

Internet Engineering Task Force (IETF)  
Request for Comments: 7012  
Obsoletes: 5102  
Category: Standards Track  
ISSN: 2070-1721

B. Claise, Ed.  
Cisco Systems, Inc.  
B. Trammell, Ed.  
ETH Zurich  
September 2013

## Information Model for IP Flow Information Export (IPFIX)

### Abstract

This document defines the data types and management policy for the information model for the IP Flow Information Export (IPFIX) protocol. This information model is maintained as the IANA "IPFIX Information Elements" registry, the initial contents of which were defined by RFC 5102. This information model is used by the IPFIX protocol for encoding measured traffic information and information related to the traffic Observation Point, the traffic Metering Process, and the Exporting Process. Although this model was developed for the IPFIX protocol, it is defined in an open way that allows it to be easily used in other protocols, interfaces, and applications. This document obsoletes RFC 5102.

### Status of This Memo

This is an Internet Standards Track document.

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

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7012>.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction .....	3
1.1. Changes since RFC 5102 .....	4
1.2. IPFIX Documents Overview .....	4
2. Properties of IPFIX Protocol Information Elements .....	5
2.1. Information Element Specification Template .....	5
2.2. Scope of Information Elements .....	7
2.3. Naming Conventions for Information Elements .....	8
3. Type Space .....	9
3.1. Abstract Data Types .....	9
3.1.1. unsigned8 .....	9
3.1.2. unsigned16 .....	9
3.1.3. unsigned32 .....	9
3.1.4. unsigned64 .....	9
3.1.5. signed8 .....	10
3.1.6. signed16 .....	10
3.1.7. signed32 .....	10
3.1.8. signed64 .....	10
3.1.9. float32 .....	10
3.1.10. float64 .....	10
3.1.11. boolean .....	10
3.1.12. macAddress .....	10
3.1.13. octetArray .....	10
3.1.14. string .....	11
3.1.15. dateTimeSeconds .....	11
3.1.16. dateTimeMilliseconds .....	11
3.1.17. dateTimeMicroseconds .....	11
3.1.18. dateTimeNanoseconds .....	11
3.1.19. ipv4Address .....	11
3.1.20. ipv6Address .....	11

3.1.21. basicList .....	11
3.1.22. subTemplateList .....	11
3.1.23. subTemplateMultiList .....	12
3.2. Data Type Semantics .....	12
3.2.1. quantity .....	12
3.2.2. totalCounter .....	12
3.2.3. deltaCounter .....	12
3.2.4. identifier .....	13
3.2.5. flags .....	13
4. Information Element Identifiers .....	13
5. Information Elements .....	14
6. Extending the Information Model .....	15
7. IANA Considerations .....	15
7.1. IPFIX Information Elements .....	16
7.2. MPLS Label Type Identifier .....	17
7.3. XML Namespace and Schema .....	17
7.4. Addition, Revision, and Deprecation .....	18
8. Security Considerations .....	19
9. Acknowledgments .....	19
10. References .....	19
10.1. Normative References .....	19
10.2. Informative References .....	20
Contributors .....	23

## 1. Introduction

The IP Flow Information Export (IPFIX) protocol serves as a means for transmitting information related to network traffic measurement. The IPFIX Protocol Specification [RFC7011] defines how Information Elements are transmitted and also specifies the encoding of a set of basic data types for these Information Elements. However, the list of Information Elements that can be transmitted by the protocol, such as Flow attributes (source IP address, number of packets, etc.) and information about the Metering Process and Exporting Process (packet Observation Point, sampling rate, Flow timeout interval, etc.), is not specified in [RFC7011].

The IANA "IPFIX Information Elements" registry [IANA-IPFIX] is the current complete reference for IPFIX Information Elements. The initial values for this registry were provided by [RFC5102].

This document complements the IPFIX Protocol Specification [RFC7011] by providing an overview of the IPFIX information model and specifying data types for it. IPFIX-specific terminology used in this document is defined in Section 2 of [RFC7011]. As in [RFC7011], these IPFIX-specific terms have the first letter of a word capitalized when used in this document.

The use of the term 'information model' is not fully in line with the definition of this term in [RFC3444], as the IPFIX information model does not specify relationships between Information Elements, nor does it specify a concrete encoding of Information Elements. For an encoding suitable for use with the IPFIX protocol, see [RFC7011]. Besides the encoding used by the IPFIX protocol, other encodings of IPFIX Information Elements can be applied, for example, XML-based encodings.

