Network Working Group
Request for Comments: 1908
Obsoletes: 1452
Category: Standards Track

SNMPv2 Working Group
J. Case
SNMP Research, Inc.
K. McCloghrie
Cisco Systems, Inc.
M. Rose
Dover Beach Consulting, Inc.
S. Waldbusser
International Network Services

January 1996

Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework

Status of this Memo

Table of Contents

1. Introduction

The purpose of this document is to describe coexistence between version 2 of the Internet-standard Network Management Framework [1-6], termed the SNMP version 2 framework (SNMPv2), and the original Internet-standard Network Management Framework (SNMPv1), which consists of these three documents:

STD 16, RFC 1155 [7] which defines the Structure of Management Information (SMI), the mechanisms used for describing and naming objects for the purpose of management.

STD 16, RFC 1212 [8] which defines a more concise description mechanism, which is wholly consistent with the SMI.

STD 15, RFC 1157 [9] which defines the Simple Network Management Protocol (SNMP), the protocol used for network access to managed objects.

2. Management Information

The SNMPv2 approach towards describing collections of managed objects is nearly a proper superset of the approach defined in the Internet-standard Network Management Framework. For example, both approaches use ASN.1 [10] as the basis for a formal descriptive notation. Indeed, one might note that the SNMPv2 approach largely codifies the existing practice for defining MIB modules, based on extensive experience with the current framework.

The SNMPv2 documents which deal with information modules are:

Structure of Management Information for SNMPv2 [1], which defines concise notations for describing information modules, managed objects and notifications;

Textual Conventions for SNMPv2 [2], which defines a concise notation for describing textual conventions, and also defines some initial conventions; and,

Conformance Statements for SNMPv2 [3], which defines concise notation for describing compliance and capabilities statements.

The following sections consider the three areas: MIB modules, compliance statements, and capabilities statements.

MIB modules defined using the current framework may continue to be used with the SNMPv2 protocol. However, for the MIB modules to conform to the SNMPv2 framework, the following changes are required:

2.1. Object Definitions

In general, conversion of a MIB module does not require the deprecation of the objects contained therein. Only if the semantics of an object truly changes should deprecation be performed.

(1) The IMPORTS statement must reference SNMPv2-SMI, instead of RFC1155-SMI and RFC-1212.

(2) The MODULE-IDENTITY macro must be invoked immediately after any IMPORTs statement.

(3) For any descriptor which contains the hyphen character, the hyphen character is removed.

(4) For any label for a named-number enumeration which contains the hyphen character, the hyphen character is removed.

(5) For any object with an integer-valued SYNTAX clause, in which the corresponding INTEGER does not have a range restriction (i.e., the INTEGER has neither a defined set of named-number enumerations nor an assignment of lower- and upper-bounds on its value), the object must have the value of its SYNTAX clause changed to Integer32.

(6) For any object with a SYNTAX clause value of an enumerated INTEGER, the hyphen character is removed from any named-number labels which contain the hyphen character.

(7) For any object with a SYNTAX clause value of Counter, the object must have the value of its SYNTAX clause changed to Counter32.

(8) For any object with a SYNTAX clause value of Gauge, the object must have the value of its SYNTAX clause changed to Gauge32.

(9) For all objects, the ACCESS clause must be replaced by a MAX-ACCESS clause. The value of the MAX-ACCESS clause is the same as that of the ACCESS clause unless some other value makes "protocol sense" as the maximal level of access for the object. In particular, object types for which instances can be explicitly created by a protocol set operation, will have a MAX-ACCESS clause of "read-create". If the value of the ACCESS clause is "write-only", then the value of the MAX-ACCESS clause is "read-write", and the DESCRIPTION clause notes that reading this object will result implementation-specific results.

(10) For all objects, if the value of the STATUS clause is "mandatory", the value must be replaced with "current".

(11) For all objects, if the value of the STATUS clause is "optional", the value must be replaced with "obsolete".

(12) For any object not containing a DESCRIPTION clause, the object must have a DESCRIPTION clause defined.

(13) For any object corresponding to a conceptual row which does not have an INDEX clause, the object must have either an INDEX clause or an AUGMENTS clause defined.

(14) For any object with an INDEX clause that references an object with a syntax of NetworkAddress, the value of the STATUS clause of both objects is changed to "obsolete".

(15) For any object containing a DEFVAL clause with an OBJECT IDENTIFIER value which is expressed as a collection of sub-identifiers, change the value to reference a single ASN.1 identifier.

