

# **Factory default Setting Capability for RESTCONF**

**draft-wu-netconf-restconf-factory-restore-01**

Qin Wu (bill.wu@Huawei.com)

Ye Niu (niuye@Huawei.com)

# Factory default Setting Capability for RESTCONF

- **Objective**

- Allows RESTCONF client to configure newly deployed devices with just its preconfigured initial state (i.e., factory default settings)(e.g., during zero touch device bootstrapping process)
- or restore the configuration to its preconfigured initial state either during **server restart process or at the time of server fatal error or malfunction.**

- **Motivation**

- RESTCONF can be implemented in a device that doesn't have NETCONF server support.
- HTTP PUT method corresponding to NC <copy-config> in RESTCONF lacks capability to copy one datastore into another datastore
- NETCONF <delete-config> operation only can delete <startup> configuration datastore and reset the <startup> to its factory default but can not delete <running> and reset the <running> to its factory default.
- NETCONF <copy-config> can be used to copy the entire content of source datastore to target datastore but can not be used to return target datastore to factory default without new factory datastore.
  - Choose for user to maintain factory default configuration is expensive

# Why Factory default Setting Capability for RESTCONF

- Limitation of RESTCONF to support factory default setting

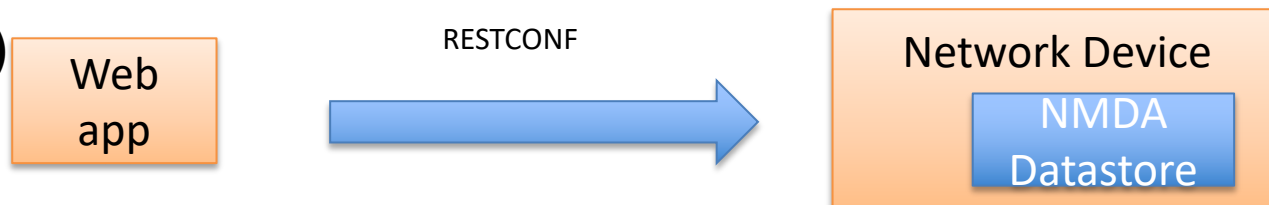
## Example (a)



- In case NC operations are supported by RESTCONF, NC operations <copy-config> can be used to copy one datastore into another datastore, but can not return target datastore into factory default.

POST /restconf/operations/ietf-netconf:copy-config HTTP/1.1

## Example (b)



- RESTCONF might implemented in a device that doesn't have NETCONF server support
- In case the client is NMDA datastore aware, HTTP PUT method in RESTCONF can be used to replace the entire content of the Datasore but not well specified to copy one datastore into another datastore.

```
Example: PUT /restconf/data HTTP/1.1 Host: example.com
Content-Type: application/yang-data+xml
<data xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
  </data>
```

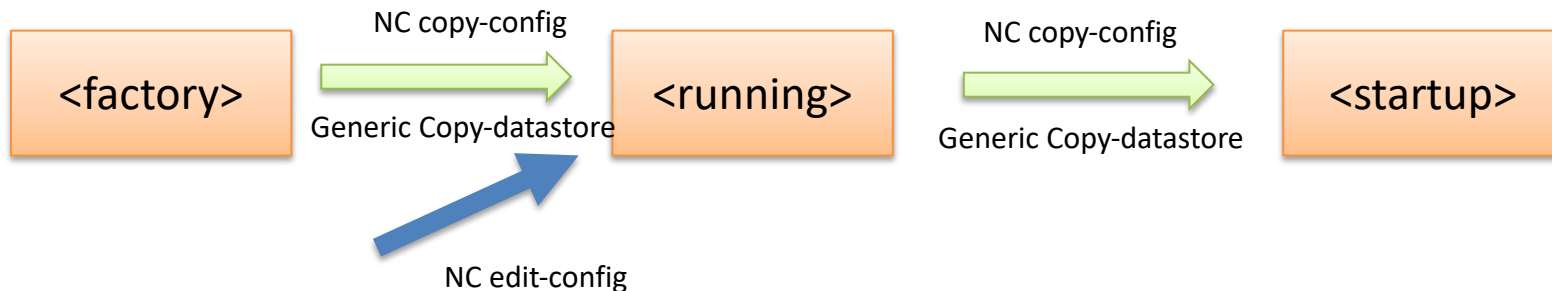
# Proposed solution for factory default

**Case A:** Device Booting or system crash, the device supports :Writable-running capability but not support :startup



The running is initialized into <factory> without <startup>. The NC Copy-config or Generic Copy-datastore can be used. In case of server crash down, the server needs to be restarted first and then the running enter into initialization stage.

