

**Factory default Setting Capability**  
**draft-wu-netconf-restconf-factory-restore-02**

Abstract

This document defines capability based extension to NETCONF and RESTCONF protocol that allows a client to configure newly deployed devices with just its preconfigured initial state (i.e., factory default settings) during zero touch bootstrapping process or reset the device with the content of factory default setting configuration.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 18, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">1.1.</a>	Terminology . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Limitations of NETCONF and RESTCONF protocol . . . . .	<a href="#">3</a>
<a href="#">2.1.</a>	NETCONF . . . . .	<a href="#">3</a>
<a href="#">2.2.</a>	RESTCONF . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Datastore . . . . .	<a href="#">4</a>
<a href="#">3.1.</a>	The factory default Datastore Resource . . . . .	<a href="#">4</a>
<a href="#">4.</a>	Factory-default capability . . . . .	<a href="#">4</a>
<a href="#">4.1.</a>	Overview . . . . .	<a href="#">4</a>
<a href="#">4.2.</a>	Dependencies . . . . .	<a href="#">5</a>
<a href="#">4.3.</a>	Capability Identifier . . . . .	<a href="#">5</a>
<a href="#">4.4.</a>	New operations . . . . .	<a href="#">5</a>
<a href="#">4.5.</a>	Modifications to Existing Operations . . . . .	<a href="#">5</a>
<a href="#">4.5.1.</a>	get-config, get-data,copy-config operation . . . . .	<a href="#">5</a>
<a href="#">5.</a>	YANG Module . . . . .	<a href="#">6</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">9</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">9</a>
<a href="#">8.</a>	Acknowledges . . . . .	<a href="#">9</a>
<a href="#">9.</a>	Normative References . . . . .	<a href="#">10</a>
	Authors' Addresses . . . . .	<a href="#">11</a>

**[1.](#) Introduction**

Manual configuration on a device can be a massively time-consuming process, especially when it's applied across multiple devices that need to be installed, configured, or maintained in the large scale network environments. On the other hand, when the device that has been installed doesn't work or face system fatal error, resetting the device is a helpful workaround. However, once you reset the device such as a router, all the settings and configurations that have been previously saved on your device will be removed. This means that you will need to set up the device again.

This document defines capability based extension to NETCONF and RESTCONF protocol that allows a client to configure newly deployed devices with just its preconfigured initial state (i.e., factory default settings) during zero touch bootstrapping process or or reset any candidate configuration with the content of factory default setting configuration.

**[1.1.](#) Terminology**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP

14 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [[RFC8342](#)] and are not redefined here:

- o operational state datastore
- o running configuration datastore
- o intended configuration datastore

The following term is defined in this document as follows:

- o factory default datastore

## **2. Limitations of NETCONF and RESTCONF protocol**

### **2.1. NETCONF**

NETCONF <delete-config> operation only can delete <startup> configuration datastore and reset the <startup> to its factory default but can not delete other datastores such as <running> and reset to its factory default.

NETCONF <copy-config> operation 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 configuration source.

NETCONF <discard-changes> operation can be used to reset candidate configuration to current running datastore but can not be used to return the device to factory default without new factory configuration source.

### **2.2. RESTCONF**

RESTCONF uses HTTP methods such as HTTP POST, PUT, PATCH, and DELETE to provide CRUD operations on a conceptual datastore containing YANG-defined data, which is compatible with a server that implements NETCONF datastores. As described in [[RFC8040](#)], the HTTP PUT method on the datastore resource can be used to replace the entire content of the datastore, however it can not be used to return any datastore (e.g., <startup>) to factory default setting, especially when the RESTCONF server is implemented in a device that does not have NETCONF support. One of reasons is RESTCONF doesn't support URL capability.

### **3. Datastore**

#### **3.1. The factory default Datastore Resource**

This document introduces a new datastore resource named 'Factory default setting' that represents datastore with its preconfigured initial state. This datastore resource is available using the following resource path:

```
{+restconf}/ds/ietf-factory-default:factory-default
```

ietf-factory-default:factory-default path component is encoded as an "identity" according to the JSON encoding rules for identities, defined in [Section 4 of \[RFC7951\]](#). Such an identity MUST be derived from the "datastore" identity defined in the "ietf-datastores" YANG module [[RFC8342](#)].