Other changes are desirable, but not necessary:

(1) Creation and deletion of conceptual rows is inconsistent using the current framework. The SNMPv2 framework corrects this. As such, if the MIB module undergoes review early in its lifetime, and it contains conceptual tables which allow creation and deletion of conceptual rows, then it may be worthwhile to deprecate the objects relating to those tables and replace them with objects defined using the new approach.

(2) For any object with a string-valued SYNTAX clause, in which the corresponding OCTET STRING does not have a size restriction (i.e., the OCTET STRING has no assignment of lower- and upper-bounds on its length), one might consider defining the bounds for the size of the object.

(3) For all textual conventions informally defined in the MIB module, one might consider redefining those conventions using the TEXTUAL-CONVENTION macro. Such a change would not necessitate deprecating objects previously defined using an informal textual convention.

(4) For any object which represents a measurement in some kind of units, one might consider adding a UNITS clause to the definition of that object.

(5) For any conceptual row which is an extension of another conceptual row, i.e., for which subordinate columnar objects both exist and are identified via the same semantics as the other conceptual row, one might consider using an AUGMENTS clause in place of the INDEX clause for the object corresponding to the conceptual row which is

an extension.

Finally, when encountering common errors in SNMPv1 MIB modules:

(1) For any non-columnar object that is instanced as if it were immediately subordinate to a conceptual row, the value of the STATUS clause of that object is changed to "obsolete".

(2) For any conceptual row object that is not contained immediately subordinate to a conceptual table, the value of the STATUS clause of that object (and all subordinate objects) is changed to "obsolete".

2.2. Trap Definitions

If a MIB module is changed to conform to the SNMPv2 framework, then each occurrence of the TRAP-TYPE macro must be changed to a corresponding invocation of the NOTIFICATION-TYPE macro:

(1) The IMPORTS statement must not reference RFC-1215.

(2) The ENTERPRISES clause must be removed.

(3) The VARIABLES clause must be renamed to the OBJECTS clause.

(4) The STATUS clause must be added.

(5) The value of an invocation of the NOTIFICATION-TYPE macro is an OBJECT IDENTIFIER, not an INTEGER, and must be changed accordingly. Specifically, if the value of the ENTERPRISE clause is not 'snmp' then the value of the invocation is the value of the ENTERPRISE clause extended with two sub-identifiers, the first of which has the value 0, and the second has the value of the invocation of the TRAP-TYPE.

2.3. Compliance Statements

For those information modules which are "standard", a corresponding invocation of the MODULE-COMPLIANCE macro must be included within the information module (or in a companion information module), and any commentary text in the information module which relates to compliance must be removed. Typically this editing can occur when the information module undergoes review.

2.4. Capabilities Statements

In the current framework, the informational document [11] uses the MODULE-CONFORMANCE macro to describe an agent's capabilities with respect to one or more MIB modules. Converting such a description for use with the SNMPv2 framework requires these changes:

(1) Use the macro name AGENT-CAPABILITIES instead of MODULE-CONFORMANCE.

(2) The STATUS clause must be added.

(3) For all occurrences of the CREATION-REQUIRES clause, note the slight change in semantics, and omit this clause if appropriate.

In order to ease the coexistence between SNMPv1 and SNMPv2, object groups defined in an SNMPv1 MIB module may be referenced by the INCLUDES clause of an invocation of the AGENT-CAPABILITIES macro: upon encountering a reference to an OBJECT IDENTIFIER subtree defined in an SNMPv1 MIB module, all leaf objects which are subordinate to the subtree and have a STATUS clause value of mandatory are deemed to be INCLUDEd. (Note that this method is ambiguous when different revisions of a SNMPv1 MIB have different sets of mandatory objects under the same subtree; in such cases, the only solution is to rewrite the MIB using the SNMPv2 SMI in order to define the object groups unambiguously.)

3. Protocol Operations

The SNMPv2 documents which deal with protocol operations are:

Protocol Operations for SNMPv2 [4], which defines the syntax and semantics of the operations conveyed by the protocol; and,

Transport Mappings for SNMPv2 [5], which defines how the protocol operations are carried over different transport services.

The following section considers two areas: the proxy behavior between a SNMPv2 entity and a SNMPv1 agent; and, the behavior of "bi-lingual" protocol entities acting in a manager role.

3.1. Proxy Agent Behavior

To achieve coexistence at the protocol-level, a proxy mechanism may be used. A SNMPv2 entity acting in an agent role may be implemented and configured to act in the role of a proxy agent.

3.1.1. SNMPv2 -> SNMPv1

When converting requests from a SNMPv2 entity acting in a manager role into requests sent to a SNMPv1 entity acting in an agent role:

(1) If a GetRequest-PDU, GetNextRequest-PDU, or SetRequest-PDU is received, then it is passed unaltered by the proxy agent.

