YANG Extension and Metadata Annotation for Immutable Flag

draft-ma-netmod-immutable-flag-04

Qiufang Ma (Huawei) Presenter
Qin Wu (Huawei)
Balazs Lengyel (Ericsson)
Hongwei Li (HPE)
Recap

• Use Cases
  – Some “config true” nodes in the schema tree are immutable
    – Irrespective how it is instantiated, it is not allowed to be created/deleted/updated
    – When data nodes exist in multiple instances in the data tree, some of which are read-only, while others are not.

• Objective
  – Allow the server to document the existing immutable configuration data and provide more visibility into immutability characteristic of particular schema or instance nodes
  – This work doesn’t mean adding immutable flag is encouraged for server implementation.
Solution Overview

• **Immutable**: A schema or instance node property indicating that the configuration data is not allowed to be created/deleted/updated.

  0 Question: Can we do better about this terminology?

  immutable YANG extension

  ```
  extension immutable {
    argument exceptions;
    ...
  }
  ```

  immutable Metadata Annotation

  ```
  md:annotation immutable {
    type boolean;
    ...
  }
  ```

  • “exceptions” is also defined to indicate specific operations (create, update, delete) are permitted
  • e.g., if a configuration data can only be created and deleted, modification is not allowed:
    im:immutable “create delete”;
Thanks to Kent, Jason, Andy, Rob, Reshad, Jan, Anthony, Joe for valuable inputs.

• Rephrase and avoid using “server MUST reject” statement, try to clarify that this document aims to provide visibility into existing immutable behavior

• Clarify that deletion to an immutable node in <running> which is instantiated in <system> and copied into <running> should always be allowed

• Clarify that write access restriction due to general YANG rules has no need to be marked as immutable
  • For example, a key leaf in a list

• Clarify how immutable flag interacts with NACM mechanism
  • When a specific data node or instance is marked as “immutable”, NACM cannot override to allow create/delete/update access.
• Add a new section to discuss the inheritance of immutability
  • Unless otherwise specified, the immutability in the hierarchy is inherited downwards towards the leaf/leaf-list nodes.
    • Specifically, if a node has child elements, non-modification means any child elements is not allowed to be created, updated and deleted.
  • The immutability of a particular contained node is allowed to be overridden.

```yaml
list application {
  im:immutable;
  key name;
  leaf name {
    type string;
  }
  leaf protocol {
    type enumeration {
      enum tcp;
      enum udp;
    }
  }
  leaf port-number {
    im:immutable "create update delete";
    type string;
  }
}
```

- The list instance is not allowed to be created/updated/deleted.
- The port-number instance is allowed to be created/updated/deleted in a list entry.
Next Step

• Request: adopt as WG item
Comments, Questions, Concerns?