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

### 1.1. Changes since RFC 5102

This document obsoletes the Proposed Standard revision of the IPFIX information model specification [RFC5102]. The following changes have been made to this document with respect to the previous document:

- At the time of this publication, technical and editorial errata reported for [RFC5102] have been reviewed and addressed as needed.
- All references to [RFC5101] have been updated to [RFC7011], reflecting changes to [RFC5101].
- Information Element definitions have been removed, as the reference for these is now [IANA-IPFIX]; a historical note on categorizations of Information Elements as defined in [RFC5102] has been retained in Section 5.
- The process for modifying [IANA-IPFIX] has been improved and is now described in [RFC7013]; Section 6 has been updated accordingly, and a new Section 7.3 provides IANA considerations for this process.
- Definitions of timestamp data types have been clarified.
- Appendices A and B have been removed.

### 1.2. IPFIX Documents Overview

The IPFIX protocol provides network administrators with access to network flow information. The architecture for the export of measured flow information out of an IPFIX Exporting Process to a Collecting Process is defined in [RFC5470], per the requirements defined in [RFC3917]. The IPFIX Protocol Specification [RFC7011]

defines how IPFIX Data Records and templates are carried via a number of transport protocols from IPFIX Exporting Processes to IPFIX Collecting Processes.

Four IPFIX optimizations/extensions are currently specified: a bandwidth-saving method for the IPFIX protocol [RFC5473], an efficient method for exporting bidirectional flows [RFC5103], a method for the definition and export of complex data structures [RFC6313], and the specification of the Protocol on IPFIX Mediators [IPFIX-MED-PROTO] based on the IPFIX Mediation Framework [RFC6183].

IPFIX has a formal description of IPFIX Information Elements -- their names, data types, and additional semantic information -- as specified in this document. The export of the Information Element types is specified in [RFC5610].

[RFC6728] specifies a data model for configuring and monitoring devices that are IPFIX and Packet Sampling (PSAMP) compliant using the Network Configuration Protocol (NETCONF), while [RFC6615] specifies MIB modules for monitoring.

In terms of development, [RFC5153] provides guidelines for the implementation and use of the IPFIX protocol, while [RFC5471] provides guidelines for testing.

Finally, [RFC5472] describes what types of applications can use the IPFIX protocol and how they can use the information provided. It furthermore shows how the IPFIX framework relates to other architectures and frameworks.

## 2. Properties of IPFIX Protocol Information Elements

### 2.1. Information Element Specification Template

Information in messages of the IPFIX protocol is modeled in terms of Information Elements of the IPFIX information model. The IPFIX Information Elements mentioned in Section 5 are specified in [IANA-IPFIX].

All Information Elements specified for the IPFIX protocol MUST have the following properties defined:

name - A unique and meaningful name for the Information Element.

elementId - A numeric identifier of the Information Element. If this identifier is used without an enterprise identifier (see [RFC7011] and the definition of enterpriseId listed below), then it is globally unique, and the list of allowed values is administered by IANA. It is used for compact identification of an Information Element when encoding Templates in the protocol.

description - The semantics of this Information Element. Describes how this Information Element is derived from the Flow or other information available to the observer. Information Elements of dataType string or octetArray that have length constraints (fixed length, minimum and/or maximum length) MUST note these constraints in their descriptions.

dataType - One of the types listed in Section 3.1 of this document or registered in the IANA "IPFIX Information Element Data Types" subregistry. The type space for attributes is constrained to facilitate implementation. The existing type space encompasses most primitive types used in modern programming languages, as well as some derived types (such as ipv4Address) that are common to this domain.

status - The status of the specification of this Information Element. Allowed values are 'current' and 'deprecated'. All newly defined Information Elements have 'current' status. The process for moving Information Elements to the 'deprecated' status is defined in Section 5.3 of [RFC7013].

Enterprise-specific Information Elements MUST have the following property defined:

enterpriseId - Enterprises may wish to define Information Elements without registering them with IANA, for example, for enterprise-internal purposes. For such Information Elements, the Information Element identifier described above is not sufficient when the Information Element is used outside the enterprise. If specifications of enterprise-specific Information Elements are made public and/or if enterprise-specific identifiers are used by the IPFIX protocol outside the enterprise, then the enterprise-specific identifier MUST be made globally unique by combining it with an enterprise identifier. Valid values for the enterpriseId are defined by IANA as Structure of Management Information (SMI) network management private enterprise numbers, defined at [IANA-PEN].

