followed by the three 8-bit integer values: 0x01 0x02 0x03, for a grand total of 4 bytes: 0x31010203.

Suppose we are at a site with many alternative alleles so AC=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]. Since there are 16 values, we have to use the long vector encoding. The type of this field is 8 bit integer with the size set to 15 to indicate that the size is the next stream value, so this has type of 0xF1. The next value in the stream is the size, as a typed 8-bit atomic integer: 0x11 with value 16 0x10. Each integer AC value is represented by it's value as a 8 bit integer. The grand total representation here is:

0xF1 0x01 0x10	8 bit integer vector with overflow size
0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10	1–16 as hexadecimal 8 bit integers

Suppose this INFO field contains the "AC=.", indicating that the AC field is missing from a record with two alt alleles. The correct representation is as the typed pair of AC followed by a MISSING vector of type 8-bit integer: 0x01.

Vectors of mixed length — In some cases genotype fields may be vectors whose length differs among samples. For example, some CNV call sets encode different numbers of genotype likelihoods for each sample, given the large number of potential copy number states, rather padding all samples to have the same number of fields. For example, one sample could have CN0:0,CN1:10 and another CN0:0,CN1:10,CN2:10. In the situation when a genotype field contain vector values of different lengths, these are represented in BCF2 by a vector of the maximum length per sample, with all values in the each vector aligned to the left, and END_OF_VECTOR values assigned to all values not present in the original vector. The BCF2 encoder / decoder must automatically add and remove these END_OF_VECTOR values from the vectors. Note that the use of END_OF_VECTOR means that it is legal to encode a vector VCF field with MISSING values.

For example, suppose I have two samples, each with a FORMAT field X. Sample A has values [1], while sample B has [2,3]. In BCF2 this would be encoded as [1, END_OF_VECTOR] and [2, 3]. Diving into the complete details, suppose X is at offset 3 in the dictionary, which is encoded by the typed INT8 descriptor 0x11 followed by the value 0x03. Next we have the type of the each format field, which here is a 2 element INT8 vector: 0x21. Next we have the encoding for each sample, A = 0x01 0x81 followed by B = 0x02 0x03. All together we have:

0x11 0x03	X dictionary offset
0x21	each value is a 2 element INT8 value
0x01 0x81	A is [1, END_OF_VECTOR]
0x02 0x03	B is [2, 3]

A **Genotype (GT) field** is encoded in a typed integer vector (can be 8, 16, or even 32 bit if necessary) with the number of elements equal to the maximum ploidy among all samples at a site. For one individual, each integer in the vector is organized as $(allele + 1) << 1 \mid phased$ where allele is set to -1 if the allele in GT is a dot '.' (thus the higher bits are all 0). The vector is padded with the END_OF_VECTOR values if the GT having fewer ploidy. We note specifically that except for the END_OF_VECTOR byte, no other negative values are allowed in the GT array.

Examples:

0/1	in standard format $(0 + 1) << 1 \mid 0$ followed by $(1 + 1) << 1 \mid 0$	0x02 04
0/1, 1/1, and 0/0	three samples encoded consecutively	0x02 04 04 04 02 02
0 | 1	$(1 + 1) << 1 \mid 1 = 0x05$ preceded by the standard first byte value 0x02	0x02 05
./.	where both alleles are missing	0x00 00
0	as a haploid it is represented by a single byte	0x02
1	as a haploid it is represented by a single byte	0x04
0/1/2	is ~~tetraploid~~triploid, with alleles	0x02 04 06
0/1 | 2	is ~~tetraploid~~ triploid with a single phased allele	0x02 04 07
0 and 0/1	pad out the final allele for the haploid individual	0x02 81 02 04

The final example is something seen on chrX when we have a haploid male and a diploid female. The male genotype vector is terminated prematurely by the END_OF_VECTOR value.