YANG Instance Data for Documenting Server Capabilities

Balazs Lengyel
Benoit Claise

draft-lengyel-netmod-yang-instance-data

2018-07-15
Documenting server capabilities

› Server capabilities are often readable via Netconf/Restconf
  – Ietf-yang-library: Modules, revisions, features, deviations, datastores
  – Ietf-netconf-monitoring: Netconf-capabilities, etc.
  – Alarms supported
  – YangPush on-change notification capabilities

› Most are set/fixed at implementation time and don’t change

› Server capabilities needed in implementation-time
  – To start the NMS implementation in parallel
  – Operator buying decision may depend on the OAM capabilities,

› Standard formal YANG based form of documentation preferred
  – SW / automation needs to read capabilities
  – Makes capabilities available both via Netconf and via documentation
    › Same format on Netconf and documents
  – Encourages easy to read uniform documentation
Preloading Data

› Define sets of default data for configuration
  - Default Access control groups
    › readOnlyUser, systemAdministrator, securityAdministrator
  - Default rule-lists, rules for the default groups

› Default monitoring routines for a node

› Document/Load these in YANG instance-data-sets
› May later be modified
Instance Data Format

- Simple – Nothing new
- XML & JSON format
  - (later maybe CBOR)
- Based on GET reply formats – already defined
- Add metadata (name, revision, description)
- Follows YANG modules that define the relevant datastore parts (target YAMs)
  - Allow partial data sets
- May contain configuration and/or state data
Changes

› Recommendation to document server capabilities changed to be just the primary use-case
  – It is a very important message. Should we really exclude it?
› Redefined using yang-data-ext
› Moved meta data into ordinary leafs/lists
  – Allows more precise definition of metadata
› Clarified encoding
  – XML declaration
  – UTF-8 encoding
  – We follow RFC 7950/7951 rules
› In case of conflict live data from Netconf/Restconf has precedence over instance data
Open Issues 1

▶ Shall we recommend as a general practice: servers SHOULD document their capabilities using instance data
  ▶ Proposal: Yes
    ▶ It is an important recommendation that might get lost otherwise
  ▶ If no:
    ▶ defining instance-data format is still important
    ▶ Would you support a new draft/bis to recommend that data in ietf-yang-library SHOULD be documented as instance-data?

▶ How the correct versions of the target models used inside instance data are defined
  – Proposal: Don’t use revisions – way to strict.
    ▶ Wait for Versioning draft and reuse their import-by-semantic-version solution.
Open Issues 2

› Should we use yang-data-ext to define the instance data module?
  › Proposal: yes. This is the perfect use case for it.
    › It does not affect the actual instance-data-file format.
  › but
    › It maybe more difficult for some tools. Exactly how?
    › It might be important for mixed modules, but not here

› Shall we add a datastore parameter for config=true data?
  – Proposal: Only if we find a use-case for it, so not now
    – The draft does not specify how instance data is used by the YANG server
    – For now all config=false data belongs to operational, all config=true data belongs to running/candidate
Open Issues 3

› Allow multiple instance-data-sets in one file
  › Proposal: No. Even in YANG one-file=one-module is the practice. The convention to use the main contained item as a file name is useful

› Create tags to hold instance data blocks in RFCs
  like <CODE BEGINS> <CODE ENDS>
  – e.g.  <INSTANCE DATA BEGINS>  <INSTANCE DATA ENDS>
  – Proposal: No, Yang examples are not using the <CODE> tags either

› Shall we include a semantic version for the instance-data-set
  – Proposal: Don’t do it, maybe later.
  – What does compatibility mean for instance data?
  – Who would use the semver numbers?
Way Forward

› Request adoption as workgroup document

› Received support on last IETF
› Received support and comments on the mailing list
Background slides
Metadata Yang Module

› module *ietf-yang-instance-data*

› Defines metadata annotations to for YANG Instance Data sets
  - instance-data-set : name of the instance data
  - Contact
  - Organization
  - Description
  - Revision
XML Format

  <name>acme-router-modules</name>
  <revision>2108-01-25</revision>
  <description>The set of modules that acme-router will contain. </description>
  <data>
    <yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
      <module-set>
        <name>basic</name>
        <module>
          <name>ietf-system</name>
          <revision>2014-08-06</revision>
          <namespace>urn:ietf:params:xml:ns:yang:ietf-system</namespace>
          <feature>authentication</feature>
          <feature>radius-authentication</feature>
        </module>
      </module-set>
    </yang-library>
  </data>
</instance-data-set>
Needed - Used

› Already used in multiple implementations
  – Based on similar principles, but slightly different formats

› Needed for YangPush
  – Proposed to be used for documenting on-change notification capabilities

› Needed for Yang-Library
  – Ietf-yang-library contains a lot of server capabilities that are usually set in implementation file, but may change
Out-of-Scope

› Which server capabilities to document
  – Separate drafts e.g. draft-lengyel-netconf-notification-capabilities
› Which yang modules to preload with data
› How instance data is loaded by the server
  – Using an instance-data-file
  – Any other way
› Life-cycle of any documented/preloaded data
  – Is it **protected** from modification, or it **may change** later
  – YangPush may supply notifications about any change