

NETCONF Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2019

W. Zheng
Q. Wu
Huawei
July 1, 2018

Inline Action Capability for NETCONF
draft-zheng-netconf-inline-action-capability-01

Abstract

This document defines capability based extension to NETCONF protocol that enables modification of <edit-config> operation and <edit-data> operation to accept action parameters and attributes and allows multiple sub-operations with inline action operation that apply to either different or the same conceptual node in the underlying data model in one transaction.

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 January 2, 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

1. Introduction	2
1.1. Terminology	3
2. Inline-Action Capability	3
2.1. Description	3
2.2. Dependencies	3
2.3. Capability Identifier	3
2.4. New Operations	3
2.5. Modifications to Existing Operations	3
2.5.1. <edit-config> and <edit-data>	3
3. Security Considerations	5
4. IANA Considerations	6
4.1. NETCONF Capability URN	6
5. Normative References	6
Authors' Addresses	7

1. Introduction

YANG 1.1 define the syntax and semantics of version 1.1 of the YANG language, which can be used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols. One key Difference from YANG 1.0, is a new statement "action", is added to YANG 1.1 to define operations connected to a specific container or list data node in a datastore. However which data node is applied to which configuration datastore is not specified under "action".

The <edit-data> operation defined in [I-D.ietf-netconf-nmda-netconf] and the <edit-config> operation defined in [RFC6241], are used to changes the contents of a writable Datastore. Containers and List entries can be created, deleted, replaced, and modified through <edit-config> by using the "operation" attribute in the container's and List's XML element. However the action is not part of <config> element in either <edit- data> operation or <edit-config> operation. Therefore the action operation and <edit-data> operation or <edit-config> operation connected to the same data node can not automatically handled in sequence in one transaction.

This document defines capability based extension to NETCONF protocol that enables modification of <edit-config> operation and <edit-data> operation to accept action parameters and allows multiple sub-operations with inline action operation that apply to different or same conceptual node in the underlying data model in one transaction.

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.

2. Inline-Action Capability

2.1. Description

The `:inline-action` capability indicates that the device supports Inline-action operation within `<edit-config>` and `<edit-data>` operation on writable configuration datastore. In other words, the device supports `<inline-action>` operation is included in `<edit-config>` and `<edit-data>` operations.

2.2. Dependencies

None.

2.3. Capability Identifier

The `:inline-action` capability is identified by the following capability string:

```
urn:ietf:params:netconf:capability:inline-action:1.1
```

2.4. New Operations

None.

2.5. Modifications to Existing Operations

2.5.1. `<edit-config>` and `<edit-data>`

The `:inline-action:1.1` capability modifies the `<edit-config>` `<edit-data>` operation to accept the `<action>` parameter and `<action>` attribute value within operation attribute.

As described in [RFC6241], "operation" attribute is defined in a element within `<config>` subtree and identify the point in the configuration to perform the operation and MAY appear on multiple elements throughout the `<config>` subtree. In this document, a new "operation" attribute value is added as follows:

inline-action: The configuration data identified by the element containing this attribute is accompanied with action operation applied to a subset of configuration within <config> subtree before edit operation is applied to the same configuration at the corresponding level in the configuration datastore identified by the <target> parameter.

In addition, the inline-action operation attribute and other "operation" attributes can apply to the same conceptual nodes in the underlying data model. The assumption is the inline-action operation attribute and other "operation" attributes applied to the same conceptual nodes will not cause unexpected operation results.

As described in [RFC6241], the config subtree is expressed as A hierarchy of configuration data as defined by one of the device's data models. The contents MUST follow the constraints of that data model, as defined by its capability definition. If inline action capability is supported, the config subtree may contain a schema node with the name "input" and a schema node with the name "output" connected to a specific container or list data node containing action element in a datastore.

Example:

```
container interfaces {
  list interface {
    key "name";
    config true;

    leaf name {
      type string;
    }

    leaf mtu {
      type uint32;
    }
  }
  action ifstatenable {
    input {
      leaf enable {
        type boolean;
        mandatory true;
      }
    }
  }
}
```

Enable ifstatistics on 1000 interfaces from the running configuration before setting the MTU to 1500 on an interface named "Ethernet0/0"

and 1000 on an interface named "Ethernet0/1" in the running configuration:

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.1">
      <top xmlns="http://example.com/schema/1.2/config">
        <interfaces>
          <interface xc:operation="merge">
            <name>Ethernet0/0</name>
            <mtu>1500</mtu>
          </interface>
          <interface>
            <name>Ethernet0/1</name>
            <mtu>1000</mtu>
          </interface>
          <action xmlns="http://example.com/schema/1.2/config">
            <ifstatenable xc:operation="action">
              <input>
                <enable>true</enable>
              </input>
            </ifstatenable>
          </action>
        </interfaces>
      </top>
    </config>
  </edit-config>
</rpc>

<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <ok/>
</rpc-reply>
```

3. Security Considerations

This document does not introduce any security vulnerability besides on defined in [RFC6241].

4. IANA Considerations

4.1. NETCONF Capability URN

IANA has created and now maintains a registry "Network Configuration Protocol (NETCONF) Capability URNs" that allocates NETCONF capability identifiers. Additions to the registry require IETF Standards Action.

IANA has added the following capabilities to the registry:

```
Index
  Capability Identifier
-----
:inline-action:1.1
  urn:ietf:params:netconf:capability:inline-action:1.1
```

5. Normative References

- [I-D.ietf-netconf-nmda-netconf]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K.,
and R. Wilton, "NETCONF Extensions to Support the Network
Management Datastore Architecture", draft-ietf-netconf-
nmda-netconf-06 (work in progress), May 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>>.
- [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>>.

Authors' Addresses

Walker Zheng
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: zhengguangying@huawei.com

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

Email: bill.wu@huawei.com