IPFIX Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2012

B. Claise
P. Aitken
S. B S
Cisco Systems, Inc.
J. Schoenwaelder
Jacobs University Bremen

March 12, 2012

Exporting MIB Variables using the IPFIX Protocol
draft-johnson-ipfix-mib-variable-export-04

Abstract

This document specifies a way to complement IPFIX Flow Records with Management Base (MIB) objects, 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 along with IPFIX Information Elements.

Status of this Memo

Copyright Notice

Table of Contents

1. Open Issues / To do list

- Skipping the length. Is a new Set ID the right solution?

- "timestamps, exporters, and other animals" -> see the mailing list.

- Question: index is an IPFIX IE that didn't appear the flow record? Do we preclude this case?

- The value of the MIB OID acting as an index may not be of fixed length and may have no default length, for example the OID can be of type string or type MIB OID.

- "we can use the IE as an index if there is one and only one similar with that length in the Template Records". To be discussed.

- use case: no index count and no index OID in the SNMP agent -> add this with the solution discussed with the DCM2.0 team.

- This also allows reduced size encoding for the indices.

- some TODO in the XML version:

  - write section: "Indexed MIB Objects with a mix of MIB OID and IPFIX Information Element"

  - insert example: "Using MIB Objects with IPFIX Structured Data"

- Describe how to choose between multiple instances of the required index field (eg, when the index is the egress interface for multicast). eg, rather than specifying the index IE by ID, we could specify it by number: the n'th field in the record.

- IPFIX Structured Data: how should it work? Add example to "sectionStructuredData".

- How does the example in 5.5 work (ifOutQLen indexed by: ifIndex) since ifIndex is not present in the record?

- How does the example in 5.8.2 work, since the ifName is indexed by ifIndex which comes after - so the value is not already known.

- Improve the examples: Add an example with the mix of IPFIX IE and OID in sectionUseIndexedwithaMixofOIDAndIPFIXIE.

- RFC 5610: explain what needs to be updated.

- ID to name mappings? -> use this for an example in section 5.

- What does this mean? : "(Consider the counter synchronization issue, non-key info should be static)".

- Tidy up the XML.

- (JS) Do we need to add something about the contextEngineID and contextName? Optionally associate context with template via options Could be done with common properties or in a flow record However, do we limit all MIB variables in a Template Record to a single context? 3 cases:

  1. if a simple SNMP agent, no contextEngineID and contextName, because it's the default

  2. the context information is valid for the entire flow record

  3. the context information is specific for each IE within the entire flow record

  question regarding 3.: only one context for an entire flow or can a flow record export MIB OID from different context? (JS): ask the IPFIX mailing list. (BC): ask internally in Cisco Action: complete the "Identifying the SNMP Context" section

- (JS) Inacio's figure: send email to the mailing list.

2. Introduction

There is growing interest in using IPFIX as a push mechanism for exporting management information. Using a push protocol such as IPFIX instead of a polling protocol like SNMP is especially interesting in situations, where large chunks of repetitive data need to be exported periodically.

While initially targeted at different problems, there is a large parallel between the information transported via IPFIX and SNMP. Furthermore, certain Management Information Base (MIB) objects are highly relevant to flows as they are understood today. For example, in the IPFIX information model [RFC5102], Information Elements coming from the SNMP world have already been specified, e.g., ingressInterface and egressInterface both refer to the ifIndex defined in [RFC2863].

Rather than mapping existing MIB objects to IPFIX Information Elements on a case by case basis, it would be advantageous to enable the export of any existing or future MIB objects as part of an IPFIX Flow Record. This way, the duplication of data models [RFC3444], both as SMI MIB objects and IPFIX Information Elements, out of the same information model [RFC3444] would be avoided.

In this document, new Template Sets for Flow Records and Options Records are specified to allow Templates to contain any combination of fields defined by traditional IPFIX Information Element(s) and/or MIB Object Identifier(s). The MIB Object Identifiers can reference either non-indexed or indexed MIB object(s). Note that the enterprise-specific MIB Object Identifiers are also supported.

When an indexed MIB object is exported, a method to identify how that MIB object is indexed is specified so that the full meaning of the information being exported can be conveyed. The specifications encompasses the different index types for the MIB Objects Identifier: indexed by one or multiple MIB variable(s), indexed by one or multiple IPFIX Information Element(s), indexed by a mix of MIB variable(s) and IPFIX Information Element(s). A set of example use cases is used to illustrate how these specifications can be used.

Some Exporters may not have the knowledge to convey the full information on how the MIB objects being exported are indexed. They may not know the index count and/or the OID's of the objects that are used to index a MIB object. In such cases the Exporter can send the the values of the index OID's identifying the instance of the object being exported as one string that conveys the instance identifier part of an object being exported. The Collecting Process may know how a MIB object is indexed by some other means, for example, it could compile this information from the MIB Module that defines exported MIB object or the Collecting Process could be hardcoded with this information for a pre-defined set of MIB objects that it is interested in. An example use case is used to illustrate this mechanism.

3. Motivation and Architectural Model

Most Flow Records contain the ingressInterface and/or the egressInterface Information Element. These Information Elements carry an ifIndex value, a MIB object defined in [RFC2863]. In order to retrieve additional information about the identified interface, a Collector could simply poll relevant objects from the device running the Exporter via SNMP, however, that approach has several problems:

- It requires implementing a mediation function between two data models, i.e., MIB objects and IPFIX Information Elements.

- Confirming the validity of simple mappings (e.g., ifIndex to ifName) requires to either check on a regular basis that the Exporter's network management system did not reload, or to impose ifIndex persistence across an Exporter's reload.

- Synchronization problems occur since counters carried in Flow Records and counters carried in SNMP messages are retrieved from the Exporter at different points in time and thus can't be correlated. In the best case, assuming very tight integration of an IPFIX Collector with and SNMP polling engine, SNMP data is retrieved shortly after Data Records have been received, which implies the sum of the active or inactive timeouts (if not null) plus the time to export the Flow Record to the Collector. If, however, the SNMP data is retrieved by a generic Network Management Station (NMS) polling interface statistics, then the time lag between IPFIX counters and SNMP counters can be significant.

