NETCONF	A. Bierman	
Internet-Draft	InterWorking Labs	
Intended status: Standards Track	June 11, 2010	
Expires: December 13, 2010		

TOC

NETCONF System Monitoring
draft-bierman-netconf-system-monitoring-00

Abstract

The NETCONF protocol provides mechanisms to manipulate configuration datastores. However, client applications often need to examine system information to determine the appropriate configuration requirements. In addition, common system events such as a change in system capabilities may impact management applications. Standard mechanisms are needed to support the monitoring of the managed system supported by a NETCONF server. This document defines a YANG module for the monitoring of system information, which allows a NETCONF client to identify system properties and receive notifications for system events.

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 http://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 December 13, 2010.

Copyright Notice

Copyright (c) 2010 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 (http://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
    1.1. Terminology
2. YANG Module for System Monitoring
    2.1. Overview
        2.1.1. <system> Container
        2.1.2. Notifications
    2.2. Definitions
3. IANA Considerations
4. Security Considerations
5. Acknowledgements
6. References
    6.1. Normative References
    6.2. Informative References
Appendix A. Change Log
    A.1. 00
§ Author's Address

---

1. Introduction

TOC

The NETCONF protocol [RFC4741] (Enns, R., “NETCONF Configuration Protocol,” December 2006.) provides mechanisms to manipulate configuration datastores. However, client applications often need to examine system information to determine the appropriate configuration requirements. In addition, common system events such as a change in system capabilities may impact management applications. Standard mechanisms are needed to support the monitoring of the managed system supported by a NETCONF server. This document defines a YANG module [I-D.ietf-netmod-yang] (Bjorklund, M., “YANG - A data modeling language for the Network Configuration Protocol (NETCONF),” June 2010.) for the monitoring of system information, which allows a NETCONF client to identify system properties and receive notifications [RFC5277] (Chisholm, S. and H. Trevino, “NETCONF Event Notifications,” July 2008.) for system events.

TOC

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).
The following terms are defined in [RFC4741] (Enns, R., "NETCONF Configuration Protocol," December 2006.):

- *client
- *datastore
- *operation
- *server

The following terms are defined in [RFC5277] (Chisholm, S. and H. Trevino, "NETCONF Event Notifications," July 2008.):

- *event
- *stream
- *subscription

The following term is defined in [I-D.ietf-netmod-yang] (Bjorklund, M., "YANG - A data modeling language for the Network Configuration Protocol (NETCONF)," June 2010.):

- *data node

---

2. YANG Module for System Monitoring

TOC

2.1. Overview

TOC

The following YANG module defines data node definitions suitable for use with NETCONF operations such as <get>, <get-config>, and <copy-config>. In addition, a small number of system events are defined for use within the NETCONF stream, and accessible to clients via the subscription mechanism in [RFC5277] (Chisholm, S. and H. Trevino, "NETCONF Event Notifications," July 2008.).

The YANG language is defined in [I-D.ietf-netmod-yang] (Bjorklund, M., "YANG - A data modeling language for the Network Configuration Protocol (NETCONF)," June 2010.). The NETCONF operations are defined in YANG in [RFC4741] (Enns, R., "NETCONF Configuration Protocol," December 2006.).

2.1.1. <system> Container

TOC

The <system> element provides commonly used vendor-specific information to identify and control a managed system:

- *sys-name: The name of the managed system.
- *sys-current-date-and-time: The current time as known to the managed system.
- *sys-boot-date-and-time: The time when the system last restarted.
- *sys-server-id: A vendor-specific string identifying the NETCONF server implementation.
- *uname: A container of common system naming information, such as the release, version, machine, and nodename of the system.

2.1.2. Notifications

TOC

This module defines some system events to notify a client application that the system state has changed.

- *sys-startup: Generated during a system restart. Lists any errors that were encountered while loading the <running> datastore during system initialization.
- *sys-config-change: Generated when the <running> configuration datastore is changed. Summarizes each edit being reported.
- *sys-capability-change: Generated when the NETCONF server capabilities are changed. Indicates which capabilities have been added, deleted, and/or modified.
- *sys-session-start: Generated when the NETCONF session is started. Indicates the identity of the user that started the session.

