International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.784
(03/2016)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

OSI management – Management functions and ODMA functions

Guidelines for implementation conformance statements proformas associated with SNMP-based management systems

Recommendation ITU-T X.784

ITU-T

International
Telecommunication
Union

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.369
IP-based networks	X.370–X.379
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems management framework and architecture	X.700–X.709
Management communication service and protocol	X.710–X.719
Structure of management information	X.720–X.729
Management functions and ODMA functions	**X.730–X.799**
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, concurrency and recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.889
Generic applications of ASN.1	X.890–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	X.1000–X.1099
SECURE APPLICATIONS AND SERVICES	X.1100–X.1199
CYBERSPACE SECURITY	X.1200–X.1299
SECURE APPLICATIONS AND SERVICES	X.1300–X.1399
CYBERSECURITY INFORMATION EXCHANGE	X.1500–X.1599
CLOUD COMPUTING SECURITY	X.1600–X.1699

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.784

Guidelines for implementation conformance statements proformas associated with SNMP-based management systems

Summary

Recommendation ITU-T X.784 proposes guidelines for implement conformance statement (ICS) proformas for simple network management protocol (SNMP)-based interface systems.

SNMP is widely used to manage nodes such as servers, workstations, routers and switches in IP networks. When SNMP-based management systems need to be interconnected, system testing is required in order to ensure operability among them. ICS proformas are a way for implementers to claim conformance to a standardized SNMP interface specification.

This Recommendation provides an overview and constructions for the management information base (MIB) and the structure of management information (SMI) and provides several proformas (tables) for each of the SMI components to be used in an interface. Instructions on how to fill in each column in the tables are also provided. Finally some examples of the SNMP interface ICS are provided in Appendix I while additional information about the conformance test methodology for SNMP-based interfaces is provided in Appendix II.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.784	2016-03-15	2	11.1002/1000/12783

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T X.784

Guidelines for implementation conformance statements proformas associated with SNMP-based management systems

1 Scope

This Recommendation provides guidelines for simple network management protocol (SNMP)-based interface implementation conformance statement (SIICS) proformas and the specification of these proformas. The SIICS is a statement made by an implementer to claim conformance to a structure of management information (SMI)-based interface definition.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.290] Recommendation ITU-T X.290 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts*.

[ITU-T X.291] Recommendation ITU-T X.291 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Abstract test suite specification*. (Twinned to [ISO 9646-2])

[ITU-T X.296] Recommendation ITU-T X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements*. (Twinned to [ISO 9646-7])

[ITU-T X.724] Recommendation ITU-T X.724 (1996), *Information technology – Open Systems Interconnection – Structure of management information: Requirements and guidelines for implementation conformance statement proformas associated with OSI management*.

[ITU-T X.781] Recommendation ITU-T X.781 (2001), *Requirements and guidelines for Implementation Conformance Statements proformas associated with CORBA-based systems*.

[ISO 9646-2] ISO/IEC 9646-2:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 2: Abstract Test Suite specification*. (Twinned to [ITU-T X.291])

[ISO 9646-7] ISO/IEC 9646-7:1995, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 7: Implementation Conformance Statements*. (Twinned to [ITU-T X.291])

[IETF RFC 1157] IETF RFC 1157 (1990), *A Simple Network Management Protocol (SNMP)*.

[IETF RFC 1212] IETF RFC 1212 (1991), *Concise MIB Definitions*.

[IETF RFC 1902] IETF RFC 1902 (1996), *Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)*.

[IETF RFC 2578]	IETF RFC 2578 (1999), *Structure of Management Information Version 2 (SMIv2)*.
[IETF RFC 2580]	IETF RFC 2580 (1999), *Conformance Statements for SMIv2*.
[IETF RFC 3410]	IETF RFC 3410 (2002), *Introduction and Applicability Statements for Internet Standard Management Framework (SNMPv3)*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 implementation conformance statement (ICS) [ITU-T X.290]: A statement made by the supplier of an implementation or system claimed to conform to a given specification, stating which capabilities have been implemented. The ICS can take several forms: protocol ICS, profile ICS, profile specific ICS, and information object ICS.

3.1.2 (ICS (proforma)) item [ITU-T X.290]: A row in an ICS (proforma) table.

3.1.3 (ICS (proforma)) question [ITU-T X.290]: The question to be answered in the intersection of an ICS item and either a support column (i.e., "Is this item supported in the context applying to this table and column?") or supported values column (i.e., "What values are supported for this item in the context applying to this table and column?") in an ICS proforma table.

3.1.4 status (value) [ITU-T X.290]: An allowed entry in the status column for an item in an ICS proforma table.

3.1.5 structure of management information (SMI) [b-ITU-T H.341]: The rules used to define objects which can be accessed via a network management protocol.

3.1.6 (support) answer [ITU-T X.290]: An allowed entry in the support or supported values columns for an item in an ICS, in answer to an ICS question.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ASN.1	Abstract Syntax Notation One
CORBA	Common Object Request Broker Architecture
ICS	Implementation Conformance Statement
IUT	Implementation Under Test
MIB	Management Information Base
OID	Object Identifier
PDU	Protocol Data Unit
SIICS	SNMP-based Interface Implementation Conformance Statement
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol

5 Conventions

None.

6 Basis of conformance test methodology for SNMP-based management system interface

6.1 Overview

Conformance relates an implementation to a standard. It states in which way systems, implemented with respect to a standard, can vary without errors occurring in their cooperation. If an implementation fulfills these requirements, then it conforms to the standard. The check of the statements is the conformance test. The starting point is the definition of conformance requirements in implementation independent interface specifications on the basis of an identification of reference points. A management interface specification shall define conformance reference points at which an object must be tested to check if it fulfils a set of conformance criteria. During the test, a number of stimuli and events are observed and evaluated at these conformance points. Management interface specifications shall include conformance statements which identify conformance reference points at every interface of the specified objects. In general, the information flow between two system components is realized through several reference points, thus the conformance test has to take into consideration the following:

a) the test of such information flow at each reference point; and

b) the test of consistency between the combinations.

Therefore, a coordinated test at all identified reference points is necessary.

6.2 Methodology of conformance test for SNMP-based interfaces

This Recommendation provides guidelines for a conformance test of SNMP-based interfaces. In order to ensure consistency of testing methods for network management interfaces, the testing methodology and process for SNMP-based interfaces shall be consistent with the testing process of a common object request broker architecture (CORBA)-based interface as defined in [ITU T-X.781]. [IETF RFC 2580] gives conformance statements for SMIv2 by defining several macros, and Appendix II provides information on the differences between [IETF RFC 2580] and this Recommendation.

7 SMI-based management interface ICS proforma

The structure of management information (SMI) is used to define interfaces of objects as a management information base (MIB) in SNMP-based systems. This clause first introduces the features of MIB and SMI, then specifies the SIICS proformas according to these language features.

7.1 MIB and SMI overview

A MIB is a database used for managing the entities in a communications network. MIB is most often associated with the simple network management protocol (SNMP).