The intended scope of this work is the addition of MIB variable(s) to IPFIX Information Elements in Flow Records, in order to complement the Flow Records with useful and already standardized information. More specifically, the case of an existing Template Record, which needed to be augmented with some MIB variables whose index was already present in the Template Record as an IPFIX Information Element: typically, a 7-tuple Flow Record containing the ingressInterface Information Element, augmented by interface counters [RFC2863], which are indexed by the respective ingressInterface values in the Flow Records.

The intended goal of this work is not a replacement of SNMP notifications, even if the specifications in this document could potentially allow this. Since IPFIX is a push mechanism, initiated from the Exporter with no acknowledgment method, this specification does not provide the ability to execute configuration changes.

The Distributed Management Expression MIB [RFC2982], which is a mechanism to create new MIB variables based on the content of existing ones, could also be advantageous in this context of this specification. Indeed, newly created MIB object (for example, the link utilization MIB variable), created with the Distributed Management Expression MIB [RFC2982] could nicely complement Flow Records.

Another advantage of exporting MIB objects via IPFIX is that IPFIX would benefit from an extended series of types to be exported. 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.

```
                 +------+  +-------+  +.........+  +.....+
                 | SNMP |  | IPFIX |  : NETCONF :  : CLI :
                 +------+  +-------+  +.........+  +.....+
                     |         |           |          |
               +--------------------------------------------+
               | Instrumentation (specified in MIB modules) |
               +--------------------------------------------+
```

Figure 1: Architectural Model

The overall architectural model is depicted in Figure 1. The IPFIX Exporter accesses the device's instrumentation, which follows the specifications contained in MIB modules. Other management interfaces such as NETCONF or the device's Command Line Interface (CLI) may provide access to the same instrumentation.

4. Terminology

IPFIX-specific terminology (Information Element, Template, Template Record, Options Template Record, Template Set, Collector, Exporter, Flow Record, etc.) used in this document is defined in Section 2 of [RFC5101]. 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)

An ASCII character sequences of decimal non-negative sub-identifier values. Each sub-identifier value MUST NOT exceed 2^32-1 (4294967295) and MUST NOT have leading zeros. Sub-identifiers are separated by single dots and without any intermediate whitespace.

MIB Object Identifier Information Element

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

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

5. MIB OID Extended Template Formats

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

5.1. MIB OID Extended Template Record Format

The format of the MIB Object Identifier Extended Template Record is shown in Figure 2. It consists of a Template Record Header and one or more Field Specifiers.

```
        +--------------------------------------------------+
        | Template Record Header                           |
        +--------------------------------------------------+
        | Field Specifier                                  |
        +--------------------------------------------------+
        | Field Specifier                                  |
        +--------------------------------------------------+
                             ...
        +--------------------------------------------------+
        | Field Specifier                                  |
        +--------------------------------------------------+
```

Figure 2: MIB Object Identifier Extended Template Record Format

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 3.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         Template ID (> 255)   |         Field Count           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: 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.

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 5.3).

5.2. MIB OID Extended Options Template Record Format

The format of the MIB Object Identifier Extended Options Template Record is shown in Figure 4. It consists of an Options Template Record Header and one or more Field Specifiers.

```
        +-----------------------------------------------------+
        | Options Template Record Header                      |
        +-----------------------------------------------------+
        | Field Specifier                                     |
        +-----------------------------------------------------+
        | Field Specifier                                     |
        +-----------------------------------------------------+
                                  ...
        +-----------------------------------------------------+
        | Field Specifier                                     |
        +-----------------------------------------------------+
```

Figure 4: 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 5.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Template ID (> 255)   |         Field Count           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Scope Field Count        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Options 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.

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 Specifiers (see Section 5.3).

Both indexed and non-indexed MIB Objects may be used as scope fields in an IPFIX Options Template Record. Each scope MIB object is included in the IPFIX Scope Field Count. When indexed MIB Objects are used, the index information is not included in the Scope Field Count since the size of the index information is already specified in the MIB Object's "index count" field (see Section 5.3.3). Examples are given in Section 6.9.

5.3. 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 Figure 6 to Figure 9 below.

5.3.1. Standard Field Specifier Format

The Field Specifier format in Figure 6, along with the associated definitions, has been copied from [RFC5101], for an easier comparison with the MIB Object Identifier Extended Field Specifier Format in Figure 7 through Figure 9.

When exporting an IANA-assigned and/or enterprise-specific IPFIX 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                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Standard Field Specifier format

Where:

E

Enterprise bit. This is the first bit of the Field Specifier. If this bit is zero, the Information Element Identifier identifies an IETF specified Information Element, and the four octet Enterprise Number field MUST NOT be present. If this bit is one, the Information Element identifier identifies an enterprise-specific Information Element, and the Enterprise Number filed MUST be present.

Information Element identifier

A numeric value that represents the type of the Information Element. Refer to [RFC5102].

Field Length

The length of the corresponding encoded Information Element, in octets. Refer to [RFC5102]. The field length may be smaller than the definition in [RFC5102] if reduced size encoding is used. The value 65535 is reserved for variable length Information Element.

Enterprise Number

IANA enterprise number [PEN] of the authority defining the Information Element identifier in this Template Record.

5.3.2. Extended Field Specifier Format for a non-indexed MIB Object

When a MIB object is to be exported, a special Information Element value is used to show that the extended Field Specifier is being used, as shown in Figure 7:

```
    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|  MIB OID Len  |    MIB Object Identifier ...  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              ... MIB Object Identifier continued              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: MIB Object Identifier Extended Field Specifier Format for a non-indexed MIB Object with an OID length < 255

Where:

E

Enterprise bit. This is the first bit of the Field Specifier. The value is always set to 0 for the MIB Object Identifier Extended Field Specifier Format, even if the MIB Object Identifier is enterprise-specific, because the MIB OID IE is an IANA standard field and is not enterprise-specific.

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 MIB Object Identifier Extended Field Specifier for the Collecting Process to be able to decode the Records.

Field Length

The definition is as [RFC5101]. Note that the Field Length can be expressed using reduced size encoding per [RFC5101].

Index Count

The number of indices for a MIB object. Set to zero for a non-indexed MIB object.

MIB Object Identifier Length

The length of the textual representation 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, where the first octet is 255 and the length is carried in the second and third octets as shown in Figure 8. If the MIB Object Identifier is longer than 254 characters then the length MUST be extended.

MIB Object Identifier

The textual representation of a MIB object identifier as defined in Section 4.

