



YANG Instance Data File Format

Balazs Lengyel

Benoit Claise

[draft-lengyel-netmod-yang-instance-data](#)

2018-11-04

Make YANG instance data available offline

- › There is a need to document YANG instance data when a live YANG server is not available
 - If data is needed already in design/implementation time
 - If a live server is not available to the user group
- › Define a file based format to document YANG instance data

Use Cases

- › Document Server Capabilities
 - It is assumed that a separate draft/RFC will define which capabilities/models to document
- › Document/load vendor defined Default Configuration
- › Document Factory Default settings
 - draft-wu-netmod-factory-default
- › Backup/Restore of configuration
- › Others?

Basic Idea

- › Re-use existing formats from <get> operation/request
- › Decorate with meta-data:
 - instance-data-set-name
 - revision-date
 - Description
 - Set of meta-data shall be extensible (augment...)
- › Define both XML and JSON format
 - Later potentially other formats e.g. CBOR
- › Both config true and false data can be included
- › Partial data sets allowed

Adopted by Workgroup

- › The draft was adopted by the workgroup
 - New title: **YANG Instance Data File Format**
 - Draft: draft-ietf-netmod-yang-instance-file-format
 - › Storing the draft is pending due to missed deadline
 - › Unchanged compared to draft-lengyel-netmod-yang-instance-data-05 except title
 - Changes since IETF 102
 - › clarified scope : This is only about the file format
 - › Use-cases are present but only as examples
 - › Added parameter to specify datastore
 - › Clarified that instance data is a snapshot valid at the time of creation, so it does not contain any later changes.

Open Issues 1

- › If we define metadata per target module, a list of target modules could be included in the metadata.
 - This depends on what additional metadata we will include.
- › How the correct versions of the target modules used inside instance data are defined
 - Don't use mandatory revision dates – way to strict.
 - Wait for Versioning draft and reuse their import-by-semantic-version solution.
 - **Proposal**: indicate for which module version(s) was the data set last updated. Just a hint, that can be considered together with the compatibility of the target modules themselves.
 - Anyone using an instance-data-set SHOULD accept compatible differences e.g.
 - › Config data for not supported modules/features

Open Issues 2

- › Should we document what YANG features does the instance data set require?
 - **Proposal**: No: that is already a use case, documenting data from the YANG library. If we would document features why not deviations, imported module revisions, etc. Gets too complicated.
- › **Augmenting metadata must be possible.**
 - As of now it looks like yang-data-ext will solve this. If not, define instance data as regular YANG instead of yd:yang-data.

Background slides

- › I don't plan to show the following slides unless needed to answer questions

Use- Case

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

Use- Case 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

Metadata Yang Module

- › module *ietf-yang-instance-data*
- › Defines metadata for YANG Instance Data sets
 - instance-data-set : name of the instance data
 - Contact
 - Organization
 - Description
 - Revision

XML Format

```
<instance-data-set xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-instance-data">
  <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</>
          <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>
```

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