SNMP is an Internet-standard protocol for managing devices on IP networks. SNMP itself does not define which information (which variables) a managed system shall offer. Rather, SNMP uses an extensible design, where the available information is defined by a MIB. The MIB describes the structure of the management data of a device subsystem.

SNMPv1 specifies five core protocol data units (PDUs): GetRequest, SetRequest, GetNextRequest, Response and Trap. Two other PDUs, GetBulkRequest and InformRequest were added in SNMPv2 and the Report PDU was added in SNMPv3. Since the implementation conformance statement (ICS) proformas are used by an implementer to claim conformance to an information model described in a MIB, so ICS proformas provided in this Recommendation are suitable to be used for any version of SNMP-based interface.

Objects in the MIB are defined using a subset of Abstract Syntax Notation One (ASN.1) called the structure of management information (SMI). The database is hierarchical (tree-structured) and each entry is addressed through an object identifier (OID).

The first version of the SMI (SMIv1) specifies the use of a number of SMI-specific data types. The second version of the SMI (SMIv2) enhances and adds to the SMIv1-specific data types and also specifies information modules. In this Recommendation, ICS proformas for SMIv2-based management interface are mainly provided since this version is commonly used nowadays when managing nodes in IP networks.

The SMI is divided into three parts: module definitions, object definitions and notification definitions [IETF RFC 2578]. According to the SMI syntax specification, the containment relationship among these constructions can be illustrated as in Figure 1.

1) Module definitions are used when describing information modules. An ASN.1 macro, MODULE-IDENTITY, is used to concisely convey the semantics of an information module. The MODULE-IDENTITY macro is used to provide contact and revision history for each information module. MIB modules, which contain definitions of inter-related managed objects, make use of the OBJECT-TYPE and NOTIFICATION-TYPE macros.

2) Object definitions are used when describing managed objects. An ASN.1 macro, OBJECT-TYPE, is used to concisely convey the syntax and semantics of a managed object.

   It is sometimes convenient for developers of management applications to impose tabular structure on an ordered collection of objects within the MIB. Each such conceptual table contains zero or more rows, and each row may contain one or more scalar objects, termed columnar objects. This conceptualization is formalized by using the OBJECT-TYPE macro to define both an object which corresponds to a table and an object which corresponds to a row in that table. A conceptual table has SYNTAX of the form SEQUENCE OF <EntryType>, where <EntryType> refers to the SEQUENCE type of its subordinate conceptual row. A conceptual row has SYNTAX of the form <EntryType>, where <EntryType> is a SEQUENCE type.

   Another type of object is leaf objects, which are not columnar objects and are not contained within a conceptual table. Instances of the leaf object are identified by appending a sub-identifier of zero to the name of that object.

3) Notification definitions are used when describing unsolicited transmissions of management information. An ASN.1 macro, NOTIFICATION-TYPE, is used to concisely convey the syntax and semantics of a notification.

   For SNMP-based interface implementation conformance testing, the SIICS proforma shall support the above SMI features such as the conceptual table, entry, column, leaf and notification. Thus SIICS proformas shall include three sub proformas: a conceptual table object support proforma, a leaf object support proforma and a notification support proforma. The conceptual table object support proforma can be divided into the conceptual table object support table, the column support table, the entry support table and the conceptual table index support table, which are used to test whether the operations (such as read, write, create) are supported for each object instance. The notification support proforma is used to test whether the notification could be sent out with the correct format. The leaf object support proforma is used to test whether the operations (such as read, write) are supported for each leaf object instance.

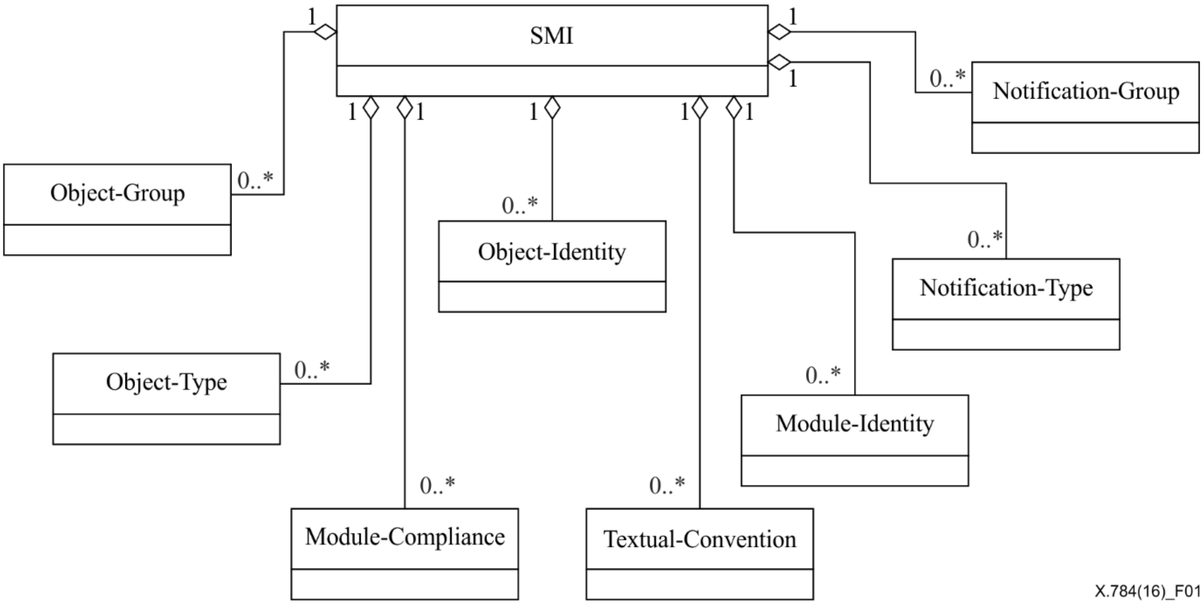

Figure 1 – Containment relationship in SMI specification

7.2 Guidelines for specification of SMI-based interface ICS proformas

Proforma specifications shall follow the style as documented in the following clauses. Proforma specifications shall provide the information required by this Recommendation. Additional tables may be included for other information, if needed.

There are three levels of documentation pertaining to SIICS, namely:

1) guidelines or Recommendation tools for the production of SIICS proformas;

2) a SIICS proforma, associated with a standard related to SNMP-based network management, which is to be filled in by a supplier of the implementation and which when filled in is a SIICS;

3) a SIICS prepared by a supplier of the implementation as part of a conformance claim to a standard related to SNMP-based network management.

7.2.1 General instructions for SMI ICS proforma specification

This Recommendation provides instructions to construct an SNMP-based management system ICS proforma specification. SIICS proformas include three sub proformas: a conceptual table object support proforma, a leaf object support proforma and a notification support proforma. All these proformas are in a tabular form which is similar to the proformas specified in [ITU-T X.781].

Clause 7.2.2 describes the SIICS profile for the interface. Clause 7.2.3 describes the SIICS for conceptual tables. Clause 7.2.4 describes the SIICS for leaves. Clause 7.2.5 describes the SIICS for notifications. Appendix I provides examples of SIICS proformas specifications, which are to be filled in by a supplier of an implementation.

