

MRMS Gridded Binary Format

7/18/2013

(Updated 4/11/2017)

The MRMS binary format consists of a simple header followed by a set of 2D or 3D data. The following documents the header and data organization:

Header:

char = 1 byte

short int = 2 bytes

int = 4 bytes

type	byte #	value
int	1 – 4	4-digit year of valid time in UTC
int	5 – 8	month of valid time in UTC
int	9 – 12	day of valid time in UTC
int	13 – 16	hour of valid time in UTC
int	17 – 20	minute of valid time in UTC
int	21 – 24	second of valid time in UTC
int	25 – 28	number of data columns (<i>NX</i>)
int	29 – 32	number of data rows (<i>NY</i>)
int	33 – 36	number of data vertical levels (<i>NZ</i>); = 1 for 2D data
4 char	37 – 40	map projection (= "LL " always)
int	41 – 44	scale factor for map projection reference values (map_scale)
int	45 – 48	true latitude 1 for map projection (scaled using map_scale)
int	49 – 52	true latitude 2 for map projection (scaled using map_scale)
int	53 – 56	true longitude for map projection (scaled using map_scale)
int	57 – 60	longitude of the most north and west grid cell's center (scaled using map_scale). See Figure 1.
int	61 – 64	latitude of the most north and west grid cell's center (scaled using map_scale). See Figure 1.

int	65 – 68	deprecated scale factor (xy_scale)
int	69 – 72	grid cell size in degrees longitude (scaled using dxy_scale)
int	73 – 76	grid cell size in degrees latitude (scaled using dxy_scale)
int	77-80	scale factor for grid cell size (dxy_scale)
NZ int	81 – X X = [80+(NZ*4)]	height of each level in meters MSL (scaled using z_scale).
int	[X+1] – [X+4]	scale factor for storing height of vertical levels (z_scale)
10 int	[X+5] – [X+44]	placeholder reserved for future use
20 char	[X+45] – [X+64]	variable name
6 char	[X+65] – [X+70]	variable unit
int	[X+71] – [X+74]	variable scale used for scaling data values (var_scale)
int	[X+75] – [X+78]	missing data value, which isn't scaled
int	[X+79] – [X+82]	number of radars (NR) contributing directly to the data field. minimum value is one.
NR * 4 char	[X+83] – [X+82+NR*4]	list of radars (4-char call signs) contributing directly to the data field. If none are applicable. NR = 1 and the string "none" is stored.

The total length of the header should be $162 + 4 * (NZ + NR)$ bytes.

For a 2D product (NZ = 1) with no contributing radars listed, the header length is 170 bytes.

A 3D product (NZ = 33) with 40 contributing radars will have a header of 454 bytes.

Data:

The data values immediately follow the header and are stored as short int (2 bytes). The byte range should be ...

$$[X + 83 + NR*4] - [X + 82 + NR*4 + NX*NY*NZ*2]$$

The data are scaled via var_scale (see header). Data is packed by level and then by row (row-major) into a 1D array. To map from 1 to 3D, use the following equation:

$$1D_index = k * NX * NY + j * NX + i$$

where i, j, k -indices correspond to the x, y, z-directions.

The origin of the data is the lowest level of the southwest corner. As k increases, so does the height. As j increases, so does the latitude. As i increases, so does longitude.

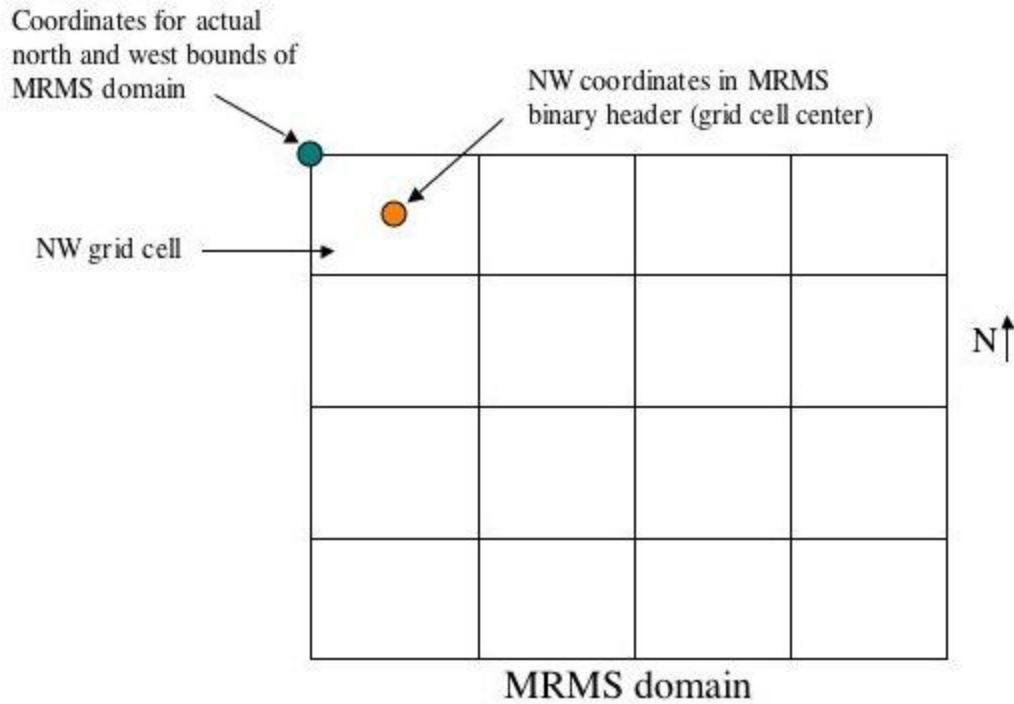


Figure 1. Illustrates values represented in bytes 57-60 and 61-64, which are the longitude/latitude coordinates (orange dot) for the northwest grid cell.