IPFIX Working Group	B. Trammell
Internet-Draft	ETH Zurich
Intended status: Best Current Practice	B. Claise
Expires: April 30, 2012	Cisco Systems, Inc.
	October 28, 2011

Guidelines for Authors and Reviewers of IPFIX Information Elements
draft-trammell-ipfix-ie-doctors-03.txt

Abstract

This document provides guidelines for the definition of IPFIX Information Elements for addition to the IANA IPFIX Information Element registry, in order to extend the applicability of the IPFIX protocol to new operations and management areas.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document provides guidelines for the extension of the applicability of the IP Flow Information Export (IPFIX) protocol to network operations and management purposes outside the initial scope defined in "IPFIX Applicability Statement" *[RFC5472]*. These new applications are largely defined by creating new Information Elements beyond those in the IANA IPFIX Information Element Registry *[iana-ipfix-assignments]*. New applications may be further specified through additional RFCs defining and describing their usage.
We intend this document to enable the expansion of the applicability of IPFIX to new areas by experts in the working group or area directorate concerned with the technical details of the protocol or application to be measured or managed using IPFIX. This expansion would occur with the consultation of IPFIX experts informally called 'IE-Doctors'. It provides guidelines both for those defining new Information Elements as well as the IE-Doctors reviewing them.

1.1. Intended Audience and Usage

This document is meant for two separate audiences. For IETF contributors extending the applicability of IPFIX, it provides a set of guidelines and best practices to be used in deciding which Information Elements are necessary for a given existing or new application, defining these Information Elements, and deciding whether an RFC should be published to further describe the application. For the IPFIX experts appointed as IE-Doctors, and for IANA personnel changing the

Information Element registry, it defines a set of acceptance criteria against which these proposed Information Elements should be evaluated. This document is not intended to guide the extension of the IPFIX protocol itself, e.g. through new export mechanisms, data types, or the like; these activities should be pursued through the publication of standards-track RFCs by the IPFIX Working Group.
This document specifies additional practices beyond those appearing in the IANA Considerations sections of existing IPFIX documents, especially the Information Model *[RFC5102]*. The practices outlined in this document are intended to guide experts when making changes to the IANA registry under Expert Review as defined in [RFC5226].

1.2. Overview of relevant IPFIX documents

[RFC5101] defines the IPFIX Protocol, the IPFIX-specific terminology used by this document, and the data type encodings for each of the data types supported by IPFIX.
[RFC5102] defines the initial IPFIX Information Model, as well as procedures for extending the Information Model. It states that new Information Elements may be added to the Information Model on Expert Review basis, and delegates the appointment of experts to an IESG Area Director. This document is intended to further codify the best practices to be followed by these experts, in order to improve the efficiency of this process.
[RFC5103] defines a method for exporting bidirectional flow information using IPFIX; this document should be followed when extending IPFIX to represent information about bidirectional network interactions in general. Additionally, new Information Elements should be annotated for their reversibility or lack thereof as per this document.
[RFC5610] defines a method for exporting information about Information Elements inline within IPFIX. In doing so, it explicitly defines a set of restrictions on the use of data types and semantics which are implied in [RFC5101] and [RFC5102]; these restrictions MUST be observed in the definition of new Information Elements, as in Section 4.4.

2. Terminology

Capitalized terms used in this document that are defined in the Terminology section of [RFC5101] are to be interpreted as defined there.
An "application", as used in this document, refers to a candidate protocol, task, or domain to which IPFIX export, collection, and/or storage is applied, beyond those within the IPFIX Applicability statement *[RFC5472]*. By this definition, PSAMP *[RFC5476]* was the first new IPFIX application after the publication of the IPFIX protocol [RFC5101].
"IANA registry", as used in this document, unless otherwise noted, refers to the IANA IPFIX Information Element Registry *[iana-ipfix-assignments]*.

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].

3. How to apply IPFIX

Though originally specified for the export of IP flow information, the message format, template mechanism, and data model specified by IPFIX lead to it being applicable to a wide variety of network management situations. In addition to flow information export, for which it was designed, and packet information export as specified by PSAMP *[RFC5476]*, any application with the following characteristics is a good candidate for an IPFIX application:

- The application's data flow is fundamentally unidirectional. IPFIX is a "push" protocol, supporting only the export of information from a sender (an Exporting Process) to a receiver (a Collecting Process). Request-response interactions are not supported by IPFIX.

- The application handles discrete event information, or information to be periodically reported. IPFIX is particularly well suited to representing events, which can be scoped in time.

- The application handles information about network entities. IPFIX's information model is network-oriented, so network management applications have many opportunities for information model reuse.

- The application requires a small number of arrangements of data structures relative to the number of records it handles. The template-driven self-description mechanism used by IPFIX excels at handling large volumes of identically structured data, compared to representations which define structure inline with data (such as XML).

