B. Korver
Xythos
L. Dusseault
Xythos
C. Warner
Netezza

Internet Draft
Document: draft-ietf-webdav-quota-02.txt
Expires: January 2004

Quota and Size Properties for DAV Collections

Status of this Memo

Abstract

    WebDAV servers are frequently deployed with quota (size)
    limitations. This Internet-Draft discusses the properties and minor
    behaviors needed for clients to interoperate with quota
    implementations on WebDAV repositories.

Table of Contents

Introduction

    Notational Conventions

    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](#)].

    Requirement for quotas

    WebDAV servers based on [[RFC2518](#)] have been implemented and deployed
    with quota restrictions on collections and users, so it makes sense
    to standardize this functionality to improve user experience and
    client interoperability. This specification requires WebDAV because
    it requires PROPFIND support and relies on the WebDAV definition of
    collections and properties, including the definitions for live and
    protected properties.

    The reasons why WebDAV servers frequently have quotas enforced are
    the same reasons why any storage system comes with quotas.

     - Sometimes the storage service charges according to quota

     - Sometimes the storage service is provided free, but the storage
    service provider has limited storage space (e.g. www.sharemation.com
    and university-provided student accounts)

     - Even in cases where the storage can be upgraded, the storage
    managers may choose to limit quota in order to encourage users to
    limit the files they store on the system and to clean up obsolete
    files. (e.g. IT departments within corporations).

    In order to work best with repositories that support quotas, client
    software should be able to determine and display the quota-available
    on collections. Further, client software should have some way of
    fairly reliably determining how much storage space is already
    counted towards that quota.

    In addition to displaying the quota-available and quota-used on
    collections, this specification does not forbid these properties on

any resource.

Solution Overview

The approach to meeting the requirements and scenarios outlined
above is to define three live properties. This specification can be

met on a server by implementing both quota-available and quota-used
on collections only. Implementing both quota-available and quota-
used on all resources is RECOMMENDED.

None of these properties need be returned in a <DAV:allprop> request
though the server may include them. However, these property names
MUST be returned in a <DAV:propname> request for a resource that
supports the properties, except in the case of infinite limits which
are explained below.

The quota-available and quota-used definitions below borrow heavily
from the quota definitions in the NFS [RFC3010] specification.

DAV:quota-available-bytes

    Name: quota-available-bytes
    Namespace: DAV:
    Purpose: Indicates the maximum amount of additional storage
    available to be allocated to a resource.
    DTD: <!ELEMENT quota-available-bytes (#PCDATA) >

    The DAV:quota-available-bytes property value is the value in octets
    representing the amount of additional disk space beyond the current
    allocation that can be allocated to this file or directory before
    further allocations will be refused. It is understood that this
    space may be consumed by allocations to other files or directories.

    Support for this property is REQUIRED on collections, and OPTIONAL
    on other resources. A server SHOULD implement this property for
    each resource that has the DAV:quota-used-bytes property.

    Clients SHOULD expect that as the quota-available on a file or
    directory approaches 0, further allocations to that file or
    directory may be refused. A value of 0 indicates that users will
    probably not be able to perform operations that write additional
    information (e.g. a PUT inside a collection), but may be able to
    replace through overwrite an existing resource of equal size.

    Note that there may be a number of distinct but overlapping limits,
    which may even include physical media limits. When reporting quota-

available, the server is at liberty to choose any of those limits
but SHOULD do so in a repeatable way. The rule may be configured
per repository, or may be choose the smallest number .

If a resource has no quota enforced or unlimited storage (infinite
limits), the server MAY choose not to return this property (404 Not
Found response in Multi-Status), although this specification
RECOMMENDS that servers return some appropriate value (e.g. the
amount of free disc space). A client cannot entirely assume that
there is no quota enforced on a resource that does not have this
property, but might as well act as if there is no quota.

The value of this property is protected. A 403 Forbidden response
is RECOMMENDED for attempts to write a protected property.

DAV:quota-used-bytes

Name: quota-used-bytes
Namespace: DAV:
Purpose: Contains the amount of storage counted against the quota on
a resource.
DTD: <!ELEMENT quota-used-bytes (#PCDATA) >

The DAV:quota-used-bytes value is the value in octets representing
the amount of space used by this file or directory and possibly a
number of other similar files or directories, where the set of
 similar meets at least the criterion that allocating space to any
file or directory in the set will count against the quota-available.
It MUST include the total count including usage derived from sub-
resources if appropriate. It SHOULD include metadata storage size
if metadata storage is counted against the quota-available.