```
    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 ...                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              ... MIB Object Identifier continued              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: MIB Object Identifier Extended Field Specifier Format for a non-indexed MIB Object with an OID length >= 255

5.3.3. Extended Field Specifier Format for an Indexed MIB Object, with an MIB OID as Index

The mechanism for "Extended Field Specifier Format for non-indexed MIB Object" in Section 5.3.2 can be used for exporting any MIB objects, including indexed MIB objects. However, per the nature of indexing in MIB module, every indexed object is specified by a new MIB Object Identifier, which in turn implies that a new Template Record must be used for every indexed object. For example, the ifInOctets for the interface represented by the interface ifIndex 1

is ifInOctets.1, the ifInOctets for the interface represented by the interface ifIndex 2 is ifInOctets.2, ... This makes the export mechanism for "Extended Field Specifier Format for non-indexed MIB Object" inefficient when used for indexed MIB objects. An example is shown in Section 6.1.

When an indexed MIB object is exported in IPFIX, either the meaning of the exported value of each index may be identified or the complete OID segment identifying the instance can be sent as one piece. When the meaning of each index is identified, this index (or indices) MUST be a MIB Object Identifier (this section) or an IPFIX Information Element (see Section 5.3.4).

A MIB Object Identifier MAY be used as an index and sent as described in Figure 9. However, if a MIB Object Identifier with an index is used as an index then its indices will not be identified.

```
      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 OID Len  |   MIB Object Identifier ...   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |              ... MIB Object Identifier continued              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |E|         MIB OID IE          |      Index Field length       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |Idx MIB OID Len|        Index MIB Object Identifier ...        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |         ... Index MIB Object Identifier continued ...         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                 ...
```

Figure 9: MIB Object Identifier Extended Field Specifier Format with a MIB Index using a normal MIB Object Identifier as index

Where:

E

Enterprise bit. This is the first bit of the Field Specifier. The value is always set to 0 for the MIB Object Identifier Extended Field Specifier Format, even if the MIB Object Identifier is enterprise-specific, because the MIB OID IE is an IANA standard field and is not enterprise-specific.

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 MIB Object Identifier Extended Field Specifier for the Collecting Process to be able to decode the Records.

Field Length

The definition is as [RFC5101]. Note that the Field Length can be expressed using reduced size encoding per [RFC5101].

Index Count

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

MIB Object Identifier Length

The length of the textual representation 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 Where the first octet is 255 and the length is carried in the second and third octets (as shown in Figure 8). If the MIB Object Identifier is longer than 254 characters then the length MUST be extended.

MIB Object Identifier

The textual representation of a MIB object identifier as defined in Section 4. For any indices identified using Information Elements the Enterprise bit can be 1, indicating that an Enterprise Number will follow the Information Element.

Index Field Length

The length of the encoded index field, in octets, per the Field Length definition in [RFC5101]. Note that the Index Field Length can be expressed using reduced size encoding per [RFC5101].

Index MIB Object Identifier Length

The length of the textual representation 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.

Index MIB Object Identifier

The textual representation of a MIB object identifier as defined in Section 4.

5.3.4. Extended Field Specifier Format for an Indexed MIB Object, with an IPFIX Information Element as Index

A possible optimization for the Extended Field Specifier Format for an Indexed MIB Object as specified in Section 5.3.3 is to use an existing IPFIX Information Element, which is already present in the Flow definition, as the index for indexed MIB Object. On the top not repeating the index, the primary advantage is to make a clear link between the Flow Record values and the MIB variable index.

For example, if a Flow Record definition contains the source IP address, the destination IP address, and the ingressInterface Information Element as Flow Keys, this implies that the IP address pairs are seen on that specific interface. If the ifInOctets, indexed by that specific interface, is added to the Flow Record, it's clear from the Flow Record, that the ifInOctets is related to the same interface. If the ifInOctets was indexed by the ifIndex (as specified in Section 5.3.3), the Collector would have to hardcode that the semantic of ifIndex MIB variable is equivalent to the ingressInterface Information Element.

When an indexed MIB object is exported in IPFIX, the index (or indices) MAY be an IPFIX Information Element(s). Note that this/ these IPFIX Information Element(s) MAY be an enterprise-specific Information Element.

Indexed MIB Objects, with IPFIX Information Elements as index, are exported as shown in Figure 10.

```
      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 OID Len  |    MIB Object Identifier ...  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |              ... MIB Object Identifier continued              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |E| Index Information Element 1 |       Index 1 Length          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |E| Index Information Element 2 |       Index 2 Length          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |       Index Information Element 2 Enterprise Number           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                 ...
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |E| Index Information Element N |       Index N Length          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |       Index Information Element N Enterprise Number           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: MIB Object Identifier Extended Field Specifier Format with an indexed MIB Object using an IPFIX Information Element as Index

Where:

E

Enterprise bit. This is the first bit of the Field Specifier. The value is always set to 0 for the MIB Object Identifier Extended Field Specifier Format, even if the MIB Object Identifier is enterprise-specific, because the MIB OID IE is an IANA standard field and is not enterprise-specific.

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 MIB Object Identifier Extended Field Specifier for the Collecting Process to be able to decode the Records.

Field Length

The definition is as [RFC5101]. The Field Length does not include the length of the index fields, since these are

specified separately. Note that the Field Length can be expressed using reduced size encoding per [RFC5101].

Index Count

The number of indices for a MIB object, and zero for a non-indexed MIB object. The index count MUST be consistent with the INDEX definition of the corresponding MIB module.

MIB Object Identifier Length

The length of the textual representation 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 where the first octet is 255 and the length is carried in the second and third octets (as shown in Figure 8). If the MIB Object Identifier is longer than 254 characters then the length MUST be extended.

MIB Object Identifier

The textual representation of a MIB object identifier as defined in Section 4.

Index Information Element 1..N

The Information Element(s) that are used as indices for the MIB Object Identifier.

Regular Information Elements, enterprise-specific Information Elements, and non-indexed MIB object identifiers may all be used as indices. However, indexed MIB object identifiers may not be used as indices because SNMP doesn't support hierarchical indexing.

Index 1..N Length

The respective index lengths for the Information Element(s) 1..N

5.3.5. Extended Field Specifier Format for an Indexed MIB Object, with one IPFIX Information Element for the OID segment identifying the instance

When MIB objects are to be exported, the Exporter may need to interact with the MIB instrumentation in an SNMP agent to obtain the

required information. For some SNMP agents, the MIB instrumentation by design does not have knowledge of the OID of the indice(s) that identify the instance of the MIB object being accessed. For example, when accessing a MIB object ifInOctets.10, the MIB instrumentation code may not know that the object ifInOctets is indexed by ifIndex, it is sufficient for it to map the value (10) of the ifIndex to an interface on the device. For such SNMP agents, the Exporter can not use the methods described in Section 5.3.3 and Section 5.3.4 without making extensive changes to the existing MIB instrumentation.

An alternate method for exporting Indexed MIB objects in such cases is to convey only the value(s) of the indice(s) that identify the instances being exported. The index count and OIDs of the indice(s) are not conveyed in the IPFIX template record. The Collecting Process is assumed to have the intelligence to understand how the exported objects are indexed. For example, it can either compile this information from the MIB Module where this object type is defined or it may be hardcoded with this information for specific MIB objects that are of interest to it. The object identifier of the indexed MIB object is split into two parts, first part is the OID prefix which is the OID of the corresponding object type and the second part is the OID segment identifying the instance. An information element called MIBInstanceIdentifier is defined for conveying the instance identification segment of an indexed MIB object's OID in string format. While the OID prefix is sent in the template record, the instance identifier segment is sent in the data record. Since the instance identifier segment of the MIB OID is in the data-record, the same template record can be used for exporting different instances of the same MIB object.

Indexed MIB objects, with MIBInstanceIdentifier as index are exported as shown in Figure 11

