Network Working Group	M. Ellison	
Internet-Draft	Ellison Software Consulting	
Intended status: Standards Track	B. Natale	
Expires: September 2, 2010	MITRE	
	March 01, 2010	

Expressing SNMP SMI Datatypes in XML Schema Definition Language
draft-ietf-opsawg-smi-datatypes-in-xsd-06.txt

Abstract

This memo defines the IETF standard expression of Structure of Management Information (SMI) base datatypes in Extensible Markup Language (XML) Schema Definition (XSD) language. The primary objective of this memo is to enable the production of XML documents that are as faithful to the SMI as possible, using XSD as the validation mechanism.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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/1id-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 September 2, 2010.

Copyright Notice

Table of Contents

1. Introduction

Numerous use cases exist for expressing the management information described by SMI Management Information Base (MIB) modules in XML [XML] (World Wide Web Consortium, "Extensible Markup Language (XML) 1.0," February 1998.). Potential use cases reside both outside and within the traditional IETF network management community. For example, developers of some XML-based management applications may want to incorporate the rich set of data models provided by MIB modules. Developers of other XML-based management applications may want to access MIB module instrumentation via gateways to SNMP agents. Such applications benefit from the IETF standard mapping of SMI datatypes to XML datatypes via XSD [XMLSchema] (World Wide Web Consortium, "XML Schema Part 1: Structures Second Edition," October 2004.), [XSDDatatypes] (World Wide Web Consortium, "XML Schema Part 2: Datatypes Second Edition," October 2004.).

MIB modules use SMIv2 [RFC2578] (McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)," April 1999.) to describe data models. For legacy MIB modules, SMIv1 [RFC1155] (Rose, M. and K. McCloghrie, "Structure and identification of management information for TCP/IP-based internets," May 1990.) was used. MIB data conveyed in variable bindings ("varbinds") within protocol data units (PDUs) of SNMP messages use the primitive, base datatypes defined by the SMI.

The SMI allows for the creation of derivative datatypes, "textual conventions" ("TCs") [RFC2579] (McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2," April 1999.). A TC has a unique name, has a syntax that either refines or is a base SMI datatype and has relatively precise application-level semantics. TCs facilitate correct application-level handling of MIB data, improve readability of MIB modules by humans and support appropriate renderings of MIB data.

Values in varbinds corresponding to MIB objects defined with TC syntax are always encoded as the base SMI datatype underlying the TC syntax. Thus, the XSD mappings defined in this memo provide support for values of MIB objects defined with TC syntax as well as for values of MIB objects defined with base SMI syntax.

Various independent schemes have been devised for expressing SMI datatypes in XSD. These schemes exhibit a degree of commonality, especially concerning numeric SMI datatypes, but these schemes also exhibit sufficient differences, especially concerning the non-numeric SMI datatypes, precluding uniformity of expression and general interoperability.

Throughout this memo, the term "fidelity" refers to the quality of an accurate, consistent representation of SMI data values and the term "faithful" refers to the quality of reliably reflecting the semantics of SMI data values. Thus defined, the characteristics of fidelity and being faithful are essential to uniformity of expression and general interoperability in the XML representation of SMI data values.

The primary purpose of this memo is to define the standard expression of SMI base datatypes in XML documents that is both uniform and interoperable. This standard expression enables Internet operators, management application developers, and users to benefit from a wider range of management tools and to benefit from a greater degree of unified management. Thus, standard expression enables and facilitates improvements to the timeliness, accuracy and utility of management information.

The overall objective of this memo, and of any related future memos as may be published, is to define the XSD equivalent [XSDDatatypes] (World Wide Web Consortium, "XML Schema Part 2: Datatypes Second Edition," October 2004.) of SMIv2 (STD58) and to encourage XML-based protocols to carry, and XML-based applications to use, the management information defined in SMIv2-compliant MIB modules. The use of a standard mapping from SMIv2 to XML via XSD validation enables and promotes the efficient

reuse of existing and future MIB modules and instrumentation by XML-based protocols and management applications.