When the factory default configuration is made accessible to the client, the Factory default datastore is essentially a read only datastore.

On devices that support non-volatile storage, the contents of <factory-default> will typically persist across reboots via that storage. The contents of the factory default datastore can be retrieved by means of <get>, <get-config>, <get-data> NETCONF operations and be used to load the entire factory default configuration into target datastore by means of <copy-config> NETCONF operation.

At reboot time, in case all writeable datastores that needs to be resored as factory default setting, the reset operation can be used by the client to trigger the device to load the saved factory default configuration into all writable target datastores in the device and make configuration update to take effect.

### **4. Factory-default capability**

#### **4.1. Overview**

A server that supports the factory-default capability can perform <get-config>, <get-data>, <copy-config><reset> operation as defined in this document.

A server implementing the :factory-default capability:

- o MUST support the ability to receive <rpc> messages that include a factory-default element and perform an operation accordingly.

- o MUST support the ability to include a factory-default element in the <rpc-reply> messages that it transmits.

#### **4.2. Dependencies**

None.

#### **4.3. Capability Identifier**

The :factory-default capability is identified by the following capability string:

```
urn:ietf:params:netconf:capability:factory-default:1.0
```

#### **4.4. New operations**

```
<reset>
```

The <reset> RPC is used to reset the device to factory default setting and make configuration update to take effect.

This operation reverts the contents of any writable datastore in the device to the contents of the factory configuration.

#### **4.5. Modifications to Existing Operations**

##### **4.5.1. get-config, get-data, copy-config operation**

The :factory default capability modifies the <copy-config> <get-config>, <get-data> operations to accept the <factory> element as a <source>, i.e., a new <factory> XML element is added to the input for the <copy-config> <get-config>, <get-data> operations. If the <factory> element is present, it controls the Particular configuration to copy/retrieve from and set the configuration source for copy/retrieve operation as <factory> datastore. For <copy-config> operation, the server MUST reset the target datastore to factory default setting according to the value of this element and return <ok> element in the NETCONF <rpc-reply> messages in success case.

If the target datastore exists, it is overwritten. Otherwise, a new one is created, if allowed. If an <copy-config> operation is invoked on a non-writable datastore, then an error is returned, as specified in "ietf-netconf-nmda".

## Example:

The client(e.g.,NMS) might send the following POST request message to invoke the "copy-config" RPC operation:

```
POST /restconf/operations/ietf-netconf:copy-config HTTP/1.1
Host: example.com
Content-Type: application/yang-data+xml
<input xmlns="https://example.com/ns/ietf-factory-default ">
  <source>
    <factory/>
  </source>
  <target>
    <running/>
  </target>
</input>
```

In this request, <source> element MUST be specified as <factory>. The server will use HTTP POST method to retrieve content of <source> corresponding to factory default setting datastore and copy the entire content to <target>. In successful case, the server might respond as follows:

```
HTTP/1.1 204 No Content
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server
```

## 5. YANG Module

```
<CODE BEGINS> file "ietf-factory-default@2018-09-03.yang"
module ietf-factory-default {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-factory-default";
  prefix fd;

  import ietf-datastores {
    prefix ds;
  }
  import ietf-netconf-acm {
    prefix nacm;
  }
  import ietf-netconf {
    prefix nc;
  }

  organization
    "IETF NETCONF (Network Configuration) Working Group";
```

## contact

"WG Web: <<https://tools.ietf.org/wg/netconf/>>

WG List: <<mailto:netconf@ietf.org>>

WG Chair: Kent Watsen  
<<mailto:kwatsen@juniper.net>>

WG Chair: Mahesh Jethanandani  
<<mailto:mjethanandani@gmail.com>>

Editor: Qin Wu  
<<mailto:bill.wu@huawei.com>>;

## description

"This module defines operations that implement factory-default capability in both NETCONF and RESTCONF protocol.

