

IPFIX Working Group
Internet-Draft
Intended Status: Standards Track
Expires: April 16, 2011

A. Johnson
B. Claise
P. Aitken
Cisco Systems, Inc.
October 16, 2010

Exporting MIB variables using the IPFIX Protocol
draft-johnson-ipfix-mib-variable-export-00

Abstract

This document specifies a way to export Management Information Base (MIB) objects within the IPFIX protocol, avoiding the need to define new IPFIX Information Elements for existing Management Information Base objects that are already fully specified.

This method requires an extension to the current IPFIX protocol. New Template Set and Options Template Sets are specified to allow the export of Simple Network Management Protocol (SNMP) MIB Objects.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on April 16, 2011.

<Johnson, Claise, Aitken> Expires April 16 2011

[Page 1]

Internet-Draft <Exporting MIB variable with IPFIX> Oct 2010
<Johnson, Claise, Aitken> Expires April 16 2011 [Page 2]

Copyright (c) 2010 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.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.	Introduction.....	4
2.	Terminology.....	6
3.	Example Use Cases.....	6
3.1	Detailing CPU Load History.....	6
3.2	Output Interface Queue Size in PSAMP Packet Report	7
4.	MIB OID Extended Template Formats.....	8
4.1	MIB OID Extended Template Record Format.....	8
4.2	MIB OID Extended Options Template Record Format...	9
4.3	MIB OID Extended Field Specifier Format.....	11
4.3.1	Standard Field Specifier Format.....	11
4.3.2	Extended Field Specifier Format for non-indexed MIB Object	12
4.3.3	Extended Field Specifier Format for Indexed MIB Object	15
4.4	Template Management.....	20
5.	The Collecting Process's Side.....	21
6.	Applicability.....	21
7.	Security Considerations.....	22
8.	IANA Considerations.....	22
9.	References.....	23
9.1	Normative References.....	23
9.2	Informative References.....	23

TO DO

ACTION: need a third example with an Options Template Set to show TBD2 in action. TBD2 is the Set ID of an Options Template Set that uses the extended Field Specifier.

1. Introduction

There is growing interest in using IPFIX to export management information, especially since periodically exporting large chunks of repetitive data from a device is often more appropriate than using a polling mechanism.

While initially targeted at different problems, there is a large parallel between the information transported via IPFIX and SNMP.

Internet-Draft <Exporting MIB variable with IPFIX> Oct 2010
Furthermore, certain Management Information Base objects are highly relevant to flows as they are understood today.

For example, in the IPFIX information model [RFC5102], several Information Elements coming from the SNMP world have already been specified. e.g. the interface's SNMP index (ifIndex, [RFC2863]), the interface's name (ifName, [RFC2863]) and the system uptime (sysUpTime, [RFC3418]). Rather than map existing SNMP MIB Object Identifiers to IPFIX Information Elements on a case by case basis, causing replication between the two models, it would be advantageous to enable the export of any existing or future SNMP objects as part of an IPFIX Data Record. However, this is not possible using the existing IPFIX Template Sets as specified in [RFC5101].

Another advantage of exporting MIB objects via IPFIX is that IPFIX would benefit from an extended series of types to be exported. Indeed, the simple and application-wide data types specified in SMIV2 [RFC2578], along with a new textual conventions, can be exported within IPFIX and then decoded in the Collector.

Yet another advantage of exporting MIB objects via IPFIX is the synchronization of the Data Record counters and the MIB counters. For example, if a Network Management Station (NMS) needs the interface counters, coming from ifTable in "The Interfaces Group MIB" [RFC2863]), at the time the Data Record ends, the NMS must poll the interface counters after receiving the Data Records. Unless synchronized Data Records export and SNMP polling is in place (which is difficult/impossible in practice because the Flow duration can not be predicted), the Flow counters and interface counters cannot be compared. With the export of the interface counters within the Data Record, this problem is avoided.

