

Compressed Land Mask (clm) and Compressed Topographic Elevation (cte) File Formats and Associated Library

Background

The compressed land mask (clm) and compressed topographic elevation (cte) files are built from Shuttle Radar Topography Mission 3 (SRTM3), SRTM30, and GTOPO30 data files. In both cases the data has been compressed twice. First using bit packing or a combination of delta coding and bit packing and, second, using the zlib compression library. Additionally the cte files are stored using the huge_io functions. These are functions that allow the use of "files" larger than 2GB on both Linux/UNIX and Windows. The functions were developed prior to broad support for large files on Linux and prior to any large file support on Windows. The huge_io functions still have a couple of advantages over using large files. First, it's easier to distribute files on DVD and, second, since Microsoft doesn't follow the POSIX standards for large file support in their C compiler, you don't have to mod your code to work on different OSes. Note that huge_io files are actually directories containing a config file (ASCII file containing the total size of the aggregate huge_io file) and one or more approximately 2GB files with the name hugefile.nnn, where nnn is a number starting at 000 and going up to 032.

Compressed Land Mask (clm) File Format

The data in the compressed land mask (clm) file is stored as follows:

Header - 16384 bytes, ASCII

[HEADER SIZE] = 16384
[CREATION DATE] =
[VERSION] =
[ZLIB VERSION] =
[END OF HEADER]

One-degree map - 64800 * 4 bytes, binary, stored as characters.

Records start at 90S,180W and proceed west to east then south to north (that is, the second record is for 90S,179W and the 361st record is for 89S,180W). Record contains 0 if all water, 1 if all land, 2 if undefined, or address if both land and water.

Data - 1's and 0's

3 bits - resolution, 0 = one second mask, 1 = 3 second mask, 2 = 30 second mask, others TBD
29 bits - size of the zlib level 9 compressed block (SB)
SB bytes - data

The data is stored as a series of single bits for water (0) and land (1). Each bit represents a one second, three second, or thirty second cell in the block. The block is a one-degree square. It will be 3600 X 3600, 1200 X 1200, 120 X 120, or 60 X 60 depending on the resolution. It is ordered in the same fashion as the srtm3 data, that is, west to east starting in the northwest corner and moving southward. The compression is compliments of the ZLIB compression library which can be found at <http://www.zlib.net/>. Many thanks to Jean-loup Gailly, Mark Adler, and all others associated with that effort.

Note that since all of the data is stored as characters or unsigned characters there are no endian issues. These files work equally well on both big and little endian systems.

Compressed Topographic Elevation (cte) File Format

The data in the compressed topographic elevation (cte) files is stored as follows:

Header - 16384 bytes, ASCII

[HEADER SIZE] = 16384
[CREATION DATE] =
[VERSION] =
[ZLIB VERSION] =
[END OF HEADER]

One-degree map - 64800 * 36 bits, double precision integer, stored as characters.

Records start at 90S,180W and proceed west to east then south to north (that is, the second record is for 90S,179W and the 361st record is for 89S,180W). Record contains 0 if all water, 1 if all land, 2 if undefined, or address if both land and water.

Data -

3 bits - Not used (originally planned for resolution in mixed files)
30 bits - size of the zlib level 9 compressed block (SB)
31 bits - size of the uncompressed block
SB bytes - data

Inside the compressed block the data has already been delta coded and bit packed. The format of the data stored in the compressed block is as follows:

16 bits - signed short, starting value, stored as characters
16 bits - signed short, bias value, stored as characters
4 bits - number of bits used to store delta coding offsets (NB)
NB bits - first offset
NB bits - offset from first
NB bits - offset from second
.
.
.
NB bits - last offset (may be the 12,960,000th, 1,440,000th, or 14,400th offset depending on resolution)

Undefined values (-32768) will be stored as (int) pow (2.0L, NB) - 1 and will not be used in the delta chain.

The deltas are computed by subtracting the previous valid value from the current value. The

data is traversed west to east for one row, then east to west for the next row, then west to east, etc. I call this the snake dance (named after the lines at Disney World ;-). After the delta is computed the bias is added to it.

Again, there are no endian issues with this data.


```
        //DO SOMETHING WITH array[i * size + j]
    }
}
cleanup_srtm_mask ();
}
```

Note that read_srtm_topo, read_srtm1_topo, read_srtm3_topo, and read_srtm30_topo have not been thoroughly tested.

Jan C. Depner

25 October 2006

jan.depner@navy.mil