*sys-session-end: Generated when the NETCONF session is terminated. Indicates the identity of the user that owned the session, and why the session was terminated.

*sys-conformed-commit: Generated when the NETCONF confirmed-commit event occurs. Indicates the current state of the confirmed-commit operation in progress.

2.2. Definitions

TOC

```
<CODE BEGINS> file="netconf-system@2010-06-10.yang"

module netconf-system {

   namespace "urn:ietf:params:xml:ns:yang:netconf-system";

   prefix ncsys;

   import ietf-yang-types { prefix yang; }
   import ietf-inet-types { prefix inet; }

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

   contact
    "WG Web:   <http://tools.ietf.org/wg/netconf/>
     WG List:  <mailto:netconf@ietf.org>

     WG Chair: Bert Wijnen
               <mailto:bertietf@bwijnen.net>

     WG Chair: Mehmet Ersue
               <mailto:mehmet.ersue@nsn.com>

     Editor: Andy Bierman
             <mailto:andyb@iwl.com>";

   description
    "This module defines an YANG data model for use with the
     NETCONF protocol that allows the NETCONF client to monitor
     common system information and receive common system events.

     Copyright (c) 2010 IETF Trust and the persons identified as
     the document authors.  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
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see
     the RFC itself for full legal notices.";
   // RFC Ed.: replace XXXX with actual RFC number and remove this note

   // RFC Ed.: remove this note
   // Note: extracted from draft-bierman-netconf-system-00.txt
```

```
 revision 2010-06-10 {
   description
     "Initial version.";
   reference
    "RFC XXXX: NETCONF System Monitoring";
 }
 // RFC Ed.: replace XXXX with actual
 // RFC number and remove this note

typedef error-type-type {
  description "NETCONF Error Type";
  type enumeration {
    enum transport {
      description "Transport layer error";
    }
    enum rpc {
      description "Operation layer error";
    }
    enum protocol {
      description "Protocol layer error";
    }
    enum application {
      description "Application layer error";
    }
  }
}

typedef error-tag-type {
  description "NETCONF Error Tag";
  type enumeration {
    // descriptions TBD; normative text in RFC 4741
    enum in-use;
    enum invalid-value;
    enum too-big;
    enum missing-attribute;
    enum bad-attribute;
    enum unknown-attribute;
    enum missing-element;
    enum bad-element;
    enum unknown-element;
    enum unknown-namespace;
    enum access-denied;
    enum lock-denied;
    enum resource-denied;
    enum rollback-failed;
    enum data-exists;
    enum data-missing;
    enum operation-not-supported;
```

```
    enum operation-failed;
    enum partial-operation;
  }
}

typedef error-severity-type {
  description "NETCONF Error Severity";
  type enumeration {
    enum error {
      description "Error severity";
    }
    enum warning {
      description "Warning severity";
    }
  }
}

typedef edit-operation-type {
  description
    "NETCONF 'operation' Attribute values";
  type enumeration {
    enum merge;
    enum replace;
    enum create;
    enum delete;
  }
  default "merge";
}

grouping sys-common-session-parms {

  leaf user-name {
    description
      "Name of the user for the session.";
    type string;
  }

  leaf session-id {
    description "Identifier of the session.";
    type uint32;    // nc:session-id-or-zero-type;
    mandatory true;
  }

  leaf remote-host {
    description
      "Address of the remote host for the session.";
    type inet:ip-address;
  }
}
```

```
container system {
  description
    "Basic objects for NETCONF system identification.";

  config false;

  leaf sys-name {
    description "The system name.";
    reference "RFC 3418, sysName object";
    type string;
    mandatory true;
  }

  leaf sys-current-date-time {
    description
      "The current system date and time.";
    type yang:date-and-time;
    mandatory true;
  }

  leaf sys-boot-date-time {
    description
      "The system date and time when the system
      last restarted.";
    type yang:date-and-time;
    mandatory true;
  }

  leaf sys-server-id {
    description
      "The vendor-specific name and version ID string
       for the NETCONF server running on this system.";
    type string;
    mandatory true;
  }

  container uname {
    description
      "Contains the broken out fields from the
      output of the 'uname' command on this machine.";
    leaf sysname {
      type string;
      description
        "The name of the operating system in use.";
    }

    leaf release {
      type string;
      description
```

```
        "The current release level of the operating
         system in use.";
    }

    leaf version {
      type string;
      description
        "The current version level of the operating
         system in use.";
    }

    leaf machine {
      type string;
      description "A description of the hardware in use.";
    }

    leaf nodename {
      type string;
      description
        "The host name of this system, as reported by
        the uname command.";
    }
  } // container uname
} // container system


notification sys-startup {
  description
    "Generated when the system restarts.
    Used for logging purposes, since no
    sessions are actually active when
    the system restarts.";

  leaf startup-source {
    description
      "The system-specific filespec used to load the
       running configuration.  This leaf will only be
       present if there was a startup  configuration file used.";
    type string;
  }

  list boot-error {
    description
      "There will be one entry for each <rpc-error>
      encountered during the load config operation.
      There is no particular order, so no key is defined.
      This list will only be present if the server is configured
      to continue on error during startup, and there were recoverable
      errors encountered during the last restart of the server.";
```

```
 leaf error-type {
   description