Most applications meeting these criteria can be supported over IPFIX. Once it's been determined that IPFIX is a good fit, the next step is determining which Information Elements are necessary to represent the information required by the application. Especially for network-centric applications, the IPFIX Information Element registry may already contain all the necessary Information Elements (see Section 6.1 for guidelines on maximizing Information Element reuse). In this case, no additional work within the IETF is necessary: simply define Templates and start exporting.
It is expected, however, that most applications will be able to reuse some existing Information Elements, but must define some additional Information Elements to support all their requirements; in this case,

see Section 4 for best practices to be followed in defining Information Elements.
Optionally, a Working Group or individual contributor may choose to publish an RFC detailing the new IPFIX application. Such an RFC should contain discussion of the new application, the Information Element definitions as in Section 4, as well as suggested Templates and examples of the use of those Templates within the new application as in Section 8.2. Section 9 defines a compact textual Information Element notation to be used in describing these suggested Templates and/or the use of IPFIX Structured Data *[I-D.ietf-ipfix-structured-data]* within the new application.

4. Defining new Information Elements

In many cases, a new application will require nothing more than a new Information Element or set of Information Elements to be exportable using IPFIX. An Information Element meeting the following criteria, as evaluated by appointed IPFIX experts, is eligible for inclusion in the Information Element registry:

- The Information Element MUST be sufficiently unique within the registry. A proposed Information Elements which is a substantial duplicate of an exiting Information Element is to be represented using the existing Element.

- The Information Element SHOULD contain minimal internal structure; complex information should be represented with multiple simple Information Elements to be exported in parallel, as in Section 4.5.

- The Information Element SHOULD be generally applicable to the application at hand, which SHOULD be of general interest to the community. Information Elements representing information about proprietary or nonstandard applications SHOULD be represented using enterprise-specific Information Elements as detailed in section 6.2 of [RFC5101].

The definition of new Information Elements requires a descriptive name, a specification of the data type as one from the IPFIX Data Type Registry, and a human-readable description written in English. This section provides guidelines on each of these components of an Information Element definition, referring to existing documentation such as [RFC5102] as appropriate.

4.1. Information Element naming

Information Element Names should be defined in accordance with section 2.3 of [RFC5102]; the most important naming conventions are repeated here for convenience.

*Names of Information Elements should be descriptive.

*Names of Information Elements MUST be unique within the IPFIX information model.

*Names of Information Elements start with non-capitalized letters.

*Composed names use capital letters for the first letter of each component (except for the first one). All other letters are non-capitalized, even for acronyms. Exceptions are made for acronyms containing non-capitalized letter, such as 'IPv4' and 'IPv6'. Examples are sourceMacAddress and destinationIPv4Address.

In addition, new Information Elements pertaining to a specific protocol SHOULD name the protocol in the first word in order to ease searching by name (e.g. "sipMethod" for a SIP method, as would be used in a logging format for SIP based on IPFIX). Similarly, new Information Elements pertaining to a specific application SHOULD name the application in the first word.

4.2. Information Element data types

IPFIX provides a set of data types covering most primitives used in network measurement and management applications. The most appropriate data type should be chosen for the Information Element type, out of the IPFIX informationElementDataTypes subregistry at [iana-ipfix-assignments].
Because IPFIX provides reduced-length encoding for Information Elements, unless an integral Information Element is derived from a fixed-width field in a measured protocol (e.g., tcpSequenceNumber, which is an unsigned32), it should be defined with the maximum possible width, generally signed64 or unsigned64. Applications can then choose to use reduced-size encoding as defined in Section 6.2 of [RFC5101] in cases where fewer than 2^64 values are necessary.
Information Elements representing time values should be exported with appropriate precision. For example, a Information Element for a time measured at second-level precision should be defined as having a dateTimeSeconds data type, instead of dateTimeMilliseconds.
The type of an Information Element MUST match the type of the data it represents. More specifically, information that could be represented as a String, but which better matches one of the other data types (e.g. an integral type for a number or enumerated type, an address type for an address) MUST be represented by the best-matching type, even if the data was represented using a different type in the source (i.e., an IPFIX application that exports Options Template Records mapping IP addresses to additional information about each host from an external database MUST use Information Elements of an address type to represent

the addresses, even if the source database represented these as strings.)
This document does NOT cover the addition of new Data Types or Data Type Semantics to the IPFIX Protocol. As such changes have important interoperability considerations and require implementation on both Collecting and Exporting Processes, they require a Standards Action as per [RFC5610]. However, note that the set of primitive types provided by IPFIX are applicable to most any appropriate application, so extending the type system is generally not necessary.

4.3. Information Element numbering

In general, when adding newly registered Information Elements to the registry, IANA SHOULD assign the lowest available Information Element identifier (the value column in [iana-ipfix-assignments] in the range 128-32767, noting that prior noncontiguous allocation may lead to unassigned Information Elements with lower Information Element identifiers than some presently assigned Information Elements. This is the case with the PSAMP Information Model *[RFC5477]*, which assigned a block of Information Elements identifiers starting at 300.
Information Element identifiers in the range 1-128 MUST NOT be assigned unless the Information Element is compatible with the NetFlow v9 protocol as described in [RFC3954]. Such Information Elements may ONLY be requested by a NetFlow v9 expert, to be designated by the IESG to consult with IANA on NetFlow v9 compatibility with IPFIX.

