

draft-clacla-netmod-model-catalog-02

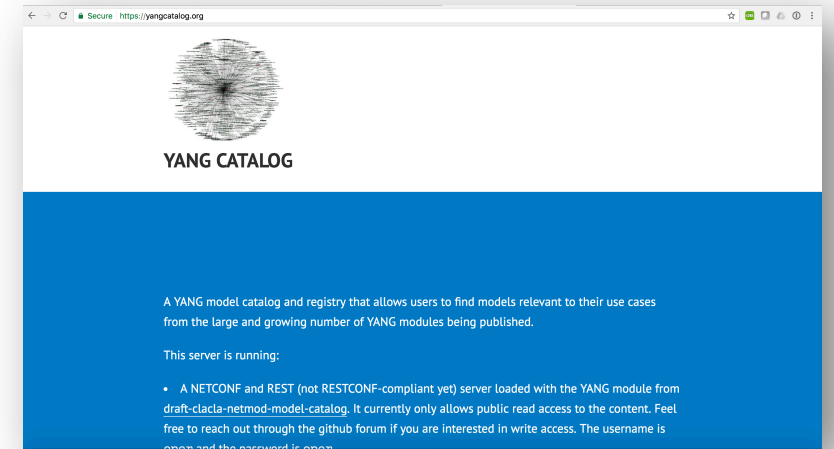
Netmod WG

November 15, 2017

Joe Clarke, Cisco
Benoît Claise, Cisco

Summary

- New use cases based on operator feedback
- YANG Doctor review to improve structure
- New metadata
 - semantic-version
 - derived-semantic-version
 - dependencies and dependents
 - belongs-to
 - tree-type
- Integration between DataTrack and YANG Catalog
- Module exploration and script generation



New Structure

Module Sub-tree

```
module: yang-catalog
+--rw catalog
+--rw modules
| +--rw module* [name revision organization]
| | +--rw name yang:yang-identifier
| | +--rw revision union
| | +--rw organization string
| | +--rw ietf
| | | +--rw ietf-wg? string
| | +--rw namespace inet:uri
| | +--rw schema? inet:uri
| | +--rw generated-from? enumeration
| | +--rw maturity-level? enumeration
| | +--rw document-name? string
| | +--rw author-email? yc:email-address
| | +--rw reference? inet:uri
| | +--rw module-classification enumeration
| | +--rw compilation-status? enumeration
| | +--rw compilation-result? inet:uri
| | +--rw prefix? string
| | +--rw yang-version? enumeration
| | +--rw description? string
| | +--rw contact? string
| | +--rw module-type? enumeration
| | +--rw belongs-to? yang:yang-identifier
| | +--rw tree-type? enumeration
| | +--rw submodule* [name revision]
| | | +--rw name yang:yang-identifier
| | | +--rw revision union
| | | +--rw schema? inet:uri
| | +--rw dependencies* [name]
| | | +--rw name yang:yang-identifier
| | | +--rw revision? union
| | | +--rw schema? inet:uri
| | +--rw dependents* [name]
| | | +--rw name yang:yang-identifier
| | | +--rw revision? union
| | | +--rw schema? inet:uri
| | +--rw semantic-version? yc:semver
| | +--rw derived-semantic-version? yc:semver
| +--rw implementations
| | +--rw implementation* [vendor platform software-version software-flavor]
| | | +--rw vendor string
| | | +--rw platform string
| | | +--rw software-version string
| | | +--rw software-flavor string
| | | +--rw os-version? string
| | | +--rw feature-set? string
| | | +--rw os-type? string
| | | +--rw feature* yang:yang-identifier
| | +--rw deviation* [name revision]
| | | +--rw name yang:yang-identifier
| | | +--rw revision union
| +--rw conformance-type? enumeration
```

Vendor Sub-tree

```
+--rw vendors
+--rw vendor* [name]
+--rw name string
+--rw platforms
+--rw platform* [name]
+--rw name string
+--rw software-versions
+--rw software-version* [name]
+--rw name string
+--rw software-flavors
+--rw software-flavor* [name]
+--rw name string
+--rw protocols
| +--rw protocol* [name]
| | +--rw name identityref
| | +--rw protocol-version* string
| | +--rw capabilities* string
+--rw modules
+--rw module* [name revision organization]
+--rw name -> /catalog/modules/module/name
+--rw revision -> /catalog/modules/module/revision
+--rw organization -> /catalog/modules/module/organization
+--rw os-version? string
+--rw feature-set? string
+--rw os-type? string
+--rw feature* yang:yang-identifier
+--rw deviation* [name revision]
| +--rw name yang:yang-identifier
| +--rw revision union
+--rw conformance-type? enumeration
```