```
   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=1 |  MIB OID Len  |    MIB Object Identifier ...  |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |              ... MIB Object Identifier continued              |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |E|    MIBInstanceIdentifier    |MIBInstanceIdentifier Len=FFFF |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: MIB Object Identifier Extended Field Specifier Format with
     an indexed MIB Object using MIBInstanceIdentifier as Index

Where:

E

Enterprise bit. This is the first bit of the Field Specifier. The value is always set to 0 for the MIB Object Identifier Extended Field Specifier Format, even if the MIB Object Identifier is enterprise-specific, because the MIB OID IE is an IANA standard field and is not enterprise-specific.

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 MIB Object Identifier Extended Field Specifier for the Collecting Process to be able to decode the Records.

Field Length

The definition is as [RFC5101]. The Field Length does not include the length of the index fields, since these are specified separately. Note that the Field Length can be expressed using reduced size encoding per [RFC5101].

Index Count

When the OID segment identifying the instance is exported as one string using the MIBInstanceIdentifier the Index Count value is always set to 1 to indicate that there is one information element conveying index values for this MIB object. Since the Collecting Process is assumed to know the INDEX definition of the MIB object, the actual index count need not be conveyed.

MIB Object Identifier Length

The length of the textual representation 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 where the first octet is 255 and the length is carried in the second and third octets (as shown in Figure 8). If the MIB Object Identifier is longer than 254 characters then the length MUST be extended.

MIB Object Identifier

The textual representation of a MIB object identifier as defined in Section 4.

E

Enterprise bit. This is the enterprise bit for the MIBInstanceIdentifier that follows. The value is always set to 0 when the MIBInstanceIdentifier is used because the MIBInstanceIdentifier is an IANA standard field and is not enterprise-specific.

MIB Instance Identifier

IPFIX Information Element, MIBInstanceIdentifier, that denotes that a MIB Instance identifier string is exported in the data record following the MIB Object's value. This instance identifier when concatenated with the MIB object type OID that was sent in the template record gives the complete OID of the MIB variable that is being exported.

5.4. Indices Considerations

When using an Indexed MIB Object, the Template Record contains the index/indices length. In some cases, this index/indices information might be redundant in the export information. For example, when the index is an Information Element already contained in the Template Record, the length is already part of the Template Record, and available to the Collecting Process for decode, as shown in the example in Section 6.6. A second example in Section 6.9 is when a specific MIB OID is already part of the Template Record as a standalone MIB object in a Template Record, and also reused as an index.

However, there are two cases where the index length is required. Therefore, for consistent decoding on the Collecting Process, the Index Length is always specified next to the index.

Situation 1: When a non-indexed MIB object is used as an index, and doesn't appear as a standalone MIB object in the Template Record, the Collecting Process might not want, per design, to access the MIB modules in order to find the length of the value for a particular MIB OID.

Situation 2: A Template Record might contain two similar Information Elements with different encoding lengths even if this situation is an unlikely real-world scenario), while an Indexed MIB Object might want

to refer to one of this Information Element as the index. However, without clearly specifying the index length, the Collecting Process would not know which length to decode the index with.

When an Information Element is used as index, there MUST be one and only one similar Information Element with the exact same length in the Template Record, so that the Collecting Process knows which Information Element value from the Flow Records to match. Note that this rule also implies that the reduced size encoding [RFC5101] of the Information Element in the index compared to the Information Element in the Template Record is not allowed. If the Collecting Process can not determine clearly which Information Element value to chose as the index because there are two (or more) Information Elements with the same length, then index MUST specified as the MIB Object Identifier.

An indexed MIB object MAY be indexed by a mix of MIB OID(s) and IPFIX Information Element(s)

5.5. Identifying the SNMP Context

Each MIB OID is looked up in a specific context, usually the default context. If exporting a MIB OID value that isn't in the default context then the context string MUST be identified and associated with the MIB OID. This can be done on a per template basis by exporting an Options Template Record.

A new IPFIX Information Element, "MIBObjectIdentifierMarker" has been allocated for this purpose. See Section 11.

5.6. 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 Field Specifier. The Template Withdrawal Message for such an Option Template must also use a Set ID field containing the value TBD2.

6. Example Use Cases

6.1. Without Using the Specifications in this Document

This example shows the need for indexed MIB objects using the example of exporting ifInOctets from Section 5.3.3.