All Information Elements specified for the IPFIX protocol either in this document or by any future extension MAY have the following properties defined:

`dataTypeSemantics` - The integral types are qualified by additional semantic details. Valid values for the data type semantics are either specified in Section 3.2 of this document or will be specified in a future extension of the information model.

`units` - If the Information Element is a measure of some kind, the units identify what the measure is.

`range` - Some Information Elements may only be able to take on a restricted set of values that can be expressed as a range (e.g., 0 through 511, inclusive). If this is the case, the valid inclusive range SHOULD be specified; values for this Information Element outside the range are invalid and MUST NOT be exported.

`reference` - Identifies additional specifications that more precisely define this item or provide additional context for its use.

The following two Information Element properties are defined to allow the management of an Information Elements registry with Information Element definitions that may be updated over time, per the process defined in Section 5.2 of [RFC7013]:

`revision` - The revision number of an Information Element, starting at 0 for Information Elements at time of definition and incremented by one for each revision.

`date` - The date of the entry of this revision of the Information Element into the registry.

A template for specifying Information Elements is given in Section 9.1 of [RFC7013].

## 2.2. Scope of Information Elements

By default, most Information Elements have a scope specified in their definitions. Within Data Records defined by Options Templates, the IPFIX protocol allows further limiting of the Information Element scope. The new scope is specified by one or more scope fields and defined as the combination of all specified scope values; see Section 3.4.2.1 on IPFIX scopes in [RFC7011].

### 2.3. Naming Conventions for Information Elements

The following naming conventions were used for naming Information Elements in this document. It is recommended that extensions of the model use the same conventions.

- o Names of Information Elements SHOULD be descriptive.
- o Names of Information Elements MUST be unique within the "IPFIX Information Elements" registry [IANA-IPFIX]. Enterprise-specific Information Elements SHOULD be prefixed with a vendor name.
- o Names of Information Elements MUST start with lowercase letters.
- o Composed names MUST use capital letters for the first letter of each component (except for the first one). All other letters are lowercase, even for acronyms. Exceptions are made for acronyms containing a mixture of lowercase and capital letters, such as 'IPv4' and 'IPv6'. Examples are "sourceMacAddress" and "destinationIPv4Address".
- o Middleboxes [RFC3234] may change Flow properties, such as the Differentiated Services Code Point (DSCP) value or the source IP address. If an IPFIX Observation Point is located in the path of a Flow before one or more middleboxes that potentially modify packets of the Flow, then it may be desirable to also report Flow properties after the modification performed by the middleboxes. An example is an Observation Point before a packet marker changing a packet's IPv4 Type of Service (TOS) field that is encoded in Information Element ipClassOfService. Then the value observed and reported by Information Element ipClassOfService is valid at the Observation Point but not after the packet passed the packet marker. For reporting the change value of the TOS field, the IPFIX information model uses Information Elements that have a name prefix "post", for example, "postIpClassOfService". Information Elements with prefix "post" report on Flow properties that are not necessarily observed at the Observation Point but that are obtained within the Flow's Observation Domain by other means considered to be sufficiently reliable, for example, by analyzing the packet marker's marking tables.



### 3. Type Space

This section describes the abstract data types that can be used for the specification of IPFIX Information Elements in Section 4. Section 3.1 describes the set of abstract data types.

Abstract data types `unsigned8`, `unsigned16`, `unsigned32`, `unsigned64`, `signed8`, `signed16`, `signed32`, and `signed64` are integral data types. As described in Section 3.2, their data type semantics can be further specified, for example, by `'totalCounter'`, `'deltaCounter'`, `'identifier'`, or `'flags'`.

#### 3.1. Abstract Data Types

This section describes the set of valid abstract data types of the IPFIX information model, independent of encoding. Note that further abstract data types may be specified by future updates to this document. Changes to the associated IPFIX "Information Element Data Types" subregistry [IANA-IPFIX] specified in [RFC5610] require a Standards Action [RFC5226].

The current encodings of these data types for use with the IPFIX protocol are defined in [RFC7011]; encodings allowing the use of the IPFIX Information Elements [IANA-IPFIX] with other protocols may be defined in the future by referencing this document.

##### 3.1.1. `unsigned8`

The type `"unsigned8"` represents a non-negative integer value in the range of 0 to 255.