The following common notations, defined in [ITU-T X.291] | [ISO/IEC 9646-2] and in [ITU-T X.296] | [ISO/IEC 9646-7], are used for the "status" value column in this Recommendation:

m	Mandatory
o	Optional
–	Not applicable or out of scope

NOTE – The notations "m", "o" are prefixed by a "c:" when nested under an optional item of the same table.

The following common notations, defined in [ITU-T X.291] | [ISO/IEC 9646-2] and in [ITU-T X.296] | [ISO/IEC 9646-7] are used for the support answer column:

Y Implemented

N Not implemented

– No answer required

Ig The item is ignored (i.e., processed syntactically but not semantically)

The SIICS proforma specification is formed by copying clauses 7.2.2, 7.2.3, 7.2.4 and 7.2.5, completing the tables except for the "Support" and "Additional information" columns and by extending the remaining tables to meet the requirements of the specification.

To form a SIICS from a SIICS proforma, the supplier of the implementation shall fill in the "Support" and if appropriate, "Additional information" columns of all the tables in the SIICS proforma.

7.2.2 SIICS profile for the interface

Table 1 is used to give the SIICS profile for the interface to be tested.

Table 1 – SIICS profile for the interface

Index	Name of table	Description of table

where:

– The "Index" field is made up of a consecutive number for readers of the SIICS to refer to each item.

– The "Name of table" field gives the name of the table.

– The "Description of table" field is the descriptor of the corresponding table.

7.2.3 Conceptual table object support proforma

The conceptual table object support proforma is used to check whether the object instance could support operations such as read, write, create. The conceptual table object support proforma can be divided into the conceptual table object support table (see Table 2), the column support table (see Table 3), the entry support table (see Table 4) and the conceptual table index support table (see Table 5).

Table 2 – Conceptual table object support table

Index	Descriptor	OID	Validity	Support all the mandatory attributes? (Y/N)	Implementation conforms to conformance statement? (Y/N)

where:

– For each conceptual table object defined in the MIB, there is a corresponding conceptual table object support table with only one row.

– The "Index" field is made up of a consecutive number for readers of the SIICS to refer to each item.

– The "Descriptor" field is a string that must be unique in the same MIB.

– The "OID" field is the OBJECT IDENTIFIER of the conceptual table object. The OID can uniquely identify the object, such as x.x.x.x (x is a nonnegative integer).

– The "Validity" field shall be filled with one of "current", "deprecated" and "obsolete", the meaning of each is as follows:

- "current" indicates that the object is being used;
- "deprecated" indicates that the object is going out of date. It's best not to use it, but the definition may support backward compatibility;
- "obsolete" indicates that the object is invalid and cannot be used.

– The "Support all the mandatory attributes?" field shall be filled with "Y" or "N", which indicates whether the system under test has implemented all the mandatory attributes included in the conceptual table object.

– The "Implementation conforms to conformance statement?" field is the test result. After checking the consistency between all the support tables related to the Conceptual table object support proforma and MIB, the tester will fill it in with "Y" or "N"..

Table 3 – Column support table

Basic Characteristics						**Conformance Requirements**			**Support operations**												**Additional information**
									Inaccessible		**Accessible for notify**		**Read**		**Write**		**Create the assignment**		**Set the default value**		
Index	Descriptor	OID	Limit and value		Validity	Status	Support	Appendix information index	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	
			Read	Write																	

where:

– For each conceptual table object defined in the MIB, there is a corresponding column support table to describe the attributes of the column objects included in the conceptual table object.

– The "Index" field is an integer and index values for each successive line must also be continuous. In the column support table, each column object shall be arranged according to the OID in lexicographical order.

– The "Descriptor" field is the descriptor of the corresponding column object.

– The "OID" field is the OBJECT IDENTIFIER of the corresponding column object.

– The "Limit and value" field is divided into two sub-columns: "read" and "write". Each column is filled with a basic data type of SMIv2, indicating the data type of the column object when reading or writing. If the data type used in the MIB is defined by TEXTUAL-CONVENTION, it shall be transformed into a corresponding basic data type before being filled in. The data types of attributes "read" and "write" of the column object are usually the same. The data types are defined in the SYNTAX clause of the object type macro.

– The "Validity" field shall be filled in with one of "current", "deprecated" or "obsolete", the meaning of each is as follows:

- "current" indicates that the object is being used;
- "deprecated" indicates that the object is going out of date. It is best not to use it, but the definition may support backward compatibility;
- "obsolete" indicates that the object is invalid and cannot be used.

– The "Status" field of "Conformance Requirements" shall be filled with one of "m", "c" or "o", which signify "mandatory", "mandatory-if-applicable" and "optional". The meaning of each is as follows:

- "mandatory" means that the column object must be implemented on any condition;
- "mandatory-if-applicable" means that the column object must be implemented on the condition that a specific protocol has been implemented or some other object group has been implemented;
- "optional" means that the column object can be chosen arbitrarily.

– The "Support" field of "Conformance Requirements" shall be filled with "Y" or "N", where "Y" means the system under test has implemented the column object and "N" means it has not.

NOTE –There is no need to test the operations of the column object if the implementer of the system under test declares that some column objects are not supported.

– The "Appendix information index" field of "Conformance Requirements" shall be filled with an integer or "-". If the column object is "mandatory-if-applicable", then the corresponding condition number in the extensibility support proforma (see clause 7.2.6) shall be filled in, otherwise "-" shall be filled in.

– The "Support operations" field indicates six operations that can be performed on the column object, corresponding to the six sub-columns. Each sub-column is further divided into "status" and "support". "Status" can be filled with "m" or "-", which means "can" or "cannot" support this operation. "Support" can be filled in "Y" or "N", which indicates whether or not the system under test can support the operation. The conditions for filling "m" in "state" are as follows:

- "Inaccessible": when the value of the defined clause MAX-ACCESS is "not-accessible".
- "Accessible for notify", "read", "write" and "create the assignment": This is decided by the MAX-ACCESS and the MIN-ACCESS (if the clause does not exist, the minimum access is decided by "accessible-for-notify"). The level of the four operations increasing in sequence, where "create the assignment" has the highest level and "Accessible for notify" has the lowest level. So whether or not to support some of the 4 operations can be ensured by defining MAX-ACCESS and MIN-ACCESS.
- "Set the default value": when the default value has been defined in the column object (there is a DEFVAL clause).

– The "Additional information" field is filled in by the user under test when the ICS proforma is created to give some supplemental information to the column object. "Additional information" can be filled in with the default value when the ICS proforma is created if the default value has been defined in the column object.

Table 4 – Entry support table

Index	**Descriptor**	**OID**	**CreateAndGo**		**CreateAndWait**		**Destroy**		**Additional information**
			Status	**Support**	**Status**	**Support**	**Status**	**Support**	

where:

- For each conceptual table object defined in the MIB, there is a corresponding entry support table to describe the attributes of the entry object in the conceptual table object.
- The "Index" field is an index number of each table object in order.
- The "Descriptor" field is the descriptor of the entry object.
- The "OID" field is the OBJECT IDENTIFIER of the entry object.