New Use Cases

- Given a module, compare its semantic version over multiple revisions to understand what types of changes (e.g., backward-incompatible changes) have been made
- Do the same given a vendor, platform, software release for all modules
- Provide a comprehensive store for metadata from which to drive tools
- Illustrate whether or not modules are NMDA-compliant

Tracking Module Changes

ietf-interfaces@2013-12-23

```
    },
  ],
  "derived-semantic-version": "1.0.0",
  "implementations": {
    "implementation": [
```

ietf-interfaces@2017-08-17

```
] ,  
  "derived-semantic-version": "2.0.0",  
  "implementations": {  
    "implementation": [  
      {
```

- Derived-semantic-version is determined using:
 1. Order all modules of the same name by revision from oldest to newest.
 2. If module A, revision N+1 has failed compilation, bump its derived semantic MAJOR version.
 3. Else, run "pyang --check-update-from" on module A, revision N and revision N+1 to see if backward-incompatible changes exist.
 4. If backward-incompatible changes exist, bump module A, revision N+1's derived MAJOR semantic version.
 5. If no backward-incompatible changes exist, compare the pyang trees of module A, revision N and revision N+1.
 6. If there are structural differences (e.g., new nodes), bump module A, revision N+1's derived MINOR semantic version.
 7. If no structural differences exist, bump module A, revision N+1's derived PATCH semantic version.

Semantic Version Diffs

1.0.0

2.0.0

ietf-interfaces-old-tree.txt	ietf-interfaces-tree.txt
<pre>module: ietf-interfaces +--rw interfaces +--rw interface* [name] +--rw name string +--rw description? string +--rw type identityref +--rw enabled? boolean +--rw link-up-down-trap-enable? enumeration {if-mib}? +--ro interfaces-state +--ro interface* [name] +--ro name string +--ro type identityref +--ro admin-status enumeration {if-mib}? +--ro oper-status enumeration +--ro last-change? yang:date-and-time +--ro if-index int32 {if-mib}? +--ro phys-address? yang:phys-address +--ro higher-layer-if* interface-state-ref +--ro lower-layer-if* interface-state-ref +--ro speed? yang:gauge64 +--ro statistics +--ro discontinuity-time yang:date-and-time +--ro in-octets? yang:counter64 +--ro in-unicast-pkts? yang:counter64 +--ro in-broadcast-pkts? yang:counter64 +--ro in-multicast-pkts? yang:counter64 +--ro in-discards? yang:counter32 +--ro in-errors? yang:counter32 +--ro in-unknown-protos? yang:counter32 +--ro out-octets? yang:counter64 +--ro out-unicast-pkts? yang:counter64 +--ro out-broadcast-pkts? yang:counter64 +--ro out-multicast-pkts? yang:counter64 +--ro out-discards? yang:counter32 +--ro out-errors? yang:counter32</pre>	<pre>module: ietf-interfaces +--rw interfaces +--rw interface* [name] +--rw name string +--rw description? string +--rw type identityref +--rw enabled? boolean +--rw link-up-down-trap-enable? enumeration {if-mib}? +--ro admin-status enumeration {if-mib}? +--ro oper-status enumeration +--ro last-change? yang:date-and-time +--ro if-index int32 {if-mib}? +--ro phys-address? yang:phys-address +--ro higher-layer-if* interface-ref +--ro lower-layer-if* interface-ref +--ro speed? yang:gauge64 +--ro statistics +--ro discontinuity-time yang:date-and-time +--ro in-octets? yang:counter64 +--ro in-unicast-pkts? yang:counter64 +--ro in-broadcast-pkts? yang:counter64 +--ro in-multicast-pkts? yang:counter64 +--ro in-discards? yang:counter32 +--ro in-errors? yang:counter32 +--ro in-unknown-protos? yang:counter32 +--ro out-octets? yang:counter64 +--ro out-unicast-pkts? yang:counter64 +--ro out-broadcast-pkts? yang:counter64 +--ro out-multicast-pkts? yang:counter64 +--ro out-discards? yang:counter32 +--ro out-errors? yang:counter32 x--ro interfaces-state x--ro interface* [name] x--ro name string x--ro type identityref x--ro admin-status enumeration {if-mib}? x--ro oper-status enumeration x--ro last-change? yang:date-and-time x--ro if-index int32 {if-mib}? x--ro phys-address? yang:phys-address x--ro higher-layer-if* interface-state-ref x--ro lower-layer-if* interface-state-ref x--ro speed? yang:gauge64 x--ro statistics x--ro discontinuity-time yang:date-and-time x--ro in-octets? yang:counter64 x--ro in-unicast-pkts? yang:counter64 x--ro in-broadcast-pkts? yang:counter64 x--ro in-multicast-pkts? yang:counter64 x--ro in-discards? yang:counter32 x--ro in-errors? yang:counter32 x--ro in-unknown-protos? yang:counter32 x--ro out-octets? yang:counter64 x--ro out-unicast-pkts? yang:counter64 x--ro out-broadcast-pkts? yang:counter64 x--ro out-multicast-pkts? yang:counter64 x--ro out-discards? yang:counter32 x--ro out-errors? yang:counter32</pre>