##### 3.1.2. `unsigned16`

The type `"unsigned16"` represents a non-negative integer value in the range of 0 to 65535.

##### 3.1.3. `unsigned32`

The type `"unsigned32"` represents a non-negative integer value in the range of 0 to 4294967295.

##### 3.1.4. `unsigned64`

The type `"unsigned64"` represents a non-negative integer value in the range of 0 to 18446744073709551615.

### 3.1.5. signed8

The type "signed8" represents an integer value in the range of -128 to 127.

### 3.1.6. signed16

The type "signed16" represents an integer value in the range of -32768 to 32767.

### 3.1.7. signed32

The type "signed32" represents an integer value in the range of -2147483648 to 2147483647.

### 3.1.8. signed64

The type "signed64" represents an integer value in the range of -9223372036854775808 to 9223372036854775807.

### 3.1.9. float32

The type "float32" corresponds to an IEEE single-precision 32-bit floating-point type as defined in [IEEE.754.2008].

### 3.1.10. float64

The type "float64" corresponds to an IEEE double-precision 64-bit floating-point type as defined in [IEEE.754.2008].

### 3.1.11. boolean

The type "boolean" represents a binary value. The only allowed values are "true" and "false".

### 3.1.12. macAddress

The type "macAddress" represents a MAC-48 address as defined in [IEEE.802-3.2012].

### 3.1.13. octetArray

The type "octetArray" represents a finite-length string of octets.

#### 3.1.14. string

The type "string" represents a finite-length string of valid characters from the Unicode coded character set [ISO.10646]. Unicode incorporates ASCII [RFC20] and the characters of many other international character sets.

#### 3.1.15. dateTimeSeconds

The type "dateTimeSeconds" represents a time value expressed with second-level precision.

#### 3.1.16. dateTimeMilliseconds

The type "dateTimeMilliseconds" represents a time value expressed with millisecond-level precision.

#### 3.1.17. dateTimeMicroseconds

The type "dateTimeMicroseconds" represents a time value expressed with microsecond-level precision.

#### 3.1.18. dateTimeNanoseconds

The type "dateTimeNanoseconds" represents a time value expressed with nanosecond-level precision.

#### 3.1.19. ipv4Address

The type "ipv4Address" represents an IPv4 address.

#### 3.1.20. ipv6Address

The type "ipv6Address" represents an IPv6 address.

#### 3.1.21. basicList

The type "basicList" supports structured data export as described in [RFC6313]; see Section 4.5.1 of that document for encoding details.

#### 3.1.22. subTemplateList

The type "subTemplateList" supports structured data export as described in [RFC6313]; see Section 4.5.2 of that document for encoding details.

### 3.1.23. subTemplateMultiList

The type "subTemplateMultiList" supports structured data export as described in [RFC6313]; see Section 4.5.3 of that document for encoding details.

## 3.2. Data Type Semantics

This section describes the set of valid data type semantics of the IPFIX information model. A subregistry of data type semantics [IANA-IPFIX] is established in [RFC5610]; the restrictions on the use of semantics below are compatible with those specified in Section 3.10 of that document. These semantics apply only to numeric types, as noted in the description of each semantic below.

Further data type semantics may be specified by future updates to this document. Changes to the associated "IPFIX Information Element Semantics" subregistry [IANA-IPFIX] require a Standards Action [RFC5226].

### 3.2.1. quantity

"quantity" is a numeric (integral or floating point) value representing a measured value pertaining to the record. This is distinguished from counters that represent an ongoing measured value whose "odometer" reading is captured as part of a given record. This is the default semantic type of all numeric data types.

### 3.2.2. totalCounter

"totalCounter" is an integral value reporting the value of a counter. Counters are unsigned and wrap back to zero after reaching the limit of the type. For example, an unsigned64 with counter semantics will continue to increment until reaching the value of  $2^{64} - 1$ . At this point, the next increment will wrap its value to zero and continue counting from zero. The semantics of a total counter is similar to the semantics of counters used in the Simple Network Management Protocol (SNMP), such as Counter32 as defined in [RFC2578]. The only difference between total counters and counters used in SNMP is that the total counters have an initial value of 0. A total counter counts independently of the export of its value.