  There are at most three operations on the entry object:

  - CreateAndGo: Create new entry and set it as available;
  - CreateAndWait: Create new entry, but the entry cannot be used;
  - Destroy: Delete all the instances related to the logical entry.

  The three operations must be performed when the command SET is given by the manager.
- The "Status" field of the three operations can be filled with "m" or "-", which means "can" or "cannot" take this operation. The rule is as follows:
  - If the data type of one of the column objects of the definition of entry is RowStatus, the entry can be created and deleted and the "state" field of CreateAndGo and CreateAndWait shall be filled with "o" to indicate that the two methods to create are optional and that at least one of the methods shall be supported. The "Status" field of Destroy shall be filled with "m", indicating that the delete operation must be supported;
  - If none of the column objects of the definition of entry has the data type of RowStatus, the table cannot be created and deleted, the "Status" field of the 3 operations shall be filled with "-".
- The "Support" field shall be filled with "Y" or "N", which indicates whether or not the system under test supports the operation.

Table 5 – Conceptual table index support table

Index	**Descriptor**	**OID**	**Index Items**							**Additional information**	**Status**	**Support**
			Type	**Sub-index**	**Descriptor**	**OID**	**Limit and value**	**Descriptor of the table that index object belongs to**	**OID of the table that index object belongs to**			

where:

- For each conceptual table object defined in the MIB, there is a corresponding table index support table to describe the index of the conceptual table object.
- The "Index" field is an index number of each table object in order.
- The "Descriptor" field is the descriptor of the table object.
- The "OID" field is the OBJECT IDENTIFIER of the table object. The "Index Items" field is made up of the index type of the table object and the column object that constitute the index of the table object.
- The "Type" field of "Index Items" can be filled with "I" or "A", indicating type of INDEX or AUGMENTS. The table index can be broken down simply into two types:
  - Type of INDEX: the index defined by the clause INDEX. The descriptor of the column object that was used as the index can be listed by the clause INDEX directly. These column objects can belong to this table object or to another table object.

- Type of AUGMENTS: the index defined by the clause AUGMENTS. The AUGMENTS clause does not list the descriptor of the column object that was used as the index, but references the descriptor of the column object of another table object, indicating this table object uses the index of the referenced table object.

– The "Sub-index" field of "Index Items" is an integer and the sub-index number of each entry is continuous. The index of the table object is made up of one or more column objects. The "Index entry" field lists these column objects in the defined sequence. "Sub-index" is the sequence number of the permutation.

– The "Descriptor" field of "Index Items" is the descriptor of the column object that was used as the index. If the index is made up of more than one part, the descriptor and OID of all the column objects will be listed in this field in the defined sequence and each entry contains one of them.

– The "OID" field of "Index Items" is the OID of the column object that was used as the index.

– "Limit and value" is the data type of the column object that was used as the index.

– The "Descriptor of the table that index object belongs to" field of "Index Items" indicates that the column object used as the index belongs to this table object or to another table object. This field shall be filled with the descriptor of the table that this index object belongs to.

– The "OID of the table that index object belongs to" field of "Index Items" field shall be filled with the OID of the table that this index object belongs to.

– The "Additional information" field shall filled with some essential additional information by the tester or user under test, such as the possible relationship between this table object and another table object and so on.

– The "Status" field is always "m", indicating the system under test must be implemented according to the table index defined in the MIB.

– The "Support" field shall be filled with "Y" or "N", indicating whether or not the system under test supports this table index.

7.2.4 Leaf object support proforma

The leaf object support proforma is used to check whether the leaf object instance could support operations such as read, write.

Table 6 – Leaf object support table

Basic properties						Conformance Requirements			Support Operations								Additional information
									Inaccessible		Accessible for notify		Read		Write		
Index	Descriptor	OID	Limit and value		Validity	Status	Support	Appendix information index	Status	Support	Status	Support	Status	Support	Status	Support	
			Read	Write													

where:

– All the leaf objects defined in the MIB share one leaf object support table, describing the attributes of the leaf object. Each leaf object occupies one entry in the table.

- The "Index" field is an integer and index values for each successive line must also be continuous. In the leaf object support table, each leaf object shall be arranged according to the OID in lexicographical order. "Index" is the sequence number.
- The "Descriptor" field is the descriptor of the corresponding leaf object.
- The "OID" field is the OBJECT IDENTIFIER of the corresponding leaf object.
- The "Limit and value" field is divided into two sub-fields: "read" and "write". Each field is filled with a basic data type of SMIv2, indicating the data type of the leaf object when reading and writing. If the data type used in the MIB is defined by TEXTUAL-CONVENTION, it shall be transformed into the corresponding basic data type before being filled in. The data types of attributes "read" and "write" of the leaf object are usually the same. The data types are defined in the SYNTAX clause of the object type macro. The SYNTAX clause of the leaf object can be redefined when the conformance statements are made by MODULE-COMPLIANCE. The data type of the WRITE-SYNTAX clause will be defined separately and this field shall be filled with the new-defined data type.
- The "Validity" field shall be filled with one of "current", "deprecated" or "obsolete", the meaning of each is as follows:
  - "current" indicates that the object is being used;
  - "deprecated" indicates that the object is going out of date. It is best not to use it, but the definition may support backward compatibility;
  - "obsolete" indicates that the object is invalid and cannot be used.
- The "Status" field of "Conformance Requirements" shall be filled with one of "m", "c" or "o", which mean respectively "mandatory", "mandatory-if-applicable" and "optional". The meaning of each is as follows:
  - "mandatory" means that the leaf object must be implemented on any condition;
  - "mandatory-if-applicable" means that the leaf object must be implemented on the condition that a specific protocol has been implemented or some other object group has been implemented;
  - "optional" means that the leaf object can be chosen arbitrarily.
- The "Support" field of "Conformance Requirements" shall be filled with "Y" or "N", which indicates whether the system under test has implemented the leaf object or not.

NOTE –There is no need to test the operations of the leaf object if the implementer of the system under test declares that some leaf objects are not supported.

- The "Appendix information index" field of "Conformance Requirements" shall be filled with an integer or "-". If the column object is "mandatory-if-applicable", then the corresponding condition number in the extensibility support proforma (see clause 7.2.6) shall be filled in, otherwise "-" shall be filled in.
- The "Support Operations" field indicates four operations that can be take on the leaf object, corresponding to the four sub-columns. Each includes "Status" and "Support" sub-columns. "Status" can be filled with "m" or "-", which means "can" or "cannot" take this operation. "Support" can be filled in "Y" or "N", which indicates whether or not the system under test can support the operation. The conditions for filling in "m" in "Status" are as follows:
  - "Inaccessible": when the value of the defined clause MAX-ACCESS is "not-accessible".
  - "Accessible for notify", "read", "write" and "create the assignment": this is decided by the MAX-ACCESS and the MIN-ACCESS clauses (if the clause does not exist, the minimum access is decided by "accessible-for-notify") defined in the leaf object.

- The "Additional information" field is filled by the user under test and this field is a supplementary specification to the leaf object.

7.2.5 Notification support proforma

The notification support proforma is used to check whether the system under test could send the notification with right format.

