System-defined Configuration

draft-ietf-netmod-system-config-01

Qiufang Ma (Huawei)  Presenter
Qin Wu (Huawei)
Chong Feng (Huawei)
Kent Watsen (Watsen Networks)
Jan Lindblad (Cisco)
Chongfeng Xie (China Telecom)
Jason Sterne (Nokia)
Recap

• Adopted by working group
  - draft-ma-netmod-with-system -> draft-ietf-netmod-system-config

• Most are editorial updates for clarity since last meeting

• The solution is *generally* stable now

  • A config true, read-only `<system>` datastore to hold config which is provided by the system itself

  • A “resolve-system” parameter which controls whether to allow a server to copy any referenced system-defined config automatically

• Some details still need further discussion
Server behavior if not given “resolve-system” parameter

OLD:
“If the ‘resolve-system’ parameter is not given by the client, the server SHOULD NOT modify <running> in any way otherwise not specified by the client.”

NEW:
“If the ‘resolve-system’ parameter is not given by the client, the server should not modify <running> in any way otherwise not specified by the client. Not using capitalized ‘SHOULD NOT’ is intentional. The intention is to bring awareness to the general need to not surprise clients with unexpected changes...”
The Ongoing Discussion around system config and keystore/truststore drafts

• Question: whether we are making this more complex than necessary to require all referenced system config must be present in <running>?

  • Both 8342 and 7950 define that <running> MUST always be valid.

  • If the referenced system configuration is not in <running>
    • <running> is implicitly valid if <intended> is valid

    • "what happens between running and intended is not covered by any standards as of today."
      • This probably would be difficult for clients to enforce offline validation of <running>
      • A legacy client without knowing the existence of <system> might break unless all the referenced system configuration is copied into <running>

• What happens if the configuration in <system> updates/disappears and a stale copy is in <running>?
Discussion #2: the precise meaning of “reference”

• The “resolve-system” parameter allows the server to copy referenced system configuration (if not present) into the target datastore
  
  • The objective is to make config valid

• Question: what does this “referenced system configuration” mean?
  
  • Present in a leafref “path” statement?
  
  • Present in an XPath expression of “when” statement?
  
  • Present in an XPath expression of “must” statement?
  
  • Defined as a “mandatory true” data node?
  
  • Defined to exactly satisfy the “min-element” constraints?
Discussion #3: Timing for the server to enforce auto-copy

- The current document states that the server’s copy referenced nodes to the target datastore MUST be enforced at the end of <edit-config>/<edit-data>/<copy-config> operations.

- Question: Is this correct?
  - The objective is to make config valid. No doubt if the target datastore is <running> (<running> must always be valid).
  - What if the target datastore is <candidate> and “test-option” element value in <edit-config> is “set” (no need to perform the validation)?
    - Can the auto-copy be deferred till <commit>/<validate> RPC?
    - Augment <commit>/<validate> to also support "resolve-system" as input parameter
  - “when” condition must be evaluated during payload parsing (see Sec.8.3 in 7950); does this mean system config present in an XPath expression of “when” statement should always be copied during <edit-config> enforcement?
Comments, Questions, Concerns?