### 3.2.3. deltaCounter

"deltaCounter" is an integral value reporting the value of a counter. Counters are unsigned and wrap back to zero after reaching the limit of the type. For example, an unsigned64 with counter semantics will continue to increment until reaching the value of  $2^{64} - 1$ . At this

point, the next increment will wrap its value to zero and continue counting from zero. The semantics of a delta counter is similar to the semantics of counters used in SNMP, such as Counter32 as defined in [RFC2578]. The only difference between delta counters and counters used in SNMP is that the delta counters have an initial value of 0. A delta counter is reset to 0 each time it is exported and/or expires without export.

#### 3.2.4. identifier

"identifier" is an integral value that serves as an identifier. Specifically, mathematical operations on two identifiers (aside from the equality operation) are meaningless. For example, Autonomous System ID 1 \* Autonomous System ID 2 is meaningless. Identifiers MUST be one of the signed or unsigned data types.

#### 3.2.5. flags

"flags" is an integral value that represents a set of bit fields. Logical operations are appropriate on such values, but other mathematical operations are not. Flags MUST always be of an unsigned data type.

### 4. Information Element Identifiers

All Information Elements defined in the IANA "IPFIX Information Elements" registry [IANA-IPFIX] have their identifiers assigned by IANA.

The values of these identifiers are in the range of 1-32767. Within this range, Information Element identifier values in the sub-range of 1-127 are compatible with field types used by NetFlow version 9 [RFC3954] for historical reasons.

In general, IANA will add newly registered Information Elements to the registry, assigning the lowest available Information Element identifier in the range of 128-32767.

Enterprise-specific Information Element identifiers have the same range of 1-32767, but they are coupled with an additional enterprise identifier. For enterprise-specific Information Elements, Information Element identifier 0 is also reserved. Enterprise-specific Information Element identifiers can be chosen by an enterprise arbitrarily within the range of 1-32767. The same identifier may be assigned by other enterprises for different purposes; these Information Elements are distinct because the Information Element identifier is coupled with an enterprise identifier.

Enterprise identifiers are to be registered as SMI network management private enterprise code numbers with IANA. The registry can be found at [IANA-PEN].

## 5. Information Elements

[IANA-IPFIX] is now the normative reference for IPFIX Information Elements. When [RFC5102] was published, it defined, in its Section 5, the initial contents of that registry.

As a historical note, Information Elements (IEs) were organized into categories in [RFC5102] according to their semantics and their applicability; these categories were not carried forward into [IANA-IPFIX] as an organizing principle. The categories (with example IEs) were:

1. Identifiers (e.g., `ingressInterface`)
2. Metering and Exporting Process Configuration (e.g., `exporterIPv4Address`)
3. Metering and Exporting Process Statistics (e.g., `exportedOctetTotalCount`)
4. IP Header Fields (e.g., `sourceIPv4Address`)
5. Transport Header Fields (e.g., `sourceTransportPort`)
6. Sub-IP Header Fields (e.g., `sourceMacAddress`)
7. Derived Packet Properties (e.g., `bgpSourceAsNumber`)
8. Min/Max Flow Properties (e.g., `minimumIpTotalLength`)
9. Flow Timestamps (e.g., `flowStartTimeMilliseconds`)
10. Per-Flow Counters (e.g., `octetDeltaCount`)
11. Miscellaneous Flow Properties (e.g., `flowEndReason`)
12. Padding (`paddingOctets`)

Information Elements derived from fields of packets or from Packet Treatment can typically serve as Flow Keys used for mapping packets to Flows. These Information Elements were placed in categories 4-7 in the original categorization.

Information Elements not serving as Flow Keys may have different values for each packet in a Flow. For Information Elements with values derived from fields of packets or from Packet Treatment, and for which the value may change from packet to packet within a single Flow, the exported value of an Information Element is by default determined by the first packet observed for the corresponding Flow; the description of the Information Element may, however, explicitly specify different semantics. This simple rule allows the writing of all Information Elements related to header fields once, when the first packet of the Flow is observed. For further observed packets

of the same Flow, only Flow properties that depend on more than one packet need to be updated; these Information Elements were placed in categories 8-11 in the original categorization.

Information Elements with a name having the "post" prefix (e.g., postIpClassOfService) do not necessarily report properties that were actually observed at the Observation Point but may be retrieved by other means within the Observation Domain. These Information Elements can be used if there are middlebox functions within the Observation Domain changing Flow properties after packets passed the Observation Point; they may also be reported directly by the Observation Point if the Observation Point is situated where it can observe packets on both sides of the middlebox.