Table 6 – Notification support table

Index	Descriptor	OID	Validity	Conformance Requirements			Sub-index	Descriptor of the object of all the fields of the notification	OID of the object of all the fields of the notification	Limit and value	Status	Support	Additional information
				Status	Support	Appendix information index							

where:

- All the notification objects defined in the MIB share one notification support table, describing the attributes of the notification object. Each notification object occupies one entry in the table.
- The "Index" field is an integer and index values for each successive line must also be continuous. In the notification support table, each notification object shall be arranged according to the OID in lexicographical order. "Index" is the sequence number.
- The "Descriptor" field is the descriptor of the corresponding notification object.
- The "OID" field is the OBJECT IDENTIFIER of the corresponding notification object.
- The "Validity" field shall be filled with one of "current", "deprecated" or "obsolete", the meaning of each is as follows:
  - "current" indicates that the object is being used;
  - "deprecated" indicates that the object is going out of date. It is best not to use it, but the definition may support backward compatibility;
  - "obsolete" indicates that the object is invalid and cannot be used.
- The "Status" field of "Conformance Requirements" shall be filled with one of "m", "c" or "o", which mean respectively "mandatory", "mandatory-if-applicable" and "optional". The meaning of each is as follows:
  - "mandatory" means that the column object must be implemented on any condition;
  - "mandatory-if-applicable" means that the column object must be implemented on the condition that a specific protocol has been implemented or some other object group has been implemented;
  - "optional" means that the column object can be chosen arbitrarily.
- The "Support" field of "Conformance Requirements" shall be filled with "Y" or "N", which indicates whether or not the system under test has implemented the notification object.

NOTE – There is no need to test the operations of the notification object if the implementer of the system under test declares that some notification objects are not supported.

- The "Appendix information index" field of "Conformance Requirements" shall be filled with an integer or "-". If the column object is "mandatory-if-applicable", then the corresponding condition number in the extensibility support proforma (see clause 7.2.6) shall be filled in, otherwise "-" shall be filled in.

– The "Sub-index" field shall be filled as the format of "index of notification. Index of the reference object of the notification". The Index of notification is same as the value in the "Index". The "Index of the reference object of the notification" is the sequence number of the object referenced by this notification in defined order.

– The "Descriptor of the object of all the fields of the notification" field shall be filled with the descriptor of the object referenced by the notification.

– The "OID of the object of all the fields of the notification" field shall be filled with the OID of the object referenced by the notification.

– The "Limit and value" field of the notification is the data type of the object referenced by notification and this field must be filled with the basic data type of SMIv2. If the data types of read and write operations are different, the field shall be filled with the data type of a read operation.

– The "Status" field is always "m", indicating the notification must send the object referenced by the definition of the notification in the defined sequence.

– The "Support" field shall be filled with "Y" or "N", indicating whether the notification of the system under test has referenced the object.

– The "Additional information" field is filled in by the user under test and this field is a supplementary specification to the notification object. This field is not required.

7.2.6 Extensibility support proforma

In the ICS sub proforma table supported by the column object, the notification object and leaf object "Status" field can declare some objects in the object group as "mandatory-if-applicable" and the condition needs to be shown in the ICS proforma. This condition is a string that has indefinite length, so it cannot be included in the sub proforma table. The ICS proforma based on the network management interface in SMIv2 shall add an appendix to store the "condition" when necessary.

The conditions of "mandatory-if-applicable" are listed in the appendix, and they are numbered in sequence. The number will be described in the "appendix information index" field of "conformance requirements" of the corresponding ICS sub proforma of all objects. Then the condition can be queried in the appendix by the number.

8 Instruction for completing the SMI-based interface ICS proformas

This clause gives the instructions for completing each column defined in clause 7.2.

8.1 Definition of "supported"

A capability is said to be supported if the implementation under test (IUT) is able to realize the specified functionality.

8.2 The "Index" column

At the left-hand of each row in the SIICS proforma, a specific number is provided in this "Index". This numbering is used to identify all possible implementation details within the SIICS proforma uniquely.

This unique number is constructed with a sequence as follows:

a) a reference to the super-clause of the item;

b) the separating character ".";

c) a unique number.

The "Index" column exhibits the containment relationship between SMI syntax structures. This containment relationship is referred to in clause 7.1 "MIB and SMI overview".

8.3 The "subIndex" column

This column has a similar meaning and the same format as the "Index" column. It is also made up of consecutive numbers.

For each row, only the "Index" column in the service support proforma stands for the item of this row. In other proformas, the "Index" column specifies the super-clause item of the item in this row while the "subIndex" column identifies the item in this row.

8.4 The "Support" column

This "Support" column shall be completed by the supplier or implementer to indicate the level of implementation of each item. The available selections for this field are listed in clause 7.2.1.

Guidelines for completing this field:

a) if an item is claimed as "supported", all the mandatory items it contains must also be supported;

b) otherwise, the "support" column can be filled with 'N';

c) if the "Status" column of an item is filled with '-', the only selection for the corresponding "support" column is '-'.

In the SIICS proforma tables, every item marked with 'm' shall be supported by the IUT.

8.5 The "Additional information" column

The "Additional information" column contains additional information provided by suppliers not contained anywhere else. To achieve SIICS, for each proforma there may some important and necessary information that is not or cannot be contained in any column of the row. In such cases, suppliers can fill in this field with this information.

8.6 The "Validity" column

The "Validity" field shall be filled with one of "current", "deprecated" or "obsolete", the meaning of each is as follows: "current" indicates that the object is being used; "deprecated" indicates that the object is going out of date. It is best not to use it, but the definition may support backward compatibility; "obsolete" indicates that the object is invalid and cannot be used.

Appendix I

Examples of the SMI-based interface ICS proforma specification

(This appendix does not form an integral part of this Recommendation.)

I.1 MIB definition example 1

In this appendix, an example is provided using the UDP-MIB definition:

```
UDP-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Counter32,
    IpAddress, mib-2                   FROM SNMPv2-SMI
    MODULE-COMPLIANCE, OBJECT-GROUP    FROM SNMPv2-CONF;
udpMIB MODULE-IDENTITY
    LAST-UPDATED "9411010000Z"
    ORGANIZATION "IETF SNMPv2 Working Group"
    CONTACT-INFO
            "        Keith McCloghrie
             Postal: Cisco Systems, Inc.
                     170 West Tasman Drive
                     San Jose, CA  95134-1706
                     US
             Phone:  +1 408 526 5260
             Email:  kzm@cisco.com"
    DESCRIPTION
            "The MIB module for managing UDP implementations."
    REVISION      "9103310000Z"
    DESCRIPTION
            "The initial revision of this MIB module was part of MIB-
            II."
    ::= { mib-2 50 }
udp      OBJECT IDENTIFIER ::= { mib-2 7 }
udpInDatagrams OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The total number of UDP datagrams delivered to UDP users."
    ::= { udp 1 }
udpNoPorts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The total number of received UDP datagrams for which there
            was no application at the destination port."
    ::= { udp 2 }
udpInErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The number of received UDP datagrams that could not be
            delivered for reasons other than the lack of an application
            at the destination port."
    ::= { udp 3 }
udpOutDatagrams OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The total number of UDP datagrams sent from this entity."
    ::= { udp 4 }
udpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF UdpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A table containing UDP listener information."
    ::= { udp 5 }
udpEntry OBJECT-TYPE
    SYNTAX      UdpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