In this document, new Template Sets for Data Records and Options Records are specified to allow Templates to contain any combination of fields defined by traditional IPFIX Information Elements and/or MIB Object Identifiers. The MIB Object Identifiers can reference either non-indexed or indexed MIB objects. When an indexed MIB object is exported, a method to identify how that MIB object was indexed is specified so that the full meaning of the information being exported can be conveyed. A set of example use cases is used to illustrate how these specifications can be used.

Internet-Draft <Exporting MIB variable with IPFIX> Oct 2010
Since IPFIX is a push mechanism, initiated from the Exporter with no acknowledgment method, this specification doesn't provide the ability to execute configuration, unlike the SNMP protocol. Instead, this specification allows adding the value of MIB objects into IPFIX Data Records.

2. Terminology

IPFIX-specific terminology used in this document is defined in section 2 of [RFC5101]. For example: Information Element, Template, Template Record, Options Template Record, Template Set, Collector, Exporter, Flow Record, etc... As in [RFC5101], these IPFIX-specific terms have the first letter of a word capitalized.

This document prefers the more generic term "Data Record" as opposed to "Flow Record" as this specification allows the export of MIB objects.

MIB Object Identifier (MIB OID)

AlphaNumeric-format variable name, denoting a variable name expressed as a sequence of decimal numbers or names separated by periods, as specified by the OBJECT IDENTIFIER in [RFC2578].

MIB Object Identifier Information Element

An IPFIX Information Element ("MIBObjectIdentifierMarker") that denotes that a MIB Object Identifier is exported in the (Options) Template Record.

3. Example Use Cases

3.1 Detailing CPU Load History

The CPU Usage of a remote network device could be monitored by configuring it to periodically send CPU usage information to a centralized Collector. In this example, the Exporter would send an IPFIX Message every 30 minutes that contained Data Records detailing the CPU 1 minute busy average at 1 minute intervals.

The table of data that is to be exported would look like:

=====	=====	=====
Timestamp		CPU BUSY PERCENTAGE
StartTime + 0 seconds		10%
StartTime + 60 seconds		14%
StartTime + 120 seconds		19%
StartTime + 180 seconds		16%
StartTime + 240 seconds		23%
StartTime + 300 seconds		29%
...		...

The Template Record for such a Data Record will detail two Information Elements:

```

flowStartSeconds from [RFC5102]:
  Value:          IE = 150
  Description:    The absolute timestamp of the first packet of
                  this Flow.
cpmCPUTotallminRev from the proprietary CISCO-PROCESS-MIB
  Value:          MIB OID = "1.3.6.1.4.1.9.9.109.1.1.1.1.7"
  Description:    The overall CPU busy percentage in the last
                  one-minute period

```

3.2 Output Interface Queue Size in PSAMP Packet Report

If a PSAMP Packet Report [RFC5476] was generated on any dropped packets on an interface then it may be desirable to know if the send queue on the output interface was full. This could be done by sending the size of the send queue in the same Data Record as the PSAMP Packet Report.

The exported data could look something like:

SRC ADDR	DST ADDR	PAK LEN	OUTPUT I/F	OUTPUT Q. SIZE
=====	=====	=====	=====	=====
192.0.2.1	192.0.2.3	150	Eth 1/0	45
192.0.2.4	192.0.2.9	350	Eth 1/0	45
192.0.2.3	192.0.2.9	650	Eth 1/0	23
192.0.2.4	192.0.2.6	350	Eth 1/1	0

The MIB object for the Output Queue Length, ifOutQLen ("1.3.6.1.2.1.2.2.1.21"), is indexed by the ifIndex interface index as detailed in the IF-MIB [RFC2863]. If, for example, the interface index of "Eth 1/0" in our example is 15, the full MIB Object Identifier for the Output Queue Length would be

Internet-Draft <Exporting MIB variable with IPFIX> Oct 2010
"1.3.6.1.2.1.2.2.1.21.15". With the specification in this document, each time a different MIB OID is specified in the Template Record, a new MIB object must be identified, hence a new Template Record. Rather than send a separate Template Record for each Interface Index, it would be much more convenient to identify the index in the Data Record itself.