4.4. Ancillary Information Element properties

Information Elements to which special semantics apply SHOULD define these semantics with one of the values in the Information Element Semantics registry, as described in Section 3.2 of [RFC5102], subject to the restrictions given in Section 3.10 of [RFC5610]; essentially, the semantics and the type must be consistent.
When defining Information Elements representing a dimensioned quantity or entity count, the units of that quantity SHOULD be defined in the units field. This field takes its values from the IANA Information Element Units registry. If an Information Element expresses a quantity in units not yet in this registry, then the unit must be added to the Units registry at the same time the Information Element is added to the Information Element registry.
Additionally, when the range of values an Information Element can take is smaller than the range implied by its data type, the range SHOULD be defined within the Information Element registry.

4.5. Internal structure in Information Elements

The definition of Information Elements with internal structure with the structure defined in the Description field is discouraged, except in the following cases:

* The Information Element is a direct copy of a structured entity in a measured protocol (e.g. the tcpControlBits Information Element for the flags byte from the TCP header)

* The Information Element represents a section of a packet of protocol entity, in raw form as captured from the wire (e.g. the mplsLabelStackSection Information Element for the MPLS label stack)

* The Information Element represents a set of flags which are tightly semantically related, where representing the flags as separate one-byte booleans would be inefficient, and which should always appear together in a data record (e.g., the anonymizationFlags Information Element for specifying optional features of anonymization techniques)

In other cases, candidate Information Elements with internal structure SHOULD be decomposed into multiple primitive Information Elements to be used in parallel. For more complicated semantics, where the structure is not identical from Data Record to Data Record, or where there is semantic dependency between multiple decomposed primitive Information Elements, use the IPFIX Structured Data *[I-D.ietf-ipfix-structured-data]* extension instead.

As an example of information element decomposition, consider an application-level identifier called an "endpoint", which represents a {host, port, protocol} tuple. Instead of allocating an opaque, structured "source endpoint" Information Element, the source endpoint should be represented by three separate Information Elements: "source address", "source port", "transport protocol". In this example, the required information elements already exist in the Information Element registry: sourceIPv4Address or sourceIPv6Address, sourceTransportPort, protocolIdentifier. Indeed, as well as being good practice, this normalization down to non-structured Information Elements also increases opportunities for reuse as in Section 6.1.

The decomposition of data with internal structure SHOULD avoid the definition of Information Elements with a meaning too specific to be generally useful, or that would result in either the export of meaningless data or a multitude of templates to handle different multiplicities. More information on multiplicities is given in the following section.

4.6. Information Element multiplicity

Some Information Elements may represent information with a multiplicity other than one; i.e., items that may occur multiple times within the data to be represented in a single IPFIX record. In this case, there are several options, depending on the circumstances:

*As specified in section 8 of [RFC5101]: "if an Information Element is required more than once in a Template, the different occurrences of this Information Element SHOULD follow the logical order of their treatments by the Metering Process." In other words, in cases where the items have a natural order (e.g., the order in which they occur in the packet), and the multiplicity is the same for each record, the information can be modeled by containing multiple instances of the Information Element representing a single item within the Template Record describing the Data Records.

*In cases where the items have a variable multiplicity, a basicList of the Information Element representing a single item can be used as in the IPFIX Structured Data *[I-D.ietf-ipfix-structured-data]* extension.

*If the multiple-item structure is taken directly from bytes observed on the wire by the Metering Process or otherwise taken from the application being measured, the multiple-item structure can be exported as a variable-length octetArray Information Element holding the raw content.

Specifically, new Information Element SHOULD NOT encode any multiplicity or ordinality information into the definition of the Information Element itself.

4.7. Enumerated Values and Subregistries

When defining an Information Element that takes an enumerated value from a set of values which may change in the future, this enumeration MUST be defined by an IANA registry or subregistry. For situations where an existing registry defines the enumeration (e.g., the IANA Protocol Numbers registry for the protocolIdentifier Information Element), that registry MUST be used. Otherwise, a new IPFIX subregistry must be defined for the enumerated value, to be modified subject to Expert Review *[RFC5226]*.

4.8. Reversibility as per RFC 5103

[RFC5103] defines a method for exporting bidirectional flows using a special Private Enterprise Number to define reverse-direction variants of IANA Information Elements, and a set of criteria for determining whether an Information Element may be reversed using this method. Since almost all Information Elements are reversible, [RFC5103] enumerates those which Information Elements which were defined at the time of its publication which are NOT reversible.
New non-reversible Information Elements SHOULD contain a note in the description stating that they are not reversible.

4.9. Promotion of Enterprise-Specific Information Elements

Some Information Elements may start their lifecycle outside the IANA registry as enterprise-specific Information Elements scoped to a Private Enterprise Number. One stated goal of enterprise-specific Information Elements is pre-standards product delivery and experimentation; should these experiments be successful and the Information Elements generally useful, these SHOULD subsequently registered with IANA.

In order to support transition from experimental registration to IANA registration, the IANA registry provides an optional "enterprise-specific IE reference" column for each Information Element. In cases of promoted enterprise-specific Information Elements, this column in the registry SHOULD contain the private enterprise and Information Element numbers of the enterprise-specific version of the Information Element.

