

Workgroup: anima Working Group  
Internet-Draft:  
draft-richardson-cbor-file-magic-00  
Published: 20 January 2021  
Intended Status: Standards Track  
Expires: 24 July 2021  
Authors: M. Richardson

Sandelman Software Works

## **On storing CBOR encoded items on stable storage**

### **Abstract**

This document proposes an on-disk format for CBOR objects that is friendly to common on-disk recognition systems like the Unix file(1) command.

### **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 <https://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 24 July 2021.

### **Copyright Notice**

Copyright (c) 2021 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 (<https://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](#)
- [2. Requirements for a Magic Number](#)
- [3. Proposal One](#)
- [4. Proposal Two](#)
- [5. Variations](#)
  - [5.1. Use a CBOR Tag on the entire file](#)
  - [5.2. Use a CBOR Tag on the CBOR Integer](#)
  - [5.3. Use a CBOR Tag on a constant CBOR Integer](#)
- [6. The Magic Number Registry](#)
- [7. Security Considerations](#)
- [8. IANA Considerations](#)
- [9. Acknowledgements](#)
- [10. Changelog](#)
- [11. References](#)
  - [11.1. Normative References](#)
  - [11.2. Informative References](#)
- [Author's Address](#)

## 1. Introduction

Since very early in computing, operating systems have sought ways to mark which files could be proposed by which programs.

For instance, the Unix `file(1)` command, which has existed since 1973 ([\[file\]](#)), has been able to identify many file formats for decades. Many systems (Linux, MacOS, Windows) will select the correct application based upon the file contents, if the system can determine it by other means. (MacOS maintains a resource fork that includes MIME information)

While having a MIME type associated with the file is a better solution in general, when files become disconnected from their type information, such as when attempting to do forensics on a damaged system, then being able to identify a file type can become very important.

It is noted that in the MIME type registration, that a magic number is asked for, if available.

A challenge for this program is often that it can be confused by the encoding vs the content. For instance, an Android "apk" used to transfer and store an application may be identified as a ZIP file.

As CBOR becomes a more and more common encoding for artifacts, identifying them as CBOR is probably not useful. This document provides a way to encode a magic number into the beginning of a CBOR format file. Two options are presented, with the intention of standardizing only one.

These proposals are invasive to how CBOR protocols are written to disk, but in both cases, the proposed envelope does not require that the tag be transferred on the wire.

Some protocols may benefit from having such a magic on the wire if they presently using a different (legacy) encoding scheme, and need to determine before invoking a CBOR decoder if the sender is using the legacy scheme, or the new CBOR scheme.

## **2. Requirements for a Magic Number**

A magic number is ideally a unique fingerprint, present in the first 4 or 8 bytes of the file, which does not change when the content change, and does not depend upon the length of the file.

Less ideal solutions have a pattern that needs to be matched, but in which some bytes need to be ignored.

## **3. Proposal One**

This proposal uses a CBOR Array of size two. The first byte is therefore 0b100\_00010 (0x82).

Array element number one is a CBOR integer in the range 0x80000000 to 0xffffffff. This number is the magic number described below in [Section 6](#)

For a magic number 0x87654321, this results in a total of a six byte sequence:

```
0b100_00010 0b000_11010 0x87 0x65 0x43 0x21
```

Array element number two is whatever the original CBOR content is supposed to be. Due the array construct with known size, there is no further syntax required.

## **4. Proposal Two**

This proposal uses a CBOR Sequence [[RFC8742](#)].

Array element number one is a CBOR integer in the range 0x80000000 to 0xffffffff. This number is the magic number described below in [Section 6](#)

For a magic number 0x87653412, this results in a total of a five byte sequence:

```
0b000_11010 0x87 0x65 0x34 0x12
```

This is followed by one or more CBOR data items of whatever type was intended.

## **5. Variations**

There are four variations.

### **5.1. Use a CBOR Tag on the entire file**

A two byte CBOR Tag could be used in proposal one to the array. This would add two bytes, bring the total flag bytes up to eight. The two byte sequence would have to start with 0b110\_11000, followed by a one byte tag value, followed by the array as described above.

### **5.2. Use a CBOR Tag on the CBOR Integer**

A two or three byte CBOR Tag could be used in proposal two, applied to the CBOR Integer.

Or, a two byte CBOR Tag could be used in proposal one, applied to the CBOR Integer, and not applied to the array. This would make the first four bytes of a CBOR encoded item recognizeably CBOR, with the next four bytes being the specific CBOR content.

### **5.3. Use a CBOR Tag on a constant CBOR Integer**

Instead of creating a new namespace (and IANA registry) for magic numbers, the CBOR Tag registry (which is very large) could be used. Rather than using the integer as the magic number, the Tag would be the magic number. Since the tag has to tag something, it could be some constant value could be tagged: a CBOR Null, or perhaps the CBOR string "cbor".

## **6. The Magic Number Registry**

In order to maintain uniqueness an IANA registry is required for the Magic Numbers.

These Magic numbers would be 4-byte numbers in a First Come/First Served registry. Applicants would be encouraged to make a selection, and it would be encouraged to make the magic number a bit descriptive in ASCII. As a historic example, the IFF ILBM [[ilbm](#)] had a formatID whose bytes were: "ILBM", or 0x49 0x4C 0x42 0x4D.

In the case where the CBOR Tag registry is used, then there are two options:

1. allow requesters to select their own four (32-bit) or eight (64-bit) tags, from the First Come First Served Registry, using the existing instructions.

2. amend the IANA instructions for [RFC8949] and carve out a 30-bit chunk of the four byte registry, or a 32-bit chunk of the eight byte registry.

While in many cases CBOR encodings strive to be as compact as possible, for the purposes of a magic number registry for objects stored on disk, the use of between eight and twelve bytes is acceptable.

## 7. Security Considerations

ZZZ

## 8. IANA Considerations

TBD

## 9. Acknowledgements

Hello.

## 10. Changelog

## 11. References

### 11.1. Normative References

- [BCP14] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8742] Bormann, C., "Concise Binary Object Representation (CBOR) Sequences", RFC 8742, DOI 10.17487/RFC8742, February 2020, <<https://www.rfc-editor.org/info/rfc8742>>.
- [RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <<https://www.rfc-editor.org/info/rfc8949>>.

### 11.2. Informative References

- [file] Wikipedia, "file (command)", 20 January 2021, <[https://en.wikipedia.org/wiki/File\\_%28command%29](https://en.wikipedia.org/wiki/File_%28command%29)>.
- [ilbm] Wikipedia, "Interleaved BitMap", 20 January 2021, <<https://en.wikipedia.org/wiki/ILBM>>.

**Author's Address**

Michael Richardson  
Sandelman Software Works

Email: [mcr+ietf@sandelman.ca](mailto:mcr+ietf@sandelman.ca)