORAGE 10 512)

4.2.2. QUOTAROOT

Data: mailbox name

zero or more quota root names

This response occurs as a result of a GETQUOTAROOT command. The first string is the mailbox and the remaining strings are the names of the quota roots for the mailbox.

Examples:

S: * QUOTAROOT INBOX ""

// The INBOX mailbox is covered by a single quota root with name "".

S: * QUOTAROOT comp.mail.mime

// The comp.mail.mime mailbox has no quota root associated with it, but one can be created.

4.3. Response Codes

4.3.1. OVERQUOTA

OVERQUOTA response code SHOULD be returned in the tagged NO response to an APPEND/COPY/MOVE when the addition of the message(s) puts the target mailbox over any one of its quota limits.

```
Example 1:  C: A003 APPEND saved-messages (\Seen) {326}
            S: + Ready for literal data
            C: Date: Mon, 7 Feb 1994 21:52:25 -0800 (PST)
            C: From: Fred Foobar <foobar@Blurdybloop.example>
            C: Subject: afternoon meeting
            C: To: mooch@owatagu.siam.edu.example
            C: Message-Id: <B27397-0100000@Blurdybloop.example>
            C: MIME-Version: 1.0
            C: Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
            C:
            C: Hello Joe, do you think we can meet at 3:30 tomorrow?
            C:
            S: A003 NO [OVERQUOTA] APPEND Failed
```

The OVERQUOTA response code MAY also be returned in an untagged NO response in the authenticated or the selected state, when a mailbox exceeds soft quota. For example, such OVERQUOTA response code might be sent as a result of an external event (e.g. LMTP delivery or COPY/MOVE/APPEND in another IMAP connection) that causes the currently selected mailbox to exceed soft quota. Note that such OVERQUOTA response code might be ambiguous, because it might relate

to the target mailbox (as specified in COPY/MOVE/APPEND) or to the currently selected mailbox. (The WG chose not to address this deficiency due to syntactic limitations of IMAP response codes and because such events are likely to be rare.) This form of the OVERQUOTA response codes MUST NOT be returned if there is no mailbox selected and no command in progress that adds a message to a mailbox (e.g. APPEND).

```
Example 2:  C: A003 APPEND saved-messages (\Seen) {326}
            S: + Ready for literal data
            C: Date: Mon, 7 Feb 1994 21:52:25 -0800 (PST)
            C: From: Fred Foobar <foobar@Blurdybloop.example>
            C: Subject: afternoon meeting
            C: To: mooch@owatagu.siam.edu.example
            C: Message-Id: <B27397-0100000@Blurdybloop.example>
            C: MIME-Version: 1.0
            C: Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
            C:
            C: Hello Joe, do you think we can meet at 3:30 tomorrow?
            C:
            S: * NO [OVERQUOTA] Soft quota has been exceeded
            S: A003 OK [APPENDUID 38505 3955] APPEND completed

Example 3:  C: A004 COPY 2:4 MEETING
            S: * NO [OVERQUOTA] Soft quota has been exceeded
            S: A004 OK [COPYUID 38505 304,319:320 3956:3958] COPY
                command completed
```

5. Resource Type Definitions

The following resource types are defined in this memo. A server supporting a resource type MUST advertise this as a CAPABILITY with a name consisting of the resource name prefixed by "QUOTA=RES-". A server MAY support multiple resource types, and MUST advertise all resource types it supports.

5.1. STORAGE

The physical space estimate, in units of 1024 octets, of the mailboxes governed by the quota root. This MAY not be the same as the sum of the RFC822.SIZE of the messages. Some implementations MAY include metadata sizes for the messages and mailboxes, other implementations MAY store messages in such a way that the physical space used is smaller, for example due to use of compression. Additional messages might not increase the usage. Client MUST NOT use the usage figure for anything other than informational purposes, for example, they MUST NOT refuse to APPEND a message if the limit less the usage is smaller than the RFC822.SIZE divided by 1024 of the message, but it MAY warn about such condition.

The usage figure may change as a result of performing actions not associated with adding new messages to the mailbox, such as SEARCH, since this may increase the amount of metadata included in the calculations.

When the server supports this resource type, it MUST also support the DELETED-STORAGE status data item.

Support for this resource MUST be indicated by the server by advertising the CAPABILITY "QUOTA=RES-STORAGE".

A resource named the same was also given as an example in RFC2087 [RFC2087]. This document provides a more precise definition.

5.2. MESSAGE

The number of messages stored within the mailboxes governed by the quota root. This MUST be an exact number, however, clients MUST NOT assume that a change in the usage indicates a change in the number of messages available, since the quota root may include mailboxes the client has no access to.

When the server supports this resource type, it MUST also support the DELETED status data item.

Support for this resource MUST be indicated by the server by advertising the CAPABILITY "QUOTA=RES-MESSAGE".

A resource named the same was also given as an example in RFC2087 [RFC2087]. This document provides a more precise definition.

5.3. MAILBOX

The number of mailboxes governed by the quota root. This MUST be an exact number, however, clients MUST NOT assume that a change in the usage indicates a change in the number of mailboxes, since the quota root may include mailboxes the client has no access to.