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in [Section 4.c](#) of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and 'OPTIONAL' in the module text are to be interpreted as described in [RFC 2119](#) (<https://tools.ietf.org/html/rfc2119>).

This version of this YANG module is part of RFC XXXX (<https://tools.ietf.org/html/rfcXXXX>); see the RFC itself for full legal notices.";

```
revision 2018-09-03 {  
  description  
    "Initial revision."  
    reference "RFC XXXX: Factory default Setting Capability for RESTCONF";  
}
```

```
identity factory {  
  
  base ds:datastore;  
  description  
    "The factory default configuration datastore."  
}
```

```
rpc reset {
  nacm:default-deny-all;
  description
    "Request to reset the device to factory default setting
    and make configuration update to take effect.
    A server SHOULD send an rpc reply to the client before
    reset the device.";
}

/*
 * Augment the copy-config operation with a
 * "factory" datastore parameter.
 */

augment "/nc:copy-config/nc:input/nc:source/nc:config-source" {
  description
    "Add factory default Datastore as source.";
  leaf factory {
    type ds:datastore-ref;
    must "derived-from-or-self(current(), 'fd:factory')" {
      error-message "config source is only applicable to factory.";
    }
  }
  description
    "The factory configuration is the config source.

    If the copy-config operation is not supported by the server on the
    specified target datastore, then the server MUST return an <rpc-
error>
    element with an <error-tag> value of 'invalid-value'.";
}
}
/*
 * Augment the get-config operation with a
 * "factory" datastore parameter.
 */

augment "/nc:get-config/nc:input/nc:source/nc:config-source" {
  description
    "Add factory default Datastore as source.";
  leaf factory {
    type ds:datastore-ref;
    must "derived-from-or-self(current(), 'fd:factory')" {
      error-message "config source is only applicable to factory.";
    }
  }
  description
    "The factory configuration is the config source.
    If the get-config operation is not supported by the server on the
    specified target datastore, then the server MUST return an <rpc-
error>
```





```
        element with an <error-tag> value of 'invalid-value.'";
    }
}
}
<CODE ENDS>
```

## **6. IANA Considerations**

This document registers one URI in the IETF XML Registry [[RFC3688](#)]. The following registration has been made:

URI: urn:ietf:params:xml:ns:yang:ietf-factory-default

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers one YANG module in the YANG Module Names Registry [[RFC6020](#)]. The following registration has been made:

name: ietf-factory-default

namespace: urn:ietf:params:xml:ns:yang:ietf-factory-default

prefix: fd

RFC: xxxx

## **7. Security Considerations**

[RFC6241] provides security considerations for the base NETCONF message layer and the base operations of the NETCONF protocol. Security considerations for the NETCONF transports are provided in the transport documents.

In addition, it is important to recognize that <reset> to the startup or running configurations is a sensitive provisioning operation, such global operations MUST disallow the changing of information that an individual does not have authorization to perform.

## **8. Acknowledges**

Thanks to Juergen Schoenwaelder, Ladislav Lhotka, Rohit R Ranade to review this draft and provide important input to this document.

## 9. Normative References

- [I-D.ietf-netconf-zerotouch]  
Watsen, K., Abrahamsson, M., and I. Farrer, "Zero Touch Provisioning for Networking Devices", [draft-ietf-netconf-zerotouch-25](#) (work in progress), September 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", [RFC 5277](#), DOI 10.17487/RFC5277, July 2008, <<https://www.rfc-editor.org/info/rfc5277>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6021](#), DOI 10.17487/RFC6021, October 2010, <<https://www.rfc-editor.org/info/rfc6021>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6470] Bierman, A., "Network Configuration Protocol (NETCONF) Base Notifications", [RFC 6470](#), DOI 10.17487/RFC6470, February 2012, <<https://www.rfc-editor.org/info/rfc6470>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

#### Authors' Addresses

Qin Wu  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: [bill.wu@huawei.com](mailto:bill.wu@huawei.com)

Ye Niu  
Huawei

Email: [niuye@huawei.com](mailto:niuye@huawei.com)