(2) If a GetBulkRequest-PDU is received, the proxy agent sets the non-repeaters and max-repetitions fields to zero, and sets the tag of the PDU to GetNextRequest-PDU.

3.1.2. SNMPv1 -> SNMPv2

When converting responses received from a SNMPv1 entity acting in an agent role into responses sent to a SNMPv2 entity acting in a manager role:

(1) If a GetResponse-PDU is received, then it is passed unaltered by the proxy agent. Note that even though a SNMPv2 entity will never generate a Response-PDU with a error-status field having a value of 'noSuchName', 'badValue', or 'readOnly', the proxy agent must not change this field. This allows the SNMPv2 entity acting in a manager role to interpret the response correctly.

If a GetResponse-PDU is received with an error-status field having a value of 'tooBig', the proxy agent will remove the contents of the variable-bindings field before propagating the response. Note that even though a SNMPv2 entity will never generate a 'tooBig' in response to a GetBulkRequest-PDU, the proxy agent must propagate such a response.

(2) If a Trap-PDU is received, then it is mapped into a SNMPv2-Trap-PDU. This is done by prepending onto the variable-bindings field two new bindings: sysUpTime.0 [6], which takes its value from the timestamp field of the Trap-PDU; and, snmpTrapOID.0 [6], which is calculated thusly: if the value of generic-trap field is 'enterpriseSpecific', then the value used is the concatenation of the enterprise field from the Trap-PDU with two additional sub-identifiers, '0', and the value of the specific-trap field; otherwise, the value of the corresponding trap defined in [6] is used. (For example, if the value of the generic-trap field is 'coldStart', then the coldStart trap [6] is used.) Then, one new binding is appended onto the variable-bindings field: snmpTrapEnterprise.0 [6], which takes its value from the enterprise field of the Trap-PDU. The destinations for the SNMPv2-Trap-PDU are determined in an implementation-dependent fashion by the proxy agent.

3.2. Bi-lingual Manager Behavior

To achieve coexistence at the protocol-level, a protocol entity acting in a manager role might support both SNMPv1 and SNMPv2. When a management application needs to contact a protocol entity acting in an agent role, the entity acting in a manager role consults a local database to select the correct management protocol to use.

In order to provide transparency to management applications, the entity acting in a manager role must map operations as if it were acting as a proxy agent.

4. Security Considerations

Security issues are not discussed in this memo.

5. Editor's Address

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
US

Phone: +1 408 526 5260
EMail: kzm@cisco.com

6. Acknowledgements

This document is the result of significant work by the four major contributors:

Jeffrey D. Case (SNMP Research, case@snmp.com)
Keith McCloghrie (Cisco Systems, kzm@cisco.com)
Marshall T. Rose (Dover Beach Consulting, mrose@dbc.mtview.ca.us)
Steven Waldbusser (International Network Services, stevew@uni.ins.com)

In addition, the contributions of the SNMPv2 Working Group are acknowledged. In particular, a special thanks is extended for the contributions of:

Alexander I. Alten (Novell)
Dave Arneson (Cabletron)
Uri Blumenthal (IBM)
Doug Book (Chipcom)
Kim Curran (Bell-Northern Research)
Jim Galvin (Trusted Information Systems)
Maria Greene (Ascom Timeplex)

Iain Hanson (Digital)
Dave Harrington (Cabletron)
Nguyen Hien (IBM)
Jeff Johnson (Cisco Systems)
Michael Kornegay (Object Quest)
Deirdre Kostick (AT&T Bell Labs)
David Levi (SNMP Research)
Daniel Mahoney (Cabletron)
Bob Natale (ACE*COMM)
Brian O'Keefe (Hewlett Packard)
Andrew Pearson (SNMP Research)
Dave Perkins (Peer Networks)
Randy Presuhn (Peer Networks)
Aleksey Romanov (Quality Quorum)
Shawn Routhier (Epilogue)
Jon Saperia (BGS Systems)
Bob Stewart (Cisco Systems, bstewart@cisco.com), chair
Kaj Tesink (Bellcore)
Glenn Waters (Bell-Northern Research)
Bert Wijnen (IBM)

7. References

[1] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.

[2] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.

[3] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.

[4] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[5] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

[6] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1907, January 1996.

[7] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", STD 16, RFC 1155, May 1990.

[8] Rose, M., and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[9] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, SNMP Research, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[10] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, (December, 1987).

[11] McCloghrie, K., and M. Rose, "A Convention for Describing SNMP-based Agents", RFC 1303, Hughes LAN Systems, Dover Beach Consulting, Inc., February 1992.