Support for this resource MUST be indicated by the server by advertising the CAPABILITY "QUOTA=RES-MAILBOX".

5.4. ANNOTATION-STORAGE

The maximum size of all annotations [RFC5257], in units of 1024 octets, associated with all messages in the mailboxes governed by the quota root.

Support for this resource MUST be indicated by the server by advertising the CAPABILITY "QUOTA=RES-ANNOTATION-STORAGE".

6. Interaction with IMAP ACL extension (RFC 4314)

This section lists [RFC4314] rights required to execute quota related commands when both RFC 4314 and this document are implemented.

Operations\Rights	l	r	s	w	i	c	x	t	e	a	Any	Non
GETQUOTA												+
GETQUOTAROOT		*										*
SETQUOTA										+		

Table 1

See Section 4 of RFC 4314 for conventions used in this table.

Legend:

+ - The right is required

***** - Only one of the rights marked with * is required

"Any" - at least one of the "l", "r", "i", "k", "x", "a" rights is required

"Non" - no rights required to perform the command

Note that which permissions are needed in order to perform GETQUOTAROOT command depends on the quota resource type being requested. For example, a quota on number of messages (MESSAGE resource type) or total size of messages (STORAGE resource type) requires "r" right on the mailbox in question, since the quota involved would reveal information about the number (or total size) of messages in the mailbox. By comparison, the MAILBOX resource type doesn't require any right.

7. 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 [RFC3501].

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.

getquota = "GETQUOTA" SP quota-root-name

getquotaroot = "GETQUOTAROOT" SP mailbox

```
quota-list =
             "(" quota-resource *(SP quota-resource) ")"

quota-resource =  resource-name SP resource-usage SP resource-limit

quota-response =  "QUOTA" SP quota-root-name SP quota-list

quotaroot-response =  "QUOTAROOT" SP mailbox *(SP quota-root-name)

setquota =  "SETQUOTA" SP quota-root-name SP setquota-list

setquota-list =  "(" [setquota-resource *(SP setquota-resource)] ")"

setquota-resource =  resource-name SP resource-limit

quota-root-name =  astring

resource-limit =  number64

resource-name =  "STORAGE" / "MESSAGE" / "MAILBOX" /

   "ANNOTATION-STORAGE" / resource-name-ext

resource-name-ext =  atom

   ;; Future resource registrations

resource-usage =  number64

   ;; must be less than corresponding resource-limit

capability-quota =  capa-quota-res / "QUOTASET"

   ;; One or more capa-quota-res must be returned.

   ;; Also "QUOTASET" can optionally be returned.

capa-quota-res =  "QUOTA=RES-" resource-name

status-att =/  "DELETED" / "DELETED-STORAGE"

   ;; DELETED status data item MUST be supported

   ;; when "QUOTA=RES-MESSAGE" capability is

   ;; advertised.

   ;; DELETED-STORAGE status data item MUST be

   ;; supported when "QUOTA=RES-STORAGE" capability
```

```
   ;; is advertised.

status-att-val =/  status-att-deleted /

   status-att-deleted-storage

status-att-deleted =  "DELETED" SP number

   ;; DELETED status data item MUST be supported

   ;; when "QUOTA=RES-MESSAGE" capability is

   ;; advertised.

status-att-deleted-storage =  "DELETED-STORAGE" SP number64

   ;; DELETED-STORAGE status data item MUST be

   ;; supported when "QUOTA=RES-STORAGE" capability

   ;; is advertised.

resp-text-code =/  "OVERQUOTA"

number64 =  <Defined in RFC 9051>
```

8. Security Considerations

Implementors should be careful to make sure the implementation of these commands does not violate the site's security policy. The resource usage of other users is likely to be considered confidential information and should not be divulged to unauthorized persons. In particular, no quota information should be disclosed to anonymous users.

As for any resource shared across users (for example a quota root attached to a set of shared mailboxes), a user that can consume or render unusable the resource can affect the resources available to the other users; this might occur, for example, by a user with permission to execute SETQUOTA setting an artificially small value.

Note that computing resource usage might incur a heavy load on the server. Server implementers should consider implementation techniques that lower load on servers, such as caching of resource usage information or usage of less precise computations when under heavy load.

9. IANA Considerations

9.1. Changes/additions to the IMAP4 capabilities registry

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

https://www.iana.org/assignments/imap4-capabilities

IANA is requested to update definition of the QUOTA extension to point to this document. IANA is also requested to add the "QUOTASET" capability to the IMAP4 capabilities registry, with this document as the reference.

IANA is requested to reserve the prefix "QUOTA=RES-" in the IMAP4 capabilities registry and add a pointer to this document and to the IMAP quota resource type registry (see Section 9.2).

IANA is requested to reserve all other capabilities starting with "QUOTA=" prefix for future IETF Stream extensions to this document.

9.2. IMAP quota resource type registry

IANA is requested to create a new registry for IMAP quota resource types. Registration policy for this registry is "Specification Required". When registering a new quota resource type, the registrant need to provide the following: Name of the quota resource type, Author/Change Controller name and email address, short description, extra (if any) required and optional IMAP commands/ responses, and a reference to a specification that describes the quota resource type in more details.

