Backward Compatibility Rules for Config=False Data

NETMOD WG Interim
2021-02-01

Presenting: Balazs Lengyel
Proposal

• RFC 7950 defines “Allowed/Not-Allowed changes”
• Yang-Versioning handles all changes so terminology used here is:
  • Backwards Compatible (BC)
  • Non-Backwards Compatible (NBC)
• RFC 7950 defines just one set of rules for both Config=False and Config=true data
• In reality compatibility rules are different for config false and true. Some examples:
  • adding a config=true, mandatory leaf NBC - adding a config=false, mandatory leaf BC
  • decreasing the value range of a config=true leaf NBC - decreasing the value range of a config=false leaf BC

• Proposal: Define separate compatibility rules for config=false data
  • All following pages are ONLY about config=false data
Principles

• Data coming from a Netconf server is not always checked by the server according to the model

• Data in the output section of notifications, actions or rpcs is governed by the same backwards compatibility rules as config=false data.

• Basic client requirements
  • A client MUST tolerate any data received (or not received) without crashing
    • A client MUST be able to discard any data that is not part of the model but is sent by the server additionally (e.g. XML elements or attributes, JSON properties)

• Client requirements for a well-designed client
  • A client SHOULD be able to handle valid parts of a received data set even if it discards other parts as invalid
  • A client SHOULD be able to handle data that is outside the valuespace defined, as long as it is of the same basic type
  • A client SHOULD be prepared to handle more items from a list or leaf-list than what is defined by the model
Basic Rules

• Adding mandatory or optional data nodes is **BC**
  • Changing an optional node to mandatory is **BC**

• Removal of optional data is **BC**
  • To be decided

• Removal of mandatory data is **NBC**
  • Changing a mandatory node to optional is **NBC**

• Expanding the valuespace is **BC**,
  • but in some cases, it might cause problems the module might still be marked semver:nbc

• Reducing the valuespace is **BC**

• Changing min/max-elements is **BC except:**
  • Changing min-elements to a lower value is **NBC** (It is like removing mandatory).
Backup Slides
Changed Rules  (compared to RFC 7950)

- Adding mandatory data nodes is BC
- Making nodes mandatory is BC
- Changing mandatory nodes to optional is NBC
- Removal of optional data node is BC
  - (Removal of mandatory data is NBC)
- Expanding the valuespace is BC
  - In some cases, it might cause problems; the module might still be marked semver:nbc
- Reducing the valuespace is BC
  - A "range", "length", or "pattern" statement may decrease the allowed value space.
- Changing min-elements to a lower value is NBC
  - (It is like removing mandatory).
- Changing min-elements to a higher value is BC.
  - Clients are expected to discard superfluous entries
- Changing max-elements statement is BC
- Adding or removing a base statement from an "identityref" is BC
Unchanged Rules (compared to RFC 7950)

• An enumeration type may have new enums added ...
• A bits type may have new bits added ...
• A range", length, or pattern statement may expand the allowed value space.
• A default, units, reference statement may be added
• A must or when statement may be removed, or its constraint relaxed.
• A removing or changing a min-elements or max-elements statement to allow less or more elements then previously is BC
Unchanged Rules (compared to RFC 7950)

• Changing a description statement without changing the semantics of the definition is BC
• Adding a "base" statement to an "identity" statement is BC
• Adding new typedefs, groupings, rpcs, notifications, extensions, features, and identities is BC.
• Adding optional data nodes is BC
• Adding a case statement is BC
• Changing a node that represented state data to represent configuration, provided it is not mandatory is BC
• An "if-feature" statement may be removed.
• Changing a nodes status to from current to deprecated is BC
• Changing a nodes status to obsolete is NBC
  • Same as the proposed change for config=true
Unchanged Rules (compared to RFC 7950)

• Changing a "type" statement may be replaced with another "type" statement is BC
• Replacing any set of data definition nodes with another set of syntactically and semantically equivalent nodes is BC
• Changing module/submodule boundaries without changing the semantics is BC
• Changing the "prefix" statement is BC
• Reordering data is BC
  • Same as the proposed change for config=true
• Removing obsolete state data nodes is BC
  • Same as the proposed change for config=true
• Any other not mentioned change is NBC