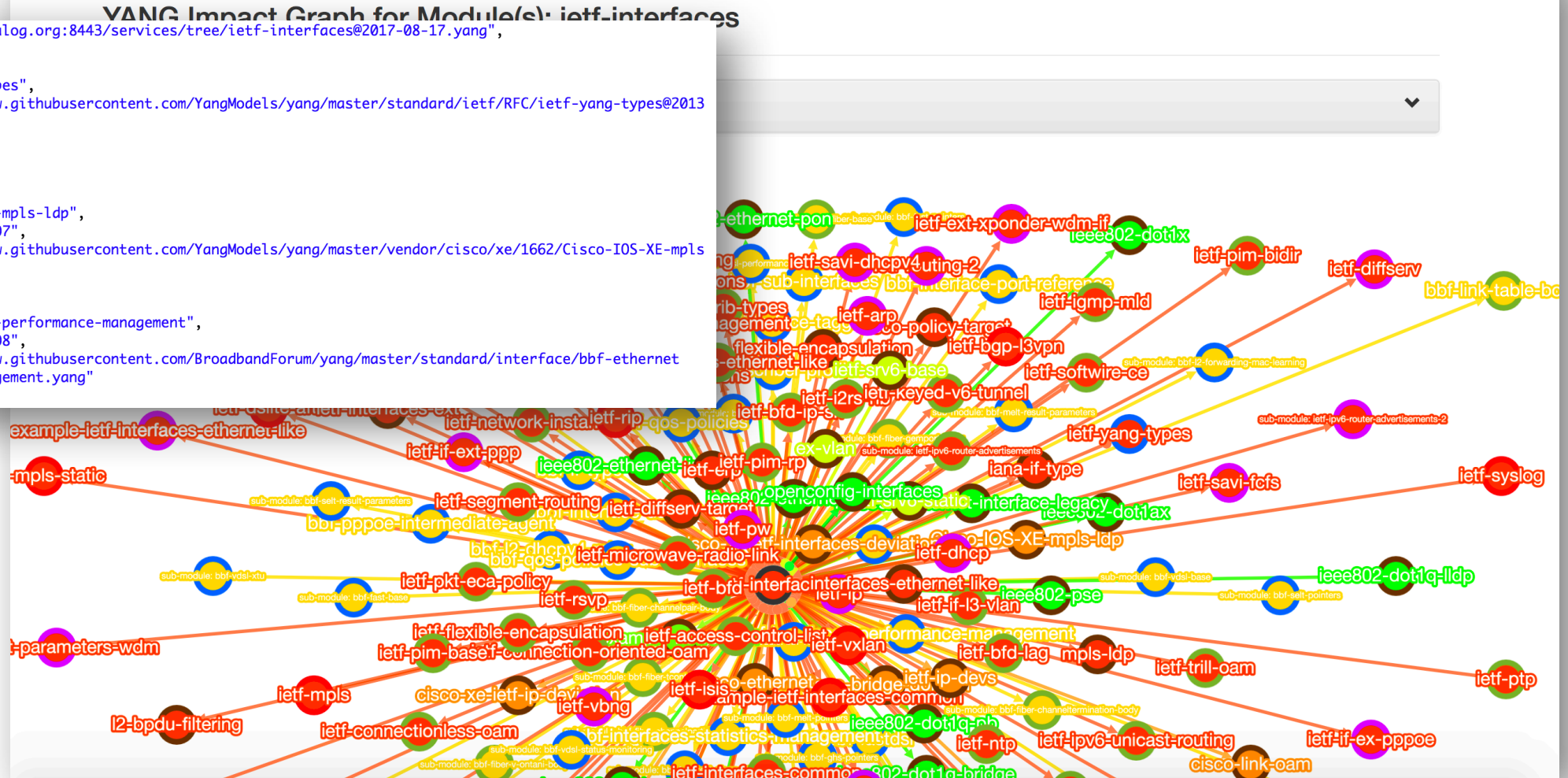
The Catalog can provide links to diff the module's tree and full structure

