

# Post WG LC NMDA datastore architecture draft

draft-ietf-netmod-revised-datastores-06  
(changes since -03)

Rob Wilton (Cisco), on behalf of NMDA authors

[rwilton@cisco.com](mailto:rwilton@cisco.com)

IETF 100, Singapore, Netmod WG

# Contents

1. WG LC summary
2. Summary of all changes since -03 version  
(Includes changes due to the WG LC)
3. Details on more significant changes
4. Recent Issues

# WG LC summary

- 18 issues raised/tracked on:  
<https://github.com/netmod-wg/datastore-dt/issues>
- 16 issues closed with resolution text sent to submitter and WG alias.
- 2 issues from WG LC listed as open:
  - RFC 2119 language – just waiting for reviewers to confirm
  - Actions and RPCs ...

# Summary of changes (since -03)

1. New objectives section added.
2. Updated to use RFC 2119 language.
3. Origin meta data is restricted to “config true” subset of <operational>.
4. Definition of “configuration transformations”.
5. Defined “datastore schema” and clarified relationship between <running>, <intended>, and <operational>.
6. Clarified behaviour of Actions/RPC operations (open issue).

# (3) Origin metadata

- Previously applied to all contents of <operational>.
- Now only applies to “config true” subset of <operational>.
- Three reasons why:
  1. It is hard to define origin for “config false” nodes.
  2. Difficult to implement.
  3. We want a simple efficient encoding:  
I.e. if not explicitly specified, the origin of a data node defaults to the parent node’s origin.

# (4) Configuration transformations

- New definition added:
  - **configuration transformation:** The addition, modification or removal of configuration between the <running> and <intended> datastores. Examples of configuration transformations include the removal of inactive configuration and the configuration produced through the expansion of templates.
- <intended> is defined as being after all configuration transformations have been processed to <running>.
- “Inactive configuration” and “templating” are now only used as examples – i.e. non normative text.



# (5) Datastore schema & conformance

- New definition added:
  - **datastore schema: The combined set of schema nodes for all modules supported by a particular datastore, taking into consideration any deviations and enabled features for that datastore.**
- All conventional datastores **MUST** have the same ds schema
- **<operational>** ds schema is a superset of conventional, but nodes may be not supported by deviation.
- Note, RFC 7950 does not actually define the term “schema”:
  - Schema mount is facing similar issues.
  - **Should we resolve try and resolve these definitions quickly and include them in the datastore draft?**



# (5) Updated datastore definitions

## **<running>:**

- MAY include configuration that requires further transformation before it can be applied,
- is defined as always being valid,
- Whenever <running> is updated, then so is intended.

## **<intended>:**

- Is after all configuration transformations
- <intended> is always valid
- May change independently of running if/when a configuration transformation changes.
- The contents of <intended> are also related to the "config true" subset of <operational>.

# (5) Updated datastore definitions

## <operational>:

- Schema is a superset of all configuration datastores, except deviations 'delete's.
- Defines "in use", avoid returning irrelevant state (such as protocols not configured).
- Semantic constraints MAY be violated (including list keys).
- Syntactic constraints MUST NOT be violated

# Actions/RPCs

The problem is which datastore is used to:

1. Evaluate action ancestor nodes
2. Evaluate action input/output parameter leafref, instance-identifier, must, when
3. Evaluate rpc input/output parameter leafref, instance-identifier, must, when

**Not related to what the action/RPC actually does, which is unconstrained.**

# Actions/RPCs (2)

Proposed solution:

- Always use <operational> for 1, 2.

In future (if required):

- Could extend protocols, and perhaps YANG, to allow Actions/RPCs to be targeted to other datastores (which would indicate which datastore any parameters are evaluated against).

# Next Steps

Get this draft to complete WG LC ASAP