INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2014/M34358
July 2014, Sapporo, Japan

Source	**Telecom ParisTech**
Status	**For consideration at the 109th MPEG Meeting**
Title	**On Movie Fragment Relative Addressing**
Author	Cyril Concolato, Jean Le Feuvre

1. Introduction

In the context of the W3C Media Source Extensions specification, it was proposed (https://www.w3.org/Bugs/Public/show_bug.cgi?id=24345) to relax some rules on MSE regarding ISOBMFF segments and the default-base-is-moof flag. In trying to understand what was meant and its consequences, we discovered that definition of movie fragments in the ISOBMFF specification was not clear and could be improved. This contribution brings editorial improvements.

2. Recap of the algorithm

In this section, we formalize the algorithm used to merge fragmented data.

Input:

- tfhd defines 2 flags:
    - default-base-is-moof (`dbim`)
    - base-data-offset-present (`bdop`)
    - an optional field base-data-offset (`bdov`)
- trun defines 1 flag:
    - data-offset-present (`dop`)
    - and one optional field data-offset (`do`).

Purpose:

A ISOBMFF fragment player needs to compute the byte position (called offset) in the **file** (considering the possibly virtual file made of the concatenation of all segments) of the first byte of a run number n, of a track fragment number j in the movie fragment, i.e.:

```
run(j, n, 0) = offset;
```

The algorithm (used in MP4Box.js) to compute `offset` is as follows:

```
if (j==0 && n == 0) {
  if (dop) {
    offset = bdo + do; // If the data-offset is present, it is
                       // relative to the base-data-offset
                       // established in the track
                       // fragment header
  } else {
    offset = bdo; // the data for this run starts the
                  // base-data-offset defined by the
                  // track fragment header
} else {
 offset = run(j, n-1, end) + 1; // this run starts
                                // immediately after the
                                // data of the previous run
}
```

and `bdo` is computed as follows:

```
if (!bdop) {
  if (!dbim) {
    if (k == 0) { // the first track in the movie fragment
      bdo = moof (0); // the position of the first byte of the
                      // enclosing Movie Fragment Box
    } else {
      bdo = run(j-1, last, end) + 1;// end of the data
                                    // defined by the
                                    // preceding track
                                    // (irrespective of the
                                    // track id) fragment in
                                    // the moof
    }
  } else {
    bdo = moof(0);
  }
} else {
  bdo = bdov;
}
```

3. Text problems and proposal

The problems in the current specification text are:

- The `dbim` semantics defines what happens "if base-data-offset-present is zero" but not if `bdop` is 1. We think it should say it is ignored, the value given in the box has precedence.
- It does not say how to imbricate the ifs between `dbim` and `dbop`. We propose to use the above algorithm, which is based on the fact that the `dbop` semantics existed before `dbim`.
- The term "the default" used in the definiton of `bdop` is ambiguous.
- The term "fragment" in the definition of `bdop` is meant to be "track fragment" (since it could be confused with "`movie fragment`")

Aside from that, the DASH spec says:

"The ‘moof’ boxes shall use movie-fragment relative addressing for media data that does not use external data references, the flag ‘default-base-is-moof’ shall be set, and data-offset shall be used, i.e. base-data-offset-present shall not be used."

The term "movie-fragment relative addressing" is not defined. The MSE used to use that term, and that's what they are trying to fix. This term should be defined in the ISOBMFF specification and then references should be made from the DASH specification and the MSE ISOBMFF byte stream specification.