

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 2, 2009

B. Natale
MITRE
October 29, 2008

Expressing SNMP SMI Datatypes in XML Schema Definition Language
draft-ietf-opsawg-smi-datatypes-in-xsd-04.txt

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/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 May 2, 2009.

Abstract

This memo (when approved as a standards-track RFC) 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.

Table of Contents

| | | |
|-----------------------------|----------------------------------------------------------|--------------------|
| 1. | Introduction | 3 |
| 2. | Conventions | 5 |
| 3. | Requirements | 6 |
| 4. | XSD for SMI Base Datatypes | 7 |
| 5. | Rationale | 10 |
| 5.1. | Numeric Datatypes | 10 |
| 5.2. | OctetString | 10 |
| 5.3. | Opaque | 11 |
| 5.4. | IpAddress | 12 |
| 5.5. | ObjectIdentifier | 12 |
| 6. | Security Considerations | 14 |
| 7. | IANA Considerations | 15 |
| 7.1. | SMI Base Datatypes Namespace Registration | 15 |
| 7.2. | SMI Base Datatypes Schema Registration | 15 |
| 8. | Acknowledgements | 16 |
| 9. | References | 17 |
| 9.1. | Normative References | 17 |
| 9.2. | Informational References | 17 |
| Appendix A. | Open Issues | 18 |
| Appendix B. | Change Log | 19 |
| | Author's Address | 20 |
| | Intellectual Property and Copyright Statements | 21 |

1. Introduction

Numerous uses exist -- both within and outside the traditional IETF network management community -- for the expression of management information described in and accessible via SMI Management Information Base (MIB) modules as XML documents [[ref.XML](#)]. For example, XML-based management applications which want to incorporate MIB modules as data models and/or to access MIB module instrumentation via gateways to SNMP agents will benefit from an IETF standard mapping of SMI datatypes and structures to XML documents via XSD.

MIB data models are described using SMIV2 [[RFC2578](#)] and, for legacy MIBs, SMIV1 [[RFC1155](#)]. MIB data is conveyed in variable bindings ("varbinds") within protocol data units (PDUs) within SNMP messages using the base/primitive datatypes defined in the SMI.

The SMI allows for creation of derivative datatypes, termed "textual conventions" ("TCs"), each of which has a unique name, a syntax which is or refines a primitive SMI datatype, and relatively precise application-level semantics. TCs are used principally to facilitate correct application-level handling of MIB data and for the convenience of humans reading MIB modules and appropriately rendered MIB data output. Values in varbinds corresponding to MIB objects with TC syntaxes are always encoded as the primitive SMI datatype underlying the TC syntax. Thus, the XSD mappings defined in this memo will support MIB objects with TC syntax as well as those with base SMI syntax.

Various independent schemes have been devised for expressing the SMI datatypes and TCs in XSD [[ref.XMLSchema](#)]. These schemes have exhibited a degree of commonality (especially concerning the numeric SMI datatypes), but also sufficient differences (especially concerning the non-numeric SMI datatypes) to preclude general interoperability.

The primary purpose of this memo is to define a standard expression of SMI base datatypes in XSD to ensure uniformity and general interoperability in this respect. Internet operators, management tool developers, and users will benefit from the wider selection of management tools and the greater degree of unified management -- with attendant improvements in timeliness and accuracy of management information -- which such a standard will facilitate.

This memo is the first in a set of three related and (logically) ordered specifications:

1. SMI Base Datatypes [[RFC2578](#)] in XSD
2. SMI MIB Structure [[RFC2578](#)] in XSD
3. SNMP Textual Conventions [[RFC2579](#)] in XSD

As a set, these documents define the XSD equivalent of SMIV2 to encourage XML-based protocols to carry, and XML-based applications to use, the information modeled in SMIV2-compliant MIB modules.

This work defines XSD equivalents of the datatypes and data structures [[RFC2578](#)] and the textual conventions [[RFC2579](#)] defined in the SMIV2 standard (STD58) to encourage efficient reuse of existing (including future) MIB modules and instrumentation by XML-based management protocols and applications.

The goal of fidelity to the SMIV2 standard (STD58), as specified in the "Requirements" section below, is crucial to this effort to leverage the established "rough consensus" for the precise data modeling used in MIB modules, and to leverage existing "running code" for implemented SMIV2 data models. This effort does not include redesign of SMIV2 datatypes or data structures or textual conventions to overcome known limitations -- that work can be pursued in other efforts.

2. Conventions

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](#)].

3. Requirements

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 -- SMIV1 modules, if encountered, MUST be converted (at least logically) in accordance with [Section 2.1](#), inclusive, of the "Coexistence" RFC [[RFC3584](#)].
- 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 direct XSD datatype, with the most parsimonious restrictions, which matches the foregoing requirements.
- R7. The XML output produced as a result of meeting the foregoing requirements SHOULD be the most direct (i.e., avoiding superfluous "decoration") from the perspective of readability by humans.