In fact, only how the indexed object was indexed is important. In our example we identify the Egress Interface, although for other uses it may be sufficient to know that the Output Queue Size was taken from the interface that the packet was switched out of without identifying the actual interface.

The Template Record for our example Data Record would contain the following Information Elements:

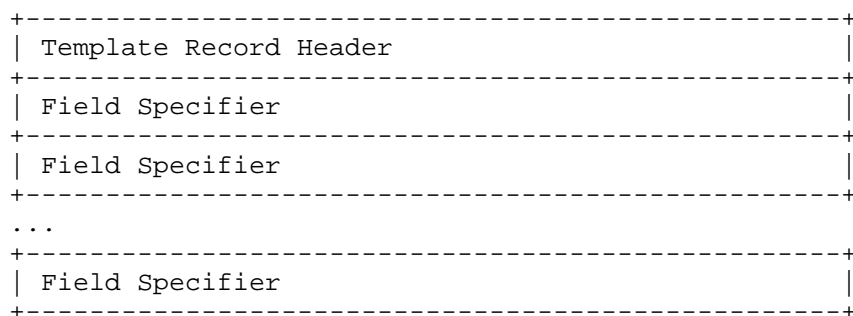
```
sourceIPv4Address
destinationIPv4Address
totalLengthIPv4
egressInterface
outboundQueueLength indexed by: egressInterface
```

4. MIB OID Extended Template Formats

Extended Template Record Formats are required to send data defined by MIB Object Identifiers. New Template Sets are required for these extended Template Record Formats.

4.1 MIB OID Extended Template Record Format

The format of the MIB Object Identifier Extended Template Record is shown in Figure A. It consists of a Template Record Header and one or more Field Specifiers.



A MIB Object Identifier Extended Template Record MUST contain at least one MIB Object Identifier Extended Field Specifier. It MAY also contain any combination of IANA-assigned and/or Enterprise-Specific Information Element identifiers as specified in [RFC5101].

The format of the Template Record Header is shown in Figure B.

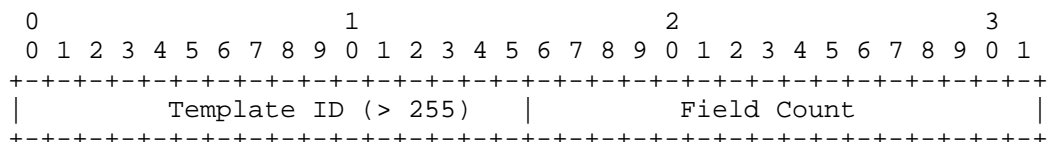


Figure B: Template Record Header Format

Where:

Template ID

Template ID of this Template Record. This value is greater than 255.

Field Count

Number of all fields in this Template Record, including the Scope Fields.

At this level of detail the layout of the Template Record Format, as specified in [RFC5101], and the MIB Object Identifier Extended Template Record Format are identical. It is only the structure of the Field Specifiers that is different (see section 4.3).

4.2 MIB OID Extended Options Template Record Format

The format of the MIB Object Identifier Extended Options Template Record is shown in Figure C. It consists of an Options Template Record Header and one or more Field Specifiers.

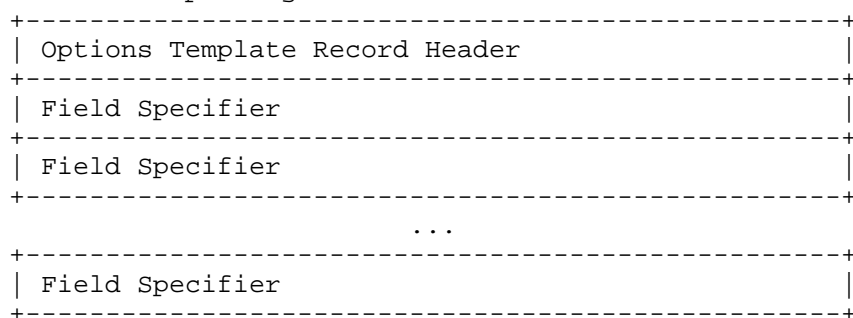


Figure C: MIB Object Identifier Options Extended Template Record Format

A MIB Object Identifier Extended Options Template Record MUST contain at least one MIB Object Identifier Extended Field Specifier, which MAY be a scope field. It MAY also contain any combination of IANA-assigned and/or Enterprise-Specific Information Element identifiers.