```
            "Information about a particular current UDP listener."
    INDEX   { udpLocalAddress, udpLocalPort }
    ::= { udpTable 1 }
UdpEntry ::= SEQUENCE {
        udpLocalAddress  IpAddress,
        udpLocalPort     INTEGER
    }
udpLocalAddress OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The local IP address for this UDP listener.  In the case of
            a UDP listener which is willing to accept datagrams for any
            IP interface associated with the node, the value 0.0.0.0 is
            used."
    ::= { udpEntry 1 }
udpLocalPort OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The local port number for this UDP listener."
    ::= { udpEntry 2 }
    END


    ENTITY-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE,
        mib-2, NOTIFICATION-TYPE
            FROM SNMPv2-SMI
        TDomain, TAddress, DisplayString, TEXTUAL-CONVENTION,
        AutonomousType, RowPointer, TimeStamp
            FROM SNMPv2-TC
        MODULE-COMPLIANCE, OBJECT-GROUP
            FROM SNMPv2-CONF;

    entityMIB MODULE-IDENTITY
        LAST-UPDATED "9605160000Z"
        ORGANIZATION "IETF ENTMIB Working Group"
        CONTACT-INFO
                "    WG E-mail: entmib@cisco.com
                     Subscribe: majordomo@cisco.com
                                msg body: subscribe entmib

                     Keith McCloghrie
                     ENTMIB Working Group Chair
                     Cisco Systems Inc.
                     170 West Tasman Drive
                     San Jose, CA 95134
                     408-526-5260
                     kzm@cisco.com

                     Andy Bierman
                     ENTMIB Working Group Editor
                     Cisco Systems Inc.
                     170 West Tasman Drive
                     San Jose, CA 95134
                     408-527-3711
                     abierman@cisco.com"
        DESCRIPTION
                "The MIB module for representing multiple logical
                entities supported by a single SNMP agent."
        ::= { mib-2 47 }
    entityMIBTraps      OBJECT IDENTIFIER ::= { entityMIB 2 }
    entityMIBTrapPrefix OBJECT IDENTIFIER ::= { entityMIBTraps 0 }

    entConfigChange NOTIFICATION-TYPE
        STATUS             current
        DESCRIPTION
                "An entConfigChange trap is sent when the value of
                entLastChangeTime changes. It can be utilized by an NMS to
                trigger logical/physical entity table maintenance polls.
                An agent must not generate more than one entConfigChange
                'trap-event' in a five second period, where a 'trap-event'
                is the transmission of a single trap PDU to a list of trap
                destinations.  If additional configuration changes occur
                within the five second 'throttling' period, then these
                trap-events shall be suppressed by the agent. An NMS shall
                periodically check the value of entLastChangeTime to detect
                any missed entConfigChange trap-events, e.g. due to
                throttling or transmission loss."
       ::= { entityMIBTrapPrefix 1 }
    END
```

I.2 SIICS Profile for UDP-MIB

According to the MIB definition above, the SIICS for this MIB definition can be illustrated as in Tables I.1 to I.7.

Table I.1 – SIICS profile for UDP-MIB

Index	Name of table	Description of table
1	UDP-MIB.udpTable conceptual table object support table	This table is used to test whether the operations (read, write, create, etc.) are supported for each object instance.
2	udpTable column support table	This table is used to test whether the operations (read, write, create, etc.) are supported for each object instance.
3	udpEntry entry support table	This table is used to test whether the operations (read, write, create, etc.) are supported for each object instance.
4	udpTable conceptual table index support table	This table is used to test whether the operations (read, write, create, etc.) are supported for each object instance.
5	Leaf object support table	This table is used to test whether the operations (read, write, etc.) are supported for each leaf object instance.
6	Notification support table	This table is used to test whether the notification could be sent out with the correct format.

I.3 SIICS for UDP-MIB

Table I.2 – UDP-MIB.udpTable conceptual table object support table

Index	Descriptor	OID	Validity	Support all the mandatory attributes? (Y/N)	Implementation conforms to conformance statement? (Y/N)
1	udpTable	1.3.6.1.2.1.7.5	current		

Table I.3 – udpTable column support table

Basic Characteristics						Conformance Requirements			Support operations												Additional information
									Inaccessible		Accessible for notify		Read		Write		Create the assignment		Set the default value		
Index	Descriptor	OID	Limit and value		Validity	Status	Support	Appendix information index	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	
			Read	Write																	
1	udpLocalAddress	1.3.6.1.2.1.7.5.1.1	TC:=IpAddress	TC:=IpAddress	current	o		-	-	-	m		m		-	-	-		-		
2	udpLocalPort	1.3.6.1.2.1.7.5.1.2	TC:=INTEGER 0..65535	TC:=INTEGER 0..65535	current	o		-	-	-	m		m		-	-	-		-		

Table I.4 – udpEntry entry support table

Index	Descriptor	OID	CreateAndGo		CreateAndWait		Destroy		Additional information
			Status	Support	Status	Support	Status	Support	
1	udpEntry	1.3.6.1.2.1.7.5.1	o		o		m		

Table I.5 – udpTable conceptual table index support table

Index	Descriptor	OID	Index Items							Additional information	Status	Support
			Type	Sub-index	Descriptor	OID	Limit and value	Descriptor of the table that index object belongs to	OID of the table that index object belongs to			
1	udpTable	1.3.6.1.2.1.7.5	I	1	udpLocalAddress	1.3.6.1.2.1.7.5.1.1	IpAddress	udpTable	1.3.6.1.2.1.7.5		m	
			I	2	udpLocalPort	1.3.6.1.2.1.7.5.1.2	INTEGER	udpTable	1.3.6.1.2.1.7.5		m	

Table I.6 – Leaf object support table

Basic properties						Conformance Requirements			Support Operations								Additional information
									Inaccessible		Accessible for notify		Read		Write		
Index	Descriptor	OID	Limit and value		Validity	Status	Support	Appendix information index	Status	Support	Status	Support	Status	Support	Status	Support	
			Read	Write													
1	UDP-MIB.udpInDatagrams	1.3.6.1.2.1.7.1	TC:=Counter32	TC:=Counter32	current			–	-		m		m		–	–	
2	UDP-MIB.udpNoPorts	1.3.6.1.2.1.7.2	TC:=Counter32	TC:=Counter32	current			–	-		m		m		–	–	
3	UDP-MIB.udpInErrors	1.3.6.1.2.1.7.3	TC:=Counter32	TC:=Counter32	current			–	-		m		m		–	–	
4	UDP-MIB.udpOutDatagrams	1.3.6.1.2.1.7.4	TC:=Counter32	TC:=Counter32	current			–	–		m		m		–	–	

Table I.7 – Notification support table

Index	Descriptor	OID	Validity	Conformance Requirements			Sub-index	Descriptor of the object of all the fields of the notification	OID of the object of all the fields of the notification	Limit and value	Status	Support	Additional information
				Status	Support	Appendix information index							
1	ENTITY-MIB.entConfigChange	1.3.6.1.2.1.47.2.0.1				–							

I.4 MIB definition example 2

In this appendix, an example is provided using the SNMPv2-MIB definition:

