# YANG Schema Comparison

draft-verdt-netmod-yang-schema-comparison-00

**NETMOD WG**

Nov 19, 2019

Netmod YANG Versioning Design Team

**Presenting: Rob Wilton**

## YANG Schema Comparison - What

- Describes algorithms to compare YANG modules and YANG schema to determine the scope of changes between different arbitrary revisions/versions
- Defines optional YANG extensions to annotate ambiguous changes to improve the accuracy of comparison
- 5th document in YANG versioning DT solution – completes the series
- Note: Is an -00 revision, aim is to scope the direction of the solution rather than be complete

## YANG Schema Comparison - Why

1. Revision labels and YANG Semver work in the mainline case, but not all cases
2. Tooling can help identify changes so new revisions can be labelled accurately
3. Clients are not impacted by changes in the parts of the schema they don’t use
4. Standard annotations can help improve the accuracy of comparison tools

## YANG Schema Comparison - Details

- Generic tree comparison algorithm defined that works on YANG schema
- Comparison is performed via identifier (rather than stmt ordering), hence reordered data nodes are allowed (except RPCs parameters)
- Algorithm can work on individual YANG modules
- Two variants of algorithm defined for YANG packages:
  - Standard version for calculating/checking package version number
  - Filtered version – that gives more deployment specific answer
- YANG extension statements help refine the comparison

# YANG Schema Comparison - Details

## Filtered version for full YANG schema

- The comparison algorithm can be filtered to give a more refined answer for particular clients. E.g., it could optionally:
  - Ignore groupings (after expansion) (e.g. helpful if grouping moved/renamed)
  - Ignore module metadata information
  - Be restricted to a subset of features
  - Be restricted to a subset of the schema (e.g. scoped by instance data)
  - Be set up to filter out editorial changes
- I.e. can give a more refined answer based on the subset of the schema that is being used

# YANG Schema Comparison – Example 1

Fixing a description (editorial rather than nbc)

**Original model**

```
module example {
  import ...

  revision 2019-11-13 {
    rev:revision-label 1.0.0;
    description "Initial revision"
  }

  container foo {
    leaf stuff {
      type string;
      description "do some stuf";
    }
  }
}
```

**New revision**

```
module example {
  import ...

  revision 2019-11-14 {
    rev:revision-label 1.0.1;
    description "Cleanup descriptions";
  }

  revision 2019-11-13 {
    rev:revision-label 1.0.0;
    description "Initial revision";
  }

  container foo {
    leaf stuff {
      type string;
      description "Do some stuff." {
        rev-ext:editorial "2019-11-14";
      }
    }
  }
}
```

# YANG Schema Comparison – Example 2

## Renaming container (e.g. as NBC change)

### Original model

```
module example {
  import ...

  revision 2019-11-13 {
    rev:revision-label 1.0.0;
    description "Initial revision"
  }

  container foo {
    leaf stuff {
      type string;
      description "do some stuf";
    }
  }
}
```

### New revision

```
module example {
  import ...

  revision 2019-11-14 {
    rev:revision-label 2.0.0;
    rev:nbc-changes;
    description "Cleanup module after review";
  }

  revision 2019-11-13 {
    rev:revision-label 1.0.0;
    description "Initial revision";
  }

  container bar {
    rev-ext:renamed-from “foo";
    leaf stuff {
      type string;
      description "do some stuf";
    }
  }
}
```

## YANG Schema Comparison – Next steps?

1. Could/should the extensions be defined in the module-versioning draft (i.e. ietf-yang-revisions)?
2. Refine exactly what extension annotations are needed/useful?
   - Do we need “rev-ext:nbc”?
3. Does “renamed-from” need to also specify a revision label?
   - Probably only useful in corner cases, e.g. if the original identifier gets reused for a different meaning/purpose
   - Probably could be deferred

- Potentially could be figured out after WG adoption?

## Backup slides

# YANG Schema Comparison – Example 3

## Renaming container (same example as BC change)

### Original model

```
module example {
  import ...

  revision 2019-11-13 {
    rev:revision-label 1.0.0;
    description "Initial revision"
  }

  container foo {
    leaf stuff {
      type string;
      description "do some stuf";
    }
  }
}
```

### New revision

```
module example {
  import ...

  revision 2019-11-14 {
    rev:revision-label 1.1.0;
    description “Rename container ‘foo’ to ‘bar’";
  }

  revision 2019-11-13 {
    rev:revision-label 1.0.0;
    description "Initial revision";
  }

  container foo {
    status deprecated;
    // as before
  }

  container bar {
    rev-ext:renamed-from "foo";
    must 'not(../foo)’ { description "Can’t configure both"; }
    leaf stuff {
      type string;
      description "do some stuf";
    }
  }
}
```