## 6. Extending the Information Model

A key requirement for IPFIX is to allow for extension of the Information Model via the "IP Flow Information Export (IPFIX) Entities" registry [IANA-IPFIX]. New Information Element definitions can be added to this registry subject to Expert Review [RFC5226], with additional process considerations as described in [RFC7013]; that document also provides guidelines for authors and reviewers of new Information Element definitions.

For new Information Elements, the type space defined in Section 3 can be used. If required, new abstract data types can be added to the "IPFIX Information Element Data Types" subregistry [IANA-IPFIX] as defined in [RFC5610]. New abstract data types and semantics are subject to Standards Action [RFC5226] and MUST be defined in IETF Standards Track documents updating this document.

Enterprises may wish to define Information Elements without registering them with IANA. IPFIX explicitly supports enterprise-specific Information Elements. Enterprise-specific Information Elements are described in Sections 2.1 and 4; guidelines for using them appear in [RFC7013].

## 7. IANA Considerations

As this document obsoletes [RFC5102], IANA has updated the references in the "IP Flow Information Export (IPFIX) Entities" registry [IANA-IPFIX], the "IPFIX MPLS label type" subregistry of that registry, the urn:ietf:params:xml:ns:ipfix-info XML namespace, and the urn:ietf:params:xml:schema:ipfix-info XML schema to refer to this document.

However, [RFC5102] still provides a historical reference for the initial entries in the "IPFIX Information Elements" registry. Therefore, IANA has kept [RFC5102] as the requestor of those Information Elements in the "IPFIX Information Elements" registry that list [RFC5102] as their requestor and added the following explanatory note to the "IPFIX Information Elements" registry:

"RFC 7012 has obsoleted RFC 5102; references to RFC 5102 in this registry remain as part of the historical record".

The Information Element Specification Template (Section 2.1) requires two new columns not present in [RFC5102]. IANA has created a new Revision column in the "IPFIX Information Elements" registry and set the Revision of existing Information Elements to 0. IANA has also created a new Date column in that registry and set the Date of all existing Information Elements to the publication date of this document.

To identify Information Elements with identifiers 127 or below as NetFlow version 9 [RFC3954] compatible, IANA has set the Name of all existing Reserved Information Elements with identifier 127 or less to "Assigned for NetFlow v9 compatibility" and the Reference of those Information Elements to [RFC3954].

As IANA now has change control of the schema used for the IANA "IPFIX Information Elements" registry [IANA-IPFIX], IANA has deprecated the previous XML schema for the description of Information Elements `urn:ietf:params:xml:schema:ipfix-info` [IPFIX-XML-SCHEMA].

To support the process described in Section 7.4, IANA has established a mailing list for communicating with the IE-DOCTORS, named `ie-doctors@ietf.org`.

The remaining subsections of this section contain no actions for IANA.

### 7.1. IPFIX Information Elements

This document refers to Information Elements, for which the Internet Assigned Numbers Authority (IANA) has created the IPFIX "Information Elements" registry [IANA-IPFIX]. The columns of this registry must, at minimum, be able to store the information defined in the template detailed in Section 2.1; it may contain other information as necessary for the management of the registry.

The process for making additions or other changes to the "IPFIX Information Elements" registry is given in Section 7.4.



## 7.2. MPLS Label Type Identifier

Information Element #46, named `mplsTopLabelType`, carries MPLS label types. Values for 5 different types have initially been defined. For ensuring the extensibility of this information, IANA has created a new subregistry for MPLS label types and filled it with the initial list from the description Information Element #46, `mplsTopLabelType`.

New assignments for MPLS label types are administered by IANA through Expert Review [RFC5226], i.e., review by one of a group of experts designated by an IETF Area Director. The group of experts must double-check the label type definitions with already-defined label types for completeness, accuracy, and redundancy. The specification of new MPLS label types MUST be published using a well-established and persistent publication medium.

## 7.3. XML Namespace and Schema

The prior version of this document [RFC5102] specified an XML schema for IPFIX Information Element definitions [IPFIX-XML-SCHEMA] that was used in the generation of the document text itself. When the IANA "IPFIX Information Elements" registry [IANA-IPFIX] was created, change control on the registry and the schema used to validate it passed to IANA.

