



# YANG Instance Data format

Balazs Lengyel, Benoit Claise

2019-11-18

[draft-ietf-netmod-yang-instance-file-format-05](#)

# Concept

- › Many use-cases where documenting YANG Instance data in files is useful
  - Documenting server capabilities
  - Preloading default configuration data
  - Documenting Factory Default Settings
  - Storing diagnostics data
  - etc.
- › Define a standard file format
  - Meta-data about the instance data set
  - XML or JSON encoding
  - content data similar to a get - reply

# Changes 03 - 05

- › removed entity-tag and last-modified timestamp
  - inherited from Restconf, but there were uncertainties about the usage
- › Added a mandatory yid-version to the header metadata to allow later updates of the instance-data-format.

```
<yid-version>1</yid-version>
```

- › Made (full) inline schema specification method optional
- › Added simplified-inline method of content-schema specification

```
<content-schema>  
  <module>  
    ietf-notification-capabilities@2019-10-22.yang  
  </module>  
</content-schema>
```

# Changes 03 - 05

- › Added a wrapping container for the content schema, and renamed the data nodes in the inline and uri cases.

```
module: ietf-yang-instance-data
  structure instance-data-set:
    +--rw name?                               string
    +--rw yid-version                          uint8
    +--rw content-schema
      |
      | +--rw (content-schema-spec)?
      |   +--:(simplified-inline)
      |     | +--rw module*                  string
      |     +--:(inline)                    {inline-content-schema}?
      |       | +--rw inline-module*         string
      |       | +--rw inline-schema         <anydata>
      |       +--:(uri)
      |         +--rw same-schema-as-file?   inet:uri
    +--rw description?                        string
    +--rw contact?                           string
    +--rw organization?                       string
    +--rw datastore?                          ds:datastore-ref
    +--rw revision* [date]
      | +--rw date                          string
      | +--rw description?                  string
    +--rw timestamp?                          yang:date-and-time
    +--rw content-data?                       <anydata>
```

# Changes 03 - 05

- › Updated according to YANG-Doctor review
- › Updated security considerations
  - does not specify any method to influence the behavior of a server
  - Use normal file handling security
- › Allowed .yin for simplified-inline schema naming
  - file extension included to keep familiar format
  - same format as YANG module file naming
- › YANG module revision in content schema is mandatory  
ONLY if present in schema defining YANG module

```
<instance-data-set xmlns="urn:...:ietf-yang-instance-data">
  <name>read-only-acm-rules</name>
  <yid-version>1</yid-version>
  <content-schema>
    <module>
      ietf-netconf-acm@2319-10-22.yang
    </module>
  </content-schema>
  <revision>
    <date>1776-07-04</date>
    <description>Initial version of data</description>
  </revision>
  <description>Access control rules for a read-only
    role.</description>
```

```
<content-data>
  <nacm xmlns="urn:...:ietf-netconf-acm">
    <enable-nacm>true</enable-nacm>
    <read-default>deny</read-default>
    <exec-default>deny</exec-default>
    <rule-list>
      <name>read-only-role</name>
      <group>read-only-group</group>
      <rule>
        <name>read-all</name>
        <module-name>*</module-name>
        <access-operation>read</access-operation>
        <action>permit</action>
      </rule>
    </rule-list>
  </nacm>
</content-data>
</instance-data-set>
```

# Open Issue module naming

- › Naming the modules in the content specification
  - In draft: `ietf-yang-library@2019-06-21.yang` or `.yin`
    - › Short - single string per module
    - › Well known
    - › No one misunderstood it
  - Do we need the file extension
    - › It is not really a file, but a module+revision we are specifying
    - › `ietf-yang-library@2019-06-21`
    - › Somewhat strange
  - Create a list with separate module-name and revision
    - › Longer, more complex
    - › list module {
      - › key “name”;
      - › leaf module name {...}
      - › leaf revision {...}
    - › }