CERP Unstructured Grid NetCDF Metadata Conventions

Version 1.2, August 13, 2014

Abstract

This document describes the Comprehensive Everglades Restoration Plan (CERP) metadata conventions for unstructured grid (UG) NetCDF files. These conventions are designed to be used for data associated with the Florida Everglades region, and will facilitate data sharing between organizations involved in Everglades science.

The CERP UG metadata conventions are heavily based on the Climate and Forecast (CF) Metadata Conventions (Version 1.4). Files that comply with the CERP UG conventions will also be compliant with the CF 1.4 conventions. The CERP UG conventions add optional and mandatory attributes to further describe the NetCDF file, and also specify some optional CF attributes as mandatory. Though the CF 1.4 conventions were intended for use with structured grid metadata, the conventions are relaxed enough that, with slight modification, they may be applied to UG metadata as well.

The changes made from the CF 1.4 conventions have been informed by existing UG definitions, such as the Regional Simulation Model (RSM) developed by the South Florida Water Management District (SFWMD).

This document does not attempt to define an entirely new standard; it lists the changes made from the CF 1.4 conventions along with best practices. Therefore, the CERP UG metadata conventions are formally described as the CF 1.4 conventions together with the changes below, which take precedence over the CF 1.4 conventions when in conflict.

Construction and Storage

There are two main components that must be present for UG construction:

1. **Nodes** - the vertices that make up the cells, defined by an (x, y) coordinate
2. **Cells** - which nodes are grouped to form each convex polygon

To properly store these two components, three variables are introduced:

1. **Cell Map** - defines for each cell (convex polygon) in the UG an ID, and a mapping from each ID to a corresponding index in the **Connections** variable.
2. **Connections** - specifies for each cell in the UG, *n* (where *n >= 3)* indices into the **Locations** variable, each corresponding to one of the nodes that define the cell, and listed in a clockwise or counter-clockwise vertex winding order.
3. **Locations** - specifies for each node in the UG, two indices, one into an **x** and one into a **y** coordinate variable. These indices are used to access the actual node coordinates stored within **x** and **y**, with respect to a defined projection.

The flow from a particular cell to its individual point coordinates is:

Cell Map → Connections → Locations → (x, y)

1. Dimensions and Variables

1.1 Required Dimensions

The following dimensions are necessary to define the required variables (see Section 1.3):

- **nodes** - the number of points (vertices) in the UG
- **cells** - the number of convex polygon cells in the UG
- **two** - the number (2), for use in defining two-dimensional arrays
- **edges** - the number of edges the convex polygon cells will have
- **x -** the number of horizontal coordinate axis values
- **y -** the number of vertical coordinate axis values

1.2 Optional Dimensions

- **time** - the number of time steps included in the file. If not provided, the dataset is assumed to be atemporal. See [**data_variable**] in Section 1.3.

1.3 Required Variables

The following variables are required for compliance with the CERP UG metadata conventions:

- [data type] [**data_variable**] (cells) - the explicit value held by each cell in an *atemporal* dataset
  OR
  [data type] [**data_variable**] (time, cells) - the explicit value held by each cell at each time step, in a *temporal* dataset. Use of the optional **time** variable is assumed in this case.
    - Supported data types include `char`, `byte`, `short`, `int`, `float` or `real`, and `double`. See: CF 1.4 Section 2.2 - Data Types.
- int **cell_map** (cells, two) - defines for each cell an ID, and a mapping for each ID to a corresponding index in **connections**
- int **connections** (cells, edges) - specifies for each cell, *n* (where *n >= 3* and is defined by the dimension **edges***)* indices into **locations**, each corresponding to one of the nodes that defines the cell. The indices should be specified in a clockwise or counter-clockwise vertex winding order.
- int **locations** (nodes, two) - specifies for each node, two indices, one into **x** and one into **y**.
- double **x** (x) - the sorted x coordinates used for nodes in the UG, with respect to a [**projection_variable**]
- double **y** (y) - the sorted y coordinates used for nodes in the UG, with respect to a [**projection_variable**]
- int [**projection_variable**] - the grid mapping (projection) of a [**data_variable**], necessary for visualization of the data in a spatial context

1.4 Optional Variables

- int **time** (time) - the actual time steps corresponding to the data in a [**data_variable**], with respect to the time stamp defined in the **units** attribute. See Section 2.2 (b).

2. Attributes

2.1 All Coordinate and Data Variables

a. The **long_name** attribute is mandatory for all coordinate and data variables. The **long_name** attribute will give a concise description of the variable, and is used as the primary means of displaying the data type in the viewing applications such as data plots and legends. Example:

```
float depth(time, y, x) ;
      depth:long_name = "water depth" ;
```

b. The **units** attribute is mandatory for all coordinate and data variables. The NetCDF library uses the UDUNITS package for units parsing. When possible, the **units** attribute for variables should use a unit recognized by this package. A list of units recognized by the UDUNITS package can be found at: http://www.unidata.ucar.edu/software/udunits/udunits-1/udunits.txt.

```
double y(y) ;
      y:units = "km" ;

double lon(y, x) ;
      lon:units = "degrees_east" ;

double Temperature(time, cells) ;
      Temperature:units = "K" ;
```

   i. This list may not cover everything, such as dimensionless units. In the case of dimensionless units, use a numerical representation of the units when possible, i.e. "1/10e6" for "parts per million".

