

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: July 10, 2016

A. Pelov, Ed.  
Acklio  
M. Veillette, Ed.  
Trilliant Networks Inc.  
Carsten Bormann  
Universitaet Bremen TZI  
Laurent Toutain  
Telecom Bretagne

January 07, 2016

## **CBOR Order-preserving Multimap - CoMAP**

### **draft-pelov-core-comap-latest**

### **Abstract**

This document describes a data structure for the Concise Binary Object Representation (CBOR) [RFC7049], which represents an ordered collection of pairs of data - CBOR Order-preserving Multimap (CoMAP). There is no native support for this type of collections neither in JSON, nor in CBOR. The current standards have two basic collection representations, namely arrays and maps. Maps provide a mapping from unique keys to values, where the order of the elements is not preserved. This document describes a data structure and its CBOR encoding which unambiguously represents key-value pairs, where the keys can be repeated and their order is preserved. This representation is based on arrays and allows native support in CBOR with efficient structure representation.

### **Status of This Memo**

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on July 10, 2016.

## Copyright Notice

Copyright (c) 2016 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.

---

## 1. Introduction

JSON has been modeled around the notions of JavaScript, with a primary goal of being simple to parse and portable across web browsers. This helped its popularity, becoming one of the de-facto standards for data marshaling across browsers. The Constrained Binary Object Representation [RFC7049], matching closely the notions formalized by JSON, has become the default data representation format for constrained devices.

However, the simplicity of the basic data types of JSON and CBOR requires the ad-hoc specification of many useful structures, where the applications by convention need to know how to interpret the structures under question. One crucial data structure missing from the default JSON/CBOR toolkit is an order-preserving map, which allows key repetition.

This document defines a way to implement the data structure of particular importance to constrained devices: order-preserving multimaps (also known as deterministic multimaps or collections of ordered pairs). The goals of CoMAPs are the following:

- No modification to CBOR. Use existing data types as building blocks.
- Provide a specification how these data types must be interpreted.
- Use an efficient representation of the structure.

CBOR Order-preserving Multimaps (CoMAPs) are implemented as CBOR Arrays, tagged with the CoMAP Tag, and interpreted as explained further in this document.

The CoMAP implementation defined in this document allows for sequential encoding/decoding of the information, e.g. it is not necessary for a sending/receiving node to have the entire CBOR object before being able to process it.

## 2. Terminology and Notation

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. Basic CBOR container for CoMAP

A collection of pairs, where the first element is the key, and the second -- the corresponding value, and in which the order of the pairs is important (e.g. should be preserved) is called an order-preserving multimap. This document defines a canonical representation of order-preserving multimaps as CBOR arrays. This structure is called CoMAP.

A CoMAP is implemented as an Array with the CoMAP CBOR Tag (IANA Action). Optionally, if the MIME type of a CoAP request or response MAY indicate that the payload is a CoMAP entity. In this case, if the first CBOR element is an Array, with no other CBOR Tags, it is considered to be a CoMAP.

The following CBOR Tag is necessary for the correct interpretation of a CoMAP:

- *CoMAP* CBOR tag (e.g. value number 6 – note requires IANA action)

### 3.1. CoMAP diagnostic notation

The following notation is used throughout this document for denoting a CoMAP:

```
<< key : value >>
```

Where the CoMAP starts with `<<` and ends with `>>`, and the key-value pairs within are separated with colon. Key-value pairs are separated by comma:

```
<< key1 : value1, key2 : value2, key3 : value3 >>
```

Note that **key1**, **key2** and **key3** can be identical. Also, their order is preserved in a CoMAP, e.g. it is guaranteed that if written in that order, they will be read in the same order.

## 4. CoMAP representation

A CoMAP is a CBOR Array with the *CoMAP tag*. The elements of the Array encode keys, values or special elements, which represent embedded CoMAPs, or multi-value representations. The CoMAP Array is decoded sequentially, starting from its first element. Any element may represent a KEY, a VALUE, or have an interpretation that will produce several KEY:VALUE pairs. This depends on its position in the sequential interpretation of the CoMAP.

