

Textual Conventions for the Representation of Floating-Point Numbers

Abstract

This memo defines a Management Information Base (MIB) module containing textual conventions (TCs) to represent floating-point numbers.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc6340>.

Copyright Notice

Copyright (c) 2011 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	2
2. The Internet-Standard Management Framework	3
3. Applicability	3
4. Structure of the MIB Module	4
4.1. MIB Modules Required for IMPORTS	4
4.2. Documents Required for REFERENCE Clauses	4
5. Definitions	4
6. Security Considerations	6
7. IANA Considerations	6
8. Contributors	6
9. References	7
9.1. Normative References	7
9.2. Informative References	7

1. Introduction

This memo defines textual conventions for the representation of floating-point numbers. All of these definitions are in terms of the IEEE "Standard for Floating-Point Arithmetic", IEEE 754-2008 [IEEE.754.2008].

The IEEE "Standard for Floating-Point Arithmetic", IEEE 754-2008 [IEEE.754.2008], provides for a variety of interchange formats for floating-point numbers. The need for three of these, namely

- o 32-bit,
- o 64-bit,
- o 128-bit,

has been recognized in network management. For example, Section 4.2.3 of the SMIng Objectives [RFC3216] elaborates the need for these three floating-point data types in network management protocols.

The selection of a floating-point format involves many considerations and trade-offs. For an introduction to the fundamentals of floating-point representations see Chapter 4 of [KNUTH]; for a discussion of these issues specifically with respect to the IEEE formats, see [GOLDBERG].

All of these textual conventions employ the binary interchange format defined in [IEEE.754.2008]. Specifically, this means that for all of them, the highest-order bit of the first byte is the sign bit, with the remaining bits of the octet string corresponding to the exponent and fraction parts, in network byte order.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. Applicability

The following list highlights some of the issues MIB designers need to consider when deciding whether to employ these textual conventions:

- o Floating-point numbers are useful if the number space needs to cover a large dynamic range. For number spaces with a limited range, fixed-point numbers can be more efficient and more precise.
- o Floating-point numbers are typically the wrong answer for data that is truly decimal or can be handled adequately by re-thinking the units and representing the scaled numbers as integers.
- o The SNMP "lexicographical" ordering for INDEX objects using these floating-point textual conventions will simply be that of the octet strings corresponding to the floating-point representations, which will not always reflect the numerical ordering of the corresponding floating-point values. Even if MIB designers take this into account, users might still find the results of a MIB "walk" surprising. Consequently, it is suggested that whenever one of these textual conventions is used for an INDEX object, that the DESCRIPTION clause should provide some warning.
- o Embedded systems sometimes lack floating-point support, which can complicate the implementation of MIB objects using floating-point numbers.
- o In choosing from among the types defined in this memo, MIB designers need to consider both the range and the precision needed, as well as recognize that it could be inefficient to use, for example, Float128TC when Float64TC would do.

- o Since these textual conventions are defined in terms of the OCTET STRING type, the SMI's mechanisms for formally setting range constraints are not available. MIB designers using these textual conventions will need to use DESCRIPTION clauses to spell out any applicable range constraints beyond those implied by the underlying IEEE types.
- o Whenever these textual conventions are used in a MIB module, the associated DESCRIPTION clause will need to clearly specify whether denormalized numbers, NaNs ("not a number") or infinities are permitted, along with any special semantics associated with these cases. This is especially important for writeable objects.

4. Structure of the MIB Module

This MIB module defines three textual conventions. It defines no MIB objects.

4.1. MIB Modules Required for IMPORTS

This MIB module employs definitions from [RFC2578] and [RFC2579].

4.2. Documents Required for REFERENCE Clauses

This MIB module contains REFERENCE clauses making reference to IEEE 754-2008 [IEEE.754.2008].

5. Definitions

FLOAT-TC-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY,
mib-2
TEXTUAL-CONVENTION

FROM SNMPv2-SMI -- RFC 2578
FROM SNMPv2-TC; -- RFC 2579

floatTcMIB MODULE-IDENTITY

LAST-UPDATED "201107270000Z" -- July 27, 2011
ORGANIZATION "IETF OPSAWG Working Group"
CONTACT-INFO "WG Email: opsawg@ietf.org"

Editor: Randy Presuhn
randy_presuhn@mindspring.com"

DESCRIPTION "Textual conventions for the representation
of floating-point numbers.

Copyright (c) 2011 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 MIB module is part of RFC 6340; see the RFC itself for full legal notices."

REVISION "201107270000Z" -- July 27, 2011
DESCRIPTION "Initial version, published as RFC 6340."
 ::= { mib-2 201 }

Float32TC ::= TEXTUAL-CONVENTION

STATUS current
DESCRIPTION "This type represents a 32-bit (4-octet) IEEE floating-point number in binary interchange format."
REFERENCE "IEEE Standard for Floating-Point Arithmetic, Standard 754-2008"
SYNTAX OCTET STRING (SIZE(4))

Float64TC ::= TEXTUAL-CONVENTION

STATUS current
DESCRIPTION "This type represents a 64-bit (8-octet) IEEE floating-point number in binary interchange format."
REFERENCE "IEEE Standard for Floating-Point Arithmetic, Standard 754-2008"
SYNTAX OCTET STRING (SIZE(8))

Float128TC ::= TEXTUAL-CONVENTION

STATUS current
DESCRIPTION "This type represents a 128-bit (16-octet) IEEE floating-point number in binary interchange format."
REFERENCE "IEEE Standard for Floating-Point Arithmetic, Standard 754-2008"
SYNTAX OCTET STRING (SIZE(16))

END

6. Security Considerations

This module does not define any management objects. Instead, it defines a set of textual conventions that can be used by other MIB modules to define management objects.

Meaningful security considerations can only be written in the MIB modules that define management objects. Therefore, this memo has no impact on the security of the Internet.

7. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER value recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
-----	-----
floatTcMIB	{ mib-2 201 }

8. Contributors

The following people provided helpful comments during the development of this document:

- o Andy Bierman
- o Martin Duerst
- o Alfred Hoenes
- o Juergen Quittek
- o Juergen Schoenwaeder
- o Dave Shield
- o Robert Story

9. References

9.1. Normative References

- [IEEE.754.2008] Institute of Electrical and Electronics Engineers, "Standard for Floating-Point Arithmetic", IEEE Standard 754, August 2008.
- [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., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

9.2. Informative References

- [GOLDBERG] Goldberg, D., "What Every Computer Scientist Should Know About Floating-Point Arithmetic", ACM Computing Surveys Volume 23, Issue 1, March 1991.
- [KNUTH] Knuth, D., "Seminumerical Algorithms", The Art of Computer Programming (Second Edition) Vol. 2, 1981.
- [RFC3216] Elliott, C., Harrington, D., Jason, J., Schoenwaelder, J., Strauss, F., and W. Weiss, "SMIng Objectives", RFC 3216, December 2001.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

Author's Address

Randy Presuhn
San Jose, CA 95120
USA

EMail: randy_presuhn@mindspring.com