The format of the Options Template Record Header is shown in Figure D.

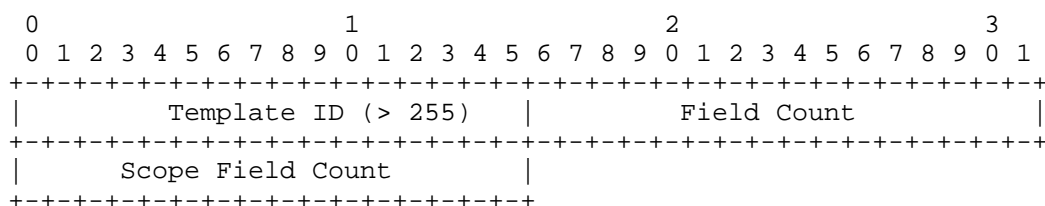


Figure D: Options Template Record Header Format

Where:

Template ID

Template ID of this Options Template Record. This value is greater than 255.

Field Count

Internet-Draft <Exporting MIB variable with IPFIX> Oct 2010
 Number of all fields in this Options Template Record,
 including the Scope Fields.

Scope Field Count

Number of scope fields in this Options Template Record.
 The Scope Fields are normal Fields except that they are
 interpreted as Scope at the Collector. The Scope Field
 Count MUST NOT be zero for an Options Template Record.

As with the Template Record Format, the only difference between
 the standard Options Template Record Format as defined in
 [RFC5101] and the MIB Object Identifier Extended Template
 Options Record Format is the structure of the Field Specifier
 (see section 4.3).

4.3 MIB OID Extended Field Specifier Format

This section specifies how the Field Specifier format in
 [RFC5101] is extended to allow fields to be defined using a
 specified MIB Object. First for a MIB Object Identifier that is
 a non-indexed MIB object, then for an indexed MIB object.

The Field Specifier formats are shown in Figures E to G below.

4.3.1 Standard Field Specifier Format

The Field Specifier format in figure E, along with the
 associated definitions, has been copied from [RFC5101], for an
 easier comparison with the MIB Object Identifier Extended Field
 Specifier Format in figures F and G.

When sending an IANA-assigned and/or Enterprise-Specific
 Information Element identifier, the Field Specifier Format is
 the same as shown below.

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
E Information Element ident.										Field Length																													
										Enterprise Number																													

E

Information Element identifier

Field Length

Enterprise Number

4.3.2 Extended Field Specifier Format for non-indexed MIB Object

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
+--+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+--+									
E										MIB OID IE										Field Length																			

```

Internet-Draft  <Exporting MIB variable with IPFIX>   Oct 2010
+-----+
|Index Count = 0|MIB Obj. ID Len|      MIB Object Identifier ... |
+-----+
|      ... MIB Object Identifier continued      |
+-----+

```

Figure F: MIB Object Identifier Extended Field Specifier Format
for a non-indexed MIB Object with an OID length < 255

Where:

E

Enterprise bit. In the special case of a MIB Object Identifier export, the Enterprise bit MUST always be 0, even if the exported MIB object is specified in a proprietary MIB, therefore containing the private enterprise number in its OID.

MIB OID IE

Special IPFIX Information Element, MIBObjectIdentifierMarker, that denotes that a MIB object is exported in the (Options) Template Record. When the MIB Object Identifier Information Element (MIB OID IE) is used, the MIB Object Identifier must be specified in the Field Specifier for the Collecting Process to be able to decode the Records.

Field Length

The definition is as [RFC5101].

Index Count

The number of indexes for a MIB object, and zero for a non-indexed MIB object.