The use of a machine-readable syntax for the registry enables the creation of IPFIX tools that can automatically adapt to extensions to the information model. It should be noted that the use of XML in Exporters, Collectors, or other tools is not mandatory for the deployment of IPFIX. In particular, Exporting Processes do not produce or consume XML as part of their operation. IPFIX Collectors MAY take advantage of the machine-readability of the information model versus hard-coding their behavior or inventing proprietary means for accommodating extensions. However, in order to avoid unnecessary load on the IANA infrastructure serving the registry, Collectors SHOULD NOT poll the IANA registry [IANA-IPFIX] directly at runtime.

The reference to the current schema is embedded in the registry [IANA-IPFIX]; this schema may change from time to time as necessary to support the maintenance of the registry. As such, the schema `urn:ietf:params:xml:schema:ipfix-info` [IPFIX-XML-SCHEMA] specified in [RFC5102] has been deprecated.

#### 7.4. Addition, Revision, and Deprecation

New assignments for the "IPFIX Information Elements" registry are administered by IANA through Expert Review [RFC5226]. These experts are referred to as IE-DOCTORS and are appointed by the IESG. The process they follow is defined in [RFC7013].

Information Element identifiers in the range of 1-127 are compatible with field types used by NetFlow version 9 [RFC3954] for historical reasons and must not be assigned unless the Information Element is compatible with the NetFlow version 9 protocol, as determined by one of the IE-DOCTORS designated by the IESG as a NetFlow version 9 expert.

Future assignments added to the "IPFIX Information Elements" registry that require subregistries for enumerated values (e.g., Section 7.2) must have those subregistries added simultaneously with the new assignment; additions to these subregistries must be subject to Expert Review [RFC5226]. Unless specified at assignment time, the experts for the subregistry will be the same as for the "IPFIX Information Elements" registry as a whole.

When IANA receives a request to add, revise, or deprecate an Information Element in the "IPFIX Information Elements" registry, it forwards the request to the IE-DOCTORS for review.

When IANA receives an approval for a request to add an Information Element definition from the IE-DOCTORS, it adds that Information Element to the registry. The approved request may include changes made by the requestor and/or reviewers as compared to the original request.

When IANA receives an approval for a request to revise an Information Element definition from the IE-DOCTORS, it changes that Information Element's definition in the registry and updates the Revision and Date columns as appropriate. The approved request may include changes from the original request. If the original Information Element was added to the registry with IETF consensus (i.e., was defined by an RFC), the revision will require IETF consensus as well.

When IANA receives an approval for a request to deprecate an Information Element definition from the IE-DOCTORS, it changes that Information Element's definition in the registry and updates the Revision and Date columns as appropriate. The approved request may include changes from the original request. If the original Information Element was added to the registry with IETF consensus (i.e., was defined by an RFC), the deprecation will require IETF consensus as well.

## 8. Security Considerations

The IPFIX information model itself does not directly introduce security issues. Rather, it defines a set of attributes that may, for privacy or business issues, be considered sensitive information.

For example, exporting values of header fields may make attacks possible for the receiver of this information; this would otherwise only be possible for direct observers of the reported Flows along the data path.

The underlying protocol used to exchange the information described here must therefore apply appropriate procedures to guarantee the integrity and confidentiality of the exported information. These protocols are defined in separate documents, specifically the IPFIX protocol document [RFC7011].

## 9. Acknowledgments

This document is substantially based on [RFC5102]. The editors thank the authors of that document; those authors are listed below as contributors. Special thanks go to Paul Aitken for the detailed review. Finally, the authors thank the IPFIX WG chairs: Nevil Brownlee and Juergen Quittek.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6313] Claise, B., Dhandapani, G., Aitken, P., and S. Yates, "Export of Structured Data in IP Flow Information Export (IPFIX)", RFC 6313, July 2011.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.
- [RFC7013] Trammell, B., and B. Claise, "Guidelines for Authors and Reviewers of IP Flow Information Export (IPFIX) Information Elements", BCP 184, RFC 7013, September 2013.