Developers of certain XML-based management applications will find this specification sufficient for their purposes. Developers of other XML-based management applications may need to make more complete reuse of existing MIB modules, requiring standard XSD documents for TCs [RFC2579] (McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2," April 1999.) and MIB structure [RFC2578] (McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)," April 1999.). Memos supporting such requirements are planned, but have not been produced at the time of this writing.

Finally, it is worthwhile to note that the goal of fidelity to the SMIv2 standard (STD58), as specified in the "Requirements" section below, is crucial to this effort. Fidelity leverages the established "rough consensus" of the precise SMIv2 data models contained in MIB modules, and leverages existing instrumentation, the "running code" implementing SMIv2 data models. This effort does not include any redesign of SMIv2 datatypes, data structures or textual conventions in order to overcome known limitations. Such work can be pursued by other efforts.

---

2. Conventions TOC

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 [RFC2119] (Bradner, s., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).

---

3. Requirements TOC

The following set of requirements is intended to produce XML documents which can be validated via the XSD defined in this specification to faithfully represent values carried "on-the-wire" in SNMP PDUs as defined by the SMI:

R1. All SMI base datatypes MUST have a corresponding XSD datatype.

R2. SMIv2 is the normative SMI for this document. Prior to mapping datatypes into XSD, legacy SMIv1 modules MUST be converted (at least logically) in accordance with Section 2.1, inclusive, of the "Coexistence" RFC [RFC3584] (Frye, R., Levi, D., Routhier, S., and B. Wijnen, "Coexistence between Version 1, Version 2, and

Version 3 of the Internet-standard Network Management Framework,"
August 2003.).

R3. The XSD datatype specified for a given SMI datatype MUST be able to represent all valid values for that SMI datatype.

R4. The XSD datatype specified for a given SMI datatype MUST represent any special encoding rules associated with that SMI datatype.

R5. The XSD datatype specified for a given SMI datatype MUST include any restrictions on values associated with the SMI datatype.

R6. The XSD datatype specified for a given SMI datatype MUST be the most logical XSD datatype, with the fewest necessary restrictions on its set of values, consistent with the foregoing requirements.

R7. The XML output produced as a result of meeting the foregoing requirements SHOULD be the most coherent and succinct representation (i.e., avoiding superfluous "decoration") from the perspective of readability by humans.

---

4. XSD for SMI Base Datatypes

This document provides XSD datatype mappings for the SMIv2 base datatypes only -- i.e., the eleven "ObjectSyntax" datatypes defined in RFC 2578. These datatypes -- via tag values defined in the SMIv2 to identify them in varbinds -- constrain values carried "on-the-wire" in SNMP PDUs between SNMP management applications and SNMP agents:

* INTEGER, Integer32

* Unsigned32, Gauge32

* Counter32

* TimeTicks

* Counter64

* OCTET STRING

* Opaque

* IpAddress

```
     *OBJECT IDENTIFIER
```

The "BITS" pseudo-type (also referred to as a "construct" in RFC 2578)
is treated as a Textual Convention, not a base datatype, for the
purpose of this document.

```
BEGIN

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:ietf:params:xml:ns:opsawg:smi:base:1.0"
targetNamespace="urn:ietf:params:xml:ns:opsawg:smi:base:1.0"
elementFormDefault="qualified"
attributeFormDefault="unqualified"
xml:lang="en">

  <xs:annotation>
    <xs:documentation>
        Mapping of SMIv2 base datatypes from RFC 2578

        Contact:      Mark Ellison
        Organization: Ellison Software Consulting
        Address:      38 Salem Road
                      Atkinson, NH 03811
                      USA
        Telephone:    +1 603-362-9270
        E-Mail:       ietf@EllisonSoftware.com

        Contact:      Bob Natale
        Organization: MITRE
        Address:      300 Sentinel Drive
                      6th Floor
                      Annapolis Junction MD 20701
                      USA
        Telephone:    +1 301-617-3008
        E-Mail:       rnatale@mitre.org

        Last Updated: 201002260000Z

        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 XML Schema Definition (XSD)
        document is part of RFC XXXX; see the RFC itself for
        full legal notices."
RFC Editor - please replace XXXX with the value allocated
```

```
      for publication as an RFC.

        </xs:documentation>
      </xs:annotation>

      <xs:simpleType name="INTEGER">
        <xs:restriction base="xs:int"/>
      </xs:simpleType>

      <xs:simpleType name="Integer32">
        <xs:restriction base="xs:int"/>
      </xs:simpleType>

      <xs:simpleType name="Unsigned32">
        <xs:restriction base="xs:unsignedInt"/>
      </xs:simpleType>

      <xs:simpleType name="Gauge32">
        <xs:restriction base="xs:unsignedInt"/>
      </xs:simpleType>

      <xs:simpleType name="Counter32">
        <xs:restriction base="xs:unsignedInt"/>
      </xs:simpleType>

      <xs:simpleType name="TimeTicks">
        <xs:restriction base="xs:unsignedInt"/>
      </xs:simpleType>

      <xs:simpleType name="Counter64">
        <xs:restriction base="xs:unsignedLong"/>
      </xs:simpleType>

      <xs:simpleType name="OctetString">
        <xs:restriction base="xs:hexBinary">
          <xs:maxLength value="65535"/>
        </xs:restriction>
      </xs:simpleType>

      <xs:simpleType name="Opaque">
        <xs:restriction base="xs:hexBinary"/>
      </xs:simpleType>

      <xs:simpleType name="IpAddress">
        <xs:restriction base="xs:string">
          <xs:pattern value=
          "((0|(1[0-9]{0,2})|
            (2(([0-4][0-9]?)|(5[0-5]?)|([6-9]?)))|
            ([3-9][0-9]?))\.){3}
            (0|(1[0-9]{0,2})|
```

```
             (2(([0-4][0-9]?)|(5[0-5]?)|([6-9]?)))|
             ([3-9][0-9]?))"/>
          </xs:restriction>
        </xs:simpleType>

        <xs:simpleType name="ObjectIdentifier">
          <xs:restriction base="xs:string">
            <xs:pattern value=
            "(([0-1](\.[1-3]?[0-9]))|
              (2\.(0|([1-9]\d*))))
              (\.(0|([1-9]\d*))){0,126}"/>
          </xs:restriction>
        </xs:simpleType>

      </xs:schema>
      END
```

---

5. Rationale

The XSD datatypes, including any specified restrictions, were chosen
based on fit with the requirements specified earlier in this document,
and with attention to simplicity while maintaining fidelity to the SMI.
Also, the "canonical representations" (i.e., refinements of the
"lexical representations") documented in the W3C XSD specification
[XMLSchema] (World Wide Web Consortium, "XML Schema Part 1: Structures
Second Edition," October 2004.), [XSDDatatypes] (World Wide Web
Consortium, "XML Schema Part 2: Datatypes Second Edition,"
October 2004.) are assumed.

5.1. Numeric Datatypes

All of the numeric XSD datatypes specified in the previous section --
INTEGER, Integer32, Unsigned32, Gauge32, Counter32, TimeTicks, and
Counter64 -- comply with the relevant requirements

    *They cover all valid values for the corresponding SMI datatypes.

    *They comply with the standard encoding rules associated with the
     corresponding SMI datatypes.

    *They inherently match the range restrictions associated with the
     corresponding SMI datatypes.

*They are the most direct XSD datatypes which exhibit the
  foregoing characteristics relative to the corresponding SMI
  datatypes (which is why no "restriction" statements -- other than
  the "base" XSD type -- are required in the XSD).

*The XML output produced from the canonical representation of
  these XSD datatypes is also the most direct from the perspective
  of readability by humans (i.e., no leading "+" sign and no
  leading zeros).

Special note to application developers: Compliance with this schema in
an otherwise correct translation from raw ("on-the-wire"
representation) SNMP MIB data produces values that are faithful to the
original. However, the Gauge32, Counter32, Counter64, and TimeTicks
datatypes have special application semantics that must be considered
when using their raw values for anything other than display, printing,
storage, or transmission of the literal value. RFC 2578 provides the
necessary details.