Note that there may be a number of distinct but overlapping sets of
files or directories for which a quota-used is maintained (e.g. all
files with a given owner , all files with a given group owner ,
etc.). The server is at liberty to choose any of those sets but
SHOULD do so in a repeatable way. The rule may be configured per
repository.

Support for this property is REQUIRED on collections, and OPTIONAL
on other resources. A server SHOULD implement this property for
each resource that has the DAV:quota-available-bytes property.

Support for this property enhances the client experience, because
together with DAV:quota-available-bytes, the client has a chance of

managing its files to avoid running out of allocated storage space.
Clients may not be able to calculate the value as accurately on
their own, depending on how total space used is calculated by the
server.

DAV:quota-assigned-bytes

    Name: quota-assigned-bytes
    Namespace: DAV:
    Purpose: Indicates the amount of storage assigned.
    DTD: <!ELEMENT quota-bytes (#PCDATA) >

    The DAV:quota-assigned-bytes property value is the amount of storage
    space potentially either assigned to or requested for this file or
    directory, measured in octets. DAV:quota-assigned-bytes is primarily
    intended to support implementations that allow quota to be
    PROPPATCHed or configured by some other means.

    The value of this property will usually be protected, although a
    user with sufficient privileges may be permitted to change the

    value. The property is useful even if it is protected. A 403
    Forbidden response is RECOMMENDED for attempts to write a protected
    property. This property will usually be read-only for instance
    because a user s quota should generally not configurable by other
    users.

    Support for this property is OPTIONAL.

    Note that a resource may show more quota-used than its quota-
    assigned appears to allow, and that quota-available MUST never be
    greater than the value of quota-assigned.

    As an example of quota-assigned, imagine a quota system where each
    collection may have a quota assigned and where a resource contained
    in a collection is subject to the quota constraints of all parent
    collections. Assume the administrator creates a collection A and
    gives it a quota-assigned of 500KB and then creates a sub-
    collections B which is given quota-assigned of 1000KB. In this
    case, the initial quota-available for B is 500KB, not 1000KB, since
    the constraint on A applies to B as well.

	DAV:quota-assigned-bytes	DAV:quota-available-bytes
/A	500KB	500KB
/A/B	1000KB	500KB

    Note that this is only one example quota system, and that other

quota systems are possible.

Example PROPFIND request and response

    Request:

      PROPFIND /~milele/public/ HTTP/1.1
      Depth: 0
      Host: www.sharemation.com
      Content-Type: text/xml
      Content-Length: xxx

      <?xml version="1.0" ?>
      <D:propfind xmlns:D="DAV:">
      <D:prop><D:quota-available-bytes><D:quota-used-bytes></D:prop>
      </D:propfind>

    Response:

      HTTP/1.1 207 Multi-Status
      Date: Tue, 16 Oct 2001 22:13:39 GMT
      Content-Length: xxx
      Content-Type: text/xml; charset=UTF-8

      <?xml version="1.0" encoding="utf-8" ?>
      <D:multistatus xmlns:D="DAV:">
      <D:response>

        <D:href>http://www.sharemation.com/~milele/public/</D:href>
        <D:propstat>
          <D:prop>
            <D:quota-available-bytes>596650</D:quota-available-bytes>
            <D:quota-used-bytes>403350</quota-used-bytes>
          </D:prop>
          <D:status>HTTP/1.1 200 OK</D:status>
        </D:propstat>
      </D:response>
      </D:multistatus>

Error reporting

    WebDAV (RFC2518) defines the status code 507 (Insufficient Storage).
    This status code SHOULD be used when a client request (e.g. a PUT,
    PROPFIND, MKCOL, MOVE or COPY) is forbidden because it would exceed
    their allotted quota. In order to differentiate the response from
    other storage problems, the server SHOULD include an XML error body
    as defined by DeltaV [RFC3253] with the <DAV:storage-quota-reached/>

precondition tag.