MIB Object Identifier Length

The length of the MIB Object Identifier that follows. This is encoded in the same manner as the variable length encoding in [RFC5101]. If the length of the MIB Object Identifier is greater than or equal to 255 octets, the length is encoded into 3 octets before the MIB Object Name.

Internet-Draft <Exporting MIB variable with IPFIX> Oct 2010
The first octet is 255 and the length is carried in the
second and third octets (as shown in Figure H).

MIB Object Identifier

An alphanumeric-format variable name which denotes a
variable name expressed as a sequence of decimal numbers or
names separated by periods, as specified by the OBJECT
IDENTIFIER in [RFC2578].

If the MIB Object Identifier is longer than 254 characters then
the length MUST be extended:

0										1										2										3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																		
E										MIB OID IE																				Field Length																			
Index Count = 0										255																				MIB Object Identifier Length																			
										MIB Object Identifier ...																																							

Figure G: MIB Object Identifier Extended Field Specifier Format
with OID length >= 255

Figure H shows the exported Template Set detailing the Template
Record for exporting CPU Load (see section 3.1).

0										1										2										3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																		
										Set ID = TBD1																				Length = 47																			
										Template ID = 256																				Field Count = 2																			
E										IE = flowStartSeconds																				Field Length = 4																			
0										MIBObjectIdentifierMarker																				Field Length 1																			
Index Count = 0										MIB OID Len=29																				MIB Object Identifier ...																			
										= "1.3.6.1.4.1.9.9.109.1.1.1.1.7"																																							

```

Internet-Draft  <Exporting MIB variable with IPFIX>   Oct 2010
+-----+
| ... MIB Object Identifier continued ... |
+-----+
| ... MIB Object Identifier continued ... |
+-----+
| ... MIB Object Identifier continued ... |
+-----+
| ... MIB Object Identifier continued ... |
+-----+
| ... MIB Object Identifier continued ... |
+-----+
| MIB Object Identifier continued |
+-----+

```

Figure H: Example of CPU Load Template Set

4.3.3 Extended Field Specifier Format for Indexed MIB Object

When an indexed MIB object is exported in IPFIX, the meaning of the exported value each index SHOULD be identified. This index (or indexes) MAY be an IPFIX Information Element or MIB Object Identifier.. Note that the IPFIX Information Element MAY be an enterprise-specific Information Element.

