

YANG annotation for schema

IETF 96

July 18, 2016

<https://datatracker.ietf.org/doc/draft-agv-netmod-yang-compiler-metadata/>

<https://datatracker.ietf.org/doc/draft-agv-netmod-yang-annotation-ds-and-derived/>

Anil Kumar S N (anil.sn@huawei.com)

Gaurav Agrawal (gaurav.agrawal@huawei.com)

Vinod Kumar S (vinods.kumar@huawei.com)



Agenda

History

Problem statement

Requirement

Proposed Mechanism

Inputs from WG mailing list

History

Origin of YANG

- NETCONF treated device as a logical data store.
- YANG was the **Schema definition language** of Logical datastore.

YANG was extended to represent the schema of data exchanged in:

- ietf-netconf-**restconf**
- ietf-netmod-yang-**json**
- vanderstok-core-**comi**

YANG was limited to:

- Schema representation for **network management protocol**.
- Semantically limited to **configurational or state data**.

Problem Statement

Background

- NETCONF enhanced NMS to operate on device like a Datastore, and YANG enabled NMS to **programmatically access the Datastore**.
- Similarly a lot of tool chains have been built using YANG, due to its simplicity and efficiency in schema representation.

YANG has become an integral part in developing tool chains for:

- **Automation** of NMS / Controller development.
- **Faster and error free** APP development in NMS / controller env.

Problem Statement

YANG is not extensible to represent all the Datastore nuances used in the NMS / controller development.

Example:

- Index requirements beyond key leaves.
- Automated index generation.
- Data change trigger requirements.

Requirement

Requirement

- Extend YANG's capability to represent Datastore nuances.
- Should be compiler / utilities / platform / language independent.
- Should not modify the current YANG module / sub-module used to represent network management schema.
- Should be compliant with YANG 1.0 / 1.1

Proposed Mechanism

extension **compiler-annotation** {
argument **target**;
description

"This extension allows for defining compiler annotations for any body-stmts. The 'ca:compiler-annotation' statement contains annotations applicable to its target statement identified by the argument.

It's purpose is to provide additional information to compiler about implementation of the modeled information.

The argument is a string that identifies a node in the schema tree. This node is called the compiler

annotation's

target node. The target node **MUST** be a body-stmt as defined in RFC6020bis.

```
} // compiler-annotation  
} //module agv-yang-compiler-annotation
```

sub-statement	RFC 6020bis section	cardinality
description	7.21.3	0..1
if-feature	7.20.2	0..n
reference	7.21.4	0..1
status	7.21.2	0..1
units	7.3.3	0..1
...	Current Section 5	1..n

Table 2: Substatements of "ca:compiler-annotation".

Proposed Mechanism Example

```
module example-yang {
  .....
```

```
  container candidate-servers {
    list server {
      key "name";
      unique "ip port";
      leaf name {
        type string;
      }
      leaf ip {
        type inet:ip-address;
      }
      leaf port {
        type inet:port-number;
      }
    }
  }
}
```

```
module example-compiler-annotation {

  ca:compiler-annotation /candidate-servers/server {
    ds:auto-key {
      Leaf server-id{
        type int32;
        Range 100..max;
      }
    }
  }
}
```

Inputs from WG mailing list

Comments / Feedback

Action / Clarification

YANG annotations, should be a valid as per YANG 1.0

Action: Draft 01 version is updated to annotate using YANG 1.0's extensions

YANG annotations, could be done with normal extension

Clarification: Yes, it is updated to use extension, but needs to be standardized.

Implementation specific as code generation is out of scope of IETF.

Clarification: We wanted to standardize a mechanism in YANG, to enable the NMS / controller app to mention additional information about the data store usage.

In line annotation is not advisable.

Action: Draft 01 version is updated to maintain the annotations in separate YANG file.

Need for standard extensions tied to a standard module is understandable, but not tool specific annotations.

Clarification: We wanted to standardize a mechanism in YANG, to enable the NMS / controller app to mention additional information about the data store usage.

Summary

- Extends YANG to represent the schema of Applications in controller/ NMS.
- Removes multiple proprietary extension for same requirement across multiple platforms.
 - Removes platform specific tool chain usage for applications.
 - Enables Operators to reuse the same application, accross different controller / tool chains.

Implemented the proposed mechanism for compiler annotation as a part of IETF 96 Hackathon.

THANK YOU

