

Network Working Group
Internet-Draft
Expires: February 18, 2003

D. Cridland
Clues
A. Melnikov
ACI
August 20, 2002

IMAP4rev1 QUOTA Extension
draft-cridland-imap-quota-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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 February 18, 2003.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

The QUOTA extension of the Internet Message Access Protocol IMAP4 [2] permits administrative limits on resource usage (quotas) to be manipulated through the IMAP protocol.

This memo replaces [RFC2087](#) [4], but attempts to remain backwards compatible whenever possible.

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 memo.

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 [RFC2119](#) [5].

Other capitalised words are IMAP4 [2] keywords or keywords from this document.

2. Introduction and Overview

The QUOTA extension is present in any IMAP4rev1 server which advertises any CAPABILITY beginning with "QUOTA=".

The capability "QUOTA", with no "=", denotes a [RFC2087](#) [4] compliant server. Some commands and responses are not present in such servers, and clients MUST NOT rely on their presence in the absence of any capability beginning "QUOTA=".

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](#) [4] 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 IMAP4 [2]. These MUST be registered with IANA, or begin with "X-", which indicates an experimental resource. Implementation specific resources MUST be registered with IANA, and begin with "V-".

Supported resource names MUST be advertised as a capability, by prepending the resource name with "QUOTA=RES-". Server 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 no usage of a resource. Usage integers MUST NOT represent proportional use, such that a client can compare available resource between two separate quota roots or servers with reasonable accuracy.

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 consisting of the resource name prefixed by "QUOTA=RES-". For compatibility with RFC2087 [4], a client which discovers resources available on the server which are not advertised through this mechanism MUST treat the resource as if it were completely opaque, and without any meaning.

The resources STORAGE (Section 5.1), MESSAGES (Section 5.2) and MAILBOXES (Section 5.3) are defined in this memo.

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 [2]. It MUST be treated as an opaque string by any clients which do not have a priori knowledge of the server implementation.

Quota roots are used since not all implementations may be able to calculate usage, or apply quotas, on arbitrary mailboxes or mailbox hierarchies. A client might be able to determine how a quota root relates to the mailboxes it governs by looking at any mapping which MAY be given in a QUOTAMAP ([Section 4.2.3](#)) response.

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 MESSAGES ([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
OPTIONAL untagged response: SUPPORTEDQUOTARES

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. GETQUOTA command MAY also return an untagged SUPPORTEDQUOTARES response that lists all resource types that can be set on the quota root. If the SUPPORTEDQUOTARES response is not returned by the server, this means that all resource types returned in CAPABILITY response (i.e. all capability items with "QUOTA=RES-" prefix) are applicable to the quota root.

Example:

```
S: * CAPABILITY [...] QUOTA QUOTA=RES-STORAGE [...]
[...]
C: G0001 GETQUOTA "!partition/sda4"
S: * QUOTA "!partition/sda4" (STORAGE 104 10923847)
S: * SUPPORTEDQUOTARES "!partition/sda4" STORAGE
S: G0001 OK Getquota complete
```

4.1.2 GETQUOTAROOT

Arguments: mailbox name

Responses: REQUIRED untagged responses: QUOTAROOT, QUOTA
OPTIONAL untagged responses: QUOTAMAP

Result: OK - getquotaroot completed
NO - getquotaroot error: no such mailbox, permission denied
BAD - command unknown or arguments invalid

The GETQUOTAROOT command takes the name of a mailbox 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 and MAY return an untagged QUOTAMAP response that describes a relationship between the quota root and the mailbox (mapping).

//Should we remove some information from QUOTAMAP as it is already returned in QUOTAROOT?