Tracking Dependencies and Dependents

```

"yang-tree": "https://yangcatalog.org:8443/services/tree/ietf-interfaces@2017-08-17.yang",
"dependencies": [
  {
    "name": "ietf-yang-types",
    "schema": "https://raw.githubusercontent.com/YangModels/yang/master/standard/ietf/RFC/ietf-yang-types@2013-07-15.yang"
  }
],
"dependents": [
  {
    "name": "Cisco-IOS-XE-mpls-ldp",
    "revision": "2017-02-07",
    "schema": "https://raw.githubusercontent.com/YangModels/yang/master/vendor/cisco/xe/1662/Cisco-IOS-XE-mpls-ldp.yang"
  },
  {
    "name": "bbf-ethernet-performance-management",
    "revision": "2017-05-08",
    "schema": "https://raw.githubusercontent.com/BroadbandForum/yang/master/standard/interface/bbf-ethernet-performance-management.yang"
  }
],

```



Checking for NMDA-compatibility

```
"name": "ieee802-ethernet-pse",
"revision": "2017-09-11",
"organization": "ieee",
"namespace": "urn:ieee:std:802.3:yang:ieee802-ethernet-pse",
"schema": "https://raw.githubusercontent.com/YangModels/yang/master/standard/ieee/802.3/draft/ieee802-ethernet-pse.yang",
"generated-from": "not-applicable",
"module-classification": "unknown",
"compilation-status": "passed-with-warnings",
"compilation-result": "https://yangcatalog.org/results/ieee802-ethernet-pse@2017-09-11_ieee.html",
"prefix": "pse",
"yang-version": "1.0",
"description": "This module contains YANG definitions for configuring and managing ports with Power Over Ethernet feature defined by IEEE 802.3. It provides functionality roughly equivalent to that of the POWER-ETHERNET-MIB defined in RFC3621.",
"contact": "Web URL: http://www.ieee802.org/3/cf/",
"module-type": "module",
"tree-type": "split",
```

```
"name": "ieee802-dot1q-tpmr",
"revision": "2017-09-07",
"organization": "ieee",
"namespace": "urn:ieee:std:802.1Q:yang:ieee802-dot1q-tpmr",
"schema": "https://raw.githubusercontent.com/YangModels/yang/master/standard/ieee/802.1/draft/ieee802-dot1q-tpmr.yang",
"generated-from": "not-applicable",
"module-classification": "unknown",
"compilation-status": "passed",
"compilation-result": "",
"prefix": "dot1q-tpmr",
"yang-version": "1.0",
"description": "This YANG module describes the bridge configuration model for the Two Port MAC Relay Bridges.",
"contact": "WG-URL: http://grouper.ieee.org/groups/802/1/\nWG-EMail: stds-802-1@ieee.org\n\nContact: IEEE 802.1 Working Group Chair\nPostal: C/O IEEE 802.1 Working Group\nLane\nP.O. Box 1331\nPiscataway\nNJ 08855-1331\nUSA\nE-mail: STDS-802-1-L@LISTSERV.IEEE.ORG",
"module-type": "module",
"tree-type": "nmda-compatible",
```


Integration of YANG Catalog with DataTracker

YANG module for yangcatalog.org

draft-clacla-netmod-model-catalog-02

Status IESG evaluation record IESG writeups Email expansions History

Versions 00 01 02



Document Type Active Internet-Draft (individual)

Last updated 2017-10-02

Stream (None)

Intended RFC status (None)

Formats plain text xml pdf html bibtex

Yang Validation 0 errors, 1 warnings.

