



NFSv4 Interim WG Meeting Data Retention 2006-09

Mike Eisler

email2mre-ietf@yahoo.com

For Alan Yoder and David Black

- ▶ **New regulations regarding rules for retaining data for years beyond point of creation or last change**
- ▶ **Harder to comply with regulations without technology to provide “mistake proofing”**
- ▶ **Several Data Retention products out there today:**
 - EMC Centera
 - NetApp SnapLock
- ▶ **Products effectively implement Write Once, Read Many semantics over read/write magnetic disks**

- ▶ **Alan Yoder and David Black have made a formal proposal based on related work and discussion in SNIASNIA**
 - `draft-yoder-nfsv4-retention-00.txt`
- ▶ **Appears to model the SNIA Data Retention APIs**
- ▶ **Impact on NFSv4.1 (next slide)**

- ▶ **7 new attributes**
 - **rtime** – duration of retention
 - **rbtime** – when retention starts
 - **rbtime + rtime** is when the file “expires”
 - **etime** – like rtime for “event based” retention
 - **ebtime** – when event based retention starts
 - **retention_flags**
 - **RET4_SIMPLE_RETENTION_ENABLED**
 - **RET4_EVENT_RETENTION_ENABLED**
 - **RET4_SIMPLE_RETENTION_INFINITE**
 - **RET4_EVENT_RETENTION_INFINITE**
 - **admin_hold_flags** – indicates an injunction is in effect that override expiration of a file
 - **svr_ret_capabilities_flags**
 - **RET4_SIMPLE_RETENTION**
 - **RET4_EVENT_RETENTION**
 - **RET4_SIMPLE_RETENTION_INFINITE**
 - **RET4_EVENT_RETENTION_INFINITE**
 - **RET4_ADMIN_HOLD**
- ▶ **New ACL permission: ACE4_WRITE_RETENTION**
 - Provides permission to set and change any of the first 6 attributes
- ▶ **Recent discussion within SNIA is that the permissions for enabling retention and setting retention duration ought to be different than the permissions for setting holds**

- ▶ **References to XAM, which is not yet publicly available**
 - WG chairs in both SNIA (David Black) and IETF (Spencer Shepler) agree that IETF NFSv4 RFC must be independent of SNIA XAM spec.
- ▶ **Retention needs better definition**
 - e.g. does it include file contents and name, what about named attributes?
 - SNIA defines retention on an XSET which contains data plus “fixed” properties. It is permissible to define an XSET plus some named attributes.
 - SNIA leaves file name lock down as a TBD, as to be defined by the NFSv4.1 protocol
 - XAM uses UUIDs not filenames, and no one is proposing adding UUIDs as alternate namespace for NFS (right?)
- ▶ **Are these enforced on the client or server side?**
 - The server is expected to retain the file independent of what a potentially-clueless client tries to do

- ▶ I-D mandates that the defaults for `rbtime` and `ebtime` be zero. Begin times of January 1, 1970 look weird. I-D also mandates that begin time be set to the current time of day when the retention is enabled.

- ▶ Simple and event based retention could get the attributes via:

```
struct retention_attr_get4 {
    uint64_t rag_duration;
    nfstime4 rag_begin_time<1>; /* if zero length, begin time not enabled
    */
};
```

and set via:

```
struct retention_attr_set4 {
    bool   ras_enable;
    uint64_t ras_duration<1>; /* if zero length, not changing duration */
};
```

- ▶ This way, it's the client's API problem to report a begin time when retention not yet enabled.

▶ **I-D notes:**

“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.”

- ▶ **The needs of the API are different from the needs of the protocol.**
 - The protocol needs to be simple.
 - Fewer attributes, and fewer linkages between attributes is less error prone
 - As has been discussed, the NFSv4.1 WG doesn't get to see the APIs until SNIA sees fit to release them, so API can't be a prime concern for the WG
- ▶ **NFSv4.1 should not support `_INFINITE` flags as explicit attributes, and instead should reserve a value of `0xFFFFFFFFFFFFFFFF` to mean infinite duration**
 - The API on the client can map this how it sees fit (e.g. map the magic constant to the APIs `_INFINITE` flag)
 - `retention_flags` can now be deleted

- ▶ **srvr_ret_capabilities_flags** is replicating the function of **supp_attr**
 - So drop **srvr_ret_capabilities_flags**
- ▶ **Summary of retention after proposed changes to the I-D**
 - Two “get only” attributes for retention times
 - Two “set only” attributes for retention times
 - One **admin_hold_flags** attribute
 - Effectively reduces 7 attributes to 3
 - One (or two) permission bits in an ACE