The general structure is the following:

```
[key1, value1, key2, value2, key3, value3]
```

If the element to be processed SHOULD be a KEY, but is a CBOR Array, then it has a special interpretation:

```
[key1, value1, ARRAY, . . .]
```

Depending on the size of the Array, it could have three interpretations:

1. Empty array (size 0): No-operation (skip this element)
2. Array of size 1: Its element is the KEY. Used for having simple Arrays as KEYS. The next element is the value
3. Array of size 2: Multiply the first element to the second one. May produce several KEY:VALUE pairs. For example, if the first element is an Array of 3 elements (A, B, C), and the second element is an Array of 6 elements (1, 2, 3, 4, 5, 6), this will produce 6 KEY-VALUE pairs: A:1, B:2, C:3, A:4, B:5, C:6.
  - a. If one of the two elements is not an Array, its value is multiplied by the other Array, e.g. [1, [1,2,3,4,5]] will produce 5 KEY:VALUE pairs: 1:1, 1:2, 1:3, 1:4, 1:5.
4. Array of size bigger or equal to 3: An embedded CoMAP. The first element is the KEY, and the remaining define the CoMAP, which is the VALUE

This is the algorithm, which is used to decode a CoMAP:

1. If no more elements in **Array A**
  - a. Decoding complete with success
  - b. END**
2. **a1** := Get next element from **Array A**
3. If **a1** is an **Array**
  - a. If **sizeof(a1) == 0**
    - i. No-operation. Skip this element.
    - ii. **GOTO 1**
  - b. If **sizeof(a1) > 2**
    - i. **KEY** := The first element of **a1**
    - ii. **VALUE** := Remaining of **a1**, decoded as CoMAP
    - iii. Produce pair: **KEY:VALUE**
    - iv. **GOTO 1**
  - c. If **sizeof(a1) == 2**
    - i. **X** := The first element of **a1**
    - ii. **Y** := The second element of **a1**
    - iii. If **X** is not an **Array** AND **Y** not an **Array**
      1. **KEY** := **X**
      2. **VALUE** := **Y**
      3. Produce pair: **KEY:VALUE**
      4. **GOTO 1**
    - iv. If **X** is not an **Array**
      1. **KEY** := **X**
      2. For every element **y1** in **Array Y**
        - a. **VALUE** := **y1**
        - b. Produce pair: **KEY:VALUE**
      3. **GOTO 1**
    - v. If **Y** is not an **Array**
      1. **VALUE** := **Y**
      2. For every element **x1** in **Array X**
        - a. **KEY** := **x1**
        - b. Produce pair: **KEY:VALUE**
      3. **GOTO 1**
    - vi. While there are more elements in **Array Y**
      1. For every element **x1** of **X**
        - a. If no more elements in **Array Y**
          - i. **GOTO 1 (beginning of algorithm)**
        - b. **y1** := Get next element from **Array Y**
        - c. **KEY** := **x1**
        - d. **VALUE** := **y1**
        - e. Produce pair: **KEY:VALUE**
    - vii. **GOTO 1**
  - d. If **sizeof(E1) == 1**
    - i. **E1** := The first element of **Array E1**
4. If no more elements in **Array A**

- a. Decoding complete with warning (no KEY-VALUE pair produced in this round)
  - b. **END**
5. **E2** := Get next element from **Array A**
6. **KEY** := **E1**
7. **VALUE** := **E2**
8. Produce pair: **KEY:VALUE**
9. **GOTO 1**

The algorithm is executed sequentially until the end of the Array. If the Array terminates unexpectedly (e.g. after a KEY), the last KEY MUST be ignored.

A CoMAP can contain any type as a KEY or as a VALUE. If an ordinary Array (e.g. without any CBOR Tag) has to be used as a KEY, it MUST be encapsulated in an Array of size 1.

In particular, CoMAPs can be used as KEYS (e.g. an Array with the CoMAP Tag), or as a VALUE. In the latter case, the CoMAP must be embedded.

## 4. CoMAP examples

This section provides several illustrative examples in CoMAP diagnostic notation, the corresponding CBOR diagnostic notation, and the final CBOR encoding.

<b>CoMAP diagnostic notation</b>	<code>&lt;&lt; "key1" : "value1" &gt;&gt;</code>
<b>CBOR diagnostic notation</b>	<code>6(["key1", "value1"])</code>

<b>CoMAP diagnostic notation</b>	<code>&lt;&lt;"key1": "value1", "key2": "value2", "key3": "value3"&gt;&gt;</code>
<b>CBOR diagnostic notation</b>	<code>6(["key1", "value1", "key2", "value2", "key3", "value3"])</code>

<b>CoMAP</b>	<code>&lt;&lt;1: "one", 1: "un", 1: "uno", 2: "two", 2: "deux",</code>
--------------	--

