

# **Factory default Setting Capability for RESTCONF**

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

Qin Wu

Ye Niu

# 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) during zero touch bootstrapping process
- or restore the configuration to its preconfigured initial state either during device rooting process or at the time of system fatal error or malfunction.

- **Motivation**

- 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.
- 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

# Why Factory default Setting Capability for **NETCONF**

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

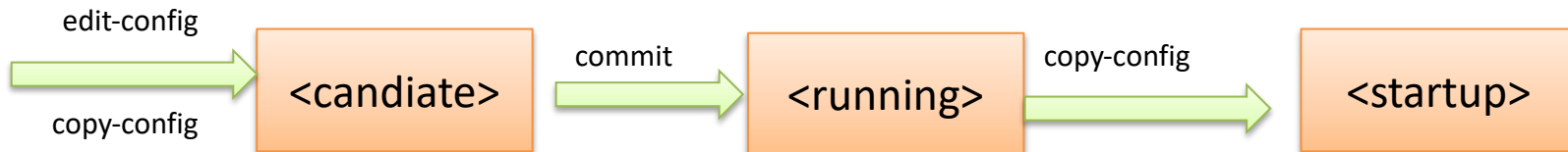
## Example (a)



## 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> exist.

# Why Factory default Setting Capability for **RESTCONF**

- RESTCONF operations are not sufficient to support factory default setting
  - 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.
  - 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.

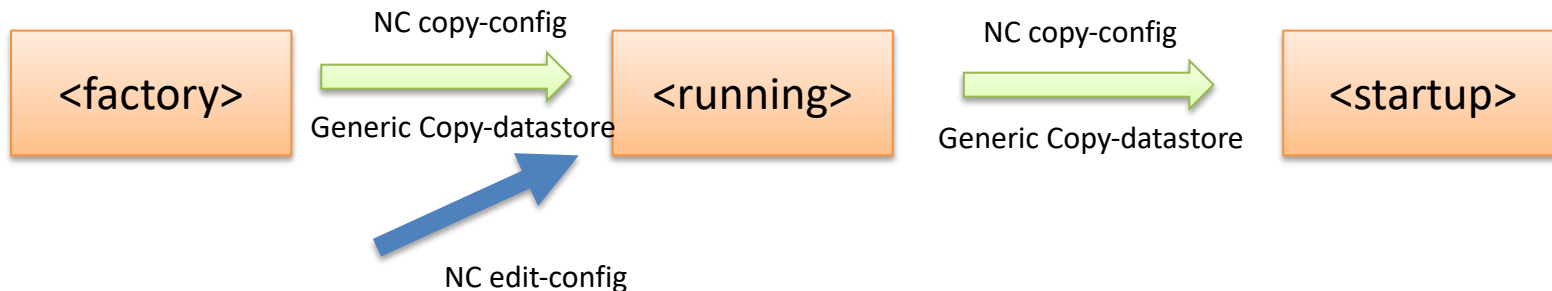
# Proposed solution for factory default

**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 system crash down, system 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

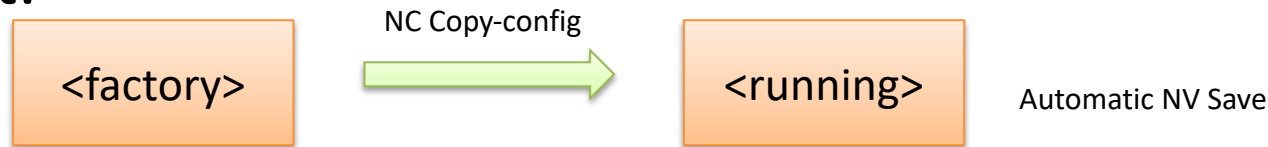


1. We need to make sure that when running is initialized into <factory>, there is no other operation such as edit-config to alter running before <factory> is copied from running to <startup>, NC <lock> and <unlock> can be used.
2. In case of system crash down, system needs to be restarted first and then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

# Why the new operation is needed?

- In some case, system restart is needed in case of system crash down. We need to initialize the device into factory default.



Suppose we use NC copy-config to initialize <running> into <factory>, we need to invoke ietf-system:system restart operation before the <running> is initialized into <factory> or after the <running> is initialized into <factory> But <copy-config> has no indication that whether system restart is needed.

In case ietf-system:system restart operation is invoked before the <running> is initialized into <factory>, zero touch bootstrapping procedure can be used to initialize the device into factory default. In this case, the source can be set to URL or the source can be set to nothing.

In case multiple datastores need to be initialized into <factory, target list can be specified in the generic <copy-datastore> operation.

- 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.
  - The generic copy-datastore can be applied to RESTCONF, NETCONF and other protocol at datastore level.

# Mailing list Discussion

- 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
  - Issue 1: Is factory-restore RPC needed?
    - NC/RC operations + reboot operations can be resumed, but reboot seems unspecified
    - letf-system: system-restart can fulfil purpose of reboot
  - Issue 2: what is conceptual dataflow between the datastores
    - How to make <running> consistent with <startup> since running altering will trigger update of <startup>
  - Issue 3: do we need to generic operation for datastore copy?
    - a few supports to define generic operation.
  - Issue 4: difference between system-restart and boot
    - Is restart in the scope of this draft?

# Next Step

- Request to accept draft as WG item
  - Got already supporter on the list