4.10. Avoiding Bad Ideas in Information Element Design

In general, the existence of a similarly-defined Information Element in the IANA registry sets a precedent which may be followed to determine whether a given proposed Information Element "fits" within the registry. Indeed, the rules specified by this document could be interpreted to mean "make new Information Elements that look like existing Information Elements". However, for reasons of history, there are several Information Elements within the IANA registry which do not follow best practices in Information Element design, and should be explicitly ignored when looking for guidance as to whether a new Information Element should be added.

Before registering a new Information Element, it must be determined that it would be sufficiently unique within the registry. This evaluation has not always been done in the past, and the existence of the Information Elements defined without this evaluation should not be taken as an example that such Information Element definition practices should be followed in the future. Specific examples of such Information Elements include initiatorOctets and responderOctets (which duplicate octetDeltaCount and its reverse per [RFC5103]) and initiatorPackets and responderPackets (the same, for packetDeltaCount).

As mentioned in Section 4.2, the type of an Information Element SHOULD match the type of data the Information Element represents. An example of how not to do this is presented by the p2pTechnology, tunnelTechnology, and encryptedTechnology Information Elements: these represent a three-state enumeration using a String. The example set by these Information Elements SHOULD NOT be followed in the definition of new Information Elements.

As mentioned in Section 4.6, an Information Element definition SHOULD NOT include any ordinality or multiplicity information. The only example of this within the IANA registry the following list of assigned IPFIX Information Elements: mplsTopLabelStackSection, mplsLabelStackSection2, mplsLabelStackSection3, mplsLabelStackSection4,

mplsLabelStackSection5, mplsLabelStackSection6 mplsLabelStackSection7, mplsLabelStackSection8, mplsLabelStackSection9, and mplsLabelStackSection10. The only distinction between those almost-identical Information Elements is the position within the MPLS stack. This Information Element design pattern met an early requirement of the definition of IPFIX which was not carried forward into the final specification -- namely, that no semantic dependency was allowed between Information Elements in the same Record -- and as such SHOULD NOT be followed in the definition of new Information Elements. In this case, since the size of the MPLS stack will vary from flow to flow, it should be exported using IPFIX Structured Data *[I-D.ietf-ipfix-structured-data]* where supported, as a basicList of MPLS label entries, or as a raw MPLS label stack using the variable-length mplsLabelStackSection Information Element.

5. The Information Element Lifecycle

Once an Information Element or set of Information Elements has been identified for a given application, Information Element specifications in accordance with Section 4 are submitted to IANA to follow the IE-DOCTORS process, as defined below. This process is also used for other changes to the registry, such as deprecation or revision, as described later in this section.

5.1. The IE-DOCTORS process

Requests to change the IANA Information Element registry or a linked subregistry are submitted to IANA, which forwards the request to a designated group of experts (IE-DOCTORS) appointed by the IETF Operations Area Directors. This group of experts reviews the request for compliance with this document, compliance with other applicable IPFIX-related RFCs, and consistency with the currently defined set of Information Elements.
IE-DOCTORS reviewers should endeavor to complete referred reviews in a timely manner. If the request is acceptable, the IE-DOCTORS signify their approval to IANA, which changes the IANA Information Element registry. If the request is not acceptable, the IE-DOCTORS can coordinate with the requestor to change the request to be compliant. The IE-DOCTORS may also choose in exceptional circumstances to reject clearly frivolous or inappropriate change requests outright.

5.2. Revising Information Elements

The Information Element status field in the Information Element Registry is defined in [RFC5102] to allow Information Elements to be 'current', 'deprecated' or 'obsolete'. No Information Elements are as of this writing deprecated or obsolete, and [RFC5102] does not define any policy for using them. Additionally, no policy is defined for revising Information Element registry entries or addressing errors

therein. To be certain, changes and deprecations within the Information Element registry are not encouraged, and should be avoided to the extent possible. However, in recognition that change is inevitable, this section is intended to remedy this situation.

The primary requirement in the definition of a policy for managing changes to existing Information Elements is avoidance of interoperability problems; IPFIX experts appointed to review changes to the Information Element Registry MUST work to maintain interoperability above all else. Changes to Information Elements already in use may only be done in an interoperable way; necessary changes which cannot be done in a way to allow interoperability with unchanged implementations MUST result in deprecation.

A change to an Information Element is held to be interoperable only when:

- it involves the correction of an error which is obviously only editorial; or
- it corrects an ambiguity in the Information Element's definition, which itself leads to non-interoperability (e.g., a prior change to ipv6ExtensionHeaders); or
- it expands the Information Element's data type without changing how it is represented (e.g., changing unsigned32 to unsigned64, as with a prior change to selectorId); or
- it defines a previously undefined or reserved enumerated value, or one or more previously reserved bits in an Information Element with flag semantics; or
- it expands the set of permissible values in the Information Element's range; or
- it harmonizes with an external reference which was itself corrected.

A non-interoperable Information Element change may also be made if it can be reasonably assumed in the eyes of the appointed experts that no unchanged implementation of the Information Element exists; this can be held to happen if a non-interoperable change to an Information Element defined shortly before is proposed to the IPFIX mailing list by the original proposer of the Information Element, and no objection is raised within a reasonable amount of time, to be defined by the expert reviewers.