```
SNMPv2-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    TimeTicks, Counter32, snmpModules, mib-2
        FROM SNMPv2-SMI
    DisplayString, TestAndIncr, TimeStamp
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
        FROM SNMPv2-CONF;
snmpMIB MODULE-IDENTITY
    LAST-UPDATED "9511090000Z"
    ORGANIZATION "IETF SNMPv2 Working Group"
    CONTACT-INFO
            "        Marshall T. Rose

             Postal: Dover Beach Consulting, Inc.
                     420 Whisman Court
                     Mountain View, CA  94043-2186
                     US

                Tel: +1 415 968 1052

             E-mail: mrose@dbc.mtview.ca.us"
    DESCRIPTION
            "The MIB module for SNMPv2 entities."
    REVISION      "9304010000Z"
    DESCRIPTION
            "The initial revision of this MIB module was published as
            RFC 1450."
    ::= { snmpModules 1 }


snmpMIBObjects OBJECT IDENTIFIER ::= { snmpMIB 1 }

system   OBJECT IDENTIFIER ::= { mib-2 1 }

sysDescr OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "A textual description of the entity.  This value shall
            include the full name and version identification of the
            system's hardware type, software operating-system, and
            networking software."
    ::= { system 1 }

sysObjectID OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The vendor's authoritative identification of the network
            management subsystem contained in the entity.  This value is
            allocated within the SMI enterprises subtree (1.3.6.1.4.1)
            and provides an easy and unambiguous means for determining
            `what kind of box' is being managed.  For example, if vendor
            `Flintstones, Inc.' was assigned the subtree
            1.3.6.1.4.1.4242, it could assign the identifier
            1.3.6.1.4.1.4242.1.1 to its `Fred Router'."
    ::= { system 2 }

sysUpTime OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The time (in hundredths of a second) since the network
            management portion of the system was last re-initialized."
    ::= { system 3 }

sysContact OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The textual identification of the contact person for this
            managed node, together with information on how to contact
            this person.  If no contact information is known, the value
            is the zero-length string."
    ::= { system 4 }

sysName OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "An administratively-assigned name for this managed node.
            By convention, this is the node's fully-qualified domain
            name.  If the name is unknown, the value is the zero-length
            string."
    ::= { system 5 }

sysLocation OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The physical location of this node (e.g., `telephone
            closet, 3rd floor').  If the location is unknown, the value
            is the zero-length string."
    ::= { system 6 }

sysServices OBJECT-TYPE
    SYNTAX      INTEGER (0..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "A value which indicates the set of services that this
            entity may potentially offers.  The value is a sum.  This
            sum initially takes the value zero, Then, for each layer, L,
            in the range 1 through 7, that this node performs
            transactions for, 2 raised to (L - 1) is added to the sum.
            For example, a node which performs only routing functions
            would have a value of 4 (2^(3-1)).  In contrast, a node
            which is a host offering application services would have a
            value of 72 (2^(4-1) + 2^(7-1)).  Note that in the context
            of the Internet suite of protocols, values shall be
            calculated accordingly:

                 layer      functionality
                   1        physical (e.g., repeaters)
                   2        datalink/subnetwork (e.g., bridges)
                   3        internet (e.g., supports the IP)
                   4        end-to-end  (e.g., supports the TCP)
                   7        applications (e.g., supports the SMTP)

            For systems including OSI protocols, layers 5 and 6 may also
            be counted."
    ::= { system 7 }

sysORLastChange OBJECT-TYPE
    SYNTAX     TimeStamp
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "The value of sysUpTime at the time of the most recent
            change in state or value of any instance of sysORID."
    ::= { system 8 }

sysORTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF SysOREntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "The (conceptual) table listing the capabilities of the
            local SNMPv2 entity acting in an agent role with respect to
            various MIB modules.  SNMPv2 entities having dynamically-
            configurable support of MIB modules will have a
            dynamically-varying number of conceptual rows."
    ::= { system 9 }

sysOREntry OBJECT-TYPE
    SYNTAX     SysOREntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "An entry (conceptual row) in the sysORTable."
    INDEX      { sysORIndex }
    ::= { sysORTable 1 }

SysOREntry ::= SEQUENCE {
    sysORIndex     INTEGER,
    sysORID        OBJECT IDENTIFIER,
    sysORDescr     DisplayString,
    sysORUpTime    TimeStamp
}

sysORIndex OBJECT-TYPE
    SYNTAX     INTEGER (1..2147483647)
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "The auxiliary variable used for identifying instances of
            the columnar objects in the sysORTable."
    ::= { sysOREntry 1 }

sysORID OBJECT-TYPE
    SYNTAX     OBJECT IDENTIFIER
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "An authoritative identification of a capabilities statement
            with respect to various MIB modules supported by the local
            SNMPv2 entity acting in an agent role."
    ::= { sysOREntry 2 }

sysORDescr OBJECT-TYPE
    SYNTAX     DisplayString
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "A textual description of the capabilities identified by the
            corresponding instance of sysORID."
    ::= { sysOREntry 3 }

sysORUpTime OBJECT-TYPE
    SYNTAX     TimeStamp
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "The value of sysUpTime at the time this conceptual row was
            last instanciated."
    ::= { sysOREntry 4 }

snmpTraps      OBJECT IDENTIFIER ::= { snmpMIBObjects 5 }

coldStart NOTIFICATION-TYPE
    STATUS  current
    DESCRIPTION
```

```
            "A coldStart trap signifies that the SNMPv2 entity, acting
            in an agent role, is reinitializing itself and that its
            configuration may have been altered."
    ::= { snmpTraps 1 }

warmStart NOTIFICATION-TYPE
    STATUS  current
    DESCRIPTION
            "A warmStart trap signifies that the SNMPv2 entity, acting
            in an agent role, is reinitializing itself such that its
            configuration is unaltered."
    ::= { snmpTraps 2 }

authenticationFailure NOTIFICATION-TYPE
    STATUS  current
    DESCRIPTION
            "An authenticationFailure trap signifies that the SNMPv2
            entity, acting in an agent role, has received a protocol
            message that is not properly authenticated.  While all
            implementations of the SNMPv2 must be capable of generating
            this trap, the snmpEnableAuthenTraps object indicates
            whether this trap will be generated."
    ::= { snmpTraps 5 }