Example:

```
S: * CAPABILITY [...] QUOTA QUOTA=RES-STORAGE QUOTA=RES-MESSAGES
[...]
[...]
C: G0002 GETQUOTAROOT INBOX
S: * QUOTAROOT INBOX "#user/alice" "!partition/sda4"
S: * QUOTAMAP "#user/alice" INBOX (USER)
S: * QUOTAMAP "!partition/sda4" INBOX ( )
S: * QUOTA "#user/alice" (MESSAGES 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.

// Should we be sending untagged QUOTA responses for all side effect changes?

// Quota root name must uniquely identifier mapping [if any]
 (different mapping must have non overlapping namespaces)

Example:

```
S: * CAPABILITY [...] QUOTA QUOTA=RES-STORAGE QUOTA=RES-MESSAGES
[...]
[...]
C: S0000 GETQUOTA "#user/alice"
S: * QUOTA "#user/alice" (STORAGE 54 111 MESSAGES 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 DELQUOTA

Arguments: quota root

resource name

Responses: no specific responses for this command

Result: OK - delquota completed
 NO - delquota error: can't delete that data
 BAD - command unknown or arguments invalid

The DELQUOTA command takes the name of a mailbox quota root and a resource name. The resource limit associated with the resource name is removed (or reset to the underlying system limit), or other resources associated with the same quote root are unaffected. (This command is different from "SETQUOTA <quota_root> (<resource_name> 0)", because the latter discards all resources associated with the quota root).

An implementation may optionally change the quota roots for any number of existing mailboxes in an implementation-defined manner.

// Should we be sending untagged QUOTA responses for all side effect changes?

Example:

```
S: * CAPABILITY [...] QUOTA QUOTA=RES-STORAGE QUOTA=RES-MESSAGES
[...]
[...]
C: S0001 DELQUOTA "#user/alice" STORAGE
S: * QUOTA "#user/alice" (MESSAGES 42 1000)
S: S0001 OK STORAGE limit removed.
C: S0002 DELQUOTA "!partition/sda4" STORAGE
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 remove system limit
```

[4.1.5 LISTQUOTA](#)

Arguments: quota root

OPTIONAL untagged responses: QUOTAMAP

Result: OK - listquota completed
 NO - listquota error: no such quota root, permission denied
 BAD - command unknown or arguments invalid

The LISTQUOTA command takes the name of a quota root and returns QUOTAMAP responses for all mailboxes accessible to the user that are governed by this quota root

//Client should be prepared to receive a lot of traffic, because this might be equivalent to performing <LIST "" *>

Example:

```
S: * CAPABILITY [...] QUOTA QUOTA=RES-STORAGE QUOTA=RES-MESSAGES
[...]
[...]
C: L0001 LISTQUOTA "#user/alice"
S: * QUOTAMAP "#user/alice" INBOX (USER)
```



```
S: * QUOTAMAP "#user/alice" "Drafts" (USER)
S: * QUOTAMAP "#user/alice" "Work/Meetings" (USER)
S: * QUOTAMAP "#user/alice" "Work/Projects" (USER)
S: L0001 OK
```

4.1.6 STATUS attribute RECOVERABLE

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 - Is it the sum of [RFC822](#).SIZEs or "How much we can recover" (depending on messages with \Deleted flag it may be different)

Example:

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

[4.2.3](#) QUOTAMAP

Data: quota root name
mailbox name
OPTIONAL mapping name

This response occurs as a result of a GETQUOTAROOT or LISTQUOTA command. It defines the relationship (mapping) between the quota root and the mailbox.

[4.2.4](#) SUPPORTEDQUOTARES

Data: quota root name
zero or more resource names

SUPPORTEDQUOTARES response occurs as a result of GETQUOTA command. It lists all resource types that can be set on the quota root. If the list of resources is missing (not empty!) this means that the server can't list supported resources and the client must try SETQUOTA. Note that SUPPORTEDQUOTARES with no resource name is different from the absent SUPPORTEDQUOTARES response. If the SUPPORTEDQUOTARES response is not returned by the server, this means that all resource types returned in CAPABILITY response (i.e. all capability items with "QUOTA=RES-" prefix) are applicable to the

quota root.

[4.3 Response Codes](#)

[4.3.1 OVERQUOTA](#)

OVERQUOTA response code is returned in NO tagged response to an APPEND/COPY when the addition of the message(s) puts 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
```

[4.4 Interaction with the ACL and ACL2 extensions](#)

Both the ACL [\[3\]](#) and ACL2 extensions define access control lists, and specific permissions which are required for certain actions.

