

# Moving Forward on NFSv4 Extension Issues

*Dave Noveck, Tom Haynes*

*NFSv4 Working Group Meeting at IETF89*

*March 5, 2014*

# Situation Overview

NFSv4.0 and MV one took a long time.

- But they were big changes with big documents

MV two was small (and had a small document) and still took a very long time

- More than three years from -01
- More than five years from -01 for server-side copy

The working group has a problem it must address

- Reform the minor versioning process
- Get rid of minor versioning (and only have freestanding extensions)
- Something else?

# Talk Overview (in terms of documents)

## draft-dnoveck-nfs-extension

- Quick Review of -00
- Needs an updated -01 soon (will expire 3/15)
- Anything else needed?

## Anticipated documents:

### - draft-ietf-nfsv4-extension

- Needs to apply to all minor versions
- Details TBD, subject to wg consensus

### - draft-id-nfsv4-extension

- Tom and Dave currently working on a individual draft (see below)
- Need to provide a framework for discussion as wg considers scope of proposed working group document

# draft-dnoveck-nfs-extension-00

An analysis of protocol extension for NFS as a whole

- XDR replacement (major versioning) for v2 → v3 → v4.0
- XDR extension (minor versioning) for v4.0 → v4.1 → v4.2

We could divide these transitions a different way

- Three implemented essentially a new protocol: v2 → v3 → v4.0 → v4.1
- One added a set of optional features to an existing protocol: v4.1 → v4.2
  - Which was the original purpose of minor versioning

Conclusions:

- Minor versioning handled a situation it wasn't designed for very well 😊
- Hasn't done so well with the situations it was intended to handle 😞
- The working group needs to improve the situation

# draft-dnoveck-nfs-extension-0x

-01 drafted and could be submitted soon

- Added the last six months of history
- More discussion of existing documents updating and obsoleting MV RFCs
  - Rfc3530bis
  - Rfc3530-migration-update
- Other suggestions?

Probably the last iteration of this document.

- Could conceivably be resurrected to reflect experience with a modified approach, if the working group adopts one.

# draft-id-nfsv4-extension

Tom and Dave working on a draft.

- Their preferred approach is to focus on reforming the minor version process
- Can still be affected by what people want to see.
- Also can comment after -00 is out.
- Or people can do their own drafts, taking a different approach to the problem

Draft's basic functions are:

- To provide a framework for discussing the shape of an eventual working group standards-track document
  - Document would cover NFSv4 protocol extension (including minor versioning)
- To suggest a path forward regarding other issues relating to NFSv4 protocol extension and minor versioning.

# Things to consider for wg stds-track document

## Core items (in I-D; need to be carried over into wg doc)

- Basics of XDR extension model
- New version of MV rules

## Likely items (in I-D; probably will be carried over into wg doc)

- Re-specify Minor Versions in terms of features (rather than protocol elements)
- Feature Discovery Mechanism
- XDR extension in documents updating/obsoleting MV RFCs
- Re-think feature status hierarchy
- Comprehensively address feature interaction issues

## Potential items (in I-D; may be carried over into wg doc)

- New minor version work flow
- Integrate pNFS Mapping types in Minor Version model

## Other items (not in I-D; might be brought into wg doc)

- New features outside minor version framework
- Feature negotiation

# Core items for draft-ietf-nfsv4-extension

## Basics of XDR extension model

- Based on draft-dnoveck-nfs-extension
- Added detail regarding:
  - Expected error returns
  - Constructs in XDR file but not properly part of the XDR protocol specification

## New version of MV rules

- Based on current rules in minorversion2 document
  - Simplified by being re-stated on the basis of the XDR extension model
- Necessary/desirable corrections discussed
  - To adjust to the fact that these rules will apply to *all* minor versions



# Likely item for draft-ietf-nfsv4-extension

## Re-specify Minor Versions in terms of features

### Distinguish “Features” and “Feature elements”

- Feature elements would include attributes, ops, flag bits, switch arms
- Features would be a set of “Feature elements” together with a behavior
- Mandatory and advisory byte-range locks could be separate features

### Features, rather than feature elements

- Would be designated required/recommended/optional (or the like)

# Likely item for draft-ietf-nfsv4-extension

## XDR extension in documents updating/obsoleting MV RFCs

Seems not to be allowed now

- Not clear why, given XDR extension model
- No explicit prohibition. Documents don't discuss the issue.
- Could be allowed in a few situations, but need appropriate restrictions
  - Makes sense to state such restrictions in wg standards-track document.

Some possible uses of such XDR extension:

- As a way to fix protocol bugs without micro-versioning
- As a way of backporting features of very limited scope (e.g. attributes).
  - Such features would be optional in target MV, even if mandatory in source

# Likely item for draft-ietf-nfsv4-extension

## Feature Discovery Mechanism

Could be accommodated by a per-fs `supp_features` attribute

- Would be mandatory in NFSv4.x (for  $x \geq 2$ ), and all later MV's
- Might be backported (as optional) in NFSv4.y (for  $y < x$ )
  - As specified in [previous slide](#).

Need way of assigning feature codes

- Features which consist of a single attribute could be handed by `supp_attributes`
- Codes for existing features would be in wg standards-track document
- New codes could be assigned by MV definition documents

# likely items for draft-ietf-nfsv4-extension

## Re-think designation hierarchy

- Figure out what “Recommended” means
  - Seems to be a synonym for “optional” only used for attributes.
- Do we need to add “Experimental”?
- Should implementation existence have a role?

## Comprehensively address feature interaction issues.

- Needed as we have more optional features
- Need to be clearer on:
  - When features are known to be compatible as opposed to not known to be incompatible
  - When a feature depends on another and when it depends on some particular allowed version of another feature.
- In many cases of MAYs and SHOULDs, different behaviors would be reported as different features, for feature discovery purposes.

# Potential items for draft-ietf-nfsv4-extension

## New minor version work flow

- Define optional features in their own standards-track documents
  - Wg could make exceptions for strongly interacting features
  - Each feature would have its own editor and be subject to wg and IESG review
  - Provides a more parallel and flexible process
- MV definition document would summarize features and normatively reference feature documents
- Specifying/mandating this might not be appropriate in a stds-track document

## Integrate pNFS Mapping types in Minor Version model

- We now have two separate axes for protocol extension
  - Provided an alternative to minor versioning
  - Might rethink in new environment
- Could integrate them by making each mapping type a feature

# Other items for draft-ietf-nfsv4-extension

New features outside minor version framework.

- Should only tackle if new minor version work flow does not provide sufficient flexibility.

## Feature negotiation

- Adds complexity
- Needs an I-D presenting a workable model before considering this for wg standards-track document
- One use is for non-support of non-mandatory callbacks. Some alternatives:
  - Support for all callbacks is mandatory, since saying NFS4\_OK is easy.
  - Support for callbacks may be optional but sever empowered to ignore NFS4ERR\_NOTSUPP.
  - All callback are feature elements within features and the client, by using the feature, has to support the associated callbacks.