If a change is permissible, it is sent to IANA, which passes it to the appointed experts for review; if there is no objection to the change from any appointed expert, IANA makes the change in the Information Element Registry. The requestor of the change is appended to the Requestor in the registry.

Each Information Element in the IANA registry has a revision number, starting at zero. Each change to an Information Element following this process increments the revision number by one. Since any revision must be interoperable according to the criteria above, there is no need for the IANA registry to store information about old revisions.

5.3. Deprecating Information Elements

Changes that are not permissible by these criteria may only be handled by deprecation. An Information Element MAY be deprecated and replaced when:

- the Information Element definition has an error or shortcoming which cannot be permissibly changed as above; or
- the deprecation harmonizes with an external reference which was itself deprecated through that reference's accepted deprecation method; or
- changes in the IPFIX Protocol or its extensions, or in community understanding thereof, allow the information represented by the Information Element to be represented in a more efficient or convenient way. Deprecation in this circumstance additionally requires the assent of the IPFIX Working Group, and should be specified in the Internet Draft(s) defining the protocol change.

A request for deprecation is sent to IANA, which passes it to the IE-DOCTORS for review, as above. When deprecating an Information Element, the Information Element description MUST be updated to explain the deprecation, as well as to refer to any new Information Elements created to replace the deprecated Information Element. The revision number of an Information Element is incremented upon deprecation. Deprecated Information Elements SHOULD continue to be supported by Collecting Processes, but SHOULD NOT be exported by Exporting Processes. The use of deprecated Information Elements SHOULD result in a log entry or human-readable warning at the Exporting and Collecting Processes. After a period of time determined in the eyes of the IE-DOCTORS experts to be reasonable in order to allow deployed Exporting Processes to be updated to account for the deprecation, a deprecated Information Element may be made obsolete. Obsolete Information Elements MUST NOT be supported by either Exporting or Collecting Processes. The receipt of obsolete Information Elements SHOULD be logged by the Collecting Process.
Names of deprecated Information Elements MUST NOT be reused. Names of obsolete Information Elements MAY be reused, but this is NOT RECOMMENDED, as it may cause confusion among users.

5.4. Versioning the entire IANA Registry

Consider a typical Collector implementation, which regularly downloads the entire registry in order to be compliant with the latest of set of supported IEs. While a registry revision number might seems advantageous for the Collector at first glance (avoiding the one by one comparison of all IE revisions), it is not necessary, as the IPFIX IANA registry specifies the date at which the registry was last updated in the "Last Updated" field. For purposes of identifying the latest set of Information Element versions specified in registry, the last revision date of the Information Element registry (available in the registry XML source, or from the Last-Modified: header of [iana-ipfix-assignments]) SHOULD be used.

6. When not to define new Information Elements

Also important in defining new applications is avoiding redundancy and clutter in the Information Element registry. Here we provide guidelines for reuse of existing Information Elements, as well as guidelines on using enterprise-specific Information Elements instead of adding Information Elements in the registry.

6.1. Maximizing reuse of existing Information Elements

Whenever possible, new applications should prefer usage of existing IPFIX Information Elements to the creation of new Information Elements. IPFIX already provides Information Elements for every common Layer 4 and Layer 3 packet header field in the IETF protocol suite, basic Layer 2 information, basic counters, timestamps and time ranges, and so on. When defining a new Information Element similar to an existing one, reviewers shall ensure that the existing one is not applicable.
Note that this guideline to maximize reuse does not imply that an Information Element that represents the same information from a packet as a existing Information Element should not be added to the registry. For example, consider the ipClassOfService (Element ID 5), ipDiffServCodePoint (Element ID 98), and ipPrecedence (Element ID 196) Information Elements. These all represent subsets of the same field in an IP version 4 packet header, but different uses of these bits. The representation in one or another of these Information Elements contains information in itself as to how the bits were interpreted by the Metering Process.
On the other hand, simply changing the context in which an Information Element will be used is insufficient reason for the definition of a new Information Element. For example, an extension of IPFIX to log detailed information about HTTP transactions alongside network-level information should not define httpClientAddress and httpServerAddress Information Elements, preferring instead the use of sourceIPv[46]Address and destinationIPv[46]Address.

Applications dealing with bidirectional interactions should use Bidirectional Flow Support for IPFIX *[RFC5103]* to represent these interactions.
Specifically, existing timestamp and time range Information Elements should be reused for any situation requiring simple time stamping of an event: for single observations, the observationTime* Information Elements from PSAMP are provided, and for events with a duration, the flowStart* and flowEnd* Information Elements suffice. This arrangement allows minimal generic time handling by existing Collecting Processes and analysis workflows. New timestamp Information Elements should ONLY be defined for semantically distinct timing information (e.g., an IPFIX-exported record containing information about an event to be scheduled in the future).
In all cases the use of absolute timestamp Information Elements (e.g. flowStartMilliseconds) is RECOMMENDED, as these Information Elements allow for maximum flexibility in processing with minimal overhead. Timestamps based on the export time header in the enclosing IPFIX Message (e.g. flowStartTimeDeltaMicroseconds) MAY be used if high-precision timing is important, export bandwidth or storage space is limited, timestamps comprise a relatively large fraction of record size, and the application naturally groups records into IPFIX Messages. Timestamps based on information which must be exported in a separate Data Record defined by an Options Template (e.g. flowStartSysUpTime) MAY be used only in the context of an existing practice of using runtime-defined epochs for the given application. New applications SHOULD avoid these structures when possible.