A Template Record for exporting the ifInOctets for the interface represented by the interface ifIndex 1 (i.e., ifInOctets.1) is shown in Figure 12. While this may be useful for exporting the single ifInOctets.1 field, clearly additional Templates are required in order to export ifInOctets.2, ifInOctets.3, etc. Therefore Indexed MIB objects (per Section 5.3.3) are required in order to export arbitrary ifInOctets.x.

```
      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 = 36          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |        Template ID = 256      |        Field Count = 1        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|IE=MIBObjectIdentifierMarker |        Field Length = 4       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |Index Count = 0|MIB OID Len=22 |    MIB Object Identifier ...  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     ... MIB Object Identifier = "1.3.6.1.2.1.2.2.1.10.1"      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued                |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: Template for exporting ifInOctets.1

6.2. Non-indexed MIB Object: Established TCP Connections

The number of established TCP connections of a remote network device could be monitored by configuring it to periodically export the number of established TCP connections to a centralized Collector. In this example, the Exporter would export an IPFIX Message every 30 minutes that contained Data Records detailing the number of established TCP connections.

The table of data that is to be exported looks like:

TIMESTAMP	ESTABLISHED TCP CONN.
StartTime + 0 seconds	10
StartTime + 60 seconds	14
StartTime + 120 seconds	19
StartTime + 180 seconds	16
StartTime + 240 seconds	23
StartTime + 300 seconds	29

Table 1: Established TCP Connections

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

1. flowStartSeconds from [RFC5102], Information Element 150: The absolute timestamp of the first packet of this Flow.

2. tcpCurrEstab from [RFC4022], Object ID "1.3.6.1.2.1.6.9": The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.

Figure 13 shows the exported Template Set detailing the Template Record for exporting the number of established TCP connections (see Section 6.2).

```
      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 = 33          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |        Template ID = 257      |        Field Count = 2        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|   IE = flowStartSeconds     |        Field Length = 4       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|IE=MIBObjectIdentifierMarker |        Field Length = 4       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |Index Count = 0|MIB OID Len=15 |    MIB Object Identifier ...  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |         ... MIB Object Identifier = "1.3.6.1.2.1.6.9"         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ... |
     +-+-+-+-+-+-+-+-+
```

Figure 13: Example of tcpCurrEstab Template Set

Figure 14 shows the start of the Data Set for exporting the number of established TCP connections (see Section 6.2).

```
      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 = 257         |         Length = 52           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    StartTime +   0 seconds                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              10                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    StartTime +  60 seconds                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              14                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    StartTime + 120 seconds                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              19                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    StartTime + 180 seconds                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              16                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    StartTime + 240 seconds                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              23                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    StartTime + 300 seconds                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              29                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: Example of tcpCurrEstab Data Set

6.3. Enterprise Specific MIB Object: Detailing CPU Load History

For the sake of demonstrating a enterprise-specific MIB object, a non-indexed MIB object is chosen for simplicity. The CPU Usage of a remote network device could be monitored by configuring it to periodically export CPU usage information, i.e. the cpmCPUTotal1minRev from the proprietary CISCO-PROCESS-MIB, Object ID "1.3.6.1.4.1.9.9.109.1.1.1.1.7", to a centralized Collector. In this example, the Exporter would export 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 looks 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%

Table 2: CPU Usage Data

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

1. flowStartSeconds from [RFC5102], Information Element 150: The absolute timestamp of the first packet of this Flow.

2. cpmCPUTotal1minRev, the overall CPU busy percentage in the last one-minute period

Figure 15 shows the exported Template Set detailing the Template Record for exporting CPU Load (see Section 6.3).

```
      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 = 258      |        Field Count = 2        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|   IE = flowStartSeconds     |        Field Length = 4       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|IE=MIBObjectIdentifierMarker |        Field Length = 1       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |Index Count = 0|MIB OID Len=29 |    MIB Object Identifier ...  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  ... MIB Object Identifier = "1.3.6.1.4.1.9.9.109.1.1.1.1.7"  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  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 15: Example of CPU Load Template Set

Note that although cpmCPUTotal1minRev is 32 bits long, reduced size encoding ([RFC5101]) has been used to encoded it within a single octet.

This example stresses that, even though the OID cpmCPUTotal1minRev is enterprise-specific, the E bit for the MIBObjectIdentifierMarker is set to "0" since the "MIBObjectIdentifierMarker" Information Element is not enterprise-specific.

The corresponding Data Set does not add any value for this example, and is therefore not displayed.

6.4. Indexed MIB Object with an OID: Output Interface Queue Size in PSAMP Packet Report

Following on the example from the previous section (see Section 6.6), if the Template Record for the example Data Record does not contain the egressInterface, the ifOutQLen must be indexed by the ifIndex

interface index as detailed in the IF-MIB [RFC2863]:

The Template Record for the example Data Record contains the following Information Elements:

1. sourceIPv4Address

2. destinationIPv4Address

3. totalLengthIPv4

4. ifOutQLen indexed by: ifIndex

Figure 16 shows the exported Template Set detailing the Template for exporting a PSAMP Report with Interface Output Queue Length (ifOutQLen) but using the ifIndex MIB object as the exported index.

```
      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 = 70          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |        Template ID = 259      |        Field Count = 4        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|   IE = sourceIPv4Address    |        Field Length = 4       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0| IE = destinationIPv4Address |        Field Length = 4       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|    IE = totalLengthIPv4     |        Field Length = 4       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|IE=MIBObjectIdentifierMarker |        Field Length = 1       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Index Count=1 |MIB OID Len=20 |    MIB Object Identifier ...  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      ... 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 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  1.3.6.1.2.1.2.2.1.1 length   |MIB OID Len=19 | MIB Obj ID ...|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |       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 cont|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: Example of a Template for a PSAMP Report with ifOutQLen using ifIndex from IF-MIB [RFC2863] as an index

Note that IPFIX reduced size encoding [RFC5101] has been used in this example to express ifOutQLen in a single octet, rather than the 32 bits specified in the IF-MIB [RFC2863].

