Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2017

M. Bjorklund
Tail-f Systems
L. Berger, Ed.
LabN Consulting, L.L.C.
June 29, 2017

YANG Tree Diagrams
draft-ietf-netmod-yang-tree-diagrams-01

Abstract

This document captures the current syntax used in YANG module Tree Diagrams. The purpose of the document is to provide a single location for this definition. This syntax may be updated from time to time based on the evolution of the YANG language.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 31, 2017.

Table of Contents

1. Introduction

YANG Tree Diagrams were first published in [RFC7223]. Such diagrams are commonly used to provided a simplified graphical representation of a data model and can be automatically generated via tools such as "pyang". (See <https://github.com/mbj4668/pyang>). This document provides the syntax used in YANG Tree Diagrams. It is expected that this document will be updated or replaced as changes to the YANG language, see [RFC7950], necessitate.

Today's common practice is include the definition of the syntax used to represent a YANG module in every document that provides a tree diagram. This practice has several disadvantages and the purpose of the document is to provide a single location for this definition. It is not the intent of this document to restrict future changes, but rather to ensure such changes are easily identified and suitably agreed upon.

An example tree diagram can be found in [RFC7223] Section 3. A portion of which follows:

```
  +--rw interfaces
  |  +--rw interface* [name]
  |     +--rw name                        string
  |     +--rw description?                string
  |     +--rw type                        identityref
  |     +--rw enabled?                    boolean
```

```
   |     +--rw link-up-down-trap-enable?   enumeration
```

The remainder of this document contains YANG Tree Diagram syntax based on output from pyang version 1.7.1.

2. Tree Diagram Syntax

This section provides the meaning of the symbols used in YANG Tree diagrams.

A full tree diagram of a module represents all elements. It includes the name of the module and sections for top level module statements (typically containers), augmentations, rpcs and notifications all identified under a module statement. Module trees may be included in a document as a whole, by one or more sections, or even subsets of nodes.

A module is identified by "module:" followed the module-name. Top level module statements are listed immediately following, offset by 4 spaces. Augmentations are listed next, offset by 2 spaces and identified by the keyword "augment" followed by the augment target node and a colon (':') character. This is followed by, RPCs which are identified by "rpcs:" and are also offset by 2 spaces. Notifications are last and are identified by "notifications:" and are also offset by 2 spaces.

The relative organization of each section is provided using a text-based format that is typical of a file system directory tree display command. Each node in the tree is prefaces with '+--'. Schema nodes that are children of another node are offset from the parent by 3 spaces. Schema peer nodes separated are listed with the same space offset and, when separated by lines, linked via a pipe ('|') character.

The full format, including spacing conventions is:

```
module: <module-name>

  +--<node>
  |  +--<node>
  |     +--<node>
```

```
     +--<node>
        +--<node>
           +--<node>
     augment <target-node>:
       +--<node>
          +--<node>
          +--<node>
             +--<node>
```



```
     rpcs:
       +--<node>
       +--<node>

     notifications:
       +--<node>
          +--<node>
          |  +--<node>
          +--<node>
```

2.1. Submodules

Submodules are represented in the same fashion as modules, but are identified by "submodule:" followed the (sub)module-name. For example:

```
submodule: <module-name>

  +--<node>
  |  +--<node>
  |     +--<node>
```

2.2. Groupings

Nodes within a used grouping are expanded as if the nodes were defined at the location of the uses statement.

2.3. Collapsed Node Representation

At times when the composition of the nodes within a module schema are not important in the context of the presented tree, peer nodes and their children can be collapsed using the notation '...' in place of the text lines used to represent the summarized nodes. For example:

```
  +--<node>
  |  ...
  +--<node>
     +--<node>
        +--<node>
```

2.4. Node Representation

Each node in a YANG module is printed as:

<status> <flags> <name> <opts> <type> <if-features>


```
  <status> is one of:
    +  for current
    x  for deprecated
    o  for obsolete

  <flags> is one of:
    rw  for configuration data
    ro  for non-configuration data
    -x  for rpcs and actions
    -n  for notifications
    mp   for schema mount points

  <name> is the name of the node
    (<name>) means that the node is a choice node
   :(<name>) means that the node is a case node

   If the node is augmented into the tree from another module, its

   name is printed as <prefix>:<name>.

  <opts> is one of:
    ?  for an optional leaf, choice, anydata or anyxml
```

```
      !  for a presence container
      *  for a leaf-list or list
      [<keys>] for a list's keys
      /  for a mounted module
      @  for a node made available via a schema mount
         parent reference
```

<type> is the name of the type for leafs and leaf-lists

If the type is a leafref, the type is printed as "-> TARGET", where TARGET is either the leafref path, with prefixed removed if possible.