-- Note the egpNeighborLoss NOTIFICATION-TYPE ::= { snmpTraps 6 }
END
```

I.5 SIICS profile for SNMPv2-MIB

According to the MIB definition above, the SIICS for this MIB definition can be illustrated as in Tables I.8 to I.14.

Table I.8 – SIICS profile for SNMPv2-MIB

Index	Name of table	Description of table
1	sysORTable conceptual table object support table	This table is used to test whether the operations (read, write, create, etc.) are supported for each object instance.
2	sysORTable column support table	This table is used to test whether the operations (read, write, create, etc.) are supported for each object instance.
3	sysOREntry entry support table	This table is used to test whether the operations (read, write, create, etc.) are supported for each object instance.
4	sysORTable conceptual table index support table	This table is used to test whether the operations (read, write, create, etc.) are supported for each object instance.
5	Leaf object support table	This table is used to test whether the operations (read, write, etc.) are supported for each leaf object instance.
6	Notification support table	This table is used to test whether the notification could be sent out with the correct format.

I.6 SIICS for the SNMPv2-MIB

Table I.9 – sysORTable conceptual table object support table

Index	Descriptor	OID	Validity	Support all the mandatory attributes? (Y/N)	Implementation conforms to conformance statement? (Y/N)
1	sysORTable	1.3.6.1.2.1.1.9	current		

Table I.10 – sysORTable column support table

Basic Characteristics						Conformance Requirements			Support operations												Additional information
									Inaccessible		Accessible for notify		Read		Write		Create the assignment		Set the default value		
Index	Descriptor	OID	Limit and value		Validity	Status	Support	Appendix information index	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	
			Read	Write																	
1	sysORIndex	1.3.6.1.2.1.1.9.1.1	TC:=INTEGER 1..2147483647	TC:=INTEGER 1..2147483647	current	o		–	m		–		–	–	–	–	–		–		
2	sysORID	1.3.6.1.2.1.1.9.1.2	TC:=OBJECT IDENTIFIER	TC:=OBJECT IDENTIFIER	current	o		–	–	–	m		m		–	–	–		–		
3	sysORDescr	1.3.6.1.2.1.1.9.1.3	TC:DisplayString=OCTET STRING	TC:DisplayString=OCTET STRING	current	o		–	–	–	m		m		–	–	–		–		
4	sysORUpTime	1.3.6.1.2.1.1.9.1.4	TC:TimeStamp=TimeTicks	TC:TimeStamp=TimeTicks	current	o		–	–	–	m		m		–	–	–		–		

Table I.11 – sysOREntry entry support table

Index	Descriptor	OID	CreateAndGo		CreateAndWait		Destroy		Additional information
			Status	Support	Status	Support	Status	Support	
1	sysOREntry	1.3.6.1.2.1.1.9.1	o		o		m		

Table I.12 – sysORTable conceptual table index support table

Index	Descriptor	OID	Index Items							Additional information	Status	Support
			Type	Sub-index	Descriptor	OID	Limit and value	Descriptor of the table that index object belongs to	OID of the table that index object belongs to			
1	sysORTable	1.3.6.1.2.1.1.9	I	1	sysORIndex	1.3.6.1.2.1.1.9.1.1	INTEGER	sysORTable	1.3.6.1.2.1.1.9		m	

Table I.13 – Leaf object support table

Basic properties						Conformance Requirements			Support Operations								Additional information
									Inaccessible		Accessible for notify		Read		Write		
Index	Descriptor	OID	Limit and value		Validity	Status	Support	Appendix information index	Status	Support	Status	Support	Status	Support	Status	Support	
			Read	Write													
1	SNMPv2-MIB.sysDescr	1.3.6.1.2.1.1.1	TC:DisplayString=OCTET STRING 0..255	TC:DisplayString=OCTET STRING 0..255	current			–	–		m		m		–	–	
2	SNMPv2-MIB.sysObjectID	1.3.6.1.2.1.1.2	TC:= OBJECT IDENTIFIER	TC:=OBJECT IDENTIFIER	current			–	–		m		m		–	–	
3	SNMPv2-MIB.sysUpTime	1.3.6.1.2.1.1.3	TC:=TimeTicks	TC:=TimeTicks	current			–	–		m		m		–	–	

Table I.13 – Leaf object support table

Basic properties						Conformance Requirements			Support Operations								Additional information
									Inaccessible		Accessible for notify		Read		Write		
Index	Descriptor	OID	Limit and value		Validity	Status	Support	Appendix information index	Status	Support	Status	Support	Status	Support	Status	Support	
			Read	Write													
4	SNMPv2-MIB.sysContact	1.3.6.1.2.1.1.4	TC:DisplayString=OCTET STRING 0..255	TC:DisplayString=OCTET STRING 0..255	current			–	–		m		m		m		
5	SNMPv2-MIB.sysName	1.3.6.1.2.1.1.5	TC:DisplayString=OCTET STRING 0..255	TC:DisplayString=OCTET STRING 0..255	current			–	–		m		m		m		
6	SNMPv2-MIB.sysLocation	1.3.6.1.2.1.1.6	TC:DisplayString=OCTET STRING 0..255	TC:DisplayString=OCTET STRING 0..255	current			–	–		m		m		m		
7	SNMPv2-MIB.sysServices	1.3.6.1.2.1.1.7	TC:=INTEGER 0..127	TC:=INTEGER 0..127	current			–	–		m		m		–	–	

Table I.13 – Leaf object support table

Basic properties						Conformance Requirements			Support Operations								Additional information
									Inaccessible		Accessible for notify		Read		Write		
Index	Descriptor	OID	Limit and value		Validity	Status	Support	Appendix information index	Status	Support	Status	Support	Status	Support	Status	Support	
			Read	Write													
8	SNMPv2-MIB.sysORLastChange	1.3.6.1.2.1.1.8	TC:TimeStamp=TimeTicks	TC:TimeStamp=TimeTicks	current			–	–		m		m		–	–	

Table I.14 – Notification support table

Index	Descriptor	OID	Validity	Conformance Requirements			Sub-index	Descriptor of the object of all the fields of the notification	OID of the object of all the fields of the notification	Limit and value	Status	Support	Additional information
				Status	Support	Appendix information index							
1	SNMPv2-MIB.coldStart	1.3.6.1.6.3.1.1.5.1				–							
2	SNMPv2-MIB.warmStart	1.3.6.1.6.3.1.1.5.2				–							
3	SNMPv2-MIB.authenticationFailure	1.3.6.1.6.3.1.1.5.5				–							

Appendix II

Additional information about the conformance test methodology for SNMP-based interfaces

(This appendix does not form an integral part of this Recommendation.)

[IETF RFC 2580] gives conformance statements for SMIv2 and provides a systematic way to describe a group of managed objects that must be implemented for conformance to a standard. [IETF RFC 2580] has defined some macros or notations to describe whether an agent can implement one or more MIB modules. It only gives SMI definitions of these macros or notations and there is no content about testing methodology, process and forms. However, all these contents are very helpful to guide an implementer to claim conformance to an SMI-based interface definition and to a tester to check the statements of the systems with respect to a standard. ITU-T has given conformance testing processes, definition style and format of the implementation conformance statement for the CORBA-based network management interface. In this Recommendation, the methodology and process of the CORBA-based interface as defined in [ITU-T X.781] is followed and several proformas (tables) for each of the SMI components to be used in SNMP-based interfaces are provided.

Bibliography

[b-ITU-T H.341] Recommendation ITU-T H.341 (1999), *Multimedia management information base*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2016