4. XSD for SMI Base Datatypes

This document concerns only the SMI base datatypes -- i.e., the eleven "ObjectSyntax" datatypes defined in [RFC2578](#). These datatypes -- via tag values defined in the SMI to identify them in varbinds -- constrain values carried "on-the-wire" in SNMP PDUs between SNMP management applications and SNMP agents:

- o INTEGER, Integer32
- o Unsigned32, Gauge32
- o Counter32
- o TimeTicks
- o Counter64
- o OctetString
- o Opague
- o IpAddress
- o ObjectIdentifier

The "BITS" pseudo-type (also referred to as a "construct" in [RFC2578](#)) is treated as a Textual Convention for the purpose of this document and, therefore, will be defined in the "SNMP Textual Conventions in XSD" 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.
    </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:pattern>
  </xs:restriction>
</xs:simpleType>
```



```
</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 specifications 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

- o They cover all valid values for the corresponding SMI datatypes.
- o They comply with the standard encoding rules associated with the corresponding SMI datatypes.
- o They inherently match the range restrictions associated with the corresponding SMI datatypes.
- o 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).
- o 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 ready 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:

- o It covers all valid values for the corresponding SMI datatype.
- o It complies with the standard encoding rules associated with the corresponding SMI datatype.
- o With the "maxLength" restriction to 65535 octets, the XSD datatype specification matches the restrictions associated with the corresponding SMI datatype.
- o 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).
- o 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:

- o It covers all valid values for the corresponding SMI datatype.
- o It complies with the standard encoding rules associated with the corresponding SMI datatype.

- o There are no restriction issues associated with using "hexBinary" for "Opaque".
- o 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).
- o 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.

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, while not mentioned in Sec. 7.1.3 of [RFC 2578](#), due to the use of Abstract Syntax Notation One (ASN.1) Basic Encoding Rules (BER) the first two components of an "OBJECT IDENTIFIER" have limited value ranges and are encoded into a single sub-id value [[Steedman](#)]. The ASN.1/BER standards specify that the numerical value of the first sub-identifier is derived from the values of the first two object identifier components in the object identifier value being encoded, using the formula: $(X*40) + Y$, where X is the value of the first object identifier component and Y is the value of the second object identifier component. This packing of the first two object identifier components recognizes that only three values are allocated from the root node, and at most 39 subsequent values from nodes reached by $X = 0$ and $X = 1$. The minimum length of an "OBJECT IDENTIFIER" is two sub-ids (with a zero-valued "OBJECT IDENTIFIER" represented as "0.0"). 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](#), we will register namespaces and schemas for all three documents in this set with the IANA XML Registry. These entries -- corresponding to this base datatypes document and the future textual conventions and MIB structure documents -- will be as follows:

- o urn:ietf:params:xml:ns:opawg:smi:base:[version_id]
- o urn:ietf:params:xml:schema:opawg:smi:base:[version_id]
- o urn:ietf:params:xml:ns:opawg:smi:tc:[version_id]
- o urn:ietf:params:xml:schema:opawg:smi:tc:[version_id]
- o urn:ietf:params:xml:ns:opawg:smi:structure:[version_id]
- o urn:ietf:params:xml:schema:opawg:smi:structure:[version_id]

The following sub-sections refer to the present document only.

7.1. SMI Base Datatypes Namespace Registration

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:opawg: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

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:opawg:smi:base:1.0

Registration Contact: The IESG.

XML: [Section 4](#) of this document.

8. Acknowledgements

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 and textual conventions 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, Mark Ellison, Rob Ennes, David Harrington, Alfred Hines, Eliot Lear, Chris Lonvick, Faye Ly, Randy Presuhn, Juergen Schoenwaelder, Andre Westerinen, and Bert Wijnen.

(Others who have been inadvertently omitted from this list should notify the editor.)

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.
- [RFC2119] Bradner, s., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, [RFC 2579](#), April 1999.

9.2. Informational References

- [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.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), January 2004.
- [Steedman] Steedman, D., "ASN.1: The Tutorial and Reference".
- [ref.XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0", W3C XML, February 1998, <<http://www.w3.org/TR/1998/REC-xml-19980210>>.
- [ref.XMLSchema] World Wide Web Consortium, "XML Schema Part 1: Structures Second Edition", W3C XML Schema, October 2004, <<http://www.w3.org/TR/xmlschema-1/>>.
- [ref.XSDDatatype] World Wide Web Consortium, "XML Schema Part 2: Datatypes Second Edition", W3C XML Schema, October 2004, <<http://www.w3.org/TR/xmlschema-2/>>.

[Appendix A.](#) **Open Issues**

- o Confirm IANA XML registration values and process.

[Appendix B](#). Change Log

- o -01 version:
 - * Incorporated mailing list comments on -00 version from Juergen Schoenwaelder
 - * Incorporated mailing list comments on -00 version from David Harrington
- o -02 version:
 - * Fixed ObjectIdentifier pattern per correction from Juergen Schoenwaelder, and text in sec. 5.5 adjusted accordingly.
 - * Moved non-normative references to Informational section per David Harrington
 - * Tightened wording in to "XSD for SMI Datatypes" section per Mark Ellison
 - * Added a note about Gauge32 and Counter32 application semantics to the "Rationale" section per Mark Ellison
 - * Security section wording tightened per David Harrington
 - * The IANA Considerations section completed--will need adjustment.
 - * Acknowledgments entries expanded and alphabetized
- o -03 version:
 - * Corrected "ten" to "eleven" in opening sentence of "XSD for SMI Datatypes" section.
 - * Removed conditional wording that previously prefaced the XSD itself.
- o -04 version:
 - * Relatively minor text fix-ups in various places, mainly in response to comments on the -03 version from Mark Ellison, Alfred Hines, Juergen Schoenwaelder, and David Harrington.

Author's Address

Bob Natale
MITRE
7515 Colshire Dr
MS H405
McLean, VA 22102
USA

Phone: +1 703-983-2505
Email: rnatale@mitre.org

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

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, THE IETF TRUST 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.

