Network Working Group   M. Bjorklund
Internet-Draft   Tail-f Systems
Intended status: Standards Track   J. Schoenwaelder
Expires: January 5, 2018   Jacobs University
                                                           P. Shafer
                                                          K. Watsen
                                                   Juniper Networks
                                                          R. Wilton
                                                      Cisco Systems
                                                       July 4, 2017

NETCONF Model for NMDA
draft-dsdt-nmda-netconf-00

Abstract

   The "Network Management Datastore Architecture" (NMDA) improves on
   NETCONF by adding new features to give more accurate handling of
   configuration and operational data. These include ability to inspect
   the current operational values of configuration data, allowing
   clients to use identical paths for retrieving the configured values
   and the operational values. These new features require additional
   operations in network management applications such as NETCONF. This
   draft details those new operations.

Status of This Memo

Copyright Notice

Table of Contents

[1](1). Introduction

   This document provides a YANG model that adds NETCONF ([[RFC6241](RFC6241)])
   support for the emerging "Network Management Datastore Architecture"
   (NMDA) [I-D.ietf-netmod-revised-datastores]. NMDA defines a
   framework for datastores, a fundamental concept binding network
   management data models to network management protocols, enabling data
   models to be written in a network management protocol agnostic way.
   NETCONF operations currently refer to the datastores defined in the
   original model, so new operations are required to allow references to
   the new datastores.

   Operations like <copy-config>, <lock> and <unlock> are augmented to
   allow them to target additional datastores.

   In addition the original <get> operation is deprecated, since the
   information it returns is no longer needed. <get>'s deficiencies
   were a major motivation for the NMDA.

1.1. Keywords

The keywords "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..

1.2. Terminology

This document uses the terminology defined by the NMDA
[I-D.ietf-netmod-revised-datastores].

2. The NMDA Model for NETCONF

This section describes the changes needed for NMDA support. These
changes are contained in a new YANG ([RFC7950]) model
"ietf-netconf-datastores".

These changes include the use of source and target parameters based
on the "datastore" identity defined in the "ietf-datastores" from
[I-D.ietf-netmod-revised-datastores]. The use of identities allows
future expansion in a way that the choice-based strategy from the
original operations (e.g. <get-config>, <edit-config>) do not.

2.1. Operations

Support for the NMDA includes two new operations defined in this
document.

2.1.1. The <get-data> Operation

The <get-data> operation retrieves data from a specific NMDA
datastore. This operation is similar to NETCONF's "get-config"
operation, but adds flexibility in naming the target datastore.

The "source" parameter indicates the datastore which is the source of
the data to be retrieved. This is a datastore identity.

The "get-data" operation mirrors the "filter" parameter of the
"get-config" operation, but it is modified to use "type anydata" for
configuration content, rather than the "get-config"'s use of "type
anyxml".

The "get-data" operation also supports the "with-defaults" parameter
as defined in [RFC6243]. The supported values follow the constraints
given by the "with-defaults" capability.

2.1.1.1. Origin Attribute

The "get-data" operation adds a new boolean parameter named "origin",
which requests that the server return the "origin" information as
detailed in the NMDA. This parameter is only valid for <operational>
and any datastores with identities derived from the "operational"
identity.

Data from <operational> can come from multiple sources. The server
should return the most accurate value for the "origin" attribute as
possible, indicating the source of the operational value.

When encoding the origin attribute for a hierarchy of returned nodes,
the origin attribute may be omitted when the value matches that of
the parent node.

2.1.2. The <edit-data> Operation

The <edit-data> operation changes the contents of a specific
datastore, similar to the <edit-config> operation, but with
additional flexibility in naming the target datastore.

The "target" parameter is a datastore identity that indicates the
desired target datastore where changes should be made.

The "edit-content" parameter from "edit-config" it is modified to
allow use "type anydata" for configuration content, rather than the
"edit-config"'s use of "type anyxml".

The "default-operation" parameter mirrors the parameter of the
"edit-config" operation.

2.2. Augmentations to the Base NETCONF Model

Several of the operations defined in the base NETCONF data model
(ietf-netconf@2011-06-01.yang) will continue to be used under the
NMDA. The <lock>, <unlock>, and <validate> operations are augmented
with a new "datastore" leaf can indicate a desired NMDA datastore.

Only writable datastores can be locked.

2.3. RPCs and Actions

RPC operations and actions can be defined in YANG modules. The
evaluation context for constraints and references in operation and
actions is <operational>.

