IMAP QUOTA Extension
draft-ietf-extra-quota-01

Abstract

   The QUOTA extension of the Internet Message Access Protocol (RFC
   3501) permits administrative limits on resource usage (quotas) to be
   manipulated through the IMAP protocol.

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

Status of This Memo

Copyright Notice

Table of Contents

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 unless specifically mentioned.

   Again, for examples, the hierarchy separator on the server is
   presumed to be "/" throughout. None of these assumptions is required
   nor recommended by this document.

   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 [RFC2119] 8174 [RFC8174] when, and only when, they appear
   in all capitals, as shown here.

   Other capitalised words are IMAP4 [RFC3501] keywords or keywords from
   this document.

2. Introduction and Overview

   The capability "QUOTA", denotes a RFC2087 [RFC2087] compliant server.
   Some commands and responses defined in this document are not present
   in such servers, 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.

   This document also reserves all other capabilities starting with
   "QUOTA=" prefix to future standard track 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-
dependant, 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, 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. Implementation specific
resources begin with "V-" .

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 32 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 constisting of the resource name prefixed by "QUOTA=RES-".
For compatability with RFC2087 [RFC2087], a client which discovers
resources available on the server which are not advertised through

this mechanism MUST treat them as if they were completely opaque, and
without any meaning.

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

3.2. Quota Root

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 memo 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 arbitary mailboxes or mailbox
hierarchies.

Not all resources may be limitable or calculatable 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 dependant 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.
   The client can try using any of the resource types returned in
   CAPABILITY response (i.e. all capability items with "QUOTA=RES-"
   prefix), however the server is not required to support any specific
   resource type 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: no such mailbox, 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
```

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.

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

DELETED-MESSAGES status data item requests the server to return the
number of messages with \Deleted flag set.

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.

```
Example:

S: * CAPABILITY [...] QUOTA QUOTA=RES-STORAGE QUOTA-RES-MESSAGE
[...]
[...]
C: S0003 STATUS INBOX (MESSAGES DELETED-MESSAGES DELETED-STORAGE)
S: * STATUS INBOX (MESSAGES 12 DELETED-MESSAGES 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 or GETQUOTAROOT
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.

Example:

S: * QUOTAROOT INBOX ""

S: * QUOTAROOT comp.mail.mime

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:

```
   S: C: A003 APPEND Drafts (\Seen $MDNSent) {310}
   S: + Ready for literal data
   C: Date: Mon, 7 Feb 1994 21:52:25 -0800 (PST)
   C: From: Fred Foobar <foobar@Blurdybloop.COM>
   C: Subject: afternoon meeting
   C: To: mooch@owatagu.siam.edu
   C: Message-Id: <B27397-0100000@Blurdybloop.COM>
   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 when a mailbox exceeds soft quota. Such responses have 2
forms. If it is followed by a tag, the tag refers to the command
that caused this (such as APPEND or COPY) and the OVERQUOTA response
code applies to the target mailbox specified by such command. If the
OVERQUOTA response code is not followed by the tag, this means that
an external event (e.g. LMTP delivery or APPEND/COPY in another IMAP
connection) caused this event and the event applies to the currently
selected mailbox. In particular, this means that such OVERQUOTA
response codes MUST NOT be returned if there is no mailbox selected
or if a mailbox other than the currently selected one exceeds soft
quota.

Example:

```
S: C: A003 APPEND Drafts (\Seen $MDNSent) {310}
S: + Ready for literal data
C: Date: Mon, 7 Feb 1994 21:52:25 -0800 (PST)
C: From: Fred Foobar <foobar@Blurdybloop.COM>
C: Subject: afternoon meeting
C: To: mooch@owatagu.siam.edu
C: Message-Id: <B27397-0100000@Blurdybloop.COM>
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 A003] Soft quota has been exceeded
S: A003 OK [APPENDUID 38505 3955] APPEND 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 mupltiple 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.

   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]. In absence of information to the contrary, clients
   conformant to this specification connecting to servers which do not
   advertise "QUOTA=RES-STORAGE", yet allow a resource named STORAGE,
   MUST NOT assume that it is the same resource. [[CREF1: Is the above
   restriction useful and will it be obeyed? If the answer is "no",
   delete it.]]

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.

   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], clients conformant to this specification connecting to
   servers which do not advertise "QUOTA=RES-MESSAGE", yet allow a
   resource named MESSAGE, MUST NOT assume that it is the same resource.
   [[CREF2: As above.]]

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

[[CREF3: Bron to check whether this is a sensible description and
whether it is needed at all:]] 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)

[[CREF4: TBD]]

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     =  number

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

resource-name-vnd  =  "V-" atom
                      ;; Vendor specific, must be registered with IANA.
                      ;; The "V-" prefix should be followed by a domain
                      name
                      ;; under vendor's control.

resource-name-ext  =  atom
                      ;; Not starting with V- and defined
                      ;; in a Standard Track or Experimental RFC

resource-names =      "(" [resource-name *(SP resource-name)] ")"

resource-usage     =  number
                      ;; must be less than corresponding resource-limit

capability-quota   =  capa-quota-res

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

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

                      [[CREF5: Should the above be optional unless the
                      server implements MESSAGE/STORAGE?  Also sync
                      this with IMAP4rev2.]]

resp-text-code     =/  "OVERQUOTA" [SP tag]
```

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.

9. IANA Considerations

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

      http://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 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 and a reference to a specification that describes the
   quota resource type in more details.

   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
   details below.

   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 quote resource type registry established above.

   IANA is requested to reserve all other capabilities starting with
   "QUOTA=" prefix for future standard track or experimental extensions
   to this document.

9.1. 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.

   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.

   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.

   Reference: Section 5.3 of RFCXXXX

   Name of the quota resource type:

   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. [[CREF6: Recheck against the final
      description of "ANNOTATION-STORAGE".]]

   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

Editors 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

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 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 3 of them.

When compared with RFC 2087, this document defines one more commonly used resource type, adds optional OVERQUOTA response code and defines two extra STATUS data items ("DELETED-MESSAGES" and "DELETED-STORAGE")

13. References

13.1. Normative References

[ABNF]   Crocker, D., Ed. and P. Overell, Ed., "Augmented BNF for
           Syntax Specifications: ABNF", RFC 5234, January 2008.

[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>.

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