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+
|E|          MIB OID IE          |          Field Length          |
+-----+
| Index Count |MIB Obj. ID Len|          MIB Object Identifier ... |
+-----+
| ... MIB Object Identifier continued |
+-----+
|E| Index Information Element 1 |E| Index Information Element 2 |
+-----+
|          Enterprise Number          |
+-----+
...

```

Figure I: MIB Object Identifier Extended Field Specifier Format with an indexed MIB Object using a IPFIX I.E. as Index

Where:

E

Enterprise bit. In the special case of a MIB Object Identifier export, the Enterprise bit MUST always be 0, even if the exported MIB object is specified in a proprietary MIB, therefore containing the private enterprise number in its OID. For any indexes identified using Information Elements the Enterprise bit can be 1, indicating that an Enterprise Number will follow the Information Element.

MIB OID IE

Special IPFIX Information Element, MIBObjectIdentifierMarker, that denotes that a MIB object is exported in the (Options) Template Record. When the MIB Object Identifier Information Element (MIB OID IE) is used, the MIB Object Identifier must be specified in the Field Specifier for the Collecting Process to be able to decode the Records.

Field Length

The definition is as [RFC5101].

Index Count

The number of indexes for a MIB object, and zero for a non-indexed MIB object.

MIB Object Identifier Length

The length of the MIB Object Identifier that follows. This is encoded in the same manner as the variable length encoding in [RFC5101]. If the length of the MIB Object Identifier is greater than or equal to 255 octets, the length is encoded into 3 octets before the MIB Object Name. The first octet is 255 and the length is carried in the second and third octets (as shown in Figure H).

MIB Object Identifier

An alphanumeric-format variable name which denotes a variable name expressed as a sequence of decimal numbers or names separated by periods, as specified by the OBJECT IDENTIFIER in [RFC2578].

A MIB Object Identifier MAY be used as an index and sent as described in Figure J. If a MIB Object Identifier with an index is used as an index then its indexes will no be identified.

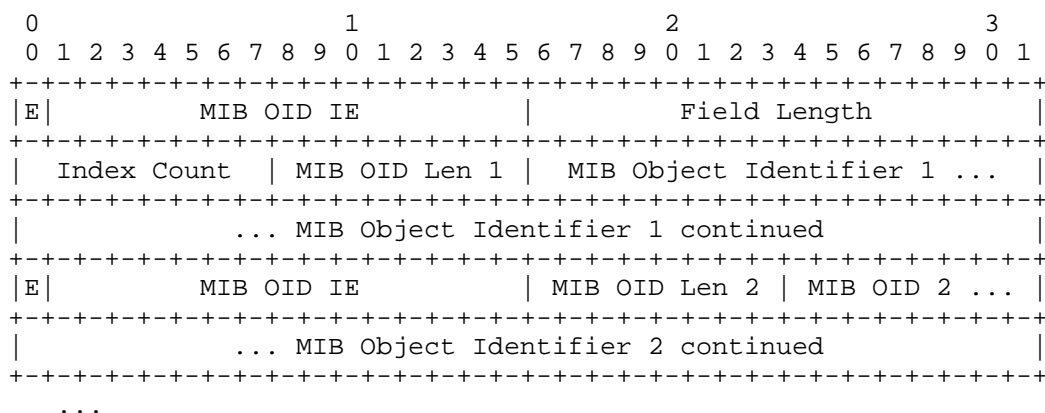


Figure J: MIB Object Identifier Extended Field Specifier Format with a MIB Index using a normal MIB Object Identifier as index

Where:

E

Enterprise bit. In the special case of a MIB Object Identifier export, the Enterprise bit MUST always be 0, even if the exported MIB object is specified in a proprietary MIB, therefore containing the private enterprise number in its OID.

MIB OID IE

Special IPFIX Information Element, MIBObjectIdentifierMarker, that denotes that a MIB object is exported in the (Options) Template Record. When the MIB Object Identifier Information Element (MIB OID IE) is used, the MIB Object Identifier must be specified in the Field Specifier for the Collecting Process to be able to decode the Records.

Field Length

Index Count

The number of indexes for a MIB object, and zero for a non-indexed MIB object.

MIB Object Identifier Length 1

The length of the MIB Object Identifier being exported. This is encoded in the same manner as the variable length encoding in [RFC5101]. If the length of the MIB Object Identifier is greater than or equal to 255 octets, the length is encoded into 3 octets before the MIB Object Name. The first octet is 255 and the length is carried in the second and third octets.

MIB Object Identifier 1

An alphanumeric-format variable name of the MIB Object Identifier being exported, which denotes a variable name expressed as a sequence of decimal numbers or names separated by periods, as specified by the OBJECT IDENTIFIER in [RFC2578].

MIB Object Identifier Length 2

The length of the MIB Object Identifier being used as an index. This is encoded in the same manner as the variable length encoding in [RFC5101]. If the length of the MIB Object Identifier is greater than or equal to 255 octets, the length is encoded into 3 octets before the MIB Object Name. The first octet is 255 and the length is carried in the second and third octets.

MIB Object Identifier 1

An alphanumeric-format variable name of the MIB Object Identifier being used as an index, which denotes a variable name expressed as a sequence of decimal numbers or names separated by periods, as specified by the OBJECT IDENTIFIER in [RFC2578].

Figure K shows the exported Template Set detailing the Template for exporting a PSAMP Report with Output Queue Size (see section 3.2).

