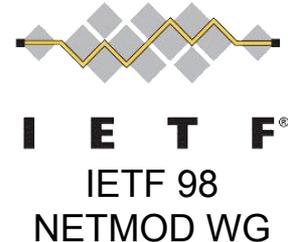


Catalog and registry for YANG models

OpenConfig network operator working group
www.openconfig.net

draft-openconfig-netmod-model-catalog-02

Anees Shaikh , Rob Shakir (Google), Kevin D'Souza (AT&T)



Recap

Motivation

- YANG model landscape is vast and growing
- many organizations / projects now publish YANG models
- users / operators need a way to find the models suited for their needs

Requirements

- “central” location to find modules, dependencies, versions, etc.
- other metadata: publisher information, implementations, access / security

What's new in -02

- informed by additional experience with implementors
- more precise definitions of release and feature bundles
- simplified module metadata, general cleanup

Observations from working with model implementors

- ensuring module interoperability is a challenge with large numbers of models
 - subsets of modules are updated simultaneously
 - difficult to identify which dependent modules need to be updated

Observations from working with model implementors

- ensuring module interoperability is a challenge with large numbers of models
 - subsets of modules are updated simultaneously
 - difficult to identify which dependent modules need to be updated
- implementors do not support models in their entirety
 - how to specify which parts of a model are required to implement a function
 - how to provide precise compliance specifications to implementors

Catalog organization

```
+--rw organizations
  +--rw organization* [name]
    +--rw name
    +--rw type?
    +--rw contact?
    +--rw modules
      |
      | ...
    +--rw release-bundles
      |
      | ...
    +--rw feature-bundles
      |
      | ...
    +--rw implementations
      |
      | ...
```

Catalog organization

```
+--rw organizations
  +--rw organization* [name]
    +--rw name
    +--rw type?
    +--rw contact?
    +--rw modules
      |
      | ...
    +--rw release-bundles
      |
      | ...
    +--rw feature-bundles
      |
      | ...
    +--rw implementations
      |
      | ...
```



```
+--rw modules
  +--rw module* [name version]
    +--rw name
    +--rw version
    +--rw namespace?
    +--rw prefix?
    +--rw revision?
    +--rw summary?
    +--rw classification
      |
      | ...
    +--rw dependencies
      |
      | ...
    +--rw access
      |
      | ...
    +--rw submodules
      |
      | ...
```

Release bundles -- ensuring compatibility

```
+--rw release-bundles
  +--rw release-bundle* [name version]
    +--rw name
    +--rw version
    +--rw members
      +--rw member* [id]
        +--rw id
        +--rw type?
        +--rw module?
        +--rw release-bundle?
        +--rw publisher?
        +--rw compatible-versions*
```

- release bundle members may be modules or other release-bundles
- `compatible-versions` is a list of member versions that work together
 - allows wildcards when using semver, e.g., `1.*.*`

Feature bundles – units of compliance

```
+--rw feature-bundles
  +--rw feature-bundle* [name version]
    +--rw name
    +--rw version
    +--rw path*
    +--rw release-bundle
      | +--rw name?
      | +--rw publisher?
      | +--rw version?
    +--rw feature-bundles
      +--rw feature-bundle* [name]
        +--rw name
        +--rw publisher?
        +--rw version?
```

- features are defined as a set of schema paths
 - users can specify exactly which paths are needed for a function
- schema paths are specified against a release-bundle to ensure compatibility
- feature-bundles may be composed from more granular features to enable reuse
 - e.g., interfaces feature-bundle could be included by ip-router and mpls-router

Additional notes

- Administration of the YANG model catalog may be distributed
 - each organization having permission to update its own entries
- Catalog is not intended to be queried on a device
 - implementations list has information about supported platforms
 - must separately query a target to learn current set of supported modules or bundles
- Catalog implementation progress at the IETF 98 hackathon !