   Example error response:

   HTTP/1.1 507 Insufficient Storage
   Content-Length: 100
   Content-Type: text/xml

   <?xml version= 1.0 >
   <error xmlns= DAV: >
   </error>

Notes

   Server implementations store and account for their data in many
   different ways. Some of the challenges:

    - Some server implementations find it prohibitive to count storage
   used for metadata, others may choose to do so for better accounting.

    - Older versions of resources may be stored as well.

    - Variants of one resource may exist with different content lengths

    - Content may be dynamically generated.

    - Resource bodies can be compressed

    - Some resources may be stored for free , not counting against
   quota.

   Since server storage accounting can vary so much, clients should
   expect the following:

    - The size of a file on the client s file system, or in a PUT
   message, may not correspond to the amount of storage required by the
   server to store the resource. Thus, the client cannot predict with
   100% accuracy whether a given file will be allowed given the storage
   quota.

    - Deleting or overwriting a resource may not free up the same
   amount of storage as indicated by the DAV:getcontentlength property
   defined in [RFC2518] for the resource. If deleting a resource does
   not free up any space, the file may have been moved to a trash
   folder or recycle bin , or retained as in versioning systems

[RFC3253].

     - The total size of a collection, DAV:quota-used-bytes, is not
    necessarily a sum of the DAV:getcontentlength properties for
    resources stored in the collection.

     - On some systems where quota is counted by collection and not by
    user, a quota on a sub-collection may be larger than the quota on
    its parent collection that contains it. For example, the quota on
    /~milele/ may be 100 MB, but the quota on /~milele/public/ may be
    unlimited. This allows the space used by /~milele/public/ to be as
    large as the quota on /~milele/ allows (depending on the other
    contents of /~milele/) even if the quota on /~milele/ is changed.
    Thus, even when the quota on a parent collection is changed, it is
    not necessarily required to change the quota on every child or
    descendant collection.

Security Considerations

    A hacker may prefer to store files in collections with a large
    quota. This isn't strictly a security concern because it doesn't
    make it any easier to store files. On the other hand, the
    DAV:quota-used-bytes property may make it easier to detect tampering
    or misuse.

    If a server chooses to make the DAV:quota-assigned-bytes writable by
    clients with sufficient authorization, then it is opening up a
    certain amount of near-administration functionality to clients.
    However, it is not required for the DAV:quota-assigned-bytes
    property to be writeable by any clients, so a server can easily
    avoid this consideration.

Internationalization Considerations

    Quota is counted in Arabic numerals expressed in strings. There are
    no internationalization considerations.

IANA Considerations

    There are no IANA considerations.

Intellectual Property

    The following notice is copied from [RFC2026], and describes the
    position of the IETF concerning intellectual property claims made

against this document.

The IETF takes no position regarding the validity or scope of any
intellectual property 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; neither does it represent that it
has made any effort to identify any such rights. Information on the
IETF's procedures with respect to rights in standards-track and
standards-related documentation can be found in BCP-11. Copies of
claims of rights made available for publication 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 implementors or users of this specification
can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any
copyrights, patents or patent applications, or other proprietary
rights which may cover technology that may be required to practice
this standard. Please address the information to the IETF Executive
Director.

Acknowledgements

References

    [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
        Requirement Levels", BCP 14, RFC 2119, March 1997.

    [RFC2026] Bradner, S., The Internet Standards Process ± Revision
        3 , BCP 9, RFC2026, October 1996.

    [RFC2518] Goland, Y., Whitehead, E., Faizi, A., Carter, S., and
        Jensen, D., "HTTP Extensions for Distributed Authoring --
        WebDAV", RFC2518, February 1999.

    [RFC3010] Shepler S., B. Callaghan, D. Robinson, R. Thurlow, C.
        Beame, M. Eisler, D. Noveck,   NFS version 4 Protocol , RFC3010,
        December 2000.

Author's Addresses

    Brian Korver
    Xythos Software, Inc.
    25 Maiden Lane, Suite 200   Phone: 1-415-248-9033
    San Francisco, CA, USA   Email: briank@xythos.com

    Lisa Dusseault
    Xythos Software, Inc.
    25 Maiden Lane, Suite 200   Phone: 1-415-248-9004
    San Francisco, CA, USA   Email: lisa@xythos.com

    Clark Warner
    Netezza Corporation
    200 Crossing Blvd.   Phone: 1-508-665-6800 x889
    Framingham, MA 01702   Email: webdav@thewarners.com

-brian
briank@xythos.com