

Schema version selection

draft-wilton-netmod-yang-ver-selection-01

NETMOD WG

November 19th, 2019

Joe Clarke (presenting), Rob Wilton, Reshad Rahman

Agenda

- Recap: goals and summary
- Changes since last revision
- Overview
- Open items
- Next steps

Goals and summary

YANG version selection - Goals

- To address requirements 3.1 and 3.2 from draft-ietf-netmod-yang-versioning-reqs-01:
 3. Requirements related to supporting existing clients in a backwards-compatible way:
 - 3.1 The solution MUST provide a mechanism to allow servers to support existing clients in a backwards-compatible way.
 - 3.2 The solution MUST provide a mechanism to support clients that expect an older version of a given module when the current version has had non-backwards-compatible changes.

YANG Version Selection - Summary

- Allows servers to do NBC changes without forcing clients to immediately migrate to the new module versions
- Makes use of YANG packages defined in draft-rwilton-netmod-yang-packages
- Provides a mechanism for servers to advertise support for multiple versions of YANG packages
- Allows clients to choose, among the ones advertised by the server, which YANG package version they use

YANG Version Selection – Summary (2)

- Servers are NOT required to concurrently support clients using different schema versions
- Servers are NOT required to support every published version of a YANG package
- Servers are NOT required to support all parts of all versioned schema. E.g. for some NBC changes, it may not be possible for the server to support both the old and new versions

Changes since last revision

Changes from rev-00 to 01

- Removed use of port number to select schema. We received comments that opening new ports in certain environments can be problematic
- A given datastore's schema can now be defined by multiple packages (more on this in the list of issues)
- For NETCONF, addition of a new capability and a new RPC for version selection

Overview

YANG Version Selection – Overview

- A versioned schema is a YANG schema with an associated YANG revision label (such as a semantic version number). For example it could be a YANG package (see draft-rwilton-netmod-yang-packages)
- A schema-set is a set of related versioned YANG schema, one or more for each supported datastore
- Servers support configuration for the default schema-set version to use for default NETCONF/RESTCONF connections
- Servers support configuration for secondary NETCONF/RESTCONF instances (using different sessions) which use a different schema-set version
- Clients choose which schema-set they use by using the corresponding RESTCONF root path or using the new NETCONF capability and RPC

YANG Version Selection – Data Tree

```
module: ietf-schema-version-selection
```

```
+--rw schema-selection
```

```
  +--rw schema-sets* [name]
```

```
    | +--rw name string
```

```
    | +--ro datastores* [datastore]
```

```
    |   +--ro datastore ds:datastore-ref
```

```
    |   +--ro packages* [package]
```

```
    |     +--ro package -> /yanglib:yang-library/pkg:package/name
```

```
    |     +--ro version? -> /yanglib:yang-library/pkg:package[pkg:name = current()/../package]/version
```

```
+--rw restconf {secondary-schema-set}?
```

```
  | +--rw schema-sets* [schema-set]
```

```
    | +--rw schema-set -> /schema-selection/schema-sets/name
```

```
    | +--rw root-path inet:uri
```

```
+--rw default-schema-set? -> /schema-selection/schema-sets/name {default-schema-set}?
```

```
rpcs:
```

```
  +---x select-schema-sets
```

```
    +---w input
```

```
      +---w schema-sets* [schema-set]
```

```
        +---w schema-set -> /schema-selection/schema-sets/name
```

YANG Version Selection – NETCONF Capability

- New NETCONF capability for a server to indicate which schema-sets the server is willing to support
- How to indicate default schema-set version? Discussed later in issues.

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
    <capability>urn:ietf:params:netconf:base:1.1</capability>
    <capability> urn:ietf:params:netconf:capability:schema-sets:1.0?
      list=example-ietf-routing@2.1.0, example-ietf-routing@1.3.1, example-vendor-xxx@9.2.3, example-vendor-xxx@8.4.2
    </capability>
  </capabilities>
</hello>
```

YANG Version Selection – NETCONF RPC

- New NETCONF `<select-schema-sets>` operation, for a client to select schema-sets which have been advertised by the server via the new capability
- Should we allow more than 1 schema-set? Discussed later in issues

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">  
  <select-schema-sets>  
    <schema-sets>  
      <schema-set>example-ietf-routing@1.3.1</schema-set>  
      <schema-set>example-vendor-xxx@9.2.3</schema-set>  
    </schema-sets>  
  </select-schema-sets>  
</rpc>
```

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">  
  <ok/> <!-- Schema set selection succeeded -->  
</rpc-reply>
```

YANG Version Selection – NETCONF RPC (2)

- Here are some examples of why a server may send a negative response to the `<select-schema-sets>` operation:
 1. A schema-set in the request is not supported at the requested version
 2. The combination of schema-sets may not be supported simultaneously by the device
 3. Another client is currently using a different schema-set@version and the device doesn't support different versions concurrently
 4. The device does not support changing schema set dynamically

YANG Version Selection – Example Config

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <schema-selection xmlns="urn:ietf:params:xml:ns:yang:ietf-schema-
version-selection" >
    <schema-sets>
      <schema-set>
        <name>example-vendor-xxx@9.3.1</name>
      </schema-set>
      <schema-set>
        <name>example-vendor-xxx@9.2.3</name>
      </schema-set>
    </schema-sets>
  </schema-selection>
</config>
```

Open Items

Item #1: Do we allow multiple schema-sets to be selected?

- The NETCONF RPC currently allows multiple schema-sets to be selected
- This could require a mechanism to resolve conflicts. E.g. if *example-ietf-l3vpn* and *example-ietf-l2vpn* require different versions of the *ietf-interfaces* module, how does a device resolve that?
- Does the device just reject requests which have conflicting requirements? And returns list of compatible packages in the response?
- Or do we allow just 1 schema-set at a time? This forces conflicts to be resolved as defined in YANG packages draft

Item #2: relationship between datastore and YANG package?

- We now have a 1:N relationship between a datastore and the package(s) defining its schema.
- With N packages for a DS, how do we handle conflicts between these packages?
- Do we go back to 1:1 relationship? *draft-rwilton-netmod-yang-packages* has concept of local packages. Or do we define a superset schema?

Item #3: Do we need a superset schema?

- A superset schema would be applicable across all datastores and would take into account deviations and supported features.
- There would be 1 superset schema per schema family (e.g. native-vendor, IETF, OpenConfig)
- If yes, does the definition belong in *draft-rwilton-netmod-yang-packages* or here?
- Only allow superset schema to be advertised (by capability) and selected (by RPC)?

Item #4: Default schema set in capabilities

- The NETCONF capability has no indication of what the default version is, e.g. in previous slide is it version 9.2.3 or 8.4.2 for example-vendor-xxx?
- Add a reserved character sequence to indicate this, e.g. 9.2.3 is the default version for example-vendor-xxx and 1.3.1 is the default version for example-ietf-routing:

```
<capability> urn:ietf:params:netconf:capability:schema-sets:1.0?list=example-ietf-routing@2.1.0, example-ietf-routing@1.3.1;DEFAULT, example-vendor-xxx@9.2.3;DEFAULT, example-vendor-xxx@8.4.2
</capability>
```

Next steps

Next steps

- Get feedback from the WG on the solution and various issues
- Ready for WG adoption?