

YANG Templates

draft-ma-netmod-yang-config-template-00

NETMOD WG

November 7, 2024

Motivation and Goal

- The use of configuration templates has always been considered as a best practice in the real world.
- The idea of templates/profiles has already been mentioned, e.g.,
 - NMDA allows clients to define configuration templates in <running>
 - Some YANG data models define a profile list that allows to be referenced.
 - Liaison from BBF on YANG scalability (<https://datatracker.ietf.org/liaison/1935/>)
- But it has not yet been standardized.
- This effort aims to propose a standard YANG template mechanism without presuming any particular YANG data model design.

Solution Overview: Defining Templates

A template data model

```
module: ietf-template
  +--rw templates
    +--rw template* [id]
      +--rw id          string
      +--rw description? string
      +--rw content?    <anydata>
      +--ro last-modified? yang:timestamp
      +--ro parent-template* -> ../../template/id
      +--ro inherited-by*  union
```

An example of an interface template

```
<templates>
  <template>
    <id>interface-type-mtu</id>
    <interface>
      <type>ianaift:ethernetCsmacd</type>
      <mtu>1500</mtu>
      <description>MTU value is set by template</description>
    </interface>
  </template>
</templates>
```

The template content:

- may include any instance nodes that wish to be reusable;
- cannot be validated when created, as it may depend on other nodes outside to satisfy YANG constraints.

Solution Overview: Inheriting/Applying Templates

Unexpanded configuration with both interface list entries inheriting the "interface-type-mtu" template

```
<interfaces xmlns:template="urn:ietf:params:xml:ns:yang:ietf-template">
  <interface template:stmt-extend="interface-type-mtu">
    <name>eth0</name>
  </interface>
  <interface template:stmt-extend="interface-type-mtu">
    <name>eth1</name>
  </interface>
</interfaces>
```

Configuration expanded from the left

```
<interfaces>
  <interface>
    <name>eth0</name>
    <type>ianaift:ethernetCsmacd</type>
    <mtu>1500</mtu>
    <description>MTU value is set by template</description>
  </interface>
  <interface>
    <name>eth1</name>
    <type>ianaift:ethernetCsmacd</type>
    <mtu>1500</mtu>
    <description>MTU value is set by template</description>
  </interface>
</interfaces>
```

A config template may be inherited:

- At any level node in the hierarchies, but the expanded result MUST be valid;
 - Expanded config is produced by a merging of explicitly provided config and template config at the corresponding level with the explicitly provided takes precedence
- By another template

Solution Overview: Overriding Templates

- The proposed solution allows a template to be overridden by other templates or config explicitly provided by the client.
- Modification/Extension
 - Explicitly provide the desired configuration when inheriting the template
 - A merge result with the explicitly provided configuration takes precedence
- Deletion
 - Declare the metadata “operation-tag” with a value “delete” for the node intended to be deleted
- “ordered-by user” list/leaf-list Reordering
 - Declare the metadata “operation-tag” with different values for the ordering control:
 - position-first: the attached entry is positioned as the first one.
 - position-last: the attached entry is positioned as the last one.
 - position-before: the attached entry is positioned before some other specified entries.
 - position-after: the attached entry is positioned after some other specified entries.

Interactions with NMDA Datastores

- Defined in <running> or <system>, depending on whether provided by operator or device.
- Unexpanded in <running> when inherited, a read back of <running> returns what is sent with the “stmt-extend” metadata attached to the specific node.
- Expanded in <intended>, which is subject to validation.
- Template definition may not be present in <operational> if it is not used, i.e., not inherited by any node in the config data tree.

Comments, Questions, Concerns?