

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 15, 2017

M. Bjorklund
Tail-f Systems
L. Berger, Ed.
LabN Consulting, L.L.C.
June 13, 2017

YANG Tree Diagrams
draft-ietf-netmod-yang-tree-diagrams-00

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 15, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Tree Diagram Syntax	2
3.	Next Steps	3
4.	IANA Considerations	4
5.	Informative References	4
	Authors' Addresses	4

[1.](#) Introduction

YANG Tree Diagrams were first published in [[RFC7223](#)]. Such diagrams are commonly used to provide 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.

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

`<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

`<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 `"{...}?"`

3. Next Steps

Authors' note: This draft is currently a bit rough. The next/future version is expected to add text covering different types of trees and when to use them; for example, complete module trees, subtrees, trees for groupings etc. Maybe also how to deal with the limited line lengths in RFCs.

4. IANA Considerations

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

5. Informative References

- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 7223](https://www.rfc-editor.org/rfc/rfc7223), 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](https://www.rfc-editor.org/rfc/rfc7950), 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