The corresponding IPFIX Data Record is shown in Figure 17. For the

sake of the example, the interface index of "Eth 1/0" is 15 and the interface index of "Eth 1/1" is 16.

```
      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 = 259         |         Length = 72           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.1                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.3                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                             150                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      45       |                     15 ...                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      ...      |                  192.0.2.4 ...                |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      ...      |                  192.0.2.9 ...                |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      ...      |                     350 ...                   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      ...      |       45      |             15 ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |             ...               |         192.0.2.3 ...         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |             ...               |         192.0.2.9 ...         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |             ...               |              650 ...          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |             ...               |       23      |    ...        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                 ... 15                        |    ...        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                 ... 192.0.2.4                 |    ...        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                 ... 192.0.2.6                 |    ...        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                 ...  350                      |       0       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              16                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17: Example of PSAMP Packet Report with the ifOutQLen using ifIndex from IF-MIB [RFC2863] as an index

6.5. Indexed MIB Object with Two OIDs: The ipIfStatsInForwDatagrams

MIB objects may be indexed by multiple indices. Note that all the indices apply to the MIB object, i.e. index 2 is not an index of index 1.

This example shows the export of ipIfStatsInForwDatagrams from the IP-MIB [RFC4293] indexed by the ipIfStatsIPVersion and ipIfStatsIfIndex which are provided as scope fields in an IPFIX option. Note that since these fields are used as indices for ipIfStatsInForwDatagrams, they don't need their own indices to be identified.

The Options Template Record for the example Data Record contains the following Information Elements:

1. ipIfStatsIPVersion (1.3.6.1.2.1.4.31.3.1.1) (scope field)

2. ipIfStatsIfIndex (1.3.6.1.2.1.4.31.3.1.2) (scope field)

3. ipIfStatsInForwDatagrams (1.3.6.1.2.1.4.31.3.1.12) (non-scope field) indexed by ipIfStatsIPVersion and ipIfStatsIfIndex

Figure 18 shows the exported Options Template Set.

```
    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 = TBD2         |          Length = 146         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |       Template ID = 260       |        Field Count = 3        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Scope Field Count = 2     |0|  MIBObjectIdentifierMarker  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Scope Field 1 Length = 1    |Index Count = 0|MIB OID Len=22 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      MIB Object Identifier = "1.3.6.1.2.1.4.31.3.1.1" ...     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           ...  MIB Object Identifier continued ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           ...  MIB Object Identifier continued ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           ...  MIB Object Identifier continued ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           ...  MIB Object Identifier continued ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |MIB Object Identifier continued|0|  MIBObjectIdentifierMarker  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     |   Scope Field 2 Length = 2    |Index Count = 0|MIB OID Len=22 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      MIB Object Identifier = "1.3.6.1.2.1.4.31.3.1.2" ...     |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |MIB Object Identifier continued|0|  MIBObjectIdentifierMarker  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |       Field Length = 4        |Index Count = 2|MIB OID Len=23 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     MIB Object Identifier = "1.3.6.1.2.1.4.31.3.1.12" ...     |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued|0|MIB OID IE...|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |... MIB OID IE | 1.3.6.1.2.1.4.31.3.1.1 Length |MIB OID Len=22 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     MIB Object Identifier = "1.3.6.1.2.1.4.31.3.1.1" ...      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |   ... MIB Object Identifier   |0|         MIB OID IE          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | 1.3.6.1.2.1.4.31.3.1.2 Length | MIB OID Len=22| MIB Obj ID ...|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     MIB Object Identifier = "1.3.6.1.2.1.4.31.3.1.2" ...      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |            ... MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   |             ... MIB Object Identifier continued ...           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             ... MIB Object Identifier continued ...           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   ... MIB Object Identifier   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: Example of an Options Template for an Indexed MIB Object with two indices.

6.6. Indexed MIB Object with an IPFIX Information Element: 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 be exporting the size of the send queue (ifOutQLen) in the same Data Record as the PSAMP Packet Report.

The exported data looks like:

SRC ADDR	DST ADDR	PAK LEN	OUTPUT I/F	OUTPUT Q. LEN (ifOutQLen)
192.0.2.1	192.0.2.3	150	Eth 1/0 (15)	45
192.0.2.4	192.0.2.9	350	Eth 1/0 (15)	45
192.0.2.3	192.0.2.9	650	Eth 1/0 (15)	23
192.0.2.4	192.0.2.6	350	Eth 1/1 (16)	0

Table 3: Packet Report with Interface Output Queue Length (ifOutQLen) Data

The MIB object for the Interface 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 the example is 15, the full MIB Object Identifier for (ifOutQLen) would be "1.3.6.1.2.1.2.2.1.21.15". Without a method to specify the index the full MIB OID would have to be used, which would mean specifying a new Template Record. Rather than export a separate Template Record for each Interface Index, it is more practical to identify the index in the Data Record itself.

In fact, only how the indexed object was indexed is necessary, although it is often useful to specify the index value. The example identifies the Egress Interface, but for other uses it may be sufficient to know that the ifOutQLen value was taken for the interface that the packet was switched out of, without identifying the actual interface.

The Template Record for the example Data Record contains the following Information Elements:

1. sourceIPv4Address

2. destinationIPv4Address

3. totalLengthIPv4

4. egressInterface

5. ifOutQLen indexed by: egressInterface

Figure 19 shows the exported Template Set detailing the Template for exporting a PSAMP Report with Interface Output Queue Length (ifOutQLen) (see Section 6.4).

```
      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 = 54          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |        Template ID = 261      |        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|IE=MIBObjectIdentifierMarker |        Field Length 4         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Index Count=1 |MIB OID Len=20 |    MIB Object Identifier ...  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      ... 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     |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  egressInterface Length = 4   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19: Example of Template for a PSAMP Report with ifOutQLen indexed by egressInterface

The corresponding IPFIX Data Record is shown in Figure 20. For the sake of the example, the interface index of "Eth 1/0" is 15 and the interface index of "Eth 1/1" is 16.

```
      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 = 261         |          Length = 84          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.1                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.3                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                             150                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                        15 (Eth 1/0)                           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              45                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.4                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.9                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                             350                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                        15 (Eth 1/0)                           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              45                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.3                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.9                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                             650                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                        15 (Eth 1/0)                           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                              23                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.4                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          192.0.2.6                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                             350                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                        16 (Eth 1/1)                           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                               0                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20: Example of PSAMP Packet Report with ifOutQLen indexed by egressInterface

6.7. Indexed MIB Objects with a mix of MIB OID and IPFIX Information Element

TODO.

