

YANG Tree Diagrams

draft-ietf-netmod-yang-tree-diagrams-01

IETF 99

Authors: Martin Björklund, Lou Berger

Repo: <https://github.com/netmod-wg/yang-tree-diagrams/>

Background: YANG Tree Representation

- Previously
 - All documents copy&paste(&change) the same basic text
 - Readers had to carefully read this text to find the differences between documents
 - No schema mount representation
- Now
 - **draft-ietf-netmod-yang-tree-diagrams**
 - Single document defining YANG Tree Representation

Changes since last meeting

- draft-ietf-netmod-yang-tree-diagrams-00
 - Adopted as WG draft
- draft-ietf-netmod-yang-tree-diagrams-01
 - Filled in some planned sections
 - Added tree representation for Schema Mount
 - Aligned examples with draft-ietf-rtgwg-ni-model-03

Changes and Additions

- Added description of full tree, including all elements
 - Matching existing conventions
 - Module, augment, rpcs, notifications
 - Submodules, groupings
 - Collapsed node representation
 - Wrapping long lines
- Remaining Changes
 - Extensions
 - Currently not shown, representation is TBD

Node Representation

`<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

Schema Mount Additions

- **mp** for schema mount points
- **/** for a mounted module
- **@** for a node made available via a schema mount parent reference
- Module (nodes/leaves/etc) marked **ro** when schema mount config leaf = false

Example

```
+--mp vrf-root?  
  |--ro rt:routing-state/  
  |   ...  
  |--ro rt:routing/  
  |   ...  
  |--ro if:interfaces@  
  |   ...  
  |--ro if:interfaces-state@  
  ...
```

NI: Module Example

```
module: ietf-network-instance
  +--rw network-instances
    +--rw network-instance* [name]
      +--rw name          string
      +--rw enabled?     boolean
      +--rw description? string
      +--rw (ni-type)?
        | +--:(l3vpn)
        | | +--rw l3vpn:l3vpn
        | | | ... // config data
        | | +--ro l3vpn:l3vpn-state
        | | | ... // state data
      +--rw (root-type)?
        +--:(vrf-root)
          +--mp vrf-root
            ...
```

Reminder: modules included under root is an implementation time choice

```
+--ro rt:routing-state/
| +--ro router-id?          yang:dotted-quad
| +--ro control-plane-protocols
| | +--ro control-plane-protocol* [type name]
| | | +--ro ospf:ospf/
| | | ...
+--rw rt:routing/
| +--rw router-id?          yang:dotted-quad
| +--rw control-plane-protocols
| | +--rw control-plane-protocol* [type name]
| | +--rw ospf:ospf/
| | | +--rw instance* [af]
| | | | +--rw areas
| | | | | +--rw area* [area-id]
| | | | | | +--rw interfaces
| | | | | | | +--rw interface* [name]
| | | | | | | | +--rw name if:interface-ref
| | | | | | | | +--rw cost?   uint16
+--ro if:interfaces@
| ...
+--ro if:interfaces-state@
| ...
```

Next Steps

- Fill in TBDs
- Confirm current tree representation
- More feedback

- WG LC before next meeting if no substantive changes?