Network Working Group
Internet-Draft
Updates: 6020 (if approved)
Intended status: Standards Track
Expires: April 21, 2014

M. Bjorklund
Tail-f Systems

October 18, 2013

YANG XPath Extensions
draft-bjorklund-netmod-yang-xpath-extensions-00

Abstract

This document introduces new YANG extension statements for defining XPath functions. These functions can be used in XPath expressions in YANG modules and in NETCONF XPath filters. A set of YANG-specific XPath functions are also defined.

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 April 21, 2014.

Table of Contents

1. Introduction

Experience with YANG [RFC6020] for data modeling has shown that using XPath for specifying constraints is very useful. Unfortunately, since XPath 1.0 has a limited set of data types, and the functions in the core function library only operates on these data types, using XPath 1.0 with other data types is often not possible, unless new XPath functions are defined.

This document defines a mechanism to formally define new XPath functions to be used in YANG modules and NETCONF [RFC6241] XPath filters, and introduces a few such XPath functions to be used for the built-in YANG types that cannot be manipulated efficiently with the core XPath functions.

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].

2. Defining and Using XPath Functions

A YANG extension statement "xpath-function" is introduced in the YANG module in Section 5. It is used to define the name, input parameters, and return type of an XPath function. For example:

```
module example-module1 {
    ...
    import ietf-yang-xpath-extensions {
        prefix yangxp;
    }

    yangxp:xpath-function string-reverse {
        yangxp:xpath-argument str {
            yangxp:xpath-type string;
        }
        yangxp:xpath-result string;
        description
            "This function reverses the string 'str' and returns
             the resulting string.";
    }
}
```

2.1. Using an XPath Function in YANG

When an XPath function defined in a YANG module is used from another module, the module that defines the function is imported, and the function is invoked using the syntax "<prefix>:<function-name>", where <prefix> is the prefix of the imported module. For example:

```
module example-module2 {
    namespace "http://example.com/example-module2";
    ...
    import example-module1 {
        prefix ex1;
    }
    ...
    leaf palindrome-of-the-day {
        type string;
        must ". = ex1:string-reverse(.)" {
            error-message "Not a palindrome.";
        }
    }
}
```

2.2. Using an XPath Function in a NETCONF Filter

An XPath function defined in a YANG module can be used in a NETCONF filter by a client if the NETCONF server advertises the :xpath capability, the capability associated with the YANG module "ietf-yang-xpath-extensions", and the capability associated with the module that defines the XPath function.

For example, suppose a NETCONF server advertises the following capabilities in its <hello> message:

```
  <!-- lines wrapped for display purposes only -->

  <capability>
    urn:ietf:params:netconf:capability:xpath:1.0
  </capability>
  <capability>
    urn:ietf:params:xml:ns:yang:ietf-yang-xpath-extensions?
       module=ietf-yang-xpath-extensions
  </capability>
  </capability>
    http://example.com/example-module2?module=example-module2
  </capability>
```

A client can then send the following request to return only interfaces whose names are palindromes:

```
  <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
       message-id="1">
    <get xmlns:ex2="http://example.com/example-module2">
      <filter type="xpath"
              select="/interface[name = ex2:reverse(name)]"/>
    </get>
  </rpc>
```

3. XPath Evaluation Context

This document updates the XPath evaluation context for YANG XPath expressions, defined in Section 6.4.1 in [RFC6020] in the following way:

- The function library is the core function library defined in [XPATH], and a function "current()" that returns a node set with the initial context node, and all functions defined by the "yangxp:xpath-function" statement in the current module, all included submodules, and all imported modules. Functions defined by "yangxp:xpath-function" are referenced as "<prefix>: <function-name>", where <prefix> is the prefix of the module that defines <function-name>.

4. XPath Functions

This document defines four YANG type-specific XPath functions, and one generic XPath function. The functions are formally defined in Section 5.

