

## Annex E

## Support for Huffman encodings

- E.1** Huffman encodings are the optimum encodings for a finite set of integer values, where the frequency with which each value will be transmitted is known.
- E.2** The encodings are self-delimiting (no length-determinant is needed) and use a small number of bits for frequent values and a larger number of bits for less frequent values.
- E.3** There are many possible Huffman encodings. For example, given any such encoding, simply change all "1"s to "0"s and vice versa, and you have a different (but just as efficient) Huffman encoding. More subtle changes can also be made to produce other Huffman encodings that are equally efficient.
- E.4** For Huffman encodings to be efficient for decoders, it is desirable that where successive integer values encode into the same number of bits, those bits should define successive integer values when interpreted as a positive integer encoding.
- E.5** An ECN Huffman encoding has been defined that has this property, and a Microsoft Word 97 macro has been produced that will generate the syntax for a "MappingIntToBits" mapping (see 19.7) which is both optimal and easy to decode.
- E.6** A version of this annex is available which contains a macro button that will take a specification of the integer values to be encoded and their frequency, and will generate in-line the formal mapping specification conforming to the ECN notation. (The version of this Annex with the associated macro is freely available from ITU website at <http://www.itu.int/ITU-T/publications/recs.html> under X.692 Recommendation, and from ISO website at [http://www.iso.ch/iso/en/ittf/PubliclyAvailableStandards/c034390\\_ISO\\_8825-3\\_2003\(E\)\\_Annex\\_E.zip](http://www.iso.ch/iso/en/ittf/PubliclyAvailableStandards/c034390_ISO_8825-3_2003(E)_Annex_E.zip)).
- E.7** The following text contains three examples of ECN Huffman specification.
- E.8** In the version with the macro, double clicking the button below:

**ECN Huffman**

will add the ECN Huffman mapping specifications to the text.

- E.9** The user of the version with the macro may wish to modify the specification of the values to be mapped and their frequencies to see the encodings that are produced in different cases.
- NOTE – In the version with macros, once encoding specifications have been produced, they can be deleted, the ECN Huffman specification changed, and the macro button again clicked.
- E.10** The informal syntax for an ECN Huffman specification should be clear from the following examples. All lines start with an ASN.1 comment marker ("--").
- E.11** The first line (if the macro is to be used) must contain exactly "ECN Huffman" preceded by two hyphens and a space, but following lines are not case sensitive and may contain more or less spaces.
- E.12** The second line is required, and specifies the lowest and highest values that are to be mapped. The range (upper bound minus lower bound) is limited to 1000, but can include negative values. Not all values in the range need to be mapped.
- E.13** Percentages are given for either single values or for ranges of values. It is not necessary for percentages to add up to 100%, but a warning is given if they do not.
- E.14** The "REST" line is optional, and provides frequencies for any values in the range not explicitly listed. If missing, then the mapped values will only be those explicitly specified.
- E.15** The final line is mandatory, and must contain "End Definition" (in upper or lower case). The formal ECN encoding specification is inserted (by the macro) after this line.

E.15.1 The first example is:

```
my-int-encoding1 #My-Special-1 ::=
{ USE #BITS
  -- ECN Huffman
  -- RANGE (-1..10)
  -- -1 IS 20%
  -- 1 IS 25%
  -- 0 IS 15%
  -- (3..6) IS 10%
  -- Rest IS 2%
  -- End Definition
```

E.15.2 The second example is:

```
my-int-encoding2 #My-Special-2 ::=
{ USE #BITS
  -- ECN Huffman
  -- RANGE (-10..10)
  -- -10 IS 20%
  -- 1 IS 25%
  -- 5 IS 15%
  -- (7..10) is 10%
  -- End Definition
```

E.15.3 The third example is:

```
my-int-encoding3 #My-Special-3 ::=
{ USE #BITS
  -- ECN Huffman
  -- RANGE (0..1000)
  -- (0..63) IS 100%
  -- REST IS 0%
  -- End Definition
```