

# Instance Data Tools

Scott Mansfield

(YANGsters)

# Introduction

- Generating instance file templates
- Validating Instance Data
- RFC 9195 and current tooling

# Generating instance file templates

- pyang has the capability to generate template instance files
- Definitions needed
  - data: Complete datastore with status data
  - config: Configuration datastore without status data
- For example
  - `pyang --format=sample-xml-skeleton --sample-xml-skeleton-doctype=data --sample-xml-skeleton-defaults turing-machine.yang`
  - Will generate a sample xml skeleton file for “data”.

pyang options for skeleton creation

Sample-xml-skeleton output specific options:

- `--sample-xml-skeleton-doctype=DOCTYPE`  
Type of sample XML document (data or config).
- `--sample-xml-skeleton-defaults`  
Insert leafs with defaults values.
- `--sample-xml-skeleton-annotations`  
Add annotations as XML comments.
- `--sample-xml-skeleton-path=SAMPLE_PATH`  
Subtree to print

# Example of instance skeleton

- Using the turing-machine example found in pyang tutorial
  - <https://github.com/mbj4668/pyang/blob/master/doc/tutorial/examples/turing-machine.yang>

```
<?xml version='1.0' encoding='UTF-8'?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <turing-machine xmlns="http://example.net/turing-machine">
    <state/>
    <head-position/>
    <tape>
      <cell>
        <coord/>
        <symbol/>
      </cell>
    </tape>
    <transition-function>
      <delta>
        <label/>
        <input>
          <state/>
          <symbol/>
        </input>
        <output>
          <state/>
          <symbol/>
          <head-move>right</head-move>
        </output>
      </delta>
    </transition-function>
  </turing-machine>
</data>
```

# Validating Instance Data

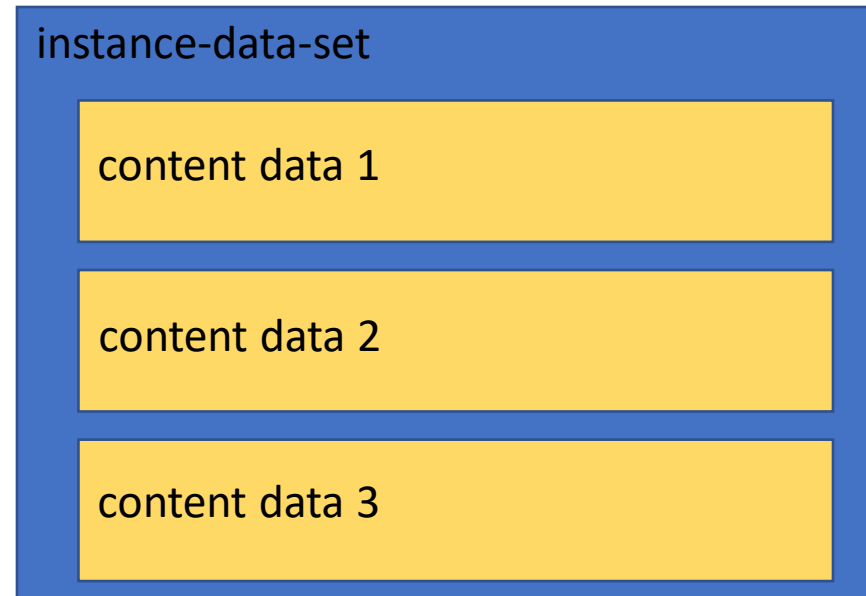
- yanglint (part of libyang) is used by many to test yang modules with representative instance data to exercise more semantic checking rather than just syntactic checking done by other tools (like pyang).
- Some information from the YANGsters FAQ on yanglint
  - [https://1.ieee802.org/yangsters/yangsters-guidelines/yangsters-faq/#Tool\\_yanglint\\_libyang](https://1.ieee802.org/yangsters/yangsters-guidelines/yangsters-faq/#Tool_yanglint_libyang)
- It is important to note, that when using XML, the XML parsers used by libyang and other tools do not enforce the XML Document restriction of one root object per document. So, tools like yanglint can operate on a file that has multiple XML fragments in it.
  - It is possible that some XML validators or editors might have trouble editing a file that is valid for use with yanglint.
  - In that case look for an option/preference that allows the parser to work on a file that includes XML fragments.
- There are other validation tools written in a variety of languages including tools that include netconf servers and other tools (like confd or yuma123).

# Validating Instance Data

- For YANGsters purposes, the use of yanglint has been the focus.
- The complexity of confd and yuma123 hasn't been shown to outweigh the ease of use of yanglint for the testing that YANGsters does.
- yanglint is in constant development, tracking new enhancements being added to the YANG language.
- There are some new features that haven't made it to yanglint yet which brings us to RFC 8791 and RFC 9195
  - <https://datatracker.ietf.org/doc/html/rfc8791>
  - <https://datatracker.ietf.org/doc/rfc9195/>

# Instance Set and Validation

- RFC 9591 is an IETF standard that includes a YANG module that provides meta-data about the instance data you want to publish
- The instance file has a <instance-data-set> that includes information about the yang modules that are used in the instance data that is included in the <content-data>
- For example →



# Tooling issues

- While RFC 9591 provides a standard way to provide information about the instance data,
- yanglint does not support one of the new features that has been added to the YANG language (structure – RFC 8791) that is use in RFC 9591
- So, in order to validate/test the content found in a instance data set
  - It is necessary to extract the content-data and test that without having the instance-data-set meta-data included.
- If interested... Go through the examples on the following slide



# Test case

- <https://github.com/samans/testing-yang/tree/main/ieee-60802/60802>
  - Run the following command “expect i7-silent.in”
    - Produces a clean run. However when you look at the output “domain-name” is still there as a node of system (which it isn’t).
    - And several other typos exist which have the namespace wrong for the ieee802-dotq1-bridge module, so apparently yanglint ignores that.
  - Extract the content-data and put it in i7-content.xml
  - Run “expect i7-content.in”
    - So if I fix that error, there are other errors that pop-up (for example there are some missing interfaces)
  - Fix all the errors in the content file
  - Run “expect i7-fixed-content.xml”
    - Produces a clean run
  - Put the instance data stuff back in to i7-fixed.xml
  - Run “expect i7-fixed.in”
    - fails and most likely has to do with the hack that had to be done to get yanglint to accept the yang module from RFC 9195
- Bottom line, the tooling needs to be fixed. Until it is, you will need to validate you instance data before you wrapper it in the RFC 9195 stuff.
- Follow on... Talk to the confd and yuma123 people and see if they support RFC 8791