

NFSv4
Internet-Draft
Intended status: Standards Track
Expires: February 26, 2007

A. Yoder
Network Appliance
D. Black
EMC
August 25, 2006

NFSv4.1 Retention Attributes
draft-yoder-nfsv4-retention-00.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-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 26, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This Internet-Draft describes additional file attributes to be used in NFSv4.1 for data retention semantics.

Requirements Language

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](#) [1].

Table of Contents

1.	NFSv4.1 Data Retention Attributes	3
1.1.	int64 rtime	3
1.2.	int64 stime	3
1.3.	int64 ertime	3
1.4.	int64 etime	4
1.5.	int8 retention_flags	4
1.6.	int8 admin_hold_flags	5
1.7.	int32 srvr_ret_capabilities_flags	5
2.	Discussion	5
3.	Further Notes	5
4.	Security Considerations	6
5.	Normative References	6
	Authors' Addresses	6
	Intellectual Property and Copyright Statements	8

[1.](#) NFSv4.1 Data Retention Attributes

The FCAS TWG of SNIA met on 1 Aug 2006 and had an extensive discussion regarding alignment of retention models between XAM and NFSv4.1. While there are some areas in which the two standards are unlikely to meet exactly, the consensus is that the following attributes are sufficient to express the retention model under consideration for XAM v1. Mike Eisler and Dave Noveck contributed to the formulation presented here.

[1.1.](#) uint64_t rtime

The minimum retention duration for a file, in seconds.

Rtime is expressed as a duration, as it may be set before the initial timestamp which delimits the beginning of the retention interval is known.

Rtime MUST have a value of zero upon file creation. It is ignored until RET4_SIMPLE_RETENTION_ENABLED is set. The rtime attribute MAY be increased by a client-side SETATTR operation. The server MUST NOT allow decrease of this quantity under any circumstances.

Permission to set and increase rtime is governed by a new NFSv4 bitmask ACE4_WRITE_RETENTION (0x00000200). Servers MAY map this to ACE4_WRITE_ATTRIBUTES (0x00000100).

[1.2.](#) nfstime4 rbtime

The timestamp specifying the beginning of the retention period. If

rbtime is set, the server MUST NOT allow deletion of or change to a file before the time given by rtime + stime.

When the client first sets the rbtime to any value, via SETATTR, the server MUST set it to the current time as perceived by the server, and atomically also set the RET4_SIMPLE_RETENTION_ENABLED flag. Once set, the server MUST NOT permit change to the rbtime or the RET4_SIMPLE_RETENTION_ENABLED flag.

Rbtime MUST have a value of zero upon file creation.

Permission to "set" rbtime is governed by ACE4_WRITE_RETENTION.

[1.3.](#) uint64_t etime

The minimum retention duration for a file, in seconds, following an application-specified event. The etime MAY be increased by a client-side SETATTR operation. The server MUST NOT allow decrease of this quantity under any circumstances.

Etime MUST have an initial value of zero. It is ignored until RET4_EVENT_RETENTION_ENABLED is set. Clients MAY set the flag

RET4_EVENT_RETENTION_ENABLED at any time.

Permission to set and increase etime is governed by ACE4_WRITE_RETENTION.

Clients SHOULD set the etime before setting the RET4_EVENT_RETENTION_ENABLED flag, to avoid windows of mutability.

[1.4.](#) nfstime4 ebtime

The timestamp specifying the beginning of the event-based retention period. If ebtime is set, the server MUST NOT allow deletion of or change to a file before the time given by ebtime + etime.

Setting the ebtime is the equivalent to an event notification operation in XAM.

Ebtime MUST have a value of zero upon file creation. This indicates that no application-specified event has yet taken place.

When the client first sets the ebtime to any value, via SETATTR, the server MUST set it to the current time as perceived by the server, and atomically also set the RET4_EVENT_RETENTION_ENABLED flag. Once set, the server MUST NOT permit change to the ebtime or the RET4_EVENT_RETENTION_ENABLED flag.

Permission to "set" ebtime is governed by ACE4_WRITE_RETENTION.

[1.5.](#) uint32_t retention_flags

Flags are as follows:

RET4_SIMPLE_RETENTION_ENABLED = 0x01

RET4_EVENT_RETENTION_ENABLED = 0x02

RET4_SIMPLE_RETENTION_INFINITE = 0x04

RET4_EVENT_RETENTION_INFINITE = 0x08

If RET4_SIMPLE_RETENTION_ENABLED is set, the server MUST do retention based on the corresponding set of attributes (rtime and stime for RET4_SIMPLE_RETENTION_ENABLED, etime and ertime for RET4_EVENT_RETENTION_ENABLED).

The server MUST NOT turn off any of these flags, once set.

RET4_SIMPLE_RETENTION_INFINITE and RET4_EVENT_RETENTION_INFINITE override all retention timestamps. The server MUST NOT allow deletion or modification of the file as long as either of these bits are set.

The server MUST NOT allow unsetting of either of these bits.