6.8. Indexed MIB Object with MIBInstanceIdentifier Information Element: ipIfStatsOutOctets

This example shows the export of ipIfStatsOutOctets from the IP-MIB [RFC4293] indexed by the ipIfStatsIPVersion and ipIfStatsIfIndex, using the MIBInstanceIdentifier Information Element to carry the index information.

The exported data looks like:

ipIfStatsIPVersion	ipIfStatsIfIndex	ipIfStatsOutOctets
1(IPv4)	10	235876
2(IPv6)	11	38688

Table 4: The number octets in IP datagrams delivered to the lower layers for transmission

The MIB object ipIfStatsOutOctets ("1.3.6.1.2.1.4.31.3.1.32"), is indexed by ipIfStatsIPVersion and ipIfStatsIfIndex as detailed in IP-MIB [RFC4293]. The instance of the ipIfStatsOutOctets for the IPv4 protocol on the interface identified by ifIndex 10 is identified in the data record with the instance identifier segment ("1.10") in string format, while the instance of the ipIfStatsOutOctets for the IPv6 protocol on the interface identified by ifIndex 11 is identified in the data record with the instance identifier segment ("2.11") in string format.

The Template Record for the example Data Records contains the following Information Elements:

1. ipIfStatsOutOctets (1.3.6.1.2.1.4.31.3.1.32)

Figure 21 shows the exported Template Set.

```
      0                   1                   2                   2
      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 = 86          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |       Template ID = 264       |        Field Count = 1        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|IE=MIBObjectIdentifierMarker |        Field Length = 4       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Index Count=1 |MIB OID Len=23 |    MIB Object Identifier ...  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  ...MIB Object Identifier = "1.3.6.1.2.1.4.31.3.1.32"         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |...MIB OID     |0|  MIBInstanceIdentifier IE   | FieldLength...|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |   ... = FFFF  |
     +-+-+-+-+-+-+-+-+
```

Figure 21: Example of a Template for an MIB Objects that use the MIBInstanceIdentifier Information Element

The corresponding IPFIX Data Record is shown in Figure 22.

Variable length encoding is used for MIBInstanceIdentifier 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 = 264        |          Length = 22          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                  ipIfStatsOutOctets = 235876                  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  Length = 4   |                  "1.10"...                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     ...       |          ipIfStatsOutOctets = 38688           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     ...       |  Length = 4   |            "2.11"...          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |              ...              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 22: Example of ipIfStatsOutOctets using ipIfStatsIPVersion and ipIfStatsIfIndex as indices

6.9. Using MIB Objects as IPFIX Options Scope fields

Both indexed and non-indexed MIB Objects may be used as IPFIX Options Scope fields as discussed in Section 5.2.

6.9.1. Using non-Indexed MIB Objects as Option Scope fields

In this example, a Cisco Telepresence system uses an IPFIX option to report bandwidth usage statistics. The ctpcLocalAddrType and ctpcLocalAddr OIDs from the CISCO-TELEPRESENCE-CALL MIB are used as scope fields to identify the Telepresence system. The ctpcLocalAddrType is expressed with a fixed size of 1 octet, while the ctpcLocalAddr is expressed using a variable length field.

These scope fields are followed by two non-scope fields containing the number of packets and bytes. IPFIX reduced size encoding is used to express each of these fields in 32 bits.

Therefore the Options Template Record for the example Data Record contains the following Information Elements:

1. ctpcLocalAddrType (1.3.6.1.4.1.9.9.644.1.2.1) (scope field)

2. ctpcLocalAddr (1.3.6.1.4.1.9.9.644.1.2.2) (scope field)

3. octetDeltaCount (non-scope field)

4. packetDeltaCount (non-scope field)

The IPFIX Options Template Record is shown in Figure 23.

```
      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 = TBD2         |          Length = 80          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |       Template ID = 262       |        Field Count = 4        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     Scope Field Count = 2     |0|  MIBObjectIdentifierMarker  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |   Scope Field 1 Length = 1    |Index Count = 0|MIB OID Len=25 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |    MIB Object Identifier = "1.3.6.1.4.1.9.9.644.1.2.1" ...    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |... MIB OID ID |0|  MIBObjectIdentifierMarker | Scope Field ...|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |...Length=65535|Index Count = 0|MIB OID Len=25 | MIB OID ID ...|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  ... MIB Object Identifier = "1.3.6.1.4.1.9.9.644.1.2.2" ...  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued                |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|     octetDeltaCount = 1     |       Field Length = 4        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0|    packetDeltaCount = 2     |       Field Length = 4        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 23: Example of an IPFIX Options Template Record using non-Indexed MIB Objects as scope fields

The corresponding IPFIX Options Data Record is shown in Figure 24.

```
   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 = 262          |          Length = 18          |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | AddrType = 1  |  Length = 4   |  ctpcLocalAddrsystemID = ...  |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |         ... 192.0.2.1         |  octetDeltaCount = nnnn ...   |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | ... octetDeltaCount continued |  packetDeltaCount = nnnn ...  |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | ... packetDeltaCount continued|
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 24: Example of an IPFIX Options Data Record using non-Indexed MIB Objects as scope fields

6.9.2. Using Indexed MIB Objects as Option Scope fields

In this example, interface statistics are reported using ifName and ifInOctets from the IF-MIB [RFC2863]. Both of these fields are indexed by the ifIndex. The ifName and ifIndex are scope fields.