**Case B:** Device Booting or system crash, the device supports :Writable-running capability and :startup capability



1. We need to make sure that when running is initialized into <factory>, there is no other operation such as edit-config to alter running before <factory> is copied from running to <startup>, NC <lock> and <unlock> can be used.
2. In case of server crash down, server needs to be restarted first and then the running enter into initialization stage.

- Introduces a new datastore resource named <Factory > that represents datastore with its preconfigured initial state.
- The <Factory > datastore resource is available using the following resource path: `{+restconf}/ds/ietf-restconf-restore:factory-default`
- Define <Factory > as readonly datastore
- Document how <factory> work together with NC/RC operations and restart operation (whether they are new or existing) to support factory default setting.

# Issue #1: Do we need a new operation factory restore?

- **Con:**
  - NC/RC operations + reboot operations can be reused, but reboot seems unspecified
- **Pro:**
  - **Limitation of RC operations:** In case RESTCONF is implemented in a device that doesn't have NETCONF server support, HTTP PUT method doesn't support copying one datastore into another datastore.
  - **General purpose Operation:** The generic copy-datastore can be applied to RESTCONF, NETCONF and other YANG based protocol at datastore level.
  - **Multiple datastores restore consistency:** In case multiple datastores need to be initialized into <factory,> target list can be specified in the generic new operation.
  - **Restart Indication:** In some case, device reboot is needed(e.g., in case of system crash down)before initializing the device into factory default. In some case the device reboot is not needed.
    - New operation can add restart parameter to indicate whether device reboot is needed.
  - **Working with zero touch:**In case the device reboot is needed and zero touch bootstrapping can be leveraged, <factory> can be set as source or source set as nothing

# Issue #2: Difference between system-restart and reboot

- **Issue #2 depends on the solution on Issue #1:**
- **System-restart:**
  - Request that **the entire system** (not just the NETCONF server) be restarted immediately.
  - A server SHOULD send an rpc reply to the client before restarting the system.
- **Reboot:**
  - Usually refer to Restart device, similar to system-restart.
- **Restart:**
  - Restart the **NETCONF server**. Does not reload the software image, but this is unspecified by NC/RC yet.
  - A server might not send an rpc reply to the client before restarting the system.
- Question: **In RC/NC operations suffice**, do we need to define restart as new RPC operation?

# Next Step

- General agreement the benefit of defining factory default datastore for device to activate
  - Follow the guidelines defined in the appendix A of RFC8342
    - Define Which YANG Modules Can Be Used in the Datastore
    - Define Which Subset of YANG-Modeled Data Applies
    - Define How Data Is Actualized
    - Define Which Protocols Can Be Used
    - Define YANG Identities for the Datastore
- Mixed opinion on defining new RC operation.
- Request to accept draft as WG item
  - Got already supporters on the list

# Why Factory default Setting Capability for NETCONF

- NC existing Operations defined in RFC6241 are not sufficient to support factory default setting

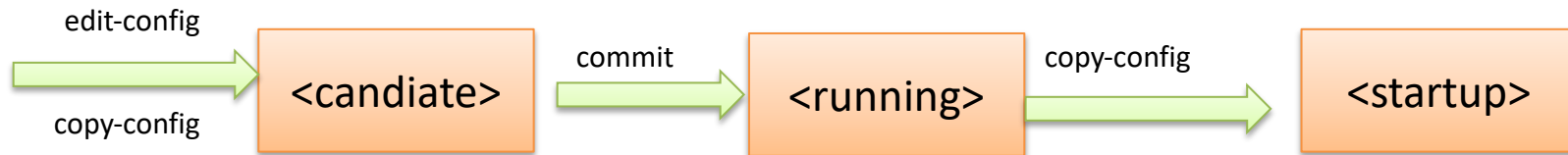
## Example (a)



## Example (b)



## Example (c)



- NC <delete-config> is only applicable to <startup> and can not be applied to <running> based on the rule defined in section 7.4 of RFC6471.
  - <delete-config> can not initialize <running> with factory default setting.
- Does Deleting <startup> configuration datastore means return the device to factory default?
  - Factory default might have a initial state or preconfigured supported models list.
- NC <Copy-config> can Create or replace an entire target configuration datastore with the complete contents of source configuration datastore.
  - Without <factory>, how do we initialize one target datastore with factory default setting?
- In addition, any change to <running> will be automatically updated to <startup> through copy-config if <startup> exists.