3. YANG Model

```
<CODE BEGINS> file "ietf-netconf-datastores@2017-06-30.yang"

module ietf-netconf-datastores {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-netconf-datastores";
  prefix ncds;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-datastores {
    prefix ds;
  }
  import ietf-netconf {
    prefix nc;
  }
  import ietf-netconf-with-defaults {
    prefix ncwd;
  }

  organization
    "IETF NETCONF Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/netconf/>

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

     Author:   Martin Bjorklund
           <mailto:mbj@tail-f.com>

     Author:   Juergen Schoenwaelder
           <mailto:j.schoenwaelder@jacobs-university.de>

     Author:   Phil Shafer
           <mailto:phil@juniper.net>

     Author:   Kent Watsen
           <mailto:kwatsen@juniper.net>

     Author:   Rob Wilton
           <rwilton@cisco.com>";
  description
    "This YANG module defines a set of NETCONF operations for the
     Network Management Datastore Architecture (NMDA).

     Copyright (c) 2017 IETF Trust and the persons identified as
```

     revision 2017-06-30 {
       description
         "Initial revision.";
       reference "RFC XXXX: NETCONF Support for NMDA";
     }

     typedef datastore {
       type identityref {
         base ds:datastore;
       }
       description
         "An NMDA datastore.";
       reference "RFC XXXX: Network Management Datastore Architecture";
     }

     rpc get-data {
       description
         "Get data from an NMDA datastore";
       input {
         leaf source {
           type ncds:datastore;
           description
             "Datastore from which to retrieve data.";
         }
         anydata filter {
           description
             "Subtree or XPath filter to use.";
           nc:get-filter-element-attributes;
         }
       }
       output {
         anydata data {
           description
             "Copy of the source datastore subset which matched
              the filter criteria (if any). An empty data

```
                   container indicates that the request did not
                   produce any results.";
            }
          }
        }
      rpc edit-data {
        description
          "Edit data in an NMDA datastore.";
        input {
          leaf target {
            type ncds:datastore;
            description
              "Datastore which data affects.";
          }
          leaf default-operation {
            type enumeration {
              enum "merge" {
                description
                  "The default operation is merge.";
              }
              enum "replace" {
                description
                  "The default operation is replace.";
              }
              enum "none" {
                description
                  "There is no default operation.";
              }
            }
            default "merge";
            description
              "The default operation to use.";
          }
          uses ncwd:with-defaults-parameters;
          choice edit-content {
            mandatory true;
            description
              "The content for the edit operation.";

            anydata config {
              description
                "Inline Config content.";
            }
            leaf url {
              if-feature nc:url;
              type inet:uri;
              description
                "URL based config content.";
```

```
            }
          }
        }
      }

      /*
       * Augment the lock and unlock operations with a
       * "datastore" parameter.
       */

      augment "/nc:lock/nc:input/nc:target/nc:config-target" {
        description
          "Add NMDA Datastore as target.";
        leaf datastore {
          type ncds:datastore;
          description
            "Datastore to lock.";
        }
      }
      augment "/nc:unlock/nc:input/nc:target/nc:config-target" {
        description
          "Add NMDA Datastore as target.";
        leaf datastore {
          type ncds:datastore;
          description
            "Datastore to unlock.";
        }
      }

      /*
       * Augment the validate operation with a
       * "datastore" parameter.
       */

      augment "/nc:validate/nc:input/nc:source/nc:config-source" {
        description
          "Add NMDA Datastore as source.";
        leaf datastore {
          type ncds:datastore;
          description
            "Datastore to validate.";
        }
      }
    }

  <CODE ENDS>
```

4. IANA Considerations

This document has no actions for IANA.

5. Security Considerations

This document has no security considerations.

6. Informative References

[I-D.ietf-netmod-revised-datastores]
          Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K.,
          and R. Wilton, "Network Management Datastore
          Architecture", draft-ietf-netmod-revised-datastores-03
          (work in progress), July 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
          Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/
          RFC2119, March 1997,
          <http://www.rfc-editor.org/info/rfc2119>.

[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,
          <http://www.rfc-editor.org/info/rfc6241>.

[RFC6243] Bierman, A. and B. Lengyel, "With-defaults Capability for
          NETCONF", RFC 6243, DOI 10.17487/RFC6243, June 2011,
          <http://www.rfc-editor.org/info/rfc6243>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",
          RFC 7950, DOI 10.17487/RFC7950, August 2016,
          <http://www.rfc-editor.org/info/rfc7950>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
          2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
          May 2017, <http://www.rfc-editor.org/info/rfc8174>.

Appendix A. Examples

Authors' Addresses

   Martin Bjorklund
   Tail-f Systems

   Email: mbj@tail-f.com

Juergen Schoenwaelder
Jacobs University

Email: j.schoenwaelder@jacobs-university.de

Phil Shafer
Juniper Networks

Email: phil@juniper.net

Kent Watsen
Juniper Networks

Email: kwatsen@juniper.net

Robert Wilton
Cisco Systems

Email: rwilton@cisco.com