Permission to set retention_flags is governed by ACE4_WRITE_RETENTION.

[1.6.](#) uint32_t admin_hold_flags

These 32 flags are used to indicate administrative holds (e.g. holds placed by judicial fiat). These override all other retention expiration mechanisms. The server MUST NOT allow deletion or

modification of the file as long as any of the bits in `admin_hold_flags` are set. Bits MAY be unset by clients with sufficient permission.

Permission to set `admin_hold_flags` is governed by a new NFSv4 bitmask `ACE4_WRITE_HOLD` (0x00000400). Servers SHOULD NOT map this to `ACE4_WRITE_ATTRIBUTES` (0x00000100).

`Admin_hold_flags` MUST have a value of zero upon file creation.

[1.7](#) `uint32_t` `svr_ret_capabilities_flags`

`RET4_SIMPLE_RETENTION` = 0x01

`RET4_EVENT_RETENTION` = 0x02

`RET4_SIMPLE_RETENTION_INFINITE` = 0x04

`RET4_EVENT_RETENTION_INFINITE` = 0x08

`RET4_ADMIN_HOLD` = 0x10

Support for retention attributes is optional. If implemented, the server MUST indicate support for simple retention, event-based retention, infinite retention and administrative holds by setting the respective bit in `svr_ret_capabilities_flags` in a GETATTR response. If the server does not support any retention attributes, it MAY return `NFS4ERR_ATTRNOTSUPP` in response to a GETATTR of `svr_ret_capabilities_flags`.

The server MUST disallow any attempts to perform a SETATTR on `svr_ret_capabilities_flags`.

[2.](#) Discussion

Support for any or all retention attributes is optional. If implemented, they MAY be implemented on a per-filesystem basis. If they are, the server MAY implement a different set of attributes, and return a corresponding `svr_ret_capabilities_flags` instance, for each filesystem.

On filesystems that do not support a given retention attribute, NFS4ERR_ATTRNOTSUPP shall be returned in response to all GETATTR and SETATTR calls for that attribute.

There are numerous interdependencies between flags which introduce requirements on which combinations of them must be supported if any are:

If the server supports rtime, then it MUST support rbtime and vice versa.

If the server supports etime, then it MUST support ebtime and vice versa.

If the server supports rtime, rbtime, etime, or ebtime, it MUST support retention_flags.

If the server supports any retention flags, it MUST also support srvr_ret_capabilities_flags.

If srvr_ret_capabilities_flags indicate simple retention, the server MUST support rtime and rbtime.

If srvr_ret_capabilities_flags indicate event retention, the server MUST support etime and ebtime.

If srvr_ret_capabilities_flags indicate ADMIN_HOLD, the server MUST support admin_hold_flags.

[3.](#) Further Notes

Notes

If both simple and event-based deletion are active on a file, as indicated by set RET4_SIMPLE_RETENTION_ENABLED and RET4_EVENT_RETENTION_ENABLED flags, deletion MUST NOT be allowed until both retention intervals have expired.

The TWG considered a delete-after datestamp instead of the datestamp/interval mechanism presented here. This can solve problems related to the variable number of seconds in durations such as months. An early reviewer from the NFSv4 WG also expressed a preference for this style. However, the etime cannot be expressed this way, as it may be set before the ebtime. An example is a requirement that a hospital record be kept three years after the death of the patient. In this case, etime would be three years' worth of seconds, and RET4_EVENT_RETENTION_ENABLED would be set, but ebtime would be kept unset until notification of the death of the patient.

Internet-Draft

NFSv4.1 Retention Attributes

August 2006

The TWG also considered allowing `ebtime` to be set in the past. In the above example, one might like to set `ebtime` to the actual time of death. However, this capability introduces a hole--a rogue client could set `ebtime` to Jan 1, 1970 and proceed to delete the file if `etime` were less than 36 years (as of 2006).

An early reviewer also questioned the need for `RET4_SIMPLE_RETENTION_INFINITE` and `RET4_EVENT_RETENTION_INFINITE`, saying that `0x7FFFFFFFFFFFFFFF` seconds is long enough. Several application writers in the TWG objected to this use of magic constants, as they already have too many of them and fear conflicts.

Clients SHOULD use extreme care in setting either `RET4_SIMPLE_RETENTION_INFINITE` or `RET4_EVENT_RETENTION_INFINITE`, as setting these cannot be undone, meaning the files will never be deletable or updateable under the retention rules.

[4.](#) Security Considerations

TBD

[5.](#) Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

Authors' Addresses

Alan Yoder
Network Appliance, Inc.
495 E Java Drive
Sunnyvale, CA 94089

Phone: +1-408-822-6919
Email: agy@netapp.com

David L. Black
EMC Corporation

176 South Street
Hopkinton, MA 01748
USA

Phone: +1-978-253-0937
Email: black_david@emc.com

Yoder & Black

Expires February 26, 2007

[Page 7]

Internet-Draft

NFSv4.1 Retention Attributes

August 2006

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of

such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

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