

Text Representation for Abstract Data Types

B. Trammell

IPFIX, IETF 86 Orlando, Thursday 14 March 2013

The Idea

- IPFIX information model independent from IPFIX protocol [RFC5102bis]
- IPFIX encodings [RFC5101bis] not appropriate for textual representations (e.g. JSON)
- All we need to extend the applicability of the information model are data type encoding mappings
- This document aims to provide them
(*work in progress*)

The Principle of Least Surprise

- If the Enclosing Context (i.e., underlying textual representation) has a native representation for a given type, use it.
- Otherwise, use common representations
- Strings escaped according to Enclosing Context's rules

Proposed Encodings

- Signed: decimal
- Unsigned: decimal or hex
 - (...or codepoint name)
- Boolean: [0FfNn] vs. [lTtYy] (*not i/8n-friendly*)
- Timestamps: ISO 8601 via RFC 3339.
- Addresses: dotted quad, MAC, RFC 5952
- Lists: aren't real ADTs, just support 63 | 3

A JSON Example

```
{  
  "flowStartMilliseconds": "2012-11-05 18:31:01.135",  
  "flowEndMilliseconds": "2012-11-05 18:31:02.880",  
  "octetDeltaCount": 195383,  
  "packetDeltaCount": 88,  
  "sourceIPv6Address": "2001:db8:c:1337::2",  
  "destinationIPv6Address": "2001:db8:c:1337::3",  
  "sourceTransportPort": 80,  
  "destinationTransportPort": 32991,  
  "protocolIdentifier": "tcp",  
  "tcpControlBits": 19,  
  "flowEndReason": 3  
}
```

Open Issues and Next Steps

- Is there interest in this work?
- How to represent floats?
- Finish document (more examples, bindings)
- Next revision for Berlin