## 10.2. Informative References

- [IANA-IPFIX]  
IANA, "IP Flow Information Export (IPFIX) Entities",  
<<http://www.iana.org/assignments/ipfix/>>.
- [IEEE.754.2008]  
Institute of Electrical and Electronics Engineers, "IEEE  
Standard for Floating-Point Arithmetic", IEEE  
Standard 754, August 2008.
- [IEEE.802-3.2012]  
Institute of Electrical and Electronics Engineers, "IEEE  
Standard for Ethernet", IEEE Standard 802.3, 2012.
- [IPFIX-MED-PROTO]  
Claise, B., Kobayashi, A., and B. Trammell, "Operation of  
the IP Flow Information Export (IPFIX) Protocol on IPFIX  
Mediators", Work in Progress, July 2013.
- [IPFIX-XML-SCHEMA]  
IANA, "IETF XML Registry",  
<<http://www.iana.org/assignments/xml-registry/>>.
- [ISO.10646]  
International Organization for Standardization,  
"Information technology - Universal Coded Character Set  
(UCS)", ISO/IEC 10646:2012, November 2012.
- [IANA-PEN] IANA, "Private Enterprise Numbers",  
<<http://www.iana.org/assignments/enterprise-numbers>>.
- [RFC20] Cerf, V., "ASCII format for Network Interchange", RFC 20,  
October 1969.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J.  
Schoenwaelder, Ed., "Structure of Management Information  
Version 2 (SMIV2)", STD 58, RFC 2578, April 1999.
- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and  
Issues", RFC 3234, February 2002.
- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between  
Information Models and Data Models", RFC 3444,  
January 2003.

- [RFC3917] Quittek, J., Zseby, T., Claise, B., and S. Zander, "Requirements for IP Flow Information Export (IPFIX)", RFC 3917, October 2004.
- [RFC3954] Claise, B., Ed., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, October 2004.
- [RFC5101] Claise, B., Ed., "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.
- [RFC5103] Trammell, B. and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)", RFC 5103, January 2008.
- [RFC5153] Boschi, E., Mark, L., Quittek, J., Stiemerling, M., and P. Aitken, "IP Flow Information Export (IPFIX) Implementation Guidelines", RFC 5153, April 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC 5470, March 2009.
- [RFC5471] Schmoll, C., Aitken, P., and B. Claise, "Guidelines for IP Flow Information Export (IPFIX) Testing", RFC 5471, March 2009.
- [RFC5472] Zseby, T., Boschi, E., Brownlee, N., and B. Claise, "IP Flow Information Export (IPFIX) Applicability", RFC 5472, March 2009.
- [RFC5473] Boschi, E., Mark, L., and B. Claise, "Reducing Redundancy in IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Reports", RFC 5473, March 2009.
- [RFC5610] Boschi, E., Trammell, B., Mark, L., and T. Zseby, "Exporting Type Information for IP Flow Information Export (IPFIX) Information Elements", RFC 5610, July 2009.

- [RFC6183] Kobayashi, A., Claise, B., Muenz, G., and K. Ishibashi, "IP Flow Information Export (IPFIX) Mediation: Framework", RFC 6183, April 2011.
- [RFC6615] Dietz, T., Ed., Kobayashi, A., Claise, B., and G. Muenz, "Definitions of Managed Objects for IP Flow Information Export", RFC 6615, June 2012.
- [RFC6728] Muenz, G., Claise, B., and P. Aitken, "Configuration Data Model for the IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Protocols", RFC 6728, October 2012.

## Contributors

Juergen Quittek  
NEC  
Kurfuersten-Anlage 36  
Heidelberg 69115  
Germany

Phone: +49 6221 90511-15  
EMail: [quittek@nw.neclab.eu](mailto:quittek@nw.neclab.eu)  
URI: <http://www.neclab.eu/>

Stewart Bryant  
Cisco Systems, Inc.  
10 New Square, Bedfont Lakes  
Feltham, Middlesex TW18 8HA  
United Kingdom

EMail: [stbryant@cisco.com](mailto:stbryant@cisco.com)

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

Phone: +44 131 561 3616  
EMail: [paitken@cisco.com](mailto:paitken@cisco.com)

Jeff Meyer  
PayPal  
2211 N. First St.  
San Jose, CA 95131-2021  
US

Phone: +1 408 976-9149  
EMail: [jemeyer@paypal.com](mailto:jemeyer@paypal.com)  
URI: <http://www.paypal.com>

## Authors' Addresses

Benoit Claise (editor)  
Cisco Systems, Inc.  
De Kleetlaan 6a b1  
1831 Diegem  
Belgium

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

Brian Trammell (editor)  
Swiss Federal Institute of Technology Zurich  
Gloriastrasse 35  
8092 Zurich  
Switzerland

Phone: +41 44 632 70 13  
EMail: trammell@tik.ee.ethz.ch