NMDA Netconf drafts

draft-ietf-netconf-nmda-netconf-01
draft-ietf-netconf-nmda-restconf-01
draft-ietf-netconf-rfc7895bis-02

Rob Wilton (Cisco), on behalf of NMDA authors
rwilton@cisco.com

IETF 100, Singapore, Netconf WG
Contents

1. Context recap
2. Summary of updates to drafts
3. Discussion of open issues
Context Recap

• Updates to NETCONF, RESTCONF, and YANG library to support NMDA datastore architecture
• Protocol updates are minimal extensions only
• Equivalent new functionality to both protocols
• YANG library changes possibly more extensive (e.g. to extend to schema mount)
• We want these drafts completed quickly
• Aiming for WG LC before next IETF
Summary of changes

1. Clarified origin meta-data encoding, and aligned the text between the two drafts

2. Clarified that testing for NMDA is achieved by querying the new YANG library using `<get-data>` on `<operational>`
   Note: they have been a request to also express this as a protocol capability, but it unclear if this is necessary or beneficial

3. RESTCONF, clarified that with-defaults does not apply to `<operational>`

4. NETCONF, added query filters based on origin and “config true”
Next steps on drafts

1. Align structure and descriptive text between the two protocol drafts
2. Close on conformance question and YANG library structure
3. WG LC
Open issues - Conformance

NMDA architecture states that:

• All conventional datastores have the same schema
• The schema for operational must be a superset of all configuration datatstores, but that data nodes may be omitted:

Hence allows:

• Features to be enabled in operational but not running (e.g. useful for router-id)
• Data nodes to be omitted from <operational> (e.g. useful for migration, accuracy or reported state)
Allowed schema differences between datastores

<running>/<intended>  <operational>

Schema node
Device level deviation
DS deviation (delete only)
Open issues - Conformance

The authors think that the current conformance is correct and required, but perhaps could be clarified in the draft.

Others have expressed a desire for simpler conformance with no per datastore features/deviations.
Open Issues - YANG Library Structure

Trying to get the structure of YANG library right:
• to cover the required functionality
• to be as simple as possible
• optimized for the ‘mainline’ case, i.e. modules supported in all datastores, without per datastore deviations.
Open Issues - YANG Library Structure

In the following 4 slides I’ll present:

1. the current YANG library (RFC 7895) tree diagram
2. the current YANG library bis tree diagram
3. a proposed simpler version of YANG library bis
4. option (3) modified to be also be easily extendable/reusable for schema mount.
YANG Library - RFC 7895 version

---ro modules-state
  +--ro module-set-id     string
  +--ro module* [name revision]
    +--ro name               yang:yang-identifier
    +--ro revision           union
    +--ro schema-url/namespace/submodule
    +--ro feature*           yang:yang-identifier
      +--ro deviation* [name revision]
        | +--ro name               yang:yang-identifier
        | +--ro revision           union
        +--ro conformance-type   enumeration
YANG Library – 7895bis-02

```text
+
  --ro yang-library
  +--ro modules
    |  +--ro module* [id]
    |     +--ro id string
    |     +--ro name yang:yang-identifier
    |     +--ro revision? revision-identifier
    |     +--ro feature* yang:yang-identifier
    |     +--ro deviation* [module]
    |        +--ro module -> ../../id
    |     +--ro conformance-type enumeration
    |     +--ro schema-url/namespace/submodule
    |        ...
  +--ro module-sets
    |  +--ro module-set* [id]
    |     +--ro id string
    |     +--ro module* -> ../../../modules/module/id
  +--ro datastores
    |  +--ro datastore* [name]
    |     +--ro name identityref
    |     +--ro module-set
    |         -> ../../../module-sets/module-set/id
  +--ro checksum string
```
YANG library – proposed

```ynd
+--ro yang-library
    +--ro modules
        |    +--ro module* [name]
        |         |    +--ro name
        |         |    +--ro revision?
        |         |    +--ro schema/namespace/submodules
        |         |    +--ro not-implemented-in*
        |         |    -> /yang-library/datastore/name
        |         |    +--ro feature* [name]
        |         |         |    +--ro name
        |         |         |    +--ro not-implemented-in*
        |         |    -> /yang-library/datastore/name
        |         |    +--ro deviation* -> ../name
        |         |    +--ro import-only-module* [name revision]
        |         |         +--ro name
        |         |         +--ro revision union
        |         |         +--ro schema-url/namespace/submodules
    +--ro datastore* [name]
        |    +--ro name
        |    +--ro checksum
```
YANG library – proposed with schema mount support

```text
+--ro yang-library
   +--ro schema* [name]
      |   +--ro name string
      |   +--ro checksum string
      |   +--ro module* [name]
      |       |   +--ro name yang:yang-identifier
      |       |   +--ro revision? yang:revision-identifier
      |       |   +-- ... as before ...
      |       |
      |       |   +-- schema-mount:mount-point* [label]
      |       |       +--ro label yang:yang-identifier
      |       |       +--ro config? boolean
      |       |       +--ro (schema-ref)
      |       |       ... 
      |       |
      |       +--ro import-only-module* [name revision]
      |       |   +-- ... as before ...
      |       |
      +--ro datastore* [name]
          +--ro name identityref
          +--ro checksum string
```