```
// But how do they interact? Presumably, QUOTAMAP responses
containing mailboxes which cannot be LISTed shouldn't be generated.
Quota Roots which govern no mailboxes to which the client has write
access should also, presumably, be hidden from the client's view?
Administration rights to set quotas?
```


5. Resource Definitions

The following resources are defined in this memo. A server supporting a resource 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 resources 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. Additional messages MAY 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.

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](#) [4], 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.

5.2 MESSAGES

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-MESSAGES".

A resource named the same was also given as an example in [RFC2087](#) [4], clients conformant to this specification connecting to servers which do not advertise "QUOTA=RES-MESSAGES", yet allow a resource

named MESSAGES, MUST NOT assume that it is the same resource.

5.3 MAILBOXES

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-MAILBOXES".

6. Quota Root Relationship Definitions

Where a specific quota root relationship, or mapping, is given in a QUOTAMAP response, a client MAY make certain assumptions about which Quota Root, and therefore which Quota, will govern an existing or newly created mailbox, without having to use LISTQUOTAROOT after creation.

Implementations MAY provide no mapping information at all, either for security reasons or because the mapping actually used does not fit one of the defined mappings.

Relationship names are atoms, as defined in IMAP4 [2], and must be registered at IANA. Relationships which are implementation specific are of limited use for interoperability, however they MUST be registered and prefixed with "V-", along with the meaning of any parameters they list.

The mapping applicable to a particular quota root and mailbox is given in the QUOTAMAP ([Section 4.2.3](#)) response.

6.1 HIER

The quota root in question applies to all inferior mailboxes of the named mailbox, and a newly created inferior mailbox will be governed by the same quota root.

6.2 SINGLE

The quota root in question applies only to the named mailbox, and to no other. This is mutually exclusive with HIER mappings.

6.3 USER

The quota root in question applies to all mailboxes owned by the same user. The definition of ownership is implementation dependant.

// Do we restrict this to the currently logged in user?

6.4 DOMAIN

The quota root in question applies to all mailboxes in the current domain on at least this server. If the server doesn't support multiple domains, GLOBAL MUST be used instead.

6.5 GLOBAL

The quota root in question applies to all mailboxes on at least this

server.

7. Formal syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) notation as specified in ABNF [1].

Non-terminals referenced but not defined below are as defined by IMAP4 [2].

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" | "MESSAGES" | "MAILBOXES" |
resource_name_priv | resource_name_vnd | resource_name_ext
resource_name_priv = "X-" atom
                  ;; Private use
resource_name_vnd  = "V-" atom
                  ;; Vendor specific, must be registered with IANA
resource_name_ext  = atom
                  ;; Not starting with either X- or 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

quotamap_response = "QUOTAMAP" SP quota_root_name SP mailbox SP "("
[mapping] ")"

suppres_response = "SUPPORTEDQUOTARES" SP quota_root_name [SP
resource_names]

mapping = "HIER" | "SINGLE" | "USER" | "DOMAIN" |
"GLOBAL" | mapping_vendor | mapping_ext

mapping_vendor = "V-" atom

;; Vendor specific, must be registered with IANA

mapping_ext = atom

;; Must be defined by an Experimental or a Standard Track RFC

delquota = "DELQUOTA" SP quota_root_name SP resource_name

capability_quota = capa_quota_res | capa_quota_mapping

capa_quota_res = "QUOTA=RES-" resource_name

capa_quota_mapping?

listquota = "LISTQUOTA" SP quota_root_name

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

//Should this be optional unless the server implements MESSAGES/
STORAGE?

resp-text-code =/ "OVERQUOTA"

References

- [1] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.
- [2] Crispin, M., "Internet Message Access Protocol - Version 4rev1", [RFC 2060](#), December 1996.
- [3] Myers, J., "IMAP4 ACL extension", [RFC 2086](#), January 1997.
- [4] Myers, J., "IMAP4 QUOTA extension", [RFC 2087](#), January 1997.
- [5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

Authors' Addresses

Dave A. Cridland
Clues Ltd

E-Mail: dave.cridland@clues.ltd.uk

URI: <http://www.clues.ltd.uk/>

Alexey Melnikov
ACI WorldWide / MessagingDirect

E-Mail: mel@messagingdirect.com

URI: <http://orthanc.ab.ca/mel/>

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgement

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

