

Netconf
Internet-Draft
Intended status: Standards Track
Expires: April 16, 2007

B. Lengyel
Ericsson Hungary
October 13, 2006

Generic Action RPC for Netconf
draft-lengyel-action-rpc-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 16, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Internet-Draft

Generic Action RPC for Netconf

October 2006

Abstract

The NETCONF protocol defines a number of operations to read and write configuration data. However, it does not define actions on the managed nodes. Such actions are useful for 1) operations that do not change the configuration data (e.g. ping) 2) reading or writing a set of data that forms a logical group but might be scattered in different parts of the management data model.

This document defines a new Netconf capability supporting a generic <action> operation. A modeling view of how the content of the generic <action> operation can be defined is also described.

Please send comments to netconf@ops.ietf.org. To subscribe, use netconf-request@ops.ietf.org.

[1.](#) Introduction

The document introduces the Generic Action Capability.

[1.1.](#) Definition of Terms

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 [RFC 2119](#).

Element: An XML Element[XML].

Managed Node: A network node, which supports NETCONF[1] and has access to management instrumentation. This is also known as the NETCONF server.

Managed Object: A collection of one or more Elements that define an abstract thing of interest.

Calling Point: A specific Managed Object in the hierarchical data model that is passed to the action as context information.

[2.](#) Generic Action Capability

[2.1.](#) Overview

The NETCONF protocol defines a number of operations to read and write configuration data. However, it does not define actions on the managed nodes. Such actions are useful for 1) operations that do not change the configuration data (e.g. ping) 2) reading or writing a set of data that form a logical group but might be scattered in different parts of the management data model.

There are two basic ways to define such actions: define each action as an individual new operation or define a generic action operation and define the details of the individual actions elsewhere. One obvious possibility is to define actions in the management data model.

Using generic action operations is a valid solution that we believe will lead to simpler implementation and usage than defining actions as separate individual operations.

Netconf RPCs will often be handled by a number of software entities before they reach their final destination, e.g. intermediate management software, software stacks from third party suppliers, interface handling software, access control enforcers etc. These entities and management software in general will be prepared for different data models but will often not be prepared for changes in the protocol itself: for new node specific action RPCs. New actions might need updates to these software entities.

Using generic actions defined in the data model will make it possible to write generic management software like a MIB browser that can issue such actions and at least check their contents based on the data model (name of the action, number and type of the parameters).

[2.1.1.1.](#) Modeling

There is a need to define what kind of generic actions can be used against a specific managed node. This definition will not be part of the protocol but should be part of the management data model of the node.

This allows the node to specify during the initial capability exchange the information model and actions it supports. Many of the actions will be device/manufacture specific just like the data model itself. It will be easy to handle the action definitions as it is generally accepted that the data model (or at least the non-standardized parts of it) will be different for different products.

[2.1.1.1.1.](#) Action Definition

Each action will probably be defined in the data model. The exact form of this definition is outside the scope of this document. The following is recommended practice.

Name (Mandatory): Each action must have a name to identify it.

Calling-Point (Optional): A specific point in the hierarchical data model that is passed to the action as context information. Often actions will be called against a specific interface, virtual router, etc. In other cases there will not be such context, in which case the calling point might be omitted.

Description (Mandatory): A description of the action.

Parameters (Optional): All parameters included in an action should be defined. For each parameter the following data should be provided:

Data type of the parameter

Optional/Mandatory

Default value

Direction the parameter is sent: IN/OUT/INOUT. OUT and INOUT parameters will be returned in the response message.

Error Results (mandatory if applicable): Possible error results including their format and what causes the error. This item might not be applicable if the action never produces error result.

[2.2.](#) Dependencies

None.

[2.3.](#) Capability Identifier

The generic-action capability is identified by the following capability string:

urn:ietf:params:netconf:capability:generic-action:1.0

[2.4.](#) New Operations

[2.4.1.](#) Action

The operation executes an action on the managed node. An action has a name, an optional calling point and a number of parameters included in the request. From the above data only the name element is mandatory. Optional data should be passed transparently to the

management entity connected to the netconf server.