Additional URLs - [Yang catalog entry for yang-catalog@2017-09-26.yang](#)
- [Yang impact analysis for draft-clacla-netmod-model-catalog](#)

Module Details for yang-catalog@2017-09-26/ietf

Specify Module

Module: yang-catalog

Get Details

[Tree View](#) | [Impact Analysis](#) | [Yang Suite](#)

Property Name Property Value

name : yang-catalog

revision : 2017-09-26

organization : ietf

YANG Impact Graph for Module(s): yang-catalog

Graph Options

Click on legend elements below to toggle highlighting on the graph.

Highlight All

Element Colors

IETF

Rim Colors

Maturity: N/A

Maturity: RATIFIED

Maturity: COMPILATION FAILED

Bottleneck to Ratification

Modules:

yang-catalog@2017-09-26.yang x

Orgs:

Recursion Levels: 0

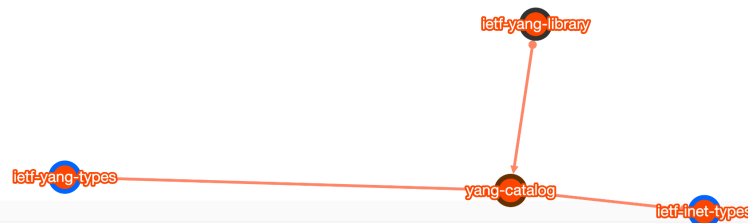
Include Ratified Standards? ☒

Include Sub-modules? ☒

Show Graph Direction: Dependencies Only

Generate

Export



YDK Integration With Yang Suite

The screenshot displays the Yang Suite interface for integrating with YDK. The top section shows the 'Yang Model Name' as 'ietf-interfaces' and a 'Load' button. Below this is a search bar and a 'YANG set' dropdown. The main area is divided into two panels: a tree view on the left and a table on the right.

Tree View:

- ietf-interfaces
 - interfaces
 - interface
 - name
 - description
 - type
 - enabled
 - link-up-down-trap-enable
 - admin-status
 - oper-status
 - last-change
 - if-index

Table:

Nodes	Value
name	GigabitEthernet0/1
description	Primary Uplink

Service Selection:

Service: ☐ codec ☒ CRUD ☐ netconf

Generate script **Clear**

ietf-interfaces: 2017-08-17

Code Editor:

```
# log debug messages if verbose argument specified
if args.verbose:
    logger = logging.getLogger("ydk")
    logger.setLevel(logging.INFO)
    handler = logging.StreamHandler()
    formatter = logging.Formatter("%(asctime)s - %(name)s - "
                                   "%(levelname)s - %(message)s")
    handler.setFormatter(formatter)
    logger.addHandler(handler)

# create NETCONF provider
provider = NetconfServiceProvider(address=device.hostname,
                                  port=device.port,
                                  username=device.username,
                                  password=device.password,
                                  protocol=device.scheme)

# create CRUD service
crud = CRUDService()

# create Codec provider and service
provider = CodecServiceProvider(type="xml")
crud = CodecService()

# decode XML, validate data and create config using NETCONF
entity = codec.decode(cd_provider, payload)
crud.create(nc_provider, entity)

exit()
# End of script
```

Module Details for ietf-interfaces@2017-08-17/ietf

Specify Module

Module: ietf-interfaces

Get Details

[Tree View](#) | [Impact Analysis](#) | [Yang Suite](#)

Next Steps

- Metadata is being added to provide per-module YANG tree output
- Tracking for draft module expiry
- Stemming from draft-clacla-netmod-yang-model-update, metadata to catalog release and feature bundles (Andy mentioned bundling last time as well)
- **Call to Action for other SDOs and vendors:** Upload modules to GitHub and use the “contribute” link to publish metadata to the Catalog
- Contribute to the Catalog → <https://yangcatalog.org/contribute.php>
- Updates about the Catalog → announce@yangcatalog.org