4.1. Function for the YANG Types "leafref" and "instance-identifier"

The function "deref" returns a node-set containing the node that a node of type "leafref" or "instance-identifier" refers to. For example:

```
  list interface {
      key name;
      leaf name { ... }
      leaf enabled {
          type boolean;
      }
      ...
  }

  leaf mgmt-interface {
      type leafref {
          path "/interface/name";
      }
      must 'yangxp:deref(.)/../enabled = "true"' {
          error-message
              "The management interface cannot be disabled.";
      }
  }
```

4.2. Function for the YANG Type "identityref"

The function "derived-from" checks if a node of type "identityref" is derived from a given identity. For example:

```
module example-interface {
    ...

    identity interface-type;

    identity ethernet {
        base interface-type;
    }

    identity fast-ethernet {
        base ethernet;
    }

    identity gigabit-ethernet {
        base ethernet;
    }

    list interface {
        key name;
        ...
        leaf type {
            type identityref {
                base interface-type;
            }
        }
        ...
    }

    augment "/interface" {
        when 'yangxp:derived-from(type,
                                  "example-interface",
                                  "ethernet")';
        // ethernet-specific definitions here
    }
}
```

4.3. Function for the YANG Type "enumeration"

The function "enum-value" returns the integer value associated with a node of type "enumeration". For example, with this data model:

```
    list alarm {
        ...
        leaf severity {
            type enumeration {
                enum cleared {
                    value 1;
                }
                enum indeterminate {
                    value 2;
                }
                enum minor {
                    value 3;
                }
                enum warning {
                    value 4;
                }
                enum major {
                    value 5;
                }
                enum critical {
                    value 6;
                }
            }
        }
    }
```

the following XPath expression selects only alarms that are of severity "major" or higher:

```
  /alarm[yangxp:enum-value(severity) >= 5]
```

4.4. Function for the YANG Type "bits"

The function "bit-is-set" checks if a node of type "bits" have a given bit set. For example, if an interface has this leaf:

```
  leaf flags {
      type bits {
          bit UP;
          bit PROMISCUOUS
          bit DISABLED;
      }
   }
```

the following XPath expression can be used to select all interfaces with the UP flag set:

```
  /interface[bit-is-set(flags, "UP")]
```

4.5. Function for strings

The function "re-match" checks if a string matches a given regular expression. The regular expressions used are the XML Schema regular expressions [XSD-TYPES]. Note that this includes implicit anchoring of the regular expression at the head and tail. For example:

```
re-match('1.22.333', '\d{1,3}\.\d{1,3}\.\d{1,3}')
```

returns true.

To count all logical interfaces called eth0.<number>, do:

```
count(/interface[re-match(name,'eth0\.\d+')])
```

5. YANG XPath Extensions Module

RFC Ed.: update the date below with the date of RFC publication and remove this note.