<if-features> is the list of features this node depends on, printed within curly brackets and a question mark "{...}?"

2.5. Extensions

TBD

3. Usage Guidelines For RFCs

This section provides general guidelines related to the use of tree diagrams in RFCs. This section covers [Authors' note: will cover] different types of trees and when to use them; for example, complete module trees, subtrees, trees for groupings etc.

3.1. Wrapping Long Lines

Internet Drafts and RFCs limit the number of characters that may in a line of text to 72 characters. When the tree representation of a node results in line being longer than this limit the line should be broken between <opts> and <type>. The type should be indented so that the new line starts below <name> with a white space offset of at least two characters. For example:

```
  notifications:
    +---n yang-library-change
       +--ro module-set-id
               -> /modules-state/module-set-id
```

The previously 'pyang' command can be helpful in producing such

output, for example the above example was produced using:

```
pyang -f tree --tree-line-length 50 < ietf-yang-library.yang
```

4. YANG Schema Mount Tree Diagrams

YANG Schema Mount is defined in [I-D.ietf-netmod-schema-mount] and warrants some specific discussion. Schema mount document is a generic mechanism that allows for mounting one data model consisting of any number of YANG modules at a specified location of another (parent) schema. Modules containing mount points will identify mount points by name using the mount-point extension. These mount-points should be identified, as indicated above using the 'mp' flag. For example:

```
module: ietf-network-instance
  +--rw network-instances
     +--rw network-instance* [name]
        +--rw name            string
        +--rw enabled?        boolean
        +--rw description?    string
        +--rw (ni-type)?
        +--rw (root-type)?
           +--:(vrf-root)
           |  +--mp vrf-root?
```

Note that a mount point definition alone is not sufficient to identify if a mount point configuration or for non-configuration data. This is determined by the yang-schema-mount module 'config' leaf associated with the specific mount point.

In describing the intended use of a module containing a mount point, it is helpful to show how the mount point would look with mounted

modules. In such cases, the mount point should be treated much like a container that uses a grouping. The flags should also be set based on the 'config' leaf mentioned above, and the mount realted options indicated above should be shown. For example, the following represents the prior example with YANG Routing and OSPF modules mounted, YANG Interface module nodes accessible via a parent-reference, and 'config' indicating true:

```
module: ietf-network-instance
  +--rw network-instances
     +--rw network-instance* [name]
        +--rw name            string
        +--rw enabled?        boolean
        +--rw description?    string
        +--rw (ni-type)?
        +--rw (root-type)?
           +--:(vrf-root)
              +--mp vrf-root?
                 +--ro rt:routing-state/
                 |  ...
                 +--rw rt:routing/
                 |  ...
                 +--ro if:interfaces@
                 |  ...
                 +--ro if:interfaces-state@
                    ...
```

The with 'config' indicating false, the only change would be to the flag on the rt:routing node:

```
                 +--ro rt:routing/
```

5. IANA Considerations

There are no IANA requests or assignments included in this document.

6. Informative References

[I-D.ietf-netmod-schema-mount]
Bjorklund, M. and L. Lhotka, "YANG Schema Mount", draft-ietf-netmod-schema-mount-05 (work in progress), May 2017.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <http://www.rfc-editor.org/info/rfc7223>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",

RFC 7950, DOI 10.17487/RFC7950, August 2016,
<http://www.rfc-editor.org/info/rfc7950>.

Authors' Addresses

Martin Bjorklund
Tail-f Systems

Email: mbj@tail-f.com

Lou Berger (editor)
LabN Consulting, L.L.C.

Email: lberger@labn.net