The following XML Schema defines the action operation:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns:act="urn:ietf:params:netconf:capability:generic-action:1.0"
  targetNamespace=
    "urn:ietf:params:netconf:capability:generic-action:1.0"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  xml:lang="en">
  <xs:annotation>
    <xs:documentation>
      Schema defining the action operation.
```

```

    </xs:documentation>
</xs:annotation>
<xs:import namespace="urn:ietf:params:xml:ns:netconf:base:1.0"
    schemaLocation="urn:ietf:params:xml:ns:netconf:base:1.0"/>

<!-- callingpoint type
    same as filterInlineType but adds some restrictions in
    documentation -->
<xs:complexType name="callingPointType">
    <xs:complexContent>
        <xs:extension base="netconf:filterInlineType">
            <xs:annotation>
                <xs:documentation>
                    The filter must select one specific XML element.
                    Subtree filter: Only a restricted subset of the subtree
                    filter may be used. The subtree filter must select exactly
                    one node based on the use of a content match node. For
                    actions the content match node will select the parent node
                    of the content match node.
                    XPATH filter: Only a restricted subset of XPATH may be
                    used. The location path must start from the root element,
                    use only the child axis and include checks on key elements.
                </xs:documentation>
            </xs:annotation>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<!-- <action> operation -->
<xs:complexType name="actionType">
    <xs:complexContent>
        <xs:extension base="netconf:rpcOperationType">
            <xs:sequence>

```

```

        <xs:element name="actionName" type="xs:string"/>
        <xs:element name="callingPoint" type="act:callingPointType"
            minOccurs="0"/>
        <xs:element name="parameters" "netconf:dataInlineType"/>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

```
<xs:element name="action" type="act:actionType"
  substitutionGroup="netconf:rpcOperation"/>
</xs:schema>
```

The following example action restarts virtual router "router1" immediately.

```
<?xml version="1.0" encoding="UTF-8"?>
<netconf:rpc
  xmlns="urn:ietf:params:netconf:capability:generic-action:1.0"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0"
  message-id="239">
  <action>
    <actionName>restart</actionName>
    <callingPoint type="xpath">
      /top/routing/virtual-router["routerName=router1"]
    </callingPoint>
    <parameters>
      <mymodel:mode xmlns:mymodel="http://mycompany.com/mymodel.xsd">
        immediately
      </mymodel:mode>
    </parameters>
  </action>
</netconf:rpc>
```

Positive Response:

If the action is successful an `<rpc-reply>` is sent. In case the action doesn't return any parameters, the `<rpc-reply>` contains a single `<ok/>` element. If there are any OUT or INOUT parameters, then an `<rpc-reply>` is sent containing a `<data>` element. The data element will contain all OUT and INOUT parameter elements.

The following reply returns a boolean parameter "restart-successful"

and an integer parameter 'run-level'.

```
<rpc-reply message-id="239"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data xmlns:mymodel=
    "http://mycompany.com/mymodel.xsd">
    <restart-successful>true</restart-successful>
    <run-level>5</run-level>
  </data>
</rpc>
```

Negative Response:

In all cases, if an error occurs, an <rpc-reply> is sent that contains an <rpc-error> element.

[2.5.](#) Modifications to Existing Operations

None.

[2.6.](#) Interactions with Other Capabilities

None.

[3.](#) Security Considerations

The same considerations as for the base Netconf Protocol [[1](#)] are valid. It is assumed that actions are only allowed for an authenticated user after passing some access control mechanism.

[4.](#) IANA Considerations

None (except maybe the capability's URI).

Lengyel

Expires April 16, 2007

[Page 10]

Internet-Draft

Generic Action RPC for Netconf

October 2006

[Appendix A](#). Change Log

[draft-00](#) Initial version

5. References

- [1] Enns, R., "NETCONF Configuration Protocol",
[draft-ietf-netconf-prot-12](#) (work in progress), March 2006.

Author's Address

Balazs Lengyel
Ericsson Hungary

Phone: +36-1-4377320

Email: balazs.lengyel@ericsson.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET

ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).