Templates

Local definitions described here refer to ECMWF local extensions to GRIB version 1, section 1.

For example, the template for local definition 1 is:

```
col 1
-----
  |
  v

  !
  !   localDefinitionTemplate_001
  !   ---------------------------
  !
  !Description                Octet   Code    Ksec1   Count
  !-----------                -----   ----    -----   -----
  !
  localDefinitionNumber       41      I1      37      -
  class                       42      I1      38      -
  type                        43      I1      39      -
  stream                      44      I2      40      -
  experimentVersionNumber     46      A4      41      -
  number                      50      I1      42      -
  total                       51      I1      43      -
  spareSetToZero              52      PAD     n/a     1
```

Lines beginning with ! are comments.

Description is a single joined_up 'word'; capital letters are used to aid readability.

Octet is the byte number in the encoded GRIB product section 1 (numbering starts at 1, a WMO convention).

Code describes the format of the encoded bytes.

Ksec1 numbering is that of a Fortran array element used by gribex.

Count may be used as a repetition factor for a coded item.

n/a means 'not applicable'.

Formats

Format	Usage
A1	A 1-byte ASCII value
A4	A 4-byte ASCII value
BYTES	A string of bytes. May be controlled by 'count' or by summing a number of byte strings
D3	A 3-byte date modified by 19000000 if necessary; see definitions 6 and 17
ENDIF	Terminates an IF_EQ, IF_GT or IF_NEQ sequence
ENDLIST	Completes a LIST which started in an earlier 'description' field. Matches the 'description' given in the 'count' field

F1	A 1-byte integer flag
I1	A 1-byte integer
I2	A 2-byte integer
I3	A 3-byte integer
I4	A 4-byte integer
IF_EQ	Tests the value held in an earlier 'description' in the template is equal to the value in 'Ksec1'. The earlier 'description' is given in the 'count' field.
IF_GT	Tests the value held in an earlier 'description' in the template is greater than the value in 'Ksec1'. The earlier 'description' is given in the 'count' field.
IF_NEQ	Tests the value held in an earlier 'description' in the template is not equal to the value in 'Ksec1'. The earlier 'description' is given in the 'count' field.
LIST	Starts a repeated sequence of template lines. The sequence may be repeated, and the repeat count is given in an earlier 'description' in the template. The earlier 'description' is given in the 'count' field. A LIST is terminated by ENDLIST.
LOCAL	Defines a local definition number to be expanded into a template. As used, for example, in ECMWF local definition 192.
LP_I1	Defines a list of single 1-byte integers. The repeat count is given in an earlier 'description' in the template. The earlier 'description' is given in the 'count' field.
LP_I2	Defines a list of single 2-byte integers. The repeat count is given in an earlier 'description' in the template. The earlier 'description' is given in the 'count' field.
LP_I3	Defines a list of single 3-byte integers. The repeat count is given in an earlier 'description' in the template. The earlier 'description' is given in the 'count' field.
LP_I4	Defines a list of single 4-byte integers. The repeat count is given in an earlier 'description' in the template. The earlier 'description' is given in the 'count' field.
LP_I4M1	Defines a list of single 4-byte integers. The repeat count is one less than the number given in an earlier 'description' in the template. The earlier 'description' is given in the 'count' field.
PAD	Defines a sequence of zero bytes for packing or zero values for unpacking or both, upto the value given in the 'count' field.
PADFROM	Defines a sequence of zero bytes upto the value given in the 'count' field starting from the byte given in 'Ksec1'.
PADMULT	Defines a sequence of zero bytes upto the value given in the 'count' field.
PADTO	Defines a sequence of zero bytes upto the value given in the 'octet' field.
S1	A signed 1-byte integer. When packed, a negative value is stored as the corresponding positive value with the most significant bit set to 1.
S2	A signed 2-byte integer. When packed, a negative value is stored as the corresponding positive value with the most significant bit set to 1.
S3	A signed 3-byte integer. When packed, a negative value is stored as the corresponding positive value with the most significant bit set to 1.
S4	A signed 4-byte integer. When packed, a negative value is stored as the corresponding positive value with the most significant bit set to 1.
SP_TO	Defines a sequence of ASCII spaces upto the octet value given in the 'octet' field.