Designated Experts should check that provided references are correct, that they describe the quota resource type being registered in sufficient details to be implementable, that syntax of any optional commands/responses is correct (e.g. ABNF validates), and their syntax/description complies with rules and limitations imposed by IMAP [RFC3501][RFC9051]. Designated Experts should avoid registering multiple identical quota resource types under different names and should provide advice to requestors about other possible quota resource types to use.

This document includes initial registrations for the following IMAP quota resource type: STORAGE (Section 5.1), MESSAGE (Section 5.2), MAILBOX (Section 5.3) and "ANNOTATION-STORAGE" (Section 5.4). See Section 9.3 for the registration templates.

9.3. Registrations of IMAP Quota Resource Types

Name of the quota resource type:
STORAGE

Author: Alexey Melnikov <alexey.melnikov@isode.com>

Change Controller: IESG <iesg@ietf.org>

Description: The physical space estimate, in units of 1024 octets, of the mailboxes governed by the quota root.

Extra required IMAP commands/responses: DELETED-STORAGE STATUS request data item and response data item

Extra optional IMAP commands/responses: N/A

Reference: Section 5.1 of RFCXXXX

Name of the quota resource type: MESSAGE

Author: Alexey Melnikov <alexey.melnikov@isode.com>

Change Controller: IESG <iesg@ietf.org>

Description: The number of messages stored within the mailboxes governed by the quota root.

Extra required IMAP commands/responses: DELETED STATUS request data item and response data item

Extra optional IMAP commands/responses: N/A

Reference: Section 5.2 of RFCXXXX

Name of the quota resource type: MAILBOX

Author: Alexey Melnikov <alexey.melnikov@isode.com>

Change Controller: IESG <iesg@ietf.org>

Description: The number of mailboxes governed by the quota root.

Extra required IMAP commands/responses: N/A

Extra optional IMAP commands/responses: N/A

Reference: Section 5.3 of RFCXXXX

Name of the quota resource type: ANNOTATION-STORAGE

Author: Alexey Melnikov <alexey.melnikov@isode.com>

Change Controller:
IESG <iesg@ietf.org>

Description: The maximum size of all annotations [RFC5257], in units of 1024 octets, associated with all messages in the mailboxes governed by the quota root.

Extra required IMAP commands/responses: N/A

Extra optional IMAP commands/responses: N/A

Reference: Section 5.4 of RFCXXXX

10. Contributors

Dave Cridland wrote lots of text in an earlier draft that became the basis for this document.

11. Acknowledgments

Editor of this document would like to thank the following people who provided useful comments or participated in discussions that lead to this update to RFC 2087: John Myers, Cyrus Daboo, Lyndon Nerenberg, Benjamin Kaduk, Roman Danyliw, Eric Vyncke.

This document is a revision of RFC 2087. It borrows a lot of text from RFC 2087. Thus work of the RFC 2087 author John Myers is appreciated.

12. Changes since RFC 2087

This document is a revision of RFC 2087. It tries to clarify the meaning of different terms used by RFC 2087. It also provides more examples, gives guidance on allowed server behaviour, defines IANA registry for quota resource types and provides initial registrations for 4 of them.

When compared with RFC 2087, this document defines two more commonly used resource type, adds optional OVERQUOTA response code and defines two extra STATUS data items ("DELETED" and "DELETED-STORAGE"). The DELETED STATUS data item must be implemented if the QUOTA=RES-MESSAGE capability is advertised. The DELETED-STORAGE STATUS data item must be implemented if the QUOTA=RES-STORAGE capability is advertised. For extensibility quota usage and quota limits are now 63 bit unsigned integers.

13. References

13.1. Normative References

[ABNF] Crocker, D., Ed. and P. Overell, Ed., "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, DOI 10.17487/RFC3501, March 2003, <https://www.rfc-editor.org/info/rfc3501>.

[RFC4314] Melnikov, A., "IMAP4 Access Control List (ACL) Extension", RFC 4314, DOI 10.17487/RFC4314, December 2005, <https://www.rfc-editor.org/info/rfc4314>.

[RFC5257] Daboo, C. and R. Gellens, "Internet Message Access Protocol - ANNOTATE Extension", RFC 5257, DOI 10.17487/RFC5257, June 2008, <https://www.rfc-editor.org/info/rfc5257>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9051] Melnikov, A., Ed. and B. Leiba, Ed., "Internet Message Access Protocol (IMAP) - Version 4rev2", RFC 9051, DOI 10.17487/RFC9051, August 2021, <https://www.rfc-editor.org/info/rfc9051>.

13.2. Informative References

[RFC2087] Myers, J., "IMAP4 QUOTA extension", RFC 2087, DOI 10.17487/RFC2087, January 1997, <https://www.rfc-editor.org/info/rfc2087>.

Author's Address

Alexey Melnikov
Isode Limited

Email: alexey.melnikov@isode.com
URI: https://www.isode.com