System-defined Configuration

draft-ietf-netmod-system-config-02

Authors:
Qiufang Ma (Huawei) Presenter
Qin Wu (Huawei)
Chong Feng (Huawei)

Contributors:
Kent Watsen (Watsen Networks)
Jan Lindblad (Cisco)
Chongfeng Xie (China Telecom)
Jason Sterne (Nokia)
Recap

• Adopted in November 2022
  • Describe the handling of the configuration data defined by the server itself

• A “system” configuration datastore (updates to NMDA)
  • Read-only to clients
  • May change dynamically (e.g., when software upgrades or hardware is installed)
  • `<system>` is merged with `<running>` to create `<intended>`
  • Some parts of system config may be modified
    • Override a system-provided value
    • Configure additional nodes inside a system-defined list entry

• Approaches to ensure referential integrity constraints are satisfied in `<running>`
  • Explicit declaration of referenced system config
  • Server auto-configuring referenced system config (“resolve-system” parameter)
High-level Document Updates Since IETF 116

• Add the precise meaning of “reference”
  • Referred as the leafref value
  • Present in an XPath expression of “when”/”must” constraints
  • Defined to satisfy the “mandatory” constraints
  • Used as an “instance-identifier” type value
  • Defined to exactly satisfy the “min-element” constraints

• Better clarify “resolve-system” parameter
  • For a node auto-copied from <system>
    • no difference from the one explicitly declared by the client
    • Will not be removed automatically even if all references to it are deleted/configuration disappears from <system>
  • The intention is not to surprise clients with unexpected changes and allow the client to have full control over <running>

• Update Fig.2 (Architectural Model of Datastores) in NMDA to incorporate <system>

• Editorial improvements
Open Discussion #1

Question: Must system nodes always be copied into <running>?

Notes:
• Both 7950 and 8342 define <running> MUST always be valid, i.e., all referential integrity constraints are satisfied.
• 8342 indicates that <intended> is “subject to validation” after configuration transformations, does this already imply <running> alone cannot be valid?

Options

1. Require referenced system nodes to always be copied into <running>
   • Only the parts that are required to make <running> valid need to be copied (e.g., the key node)
   • Question: What if the configuration in <system> updates and a stale copy is still in <running>?
     • The current document states any update of the contents in <system> will not cause the automatic update of <running>

2. Don’t require referenced system nodes to always be copied into <running>
   • <system> aware clients offline merge <system> into <running>, as well as implement the transformation to <running>
   • The problem might arise after an NMDA-client makes an “implicit” reference, and offline validation of <running> for legacy client(e.g., non-NMDA client) will break

3. Option 1 now, flip to option 2 later
   • What is the impact on legacy clients?
   • How would the client know to stop copying referenced system-nodes?
Open Discussion #2

**Question: What “origin” value do system nodes copied into <running> have?**

- The current draft doesn’t limit which origin should be used, should this be clearly and consistently defined?
- What if the copied system nodes are immutable?

```xml
<applications xmlns="urn:example:application">
  <application>
    <name>ftp</name>
    <protocol>tcp</protocol>
    <destination-port>21</destination-port>
  </application>
  <application>
    <name>tftp</name>
    <protocol>udp</protocol>
    <destination-port>69</destination-port>
  </application>
</applications>
```

**Option 1**

```xml
<applications xmlns="urn:example:application"
  xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin"
  or:origin="or:system">
  <application>
    <name>ftp</name>
    <protocol>tcp</protocol>
    <destination-port>21</destination-port>
  </application>
  <application>
    <name>tftp</name>
    <protocol>udp</protocol>
    <destination-port>69</destination-port>
  </application>
</applications>
```

**Option 2**

```xml
<applications xmlns="urn:example:application"
  xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin"
  or:origin="or:intended">
  <application>
    <name>ftp</name>
    <protocol>tcp</protocol>
    <destination-port>21</destination-port>
  </application>
  <application>
    <name>tftp</name>
    <protocol>udp</protocol>
    <destination-port>69</destination-port>
  </application>
</applications>
```
Comments, Questions, Concerns?