6.2. Applying enterprise-specific Information Elements

IPFIX provides a mechanism for defining enterprise-specific Infomation Elements, as in Section 3.2 of [RFC5101]. These are scoped to a vendor's or organization's Structure of Management Information (SMI) Private Enterprise Number, and are under complete control of the organization assigning them.
For situations in which interoperability is unimportant, new information SHOULD be exported using enterprise-specific Information Elements instead of adding new Information Elements to the registry. These situations include:

- *export of implementation-specific information, or
- *export of information derived in a commercially-sensitive or proprietary method, or
- *export of information or meta-information specific to a commercially-sensitive or proprietary application.

While work within the IETF generally does not fall into these categories, enterprise-specific Information Elements are also useful

for pre-standardization testing of a new IPFIX application. While performing initial development and interoperability testing of a new application, the Information Elements used by the application SHOULD NOT be submitted to IANA for inclusion in the registry. Instead, these experimental Information Elements SHOULD be represented as enterprise-specific until their definitions are finalized, then transitioned from enterprise-specific to IANA-defined upon finalization. To support this transition, the IANA registry provides an experimental IE reference as defined in Section 4.9.

7. Applying IPFIX to non-Flow Applications

At the core of IPFIX is its definition of a Flow, a set of packets sharing some common properties crossing an observation point within a certain time window. However, the reliance on this definition does not preclude the application of IPFIX to domains which are not obviously handling flow data according to it. Most network management data collection tasks, those to which IPFIX is most applicable, have at their core the movement of packets from one place to another; by a liberal interpretation of the common properties defining the flow, then, almost any event handled by these can be held to concern data records conforming to the IPFIX definition of a Flow.
Non-flow information defining associations or key-value pairs, on the other hand, are defined by IPFIX Options Templates. Here, the Information Elements within an Options Template Record are divided into Scope Information Elements which define the key, and non-scope Informatin Elements which define the values associated with that key. Unlike Flows, Data Records defined by Options Template are not necessarily scoped in time; these Data Records are generally held to be in effect until a new set of values for a specific set of keys is exported. While this mechanism is often used by IPFIX to export metadata about the collection infrastructure, it is applicable to any association information.
An IPFIX application can mix Data Records described either type of template in an IPFIX Message or Message stream, and exploit relationships among the Flow Keys, values, and Scopes to create interrelated data structures. See [RFC5473] for an example application of this.

8. Writing Internet-Drafts for IPFIX Applications

When a new application is complex enough to require additional clarification or specification as to the use of the defined Information Elements, this may be given in an Internet-Draft. Internet-Drafts for new IPFIX applications are best submitted to a Working Group with expertise in the area of the new application, or as independent submissions.
When defining new Information Elements in an Internet-Draft, the Internet-Draft SHOULD contain a section (or subsection) for each

Information Element, which contains the attributes in Section 4 in human-readable form. An example subsection is given below. These Information Element descriptions SHOULD NOT assign Information Element numbers, instead using placeholder identifiers for these numbers (e.g. "AAA", "BBB", "CCC", or "TBD1", "TBD2", "TBD3") and a note to IANA in the IANA Considerations section to replace those placeholders in the document with Information Element numbers when the numbers are assigned. The use of these placeholder definitions allows references to the numbers in e.g. box-and-line diagrams or template definitions as in Section 9.

8.1. Example Information Element Definition

This is an example of an Information Element definition which would appear in an Internet-Draft. The name appears in the section title.

Description: Description goes here.

Data Type: Data type goes here; obligatory

Data Type Semantics: Data type semantics, if any, go here; optional

Units: Units, if any, go here; optional

Range: Range, if not implied by the data type, goes here; optional

References: References to other RFCs or documents outside the IETF, in which additional information is given, or which are referenced by the description, go here; optional

ElementId: TBD1

8.2. Defining Recommended Templates

New IPFIX applications SHOULD NOT, in the general case, define fixed templates for export, as this throws away much of the flexibility afforded by IPFIX. However, fixed template export is permissible in the case that the export implementation must operate in a resource constrained environment, and/or that the application is replacing an existing fixed-format binary export format in a maximally compatible way. In any case, Collecting Processes for such applications SHOULD support reordered Templates or Templates with additional Information Elements.

An Internet-Draft clarifying the use of new Information Elements SHOULD include any recommended Template or Options Template Records necessary for supporting the application, as well as examples of records exported using these Template Records. In defining these Template Records, such Internet-Drafts SHOULD mention, subject to rare exceptions as above:

*that the order of Information Elements within a Template is not significant;

*that Templates on the wire for the application may also contain additional Information Elements beyond those specified in the recommended Template;

*that a stream of IPFIX Messages supporting the application may also contain Data Records not described by the recommended Templates; and

*that any reader of IPFIX Messages supporting the application MUST accept these conditions.