5.2. OctetString

This XSD datatype corresponds to the SMI "OCTET STRING" datatype.
Several independent schemes for mapping SMI datatypes to XSD have used
the XSD "string" type to represent "OCTET STRING", but this mapping
does not conform to the requirements specified in this document. Most
notably, "string" cannot faithfully represent all valid values (0 thru
255) that each octet in an "OCTET STRING" can have -- or at least
cannot do so in a way that provides for easy human readability of the
resulting XML output.
Consequently, the XSD datatype "hexBinary" is specified as the standard
mapping of the SMI "OCTET STRING" datatype. In hexBinary, each octet is
encoded as two hexadecimal digits; the canonical representation limits
the set of allowed hexadecimal digits to 0-9 and uppercase A-F.
The hexBinary representation of "OCTET STRING" complies with the
relevant requirements:

*It covers all valid values for the corresponding SMI datatype.

*It complies with the standard encoding rules associated with the
  corresponding SMI datatype.

*With the "maxLength" restriction to 65535 octets, the XSD
  datatype specification matches the restrictions associated with
  the corresponding SMI datatype.

*It is the most direct XSD datatype which exhibits the foregoing
    characteristics relative to the corresponding SMI datatype (which
    must allow for any valid binary octet value).

   *The XML output produced from the canonical representation of this
    XSD datatype is not optimal with respect to readability by
    humans; however, that is a consequence of the SMI datatype
    itself. Where human readability is more of a concern, it is
    likely that the actual MIB objects in question will be
    represented by textual conventions which limit the set of values
    that will be included in the OctetStrings and will, thus, bypass
    the hexBinary typing.

5.3. Opaque

The "hexBinary" XSD datatype is specified as the representation of the
SMI "Opaque" datatype generally for the same reasons as "hexBinary" is
specified for the "OctetString" datatype:

   *It covers all valid values for the corresponding SMI datatype.

   *It complies with the standard encoding rules associated with the
    corresponding SMI datatype.

   *There are no restriction issues associated with using "hexBinary"
    for "Opaque".

   *It is the most direct XSD datatype which exhibits the foregoing
    characteristics relative to the corresponding SMI datatype (which
    must allow for any valid binary octet value).

   *The XML output produced from the canonical representation of this
    XSD datatype is not optimal with respect to readability by
    humans; however, that is a consequence of the SMI datatype
    itself. Unmediated "Opaque" data is intended for consumption by
    applications, not humans.

5.4. IpAddress

The XSD "string" datatype is the natural choice to represent an
IpAddress as XML output. The "pattern" restriction applied in this case
results in a dotted-decimal string of four values between "0" and "255"

separated by a period (".") character. This pattern also precludes leading zeros.
Note that the SMI relies upon Textual Conventions (TCs) to specify an IPv6 address. As such, the representation of an IPv6 address as an XSD datatype is beyond the scope of this document.

5.5. ObjectIdentifier

This XSD datatype corresponds to the SMI "OBJECT IDENTIFIER" datatype. The XSD "string" datatype is also the natural choice to represent an ObjectIdentifier as XML output, for the same reasons as for the IpAddress choice. The "pattern" restriction applied in this case results in a dotted-decimal string of up to 128 elements (referred to as "sub-ids"), each holding an "Unsigned32" integer value.
Note that the first two components of an "OBJECT IDENTIFIER" each have a limited range of values as indicated in the XSD pattern restriction and as described in The ASN1.1/BER standard [ASN.1] (International Organization for Standardization, "Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)," December 1987.).
There are three values allocated for the root node, and at most 39 values for nodes subordinate to a root node value of 0 or 1.
The minimum length of an "OBJECT IDENTIFIER" is two sub-ids and the representation of a zero-valued "OBJECT IDENTIFIER" is "0.0".
Note that no explicit "minLength" restriction, which would be "3" to allow for the minimum of two sub-ids and a single separating dot, is required since the pattern itself enforces this restriction.