<b>diagnostic notation</b>	2:"due">>
<b>CBOR diagnostic notation</b>	6([1,"one",1,"un",1,"uno",2,"two",2,"deux",2,"due"])

<b>CoMAP diagnostic notation</b>	<<1: ["one","un","uno"], 2: ["two","deux","due"]>>
<b>CBOR diagnostic notation</b>	6([1, ["one","un","uno"], 2, ["two","deux","due"] ])

<b>CoMAP diagnostic notation</b>	<<1:"one", 2:"two", 3:"three", 1:"un", 2:"deux", 3:"trois", 1:"uno", 2:"due", 3:"tre">>
<b>CBOR diagnostic notation</b>	6([ [1,2,3], ["one","two","three","un","deux","trois","uno","due","tre"] ])

<b>CoMAP diagnostic notation</b>	<<1:"one", 1:"un", 1:"uno", 2:"two", 2:"deux", 2:"due", 3:"three", 3:"trois", 3:"tre">>
<b>CBOR diagnostic notation</b>	6([ [1, ["one","un","uno"]], [2, ["two","deux","due"]], [3, ["three","trois","tre"]] ])



<b>CoMAP diagnostic notation</b>	<pre>&lt;&lt; 1: &lt;&lt; "en":"one", "fr":"un", "it":"uno" &gt;&gt;, 2: &lt;&lt;"en":"two", "fr":"deux", "it":"due"&gt;&gt; &gt;&gt;</pre>
<b>CBOR diagnostic notation</b>	<pre>6([ [1,"en","one", "fr","un", "it","uno"], [2,"en","two", "fr","deux", "it","due"] ])</pre>

<b>CoMAP diagnostic notation</b>	<pre>&lt;&lt; ["key1a", "key1b"]: ["value1a", "value1b"] &gt;&gt;</pre>
<b>CBOR diagnostic notation</b>	<pre>6([ [ ["key1a", "key1b"]], ["value1a", "value1b"] ])</pre>

In the following example, the KEY is ["rema", "ining"] and the VALUE:  
[1,2,3]

<b>CoMAP diagnostic notation</b>	<pre>&lt;&lt; ["rema", "ining"]: [1,2,3] &gt;&gt;</pre>
<b>CBOR diagnostic</b>	<pre>6([ [ ["rema", "ining"]], [1,2,3] ])</pre>

<b>notation</b>	
-----------------	--

<b>CoMAP diagnostic notation</b>	<pre>&lt;&lt; ["Bart","Simpson"]: &lt;&lt;"age":8,"height":4.0,"weight":85 &gt;&gt;,  ["Homer","Simpson"]: &lt;&lt;"age":39,"height":5.11, "weight":239 &gt;&gt;  &gt;&gt;</pre>
<b>CBOR diagnostic notation</b>	<pre>6([  [ ["Bart","Simpson"], "age",8,"height",4.0,"weight",85 ],  [ ["Homer","Simpson"], "age",39,"height",5.11,"weight",239 ]  ])</pre>

<b>CoMAP diagnostic notation</b>	<pre>&lt;&lt;  1: &lt;&lt; "en":"one", "fr":"un", "it":"uno" &gt;&gt;,  2: &lt;&lt;"en":"two", "fr":"deux", "it":"due"&gt;&gt;  &gt;&gt;</pre>
<b>CBOR diagnostic notation</b>	<pre>6([  [1,"en","one", "fr","un", "it","uno"],  [2,"en","two", "fr","deux", "it","due"],  ])</pre>

<b>CoMAP diagnostic notation</b>	<pre> &lt;&lt;   &lt;&lt; "en":"one", "fr":"un", "it":"uno" &gt;&gt; :     &lt;&lt; 1:"1", 1:"1.0", 1:"0.9999" &gt;&gt;,   &lt;&lt;"en":"two", "fr":"deux", "it":"due"&gt;&gt; :     &lt;&lt; 2:"2", 2:"2.0", 2:"1.9999" &gt;&gt; &gt;&gt; </pre>
<b>CBOR diagnostic notation</b>	<pre> 6([   [ 6(["en","one", "fr","un", "it","uno"]),     1,"1", 1,"1.0", 1, "0.9999"],   [ 6(["en","two", "fr","deux", "it","due"]),     2,"2", 2,"2.0", 2,"1.9999"] ]) </pre>

## 8. Security Considerations

## 9. IANA Considerations

The allocation of a new CBOR Tag is required for CoMAPs:

CoMAP – CBOR Tag ID 6