Therefore the Options Template Record for the example Data Record contains the following Information Elements:

1. ifName (1.3.6.1.2.1.31.1.1.1.1) (scope field) indexed by ifIndex

2. ifIndex (1.3.6.1.2.1.2.2.1.1) (scope field)

3. ifInOctets (1.3.6.1.2.1.2.2.1.10) (non-scope field) indexed by ifIndex

The IPFIX Options Template Record is shown in Figure 25.

```
   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 = TBD2         |          Length = 137         |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |       Template ID 263         |         Field Count = 3       |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |     Scope Field Count = 2     |0|  MIBObjectIdentifierMarker  |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |  Scope Field 1 Length = 65535 |Index Count = 1|MIB OID Len=22 |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     |       MIB Object Identifier = "1.3.6.1.2.1.31.1.1.1.1"        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |MIB Object Identifier continued|0|  MIBObjectIdentifierMarker  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | Scope Field 1 index Length = 4|MIB OID Len=19 | MIB OID ID ...|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     ... 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   |0|  MIBObjectIdentifierMarker  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |    Scope Field 2 Length = 4   |Index Count = 0|MIB OID Len=19 |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |       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 Obj Identifier continued       |0| MIBObject...|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |...Ident Marker|       Field Length =  4       |Index Count = 1|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |MIB OID Len=20 |MIB Object Identifier="1.3.6.1.2.1.2.2.1.10"...|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           ...  MIB Object Identifier continued ...            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
  |  ... MIB OID  |0|  MIBObjectIdentifierMarker  |  Field 1 ...  |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |... index Len=4|MIB OID Len=19 |   MIB Object Identifier ...   |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |     ... 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 Obj Id |
  +-+-+-+-+-+-+-+-+
```

Figure 25: Example of an IPFIX Options Template Record using Indexed MIB Objects as scope fields

The corresponding IPFIX Options Data Record is shown in Figure 26. For the sake of the example, the interface index of "Eth 1/1" is 15 and the ifInOctets are 1000.

```
   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 = 263          |          Length = 20          |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |  Length = 7   |             ifName = "Eth 1/1" ...            |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                     ... ifName continued                      |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                         ifIndex = 15                          |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                       ifInOctets = 1000                       |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 26: Example of an IPFIX Options Data Record using Indexed MIB Objects as scope fields

6.10. Using MIB Objects with IPFIX Structured Data

It's possible to export both indexed and non-indexed MIB Objects using IPFIX Structured Data per [RFC6313] as shown in the example below.

TODO: insert example.

7. Configuration Considerations

When configuring a MIB OID for export, consideration should be given to whether the SNMP Context String should also be configurable. If a non-default Context String is used then it should be associated with the fields as per Section 5.5.

8. 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 [RFC5101].

The specifications in section 9 of [RFC5101] 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 MIB modules in order to look up the received MIB Object Identifiers and find the type and name of MIB OID fields used in received templates. It should be noted that since reduced size 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 indices then it SHOULD log an error, but it MAY use the Template Record to decode the Data Records as the associated indices are purely semantic information.

9. Applicability

Making available the many and varied items from MIB modules 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 "ifInUcastPkts" of the IF-MIB [RFC2863] 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.

10. Security Considerations

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

The access to MIB objects is controlled by the configuration of the IPFIX exporter. This is consistent with the way IPFIX controls access to other Information Elements in general. The configuration of an IPFIX exporter determines which MIB objects are included in IPFIX flow records sent to certain collectors. Network operators should take care that only MIB objects are included in IPFIX flow records that the receiving flow collector is allowed to receive.

11. IANA Considerations

11.1. New Set IDs

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 allocated in IANA's IPFIX Set IDs registry [IANA-SETS], and 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.

11.2. New Data Types

A new mibObject data type must be allocated in IANA's IPFIX Information Element Data Types registry, [IANA-DATATYPES].

11.3. New Information Elements

Two new Information Elements, "MIBObjectIdentifierMarker", and "MIBInstanceIdentifier" must be allocated in IANA's IPFIX registry, [IANA-IPFIX]:

MIB Object Identifier Marker

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

Abstract Data Type: mibObject

Data Type Semantics: identifier

ElementId: TBD

Status: current

Reference: [this document].

MIB Instance Identifier

Description: IPFIX Information Element, MIBInstanceIdentifier, that denotes that a MIB Instance identifier string is exported in the data record following the MIB Object's value. This instance identifier when concatenated with the MIB object type OID that was sent in the template record gives the complete OID of the MIB variable that is being exported.

Abstract Data Type: mibObject

Data Type Semantics: identifier

ElementId: TBD

Status: current

Reference: [this document].

12. Acknowledgements

The authors would like to thank Andrew Johnson for his collaboration on the first version of the draft.

13. References

13.1. Normative References

[IANA-DATATYPES]
IANA, "IPFIX Information Element Data Types registry", <ht

tp://www.iana.org/assignments/ipfix/
ipfix.xml#ipfix-information-element-data-types>.

[IANA-IPFIX]
IANA, "IPFIX Information Elements registry", <http://www.iana.org/assignments/ipfix/ipfix.xml>.

[IANA-SETS]
IANA, "IPFIX Set IDs registry", <http://www.iana.org/assignments/ipfix/ipfix.xml#ipfix-set-ids>.

[PEN] IANA, "Private Enterprise Numbers registry", <http://www.iana.org/assignments/enterprise-numbers>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC4293] Routhier, S., "Management Information Base for the Internet Protocol (IP)", RFC 4293, April 2006.

[RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5102] Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, "Information Model for IP Flow Information Export", RFC 5102, January 2008.

13.2. Informative References

[RFC2982] Kavasseri, R., "Distributed Management Expression MIB", RFC 2982, October 2000.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.

[RFC4022] Raghunarayan, R., "Management Information Base for the Transmission Control Protocol (TCP)", RFC 4022, March 2005.

[RFC5476] Claise, B., Johnson, A., and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, March 2009.

[RFC6313] Claise, B., Dhandapani, G., Aitken, P., and S. Yates, "Export of Structured Data in IP Flow Information Export (IPFIX)", RFC 6313, July 2011.

Authors' Addresses

Benoit Claise
Cisco Systems, Inc.
De Kleetlaan 6a b1
Diegem,   1813
Belgium

Phone: +32 2 704 5622
Email: bclaise@cisco.com

Paul Aitken
Cisco Systems, Inc.
96 Commercial Quay
Commercial Street
Edinburgh,   EH6 6LX
UK

Phone: +44 131 561 3616
Email: paitken@cisco.com

Srikar
Cisco Systems, Inc.
Mail Stop BGL13/3/, SEZ Unit, Cessna Business Park, Kadubeesanahalli
Village Varthur Hobli, Sarjapur Marathalli Outer Ring Road
Bangalore,   KARNATAKA 560 103
IN

Phone: +91 80 4426 3264
Email: srikar@cisco.com

Juergen Schoenwaelder
Jacobs University Bremen
Campus Ring 1
Bremen,   28725
Germany

Phone: +49 421 200-3587
Email: j.schoenwaelder@jacobs-university.de