```

Internet-Draft  <Exporting MIB variable with IPFIX>      Oct 2010
      0              1              2              3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Set ID = TBD1          |          Length          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Template ID = 257      |          Field Count = 5  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|0|    IE = sourceIPv4Address      |          Field Length = 4  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|0|    IE = destinationIPv4Address |          Field Length = 4  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|0|    IE = totalLengthIPv4        |          Field Length = 4  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|0|    IE = egressInterface        |          Field Length = 4  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|0|    MIBObjectIdentifierMark     |          Field Length 1    |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Index Count=1 |MIB OID Len=20 |    MIB Object Identifier ... |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          = "1.3.6.1.2.1.2.2.1.21"          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          ... MIB Object Identifier continued ...          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          ... MIB Object Identifier continued ...          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          ... MIB Object Identifier continued ...          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... MIB OID continued          |0|    IE = egressInterface  |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure K: Example of PSAMP Report with Output Queue Size

Figure L shows the exported Template Set detailing the Template for exporting a PSAMP Report with Output Queue Size but using the ifIndex MIB object as the exported index, rather than the Egress Interface Information Element.

```

      0              1              2              3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Set ID = TBD1          |          Length          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Template ID = 257      |          Field Count = 5  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|0|    IE = sourceIPv4Address      |          Field Length = 4  |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

Internet-Draft  <Exporting MIB variable with IPFIX>   Oct 2010
+-----+
|0| IE = destinationIPv4Address |           Field Length = 4           |
+-----+
|0|   IE = totalLengthIPv4      |           Field Length = 4           |
+-----+
|0|   IE = egressInterface      |           Field Length = 4           |
+-----+
|0|IE=MIBObjectIdentifierMarker |           Field Length 1           |
+-----+
| Index Count=1 |MIB OID Len=20 |       MIB Object Identifier ... |
+-----+
|                                     = "1.3.6.1.2.1.2.2.1.21"          |
+-----+
|                                     ... MIB Object Identifier continued ... |
+-----+
|                                     ... MIB Object Identifier continued ... |
+-----+
|                                     ... MIB Object Identifier continued ... |
+-----+
| ... MIB OID continued              |0|IE=MIBObjectIdentifierMarker |
+-----+
|       MIB Object Identifier = "1.3.6.1.2.1.2.2.1.1"          |
+-----+
|       ... MIB Object Identifier continued ... |
+-----+
|       ... MIB Object Identifier continued ... |
+-----+
|       ... MIB Object Identifier continued ... |
+-----+
|       ... MIB Object Identifier continued      |
+-----+

```

Figure L: Example of PSAMP Report with Output Queue Size using ifIndex from IF-MIB as an index [RFC2578]

4.4 Template Management

Templates are managed as per [RFC5101].

The Set ID field MUST contain the value TBD1 for any Template Set that contains a MIB Object Identifier Extended Field Specifier. The Template Withdrawal Message for such a Template must also use a Set ID field containing the value TBD1.

The Set ID field MUST contain the value TBD2 for any Option Template Set that contains a MIB Object Identifier Extended

Internet-Draft <Exporting MIB variable with IPFIX> Oct 2010
Field Specifier. The Template Withdrawal Message for such an
Option Template must also use a Set ID field containing the
value TBD2.

5. The Collecting Process's Side

This section describes the Collecting Process when using SCTP and PR-SCTP as the transport protocol. Any necessary changes to the Collecting Process specifically related to TCP or UDP transport protocols are specified in section 10 of RFC 5101.

The specifications in section 9 of RFC 5101 also apply to Collector's that implement this specification. In addition, the following specifications should be noted.

A Collecting Process that implements this specification MUST be able to receive Set IDs TBD1 and TBD2, as specified in this document.

A Collecting Process that implements this specification MUST have access to a MIB database in order to look up the received MIB Object IDs and find the type and name of MIB OID fields used in received templates. It should be noted that since reduced length encoding MAY be used by the Exporting Process then the Collecting Process cannot assume a received size for a field is the maximum size it should expect for that field.

If a Collecting Process receives a MIB Object ID that it cannot decode, it SHOULD log an error.

If a Collecting Process receives a MIB Object ID for an indexed MIB Object but isn't sent the appropriate number of indexes then it SHOULD log an error, but it MAY use the Template Record to decode the Data Records as the associated indexes are purely semantic information.