<CODE BEGINS> file "ietf-yang-xpath-extensions@2013-10-18.yang"

```
module ietf-yang-xpath-extensions {

  namespace "urn:ietf:params:xml:ns:yang:ietf-yang-xpath-extensions";
  prefix "yangxp";

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

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

     WG Chair: David Kessens
               <mailto:david.kessens@nsn.com>

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

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>";

  description
    "This module contains a collection of YANG extensions
     for defining XPath functions to be used in XPath
     expressions in YANG modules and NETCONF filters.

     Copyright (c) 2013 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
     (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.: update the date below with the date of RFC publication
// and remove this note.
revision 2013-10-18 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG XPath Extensions";
}

/*
 * Extensions
 */

extension xpath-function {
  argument name;
  description
    "This statement introduces an XPath function that can be
     used in 'must' and 'when' XPath expression in YANG modules,
     and in NETCONF filters.

     The statement's argument specifies the name of the XPath
     function.

     When the function is used in a YANG module, the module where
     the function is defined MUST be imported.  The function is
     referred to using the syntax '<prefix>:<name>', where <prefix>
     is the prefix of the module, and <name> is the name of the
     XPath function.

     The following substatements are used:

       +----------------------+-------------+
       | substatement         | cardinality |
       +----------------------+-------------+
       | yangxp:xpath-argument | 0..n        |
       | yangxp:xpath-result   | 1           |
       | description           | 0..1        |
       | reference             | 0..1        |
       +----------------------+-------------+

     The yangxp:xpath-argument statement defines the arguments to
     the XPath function.  The functions takes the arguments in the
     order they are defined in the YANG module.";
}

extension xpath-argument {
  argument name;
  description
```

```
        "This statement defines an argument to an XPath function.

         The statement's argument specifies the name of the XPath
         function's argument.  The mandatory substatement
         yangxp:xpath-type defines the type of the argument.

         The following substatements are used:

           +-----------------------+-------------+
           | substatement          | cardinality |
           +-----------------------+-------------+
           | yangxp:xpath-type     | 1           |
           +-----------------------+-------------+";
    }

    extension xpath-type {
      argument type-name;
      description
        "This statement defines the type of the parent statement's
         XPath object.

         The statement's argument is one of the strings:
         'node-set', 'number', 'string', or 'boolean'.";
    }

    extension xpath-result {
      argument type;
      description
        "This statement defines the type of the XPath function's
         return value.

         The statement's argument is one of the strings:
         'node-set', 'number', 'string', or 'boolean'.";
    }

    /*
     * XPath functions
     */

    /* Function for leafref and instance-identifier */

    yangxp:xpath-function deref {
      yangxp:xpath-argument nodes {
        yangxp:xpath-type node-set;
      }
      yangxp:xpath-result node-set;
      description
        "The deref() function follows the reference defined by the
```

```
          first node in document order in the argument 'nodes', and
          returns the nodes it refers to.

          If the first argument node is of type instance-identifier,
          the function returns a node-set that contains the single node
          that the instance identifier refers to, if it exists. If no
          such node exists, an empty node-set is returned.

          If the first argument node is of type leafref, the function
          returns a node-set that contains the nodes that the leafref
          refers to.

          If the first argument node is of any other type, an empty
          node-set is returned.";
      reference
        "RFC 6020: YANG, Section 9.9 and Section 9.13.";
    }

    /* Function for identityref */

    yangxp:xpath-function derived-from {
      yangxp:xpath-argument nodes {
        yangxp:xpath-type node-set;
      }
      yangxp:xpath-argument module-name {
        yangxp:xpath-type string;
      }
      yangxp:xpath-argument identity-name {
        yangxp:xpath-type string;
      }
      yangxp:xpath-result boolean;
      description
        "The derived-from() function returns true if the first node in
         document order in the argument 'nodes' is a node of type
         identityref, and its value is an identity that is derived from
         the identity 'identity-name' defined in the YANG
         'module-name'; otherwise it returns false.";
      reference
        "RFC 6020: YANG, Section 9.10.";
    }

    /* Function for enumeration */

    yangxp:xpath-function enum-value {
      yangxp:xpath-argument nodes {
        yangxp:xpath-type node-set;
      }
      yangxp:xpath-result number;
```

```
      description
        "The enum-value() function checks if the first node in
         document order in the argument 'nodes' is a node of type
         enumeration, and returns the enum's integer value.  If the
         'nodes' node-set is empty, or if the first node 'nodes' is
         not of type enumeration, it returns NaN.";
      reference
        "RFC 6020: YANG, Section 9.6.4.2.";
    }

    /* Function for bits */

    yangxp:xpath-function bit-is-set {
      yangxp:xpath-argument nodes {
        yangxp:xpath-type node-set;
      }
      yangxp:xpath-argument bit-name {
        yangxp:xpath-type string;
      }
      yangxp:xpath-result boolean;
      description
        "The bit-is-set() function returns true if the first node in
         document order in the argument 'nodes' is a node of type
         bits, and its value has the bit 'bit-name' set; otherwise
         it returns false.";
      reference
        "RFC 6020: YANG, Section 9.7.4.";
    }

    /* String function */

    yangxp:xpath-function re-match {
      yangxp:xpath-argument subject {
        yangxp:xpath-type string;
      }
      yangxp:xpath-argument pattern {
        yangxp:xpath-type string;
      }
      yangxp:xpath-result boolean;
      description
        "The re-match() function returns true if the 'subject' string
         matches the regular expression 'pattern'; otherwise it
         returns false.

         The regular expressions used are the XML Schema regular
         expressions.";

      reference
```

```
          "http://www.w3.org/TR/xmlschema-2/#regexs";
      }

   }

   <CODE ENDS>
```

6. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-yang-xpath-extensions

Registrant Contact: The NETMOD WG of the IETF.

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

This document registers a YANG module in the YANG Module Names registry [RFC6020].

```
name:         ietf-yang-xpath-extensions
namespace:    urn:ietf:params:xml:ns:yang:ietf-yang-xpath-extensions
prefix:       yangxp
reference:    RFC XXXX
```

7. Security Considerations

This document defines a formal mechanism for defining XPath functions in YANG data models, and has no security impact on the Internet.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[XPATH] Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0", World Wide Web Consortium Recommendation REC-xpath-19991116, November 1999, <http://www.w3.org/TR/1999/REC-xpath-19991116>.

[XSD-TYPES] Malhotra, A. and P. Biron, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>.

8.2. Informative References

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

Author's Address

Martin Bjorklund
Tail-f Systems

Email: mbj@tail-f.com