YABE

YABE is **Y**et **A**nother **B**inary **E**ncoding proposal as simple as JSON, but with the addition of a binary blob data value (i.e. Images) and the benefit of marshalling binary data.

Author: Christophe Meessen
Date : 19 sep 2012

Value types

Atomic data values :

1. *null* ;
2. **Boolean** : *true* & *false* ;
3. **Integer** : 64 bit integer ;
4. **Floating point** : 64 bit IEEE 754-2008 ;
5. **String** : utf8 encoded character sequence ;
6. **Blob** : array of raw bytes with a mime type string ;

Composed values :

1. **Array** : Sequence of any values ;
2. **Object** : Sequence of a pair of value identifier (string) and any value ;

Value Encoding

Each value is encoded as a tag byte identifying its type, followed by an optional value size and the value itself. When possible the size or the value are stored in the tag.

```
[tag]([size])([value])
```

```
| Value  |    Tag    | arguments          | comment
-------------------------------------------------------------------
  0..127 : [0xxxxxxx]                    : integer value 0..127
  str6   : [10xxxxxx] [byte]*            : utf8 char string
  null   : [11000000]                    : null value
  int16  : [11000001] [int16]            : 16 bit integer
  int32  : [11000010] [int32]            : 32 bit integer
  int64  : [11000011] [int64]            : 64 bit integer
  flt0   : [11000100]                    : 0. float value
  flt16  : [11000101] [flt16]            : 16 bit float
  flt32  : [11000110] [flt32]            : 32 bit float
  flt64  : [11000111] [flt64]            : 64 bit float
  true   : [11001000]                    : boolean true value
  false  : [11001001]                    : boolean false value
  blob   : [11001010] [string] [string] : mime typed byte array
  ends   : [11001011]                    : equivalent to ] or }
  none   : [11001100]                    : tag byte to be ignored
  str16  : [11001101] [len16] [byte]*    : utf8 char string
  str32  : [11001110] [len32] [byte]*    : utf8 char string
  str64  : [11001111] [len64] [byte]*    : utf8 char string
  arrayn : [11010xxx] [value]*           : 0 to 6 value array
  array  : [11010111] [value]*           : equivalent to [
  objectn: [11011xxx] [string, value]*   : 0 to 6 value object
  object : [11011111] [string, value]*   : equivalent to {
  -1..-32: [111xxxxx]                    : integer value -1..-32
```

- The tag is a one byte value ;
- Integer values from -32 to 127 are encoded as is as the tag value ;
- Integer values are encoded as little endian signed integer ;
- Floating point values are encoded in the IEEE 754-2008 format ;
- A strings is a sequence of utf8 encoded chars with the number of bytes as length ;
- A length value are encoded as little endian unsigned integer of 16, 32 or 64 bits ;
- A blob is a pair of strings, the first is a mime type and the second is a sequence or raw bytes ;
- An Array is encoded as a stream of values ;
- An Object is encoded as a stream of string and value pairs where the string is a unique identifier ;
- An Object may not have an empty string as identifier ;
- A array of object stream is ended by the *ends* tag ;
- If an array or an object have lest than 7 items, the *arrayn* or *objectn* encoding may be used where the number of items is encoded in the tag ;

YABE data Signature

Stored or transmitted YABE encoded data starts with a five byte signature. The first four bytes are the ASCII code 'Y', 'A', 'B', 'E' in that order, and the fifth byte is the version number of the encoding. This specification describes the encoding version 0.

YABE data size

The encoded data byte length is defined by the context (i.e. file or record size) or is implicit if the data is limited to one value like an array or an object.

File extension and mime type

A file name with extention .yabe and starting with the YABE tag contains YABE encoded data. The mime type is "application/yabe" (to be registered).