     "Defines the conceptual layer that the error occurred.";
   type error-type-type;
   mandatory true;
 }

 leaf error-tag {
   description
     "Contains a string identifying the error condition.";
   type error-tag-type;
   mandatory true;
 }

 leaf error-severity {
   description
     "Contains a string identifying the error severity, as
      determined by the device.";
   type error-severity-type;
   mandatory true;
 }

 leaf error-app-tag {
   description
     "Contains a string identifying the data-model-specific
      or implementation-specific error condition, if one exists.";
   type string;
 }

leaf error-path {
  description
    "Contains the absolute XPath expression identifying
     the element path to the node that is associated with
     the error being reported in a particular <rpc-error>
     element.";
  type yang:xpath1.0;
}

leaf error-message {
  description
    "Contains a string suitable for human display that
     describes the error condition.";
   type string;   // LangString;
 }

 anyxml error-info {
   description
     "Contains protocol- or data-model-specific error content.";
 }
```

```
  } // list boot-error
} // notification sys-startup


notification sys-config-change {
  description
    "Generated when the <running> configuration is changed.";
  uses sys-common-session-parms;

  list edit {
    description
      "An edit record will be present for each distinct
      edit operation on the running config.";
    leaf target {
      type instance-identifier;
      description
        "Topmost node associated with the configuration change.";
    }

    leaf operation {
      type edit-operation-type;
      description "Type of edit operation performed.";
    }
  } // list edit
} // notification sys-config-change


notification sys-capability-change {
  description
    "Generated when a <capability> is added, deleted,
     or modified.";
  container changed-by {
    description
      "Indicates who caused this capability change.
      If caused by internal action, then the
      empty leaf 'server' will be present.
      If caused by a management session, then
      the name, remote host address, and session ID
      of the session that made the change will be reported.";
    choice server-or-user {
      leaf server {
        type empty;
        description
          "If present, the capability change was caused
           by the server.";
      }

      case by-user {
        uses sys-common-session-parms;
```

```
      } // case by-user
    } // choice server-or-user
  } // container changed-by

  leaf-list added-capability {
    type inet:uri;
    description
      "List of capabilities that have just been added.";
  }

  leaf-list deleted-capability {
    type inet:uri;
    description
      "List of capabilities that have just been deleted.";
  }

  leaf-list modified-capability {
    type inet:uri;
    description
      "List of capabilities that have just been modified.";
  }
} // notification sys-capability-change


notification sys-session-start {
  description
    "Generated when a new NETCONF session is started.";
  uses sys-common-session-parms;
} // notification sys-session-start


notification sys-session-end {
  description
    "Generated when a NETCONF session is terminated.";
  uses sys-common-session-parms;

  leaf killed-by {
    when "../termination-reason = 'killed'";
    type uint32;    // nc:session-id-type;
    description
      "Session ID that issued the <kill-session>
      if the session was terminated by this operation.";
  }

  leaf termination-reason {
    type enumeration {
      enum "closed" {
        value 0;
        description
          "The session was terminated with
```

```
          the <close-session> operation.";
      }
      enum "killed" {
        value 1;
        description
          "The session was terminated with
          the <kill-session> operation.";
      }
      enum "dropped" {
        value 2;
        description
          "The session was terminated because
          the SSH session or TCP connection was
          unexpectedly closed.";
      }
      enum "timeout" {
        value 3;
        description
          "The session was terminated because
          of inactivity, either waiting for
          the <hello> or <rpc> messages.";
      }
      enum "bad-start" {
        value 4;
        description "The session startup sequence failed.";
      }
      enum "bad-hello" {
        value 5;
        description
          "The client's <hello> message was
          bad or never arrived.";
      }
      enum "other" {
        value 6;
        description
          "The session was terminated for
          some other reason.";
      }
    }
    mandatory "true";
    description "Reason the session was terminated.";
  }
} // notification sys-session-end


notification sys-confirmed-commit {
  description
    "Generated when a confirmed-commit event occurs.";
  uses sys-common-session-parms;
```

```
    leaf confirm-event {
      description
        "Indicates the event that caused the notification.";
      type enumeration {
        enum "start" {
          value 0;
          description
            "The confirm-commit procedure has started.";
        }
        enum "cancel" {
          value 1;
          description
            "The confirm-commit procedure has been canceled,
            due to the session being terminated.";
        }
        enum "timeout" {
          value 2;
          description
            "The confirm-commit procedure has been canceled,
            due to the confirm-timeout interval expiring.
            The common session parameters will not be present
            in this sub-mode.";
        }
        enum "extend" {
          value 3;
          description
            "The confirm-commit timeout has been extended.";
        }
        enum "complete" {
          value 4;
          description
            "The confirm-commit procedure has been completed.";
        }
      }
      mandatory "true";
    }
  } // notification sys-confirmed-commit

}

<CODE ENDS>
```