Definitions of recommended Template Records for flow-like information, where the Flow Key is well-defined, SHOULD indicate which of the Information Elements in the recommended Template are Flow Keys. Recommended Templates are defined, for example, in [RFC5476] for PSAMP packet reports (section 6.4) and extended packet reports (section 6.5). Recommended Options Templates are defined extensively throughout the IPFIX documents, including in the protocol document itself *[RFC5101]* for exporting export statistics; in the file format *[RFC5655]* for exporting file metadata; and in Mediator intermediate process definitions such as [I-D.ietf-ipfix-anon] for intermediate process metadata. The discussion in these examples is a good model for recommended template definitions.

9. A Textual Format for Specifying Information Elements and Templates

The examples given above are all expressed using bitmap diagrams of the respective Templates. These are illustrative of the wire representation of simple Templates, but not particularly readable for more complicated recommended Templates, provide no support for rapid implementation of new Templates, and do not adequately convey the optional nature of ordering and additional Information Elements as above. Therefore, we define a RECOMMENDED textual format for specifying Information Elements and Templates in Internet-Drafts in this section.
Here we define a simple textual syntax for describing IPFIX Information Elements and IPFIX Templates, with human readability, human writability, compactness, and ease of parser/generator implementation without requiring external XML support as design goals. It is intended both for use in human communication (e.g., in new Internet-Drafts containing higher-level descriptions of IPFIX Templates, or describing sets of new IPFIX Information Elements for supporting new applications of the protocol) as well as at runtime by IPFIX implementations.

9.1. Information Element Specifiers

The basis of this format is the textual Information Element Specifier, or IESpec. An IESpec contains each of the four important aspects of an Information Element: its name, its number, its type, and its size, separated by simple markup based on various types of brackets. Fully-qualified IESpecs may be used to specify existing or new Information Elements within an Information Model, while either fully-qualified or partial IESpecs may be used to define fields in a Template.
Bare words are used for Information Element names, and each aspect of information associated with an Information Element is associated with a type of brackets:

*() parentheses for Information Element numbers,

*<> angles for Information Element data types, and

*[] square brackets for Information Element sizes.

*{} curly braces contain an optional space-separated list of context identifiers to be associated with an Information Element, as described in more detail in Section 9.2

The symbol + is reserved for Information Element nesting within structured data elements; these are described in and Section 9.3, respectively.
Whitespace in IESpecs is insignificant; spaces can be added after each element in order, e.g., to align columns for better readability.
The basic form of a fully-qualified IESpec for an IANA-registered Information Element is as follows:
name(number)<type>[size]
where 'name' is the name of the Information Element in UTF-8, 'number' is the Information Element as a decimal integer, 'type' is the name of the data type as in the IANA informationElementDataTypes registry, and 'size' is the length of the Information Element in octets as a decimal integer, where 65535 or the string 'v' signifies a variable-length Information Element. [size] may be omitted; in this case, the data type's native or default size is assumed.
The basic form of a fully-qualified IESpec for an enterprise-specific Information Element is as follows:
name(pen/number)<type>[size]
where 'pen' is the Private Enterprise Number as a decimal integer.
A fully-qualified IESpec is intended to express enough information about an Information Element to decode and display Data Records defined by Templates containing that Information Element. Range, unit, semantic, and description information, as in [RFC5610], is not supported by this syntax.
Example fully-qualified IESpecs follow:

*octetDeltaCount(1)<unsigned64>[8]

*octetDeltaCount(1)<unsigned64> (unsigned64 is natively 8 octets long)

*sourceIPv4Address(8)<ipv4Address>

*wlanSSID(146)<string>[v]

*sipRequestURI(35566/403)<string>[65535]

A partial IESpec is any IESpec that is not fully-qualified; these are useful when defining templates. A partial IESpec is assumed to take missing values from its canonical definition, for example, the IANA registry. At minimum, a partial IESpec must contain a name, or a number. Any name, number, or type information given with a partial IESpec must match the values given in the Information Model; however, size information in a partial IESpec overrides size information in the Information Model; in this way, IESpecs can be used to express reduced-length encoding for Information Elements.
Example partial IESpecs follow:

*octetDeltaCount

*octetDeltaCount[4] (reduced-length encoding)

*(1)

*(1)[4] (reduced length encoding; note that this is exactly equivalent to an Information Element specifier in a Template)

9.2. Specifying Templates

A Template can then be defined simply as an ordered, newline-separated sequence of IESpecs. IESpecs in example Templates illustrating a new application of IPFIX SHOULD be fully-qualified. Flow Keys may be optionally annotated by appending the {key} context to the end of each Flow Key specifier. A template counting packets and octets per five-tuple with millisecond precision in IESpec syntax is shown below.
An Options Template is specified similarly. Scope is specified appending the {scope} context to the end of each IESpec for a Scope IE. Due to the way Information Elements are represented in Options Templates, all {scope} IESpecs must appear before any non-scope IESpec. The Flow Key Options Template defined in section 4.4 of [RFC5101] in IESpec syntax is shown below:

9.3. Specifying IPFIX Structured Data

IESpecs can also be used to illustrate the structure of the information exported using the IPFIX Structured Data extension *[I-D.ietf-ipfix-structured-data]*. Here, the semantics of the structured data elements are specified using contexts, and the information elements within each structured data element follow the structured data element, prefixed with + to show they are contained therein. Arbitrary nesting of structured data elements is possible by using multiple + signs in the prefix. For example, a basic list of IP addresses with "one or more" semantics would be expressed using parially qualified IESpecs as follows:
And an example subTemplateList itself containing a basicList is shown below:
This describes a subTemplateMultilist containing all of the expressed set of source-destination pairs, where the source address itself could be one of any number in a basicList (e.g., in the case of SCTP multihoming).
The contexts associable with structured data Information Elements are the semantics, as defined in section 4.4 of [I-D.ietf-ipfix-structured-data]; a structured data Information Element without any context is taken to have undefined semantics. More information on the application of structured data is available in [I-D.ietf-ipfix-structured-data].

10. Security Considerations

The security aspects of new Information Elements must be considered in order not to give a potential attacker too much information. For example, the "A Framework for Packet Selection and Reporting" [RFC5474] concluded in section 12.3.2 that the hash functions private parameters should not exported within IPFIX.
If some security considerations are specific to an Information Element, they MUST be mentioned in the Information Element description. For example, the ipHeaderPacketSection in the IPFIX registry mentions: "This Information Element, which may have a variable length, carries a series of octets from the start of the IP header of a sampled packet. With sufficient length, this element also reports octets from the IP payload, subject to [RFC2804]. See the Security Considerations section."
These security considerations MAY also be stressed in an accompanying Internet-Draft, as in Section 8. For example, the "Packet Sampling (PSAMP) Protocols Specification" [RFC5476] specifies: "In the basic Packet Report, a PSAMP Device exports some number of contiguous bytes from the start of the packet, including the packet header (which includes link layer, network layer and other encapsulation headers) and some subsequent bytes of the packet payload. The PSAMP Device SHOULD NOT export the full payload of conversations, as this would mean wiretapping [RFC2804]. The PSAMP Device MUST respect local privacy laws."

11. IANA Considerations

With respect to the management of the IPFIX Information Element registry and associated subregistries located at [iana-ipfix-assignments], this document defines a process for IANA in Section 5.1, and includes a set of guidelines for IANA for applying this process in Section 4, Section 5, and Section 6.
In addition, in order to support more effective management of the Information Element lifecycle as defined in Section 5, it specifies the addition of three new columns for this registry:

Revision: a serial revision number for each Information Element, beginning at 0 for all presently existing and newly created Information Elements.

Date: the date at which the Information Element was created or last modified.

Enterprise-specific reference: for Information Elements which where deployed as enterprise-specific Information Elements for experimentation and testing, and subsequently registered in the IANA registry, specifies the private enterprise number (PEN) and IE number of the equivalent experimental IE.

12. Acknowledgements

The authors would like to acknowledge the FP7 PRISM and DEMONS projects for their material support of this work.

13. References

13.1. Normative References

[RFC3954]	Claise, B., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, October 2004.
[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.
[RFC5103]	Trammell, B. and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)", RFC 5103, January 2008.
[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.
[RFC2119]	

	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC5226]	Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

13.2. Informative References

[RFC2804]	IABIESG, "IETF Policy on Wiretapping", RFC 2804, May 2000.
[RFC3917]	Quittek, J., Zseby, T., Claise, B. and S. Zander, "Requirements for IP Flow Information Export (IPFIX)", RFC 3917, October 2004.
[RFC4181]	Heard, C., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, September 2005.
[RFC5153]	Boschi, E., Mark, L., Quittek, J., Stiemerling, M. and P. Aitken, "IP Flow Information Export (IPFIX) Implementation Guidelines", RFC 5153, April 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.
[RFC5474]	Duffield, N., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M. and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, March 2009.
[RFC5476]	Claise, B., Johnson, A. and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, March 2009.
[RFC5477]	Dietz, T., Claise, B., Aitken, P., Dressler, F. and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, March 2009.
[RFC5655]	Trammell, B., Boschi, E., Mark, L., Zseby, T. and A. Wagner, "Specification of the IP Flow Information Export (IPFIX) File Format", RFC 5655, October 2009.

[I-D.ietf-ipfix-structured-data]	Claise, B, Dhandapani, G, Yates, S and P Aitken, "Export of Structured Data in IPFIX", Internet-Draft draft-ietf-ipfix-structured-data-06, May 2011.
[I-D.ietf-ipfix-anon]	Boschi, E and B Trammell, "IP Flow Anonymization Support", Internet-Draft draft-ietf-ipfix-anon-06, January 2011.
[iana-ipfix-assignments]	Internet Assigned Numbers Authority, , "IP Flow Information Export Information Elements (http://www.iana.org/assignments/ipfix/ipfix.xml)", .

Authors' Addresses

Brian Trammell Trammell Swiss Federal Institute of Technology Zurich Gloriastrasse 35 8092 Zurich, Switzerland Phone: +41 44 632 70 13 EMail: trammell@tik.ee.ethz.ch

Benoit Claise Claise Cisco Systems, Inc. De Kleetlaan 6a b1 1831 Diagem, Belgium Phone: +32 2 704 5622 EMail: bclaise@cisco.com