---

6. Security Considerations

Security considerations for any given SMI MIB module are likely to be relevant to any XSD/XML mapping of that MIB module; however, the mapping defined in this document does not itself introduce any new security considerations.
If and when proxies or gateways are developed to convey SNMP management information from SNMP agents to XML-based management applications via XSD/XML mapping of MIB modules based on this specification and its planned siblings, special care will need to be taken to ensure that all applicable SNMP security mechanisms are supported in an appropriate manner yet to be determined.

---

7. IANA Considerations

In accordance with RFC 3688 [RFC3688] (Mealling, M., "The IETF XML Registry," January 2004.), we request the following namespace and schema registrations associated with this document in the IANA XML Registry:

    *urn:ietf:params:xml:ns:opsawg:smi:base:[version_id]

    *urn:ietf:params:xml:schema:opsawg:smi:base:[version_id]

7.1. SMI Base Datatypes Namespace Registration   TOC

This document registers a URI for the SMI Base Datatypes XML namespace in the IETF XML registry. Following the format in RFC 3688, IANA has made the following registration:
URI: urn:ietf:params:xml:ns:opsawg:smi:base:1.0
Registration Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

7.2. SMI Base Datatypes Schema Registration   TOC

This document registers a URI for the SMI Base Datatypes XML schema in the IETF XML registry. Following the format in RFC 3688, IANA has made the following registration:
URI: urn:ietf:params:xml:schema:opsawg:smi:base:1.0
Registration Contact: The IESG.
XML: Section 4 of this document.

---

8. Acknowledgements   TOC

Dave Harrington provided strategic and technical leadership to the team which developed this particular specification. Yan Li did much of the research into existing approaches that was used as a baseline for the recommendations in this particular specification.
This document owes much to draft-romascanu-netconf-datatypes-xx and to many other sources (including libsmi and group discussions on the NETCONF mailing lists) developed by those who have researched and published candidate mappings of SMI datatypes to XSD.
Individuals who participated in various discussions of this topic at IETF meetings and on IETF mailing lists include: Ray Atarashi,

Yoshifumi Atarashi, Andy Bierman, Sharon Chisholm, Avri Doria, Rob
Ennes, Mehmet Ersue, David Harrington, Alfred Hines, Eliot Lear, Chris
Lonvick, Faye Ly, Randy Presuhn, Juergen Schoenwaelder, Andre
Westerinen, and Bert Wijnen.

---

9. References

9.1. Normative References

[RFC1155]	Rose, M. and K. McCloghrie, "Structure and identification of management information for TCP/IP-based internets," STD 16, RFC 1155, May 1990 (TXT).
[RFC2119]	Bradner, s., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2578]	McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)," STD 58, RFC 2578, April 1999.
[RFC3584]	Frye, R., Levi, D., Routhier, S., and B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework," BCP 74, RFC 3584, August 2003 (TXT).
[XML]	World Wide Web Consortium, "Extensible Markup Language (XML) 1.0," W3C XML, February 1998.
[XMLSchema]	World Wide Web Consortium, "XML Schema Part 1: Structures Second Edition," W3C XML Schema, October 2004.
[XSDDatatypes]	World Wide Web Consortium, "XML Schema Part 2: Datatypes Second Edition," W3C XML Schema, October 2004.

9.2. Informational References

[ASN.1]	International Organization for Standardization, "Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)," International Standard 8825, December 1987.
[RFC2579]	

	McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2," STD 58, RFC 2579, April 1999.
[RFC3688]	Mealling, M., "The IETF XML Registry," BCP 81, RFC 3688, January 2004 (TXT).

Authors' Addresses

	Mark Ellison
	Ellison Software Consulting
	38 Salem Road
	Atkinson, NH 03811
	USA
Phone:	+1 603-362-9270
Email:	ietf@ellisonsoftware.com
	Bob Natale
	MITRE
	300 Sentinel Drive
	6th Floor
	Annapolis Junction, MD 20701
	USA
Phone:	+1 301-617-3008
Email:	rnatale@mitre.org