6. Applicability

Making available the many and varied items from the MIBs opens up a wide range of possible applications for the IPFIX protocol, some quite different from the usual flow information. Some potential enhancements for traditional applications are detailed below:

Some monitoring applications periodically export an interface id to interface name mapping using IPFIX Options Templates. This could be expanded to include the MIB Object "1.3.6.1.2.1.2.ifTable.ifEntry.ifInUcastPkts" indexed using the ingressInterface Information Element, as a index. This would give the input statistics for each interface which can be compared to the flow information to ensure the sampling rate is expected. Or, if there is no sampling, to ensure that all the expected packets are being monitored.

7. Security Considerations

For this extension to the IPFIX protocol, the same security considerations as for the IPFIX protocol apply [RFC5101].

However, the Metering Process MUST check whether or not the MIB variables can be accessed, and hence exported with IPFIX. Therefore a read or read-write community string in SNMPv1 and SNMPv2c, or a principal in SNMPv3, MUST be associated with the Metering Process.

If the management entity supports the View-based Access Control Model (VACM) for the SNMP [RFC3415], then the Metering Process MUST validate with the View-Based Access Control [RFC3415] that the MIB object can accessed before exporting his content.

If there is a view in case of SNMPv1 and SNMPv2c, the Metering Process MUST validate that the MIB object can accessed before exporting his content.

Whether the Exporter allows or not the configuration of Template that contains an unauthorized MIB object is implementation specific.

8. IANA Considerations

IPFIX Messages use two fields with assigned values. These are the IPFIX Version Number, indicating which version of the IPFIX Protocol was used to export an IPFIX Message, and the IPFIX Set ID, indicating the type for each set of information within an IPFIX Message.

The previously reserved Set ID values of TBD1 and TBD2 are used as specified in this document. All other Set ID values are reserved for future use. Set ID values above 255 are used for Data Sets.

Internet-Draft <Exporting MIB variable with IPFIX> Oct 2010
A new Information Element, "MIBObjectIdentifierMarker", needs to
be reserved.

9. References

9.1 Normative References

- [RFC2119] S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, BCP 14, RFC 2119, March 1997
- [RFC2578] McCloghrie, K. Perkins, D., Schoenwaelder, J., Structure of Management Information Version 2 (SMIV2), RFC 2578, April 1999
- [RFC2863] McCloghrie, K., Kastenholz, F., Interfaces Group MIB, RFC 2863, June 2000
- [RFC3415] Wijnen, B., Presuhn, R., and K. McCloghrie, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), RFC3415, December 2002
- [RFC3418] Presuhn, R., Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), RFC3418, December 2002
- [RFC5101] B. Claise et Al, IPFIX Protocol Specification, RFC 5101, January 2008
- [RFC5102] J. Quittek, S. Bryant, B. Claise, J. Meyer, Information Model for IP Flow Information Export, RFC5102, January 2008
- [PEN] IANA Private Enterprise Numbers registry
<http://www.iana.org/assignments/enterprise-numbers>.

9.2 Informative References

Internet-Draft <Exporting MIB variable with IPFIX> Oct 2010
[RFC5476] Claise, B., Johnson, A, and J. Quittek,
Packet Sampling (PSAMP) Protocol Specifications,
RFC 5476, Mars 2009

Author's Addresses

Andrew Johnson
Cisco Systems (Scotland) Ltd.
96 Commercial Quay
Commercial Street
Edinburgh, EH6 6LX, United Kingdom
Phone: +44 131 561 3641
Email: andrjohn@cisco.com

Benoit Claise
Cisco Systems
De Kleetlaan 6a b1
Diegem 1813
Belgium
Phone: +32 2 704 5622
Email: bclaise@cisco.com

Paul Aitken
Cisco Systems (Scotland) Ltd.
96 Commercial Quay
Commercial Street
Edinburgh, EH6 6LX, United Kingdom
Phone: +44 131 561 3616
Email: paitken@cisco.com