TOC

3. IANA Considerations

TBD

---

4. Security Considerations

TOC

This document defines a YANG module for reporting of particular system information and system events. Although unlikely, it is possible that data obtained from this module could be used in an attack of some kind, although no specific information in this module is considered sensitive.
TBD: follow Security Consideration guidelines from new template text.

---

5. Acknowledgements

TOC

Some data node definitions in this document are based on information provided by the unix 'uname' program (origin unknown).
This module is based on the yuma-system.yang module, which can be found at: http://www.netconfcentral.org/modules/yuma-system.

---

6. References

TOC

6.1. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC3688]	Mealling, M., “The IETF XML Registry,” BCP 81, RFC 3688, January 2004 (TXT).
[RFC4741]	Enns, R., “NETCONF Configuration Protocol,” RFC 4741, December 2006 (TXT).
[RFC5277]	Chisholm, S. and H. Trevino, “NETCONF Event Notifications,” RFC 5277, July 2008 (TXT).
[I-D.ietf-netmod-yang]	Bjorklund, M., “YANG - A data modeling language for the Network Configuration Protocol (NETCONF),” draft-ietf-netmod-yang-13 (work in progress), June 2010 (TXT).

[I-D.ietf-netmod-yang-types]	Schoenwaelder, J., "Common YANG Data Types," draft-ietf-netmod-yang-types-09 (work in progress), April 2010 (TXT).

6.2. Informative References

TOC

[RFC3418]	Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)," STD 62, RFC 3418, December 2002 (TXT).

Appendix A. Change Log

TOC

-- RFC Ed.: remove this section before publication.

A.1. 00

TOC

Initial version.

Author's Address

TOC

	Andy Bierman
	InterWorking Labs
	Scotts Valley, CA
	USA
Phone:	+1 831 460 7010
Email:	andyb@iwl.com