

YANG, NETCONF, RESTCONF

Hackathon 100

November 12, 2017

Summary

- Yang Suite integration with yangcatalog.org
- YDK integration with Yang Suite
- YANG module semantic versioning and diff views from YANG Catalog
- REST API for the YANG Regex validator
- Automated YANG test harness against confd and netconfd
- Integration of YANG Catalog with DataTracker (Henrik)

Yang Suite Integration With yangcatalog.org

The image shows two browser windows illustrating the integration between Yang Suite and yangcatalog.org.

Top Window (Left): Module Details for ietf-interfaces@2017-08-17/ietf

- Specify Module:** The module "ietf-interfaces" is selected.
- Get Details:** A button to fetch details.
- Links:** Tree View, Impact Analysis, and Yang Suite (highlighted with an orange box).

Bottom Window (Right): yangcatalog.org:8000/ydk/

- Yang Model Name:** ietf-interfaces
- Nodes:** A tree view of the YANG model structure:
 - ietf-interfaces
 - interfaces
 - interface
 - name
 - description
 - type
 - enabled
 - link-up-down-trap-enable
 - admin-status
 - oper-status
 - last-change
 - if-index
 - phys-address
 - higher-layer-if
 - lower-layer-if
 - speed
 - statistics
 - interfaces-state
 - Datastores:** candidate
 - Service:** codec, CRUD, netconf (netconf is selected)
 - Buttons:** Generate script, Clear

A large blue arrow points from the Yang Suite link in the top window to the YDK interface in the bottom window, indicating the integration flow.

YDK Integration With Yang Suite

The screenshot shows the Yang Suite interface with the following details:

Yang Model Name: ietf-interfaces **Load**

Nodes (Tree View):

- ietf-interfaces
 - interfaces
 - interface
 - name: gigabitEthernet0/1
 - description: Primary Uplink
 - type
 - enabled
 - link-up-down-trap-enable
 - admin-status
 - oper-status
 - last-change
 - if-index
 - phys-address
 - higher-layer-if
 - lower-layer-if
 - speed
 - statistics
 - interfaces-state

Service: codec CRUD netconf

Generate script **Clear**

ietf-interfaces: 2017-08-17

```
# 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
nc_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
cd_provider = CodecServiceProvider(type="xml")
codec = 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
```

Semantic Version Diffs

1.0.0

```
typedef Qos-caps-operation-enum {
    type enumeration {
        enum add {
            value 0;
            description "Add";
        }
    }
}

grouping QOS-PI-OPER-INPUT {
    description
        "Common node of shared-policy-instance,
        member-interface, interface,
        nv-satellite-interface, satellite-id";
}

container input {
    description
        "A piece of QoS policy-map operational data for
        an interface";
}

uses STATISTICS;
}

grouping VO-Q-STATS {
    description
        "Common node of locationvo-q, output-vo-q,
        vo-qoutput";
}

container vo-q-stats {

skipping to change at line 977
}

grouping OOS-PI-OPER-OUTPUT {

```

2.0.0

```
typedef Qos-caps-operation-enum {
    type enumeration {
        enum add {
            value 0;
            description "Add";
        }
    }
}

grouping QOS-PI-OPER-INPUT {
    description
        "Common node of shared-policy-instance,
        member-interface, interface,
        nv-satellite-interface, satellite-id";
}

container input {
    description
        "A piece of QoS policy-map operational data for
        an interface";
}

container service-policy-names {
    description "Operational data for all Policy instance";
    list service-policy-instance {
        key "service-policy-name";
        description
            "QoS policy-map operational data for a
            particular Policy ";
        leaf service-policy-name {
            type xr:Cisco-ios-xr-string;
            description "Name of the policy instance";
        }
    }
}

uses STATISTICS;
}

grouping VO-Q-STATS {
    description
        "Common node of locationvo-q, output-vo-q,
        vo-qoutput";
}

container vo-q-stats {

skipping to change at line 982
}

grouping OOS-PI-OPER-OUTPUT {

```

REST API For Regex Validator

The screenshot shows a web browser displaying the YANGre W3Cre API documentation. The URL in the address bar is <https://yangcatalog.org/yangre/v1/>. The page title is "git create a tag". The top navigation bar includes links to various Cisco resources like Casekwy, NMS Resource Center, CSC EEM Space, EEM Best Practices, EEM Versions, EEM Built-in "Action...", How To Introduce L..., Migration code to m..., How To Run an EEM..., Tech Zone, and APIC-EM Community. Below the navigation bar, there are links to the Cisco Support Community and Cisco social media pages (Facebook, Twitter, YouTube). The main content area has a green header with the text "swagger" and "Explore". The main title is "YANGre W3Cre API 1.0.0". Below it, there is a note "[Base URL: yangcatalog.org/yangre/v1]" and a link to "https://raw.githubusercontent.com/plewyllie/yangre-gui/master/swagger.json". There are also links to "Contact the developer" and "Apache 2.0". A dropdown menu for "Schemes" is set to "HTTPS". The "default" section contains two POST requests: one for "/yangre" which validates a yang regexp, and another for "/w3c" which validates a w3c regexp.

Automated Test Harness With Servers

Objectives

- Implement Network Management Datastore Architecture (NMDA) support for netconfd.
- NMDA support testcase based on litenc client and ietf-interfaces, ietf-ip, ietf-routing model.
- Add multisession support to libyangrpc.
- litenc: Implement netconf:base:1.1 capability (<https://tools.ietf.org/html/rfc6241>).

Progress

- The testcases (running against confd and netconfd based raw NETCONF session scripting):
 - Non-NMDA - <https://sourceforge.net/p/yuma123/git/ci/master/tree/netconf/test/netconfd/ietf-routing/>
 - NMDA - <https://sourceforge.net/p/yuma123/git/ci/master/tree/netconf/test/netconfd/ietf-routing-bis/> (in progress)
- Server side implementation (in progress, complexity of more than 10 volatile drafts actively developed to implement support for the testcase)

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

draft-clacla-netmod-model-catalog 00 01 02

Jul 2017 Aug 2017 Oct 2017

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: COMPILATION FAILED
Maturity: RATIFIED Bottleneck to Ratification

Modules: yang-catalog@2017-09-26.yang

Orgs:

Recursion Levels: 0 Include Ratified Standards?

Include Sub-modules? Show Graph Direction: Dependencies Only

Generate Export

The graph displays the following connections:

- ietf-yang-types (blue circle) connects to yang-catalog (black circle).
- yang-catalog (black circle) connects to ietf-yang-library (red circle).
- ietf-net-types (blue circle) connects to yang-catalog (black circle).

How Much Was Done At This Hackathon?