2.2 Coordinate Variables

a. The **standard_name** attribute is mandatory for all coordinate variables. The **standard_name** attribute is used to determine the specific axis of the variable as it pertains to the coordinate system. The standard names for Cartesian projected coordinate systems are "projected_x_coordinate" and "projected_y_coordinate". The standard names for geographic coordinate systems are "latitude" and "longitude". The standard names for the rotated pole projection are "grid_latitude" and "grid_longitude".

```
double y(y) ;
      y:standard_name = "projection_y_coordinate" ;

double lon(y,x) ;
      lon:standard_name = "longitude" ;
```

b. Time is regarded in these conventions as an optional coordinate variable. The rules of attribution for coordinate variables will apply to all time coordinate variables that are present in a NetCDF file. See: CF 1.4 Section 4.4 - Time Coordinate.

2.3 Data Variables

a. The mandatory **mapping** attribute indicates the variable that maps indices from the data variable to a cell ID and an index into the **connections** variable. See **cell_map** in Section 1.3.
b. The mandatory **connectivity** attribute indicates the variable that specifies the nodes which define each cell. See **connections** in Section 1.3.
c. The mandatory **positions** attribute indicates the variable that specifies the coordinate indices, so that the location of a particular node can be determined. See **locations** in Section 1.3.
d. The mandatory **coordinates** attribute specifies the names of all related coordinate variables in a space-delimited string. Specifically, the format of the string will depend on whether the data is temporal:
    i. If the data contains a temporal dimension, the name of the time coordinate variable must occur first in the string. Spatial coordinates should follow as specified below.
    ii. The names of spatial coordinate variables may be listed in any order, such that the chosen order matches that of the indices of the variable indicated by the **positions** attribute.

```
double data_var_with_time(time, cells) ;
      data_var_with_time:positions = "locations" ;
      data_var_with_time:coordinates = "t y x" ; // for a data
      variable containing a temporal dimension, locations(#,0)
      corresponds to the y variable, locations (#,1) corresponds
      to the x variable.

double data_var_no_time(cells) ;
      data_var_no_time:positions = "locations" ;
      data_var_no_time:coordinates = "x y" ; // for a data
      variable without a temporal dimension, locations(#,0)
      corresponds to the x variable, locations(#,1) corresponds to
      the y variable.
```

e. To facilitate compatibility with the ESRI ArcGIS software package as well as other open source GIS software packages, all data variables must include an attribute describing the coordinate system in a form known as a Well Known Text string (WKT). The WKT string must be stored in the **esri_pe_string** attribute as shown in the example below:

```
depth:esri_pe_string =
      "PROJCS[\"NAD_1983_UTM_Zone_17N\",GEOGCS[\"GCS_North_America
      n_1983\",DATUM[\"D_North_American_1983\",SPHEROID[\"GRS_1980
      \",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT[
      \"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Merc
      ator\"],PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"Fa
      lse_Northing\",0.0],PARAMETER[\"Central_Meridian\",-81.0],PA
      RAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_Ori
      gin\",0.0],UNIT[\"Meter\",1.0]]" ;
```

    i. For more information on WKT strings, see: http://www.geoapi.org/2.0/javadoc/org/opengis/referencing/doc-files/WKT.html

f. To further describe the characteristics of the data, the **cell_methods** attribute can be used. This attribute can describe the statistical nature of the data, such as mean, maximum, instantaneous (point), etc. The value of the **cell_methods** attribute must be selected from the list of cell methods in the CF conventions document. See: CF 1.4 Appendix E. Cell Methods.

```
float pressure(time, stations) ;
      pressure:long_name = "pressure" ;
      pressure:units = "kPa" ;
      pressure:cell_methods = "time: point" ;

float maxtemp(time, stations) ;
      maxtemp:long_name = "temperature" ;
      maxtemp:units = "K" ;
      maxtemp:cell_methods = "time: maximum" ;

float ppn(time, stations) ;
      ppn:long_name = "depth of water-equivalent precipitation" ;
      ppn:units = "mm" ;
      ppn:cell_methods = "time: sum" ;
```

g. The **_FillValue** attribute must be used to specify missing data values in a data variable. The value of this attribute should be the same as the missing data representation in the variable's actual data. See: CF 1.4 Section 2.5.1 - Missing Data.

h. The **standard_name** attribute is *recommended* for data variables. The **standard_name** attribute is used to determine if datasets are of a comparable quality. A list of common standard names can be found at: http://cf-pcmdi.llnl.gov/documents/cf-standard-names/standard-name-table/18/cf-standard-name-table.html. If an acceptable standard name cannot be found in this list, then the **standard_name** attribute should not be used. A standard name suitable to the data type may be created following the *Guidelines for Construction of CF Standard Names* document found at the above link and submitted to the CF mailing list for inclusion in the list.

i. To facilitate data visualization in spatial contexts, the inclusion of **min** and **max** attributes is suggested for each data variable. The values of these attributes should correspond to the minimum and maximum values contained in the data, respectively.

2.4 Grid Mapping (Projection) Variables

a. All data variables must be accompanied by a grid mapping variable matching the **grid_mapping** attribute in the data variable. This is necessary to be able to view data in the correct spatial context. See: CF 1.4 Appendix F. Grid Mappings.

b. All grid mapping variables must have attributes describing the ellipsoid used for the mapping. There are two possible combinations of attributes that can completely describe an ellipsoid:
   i. **semi_major_axis** and **inverse_flattening**

```
int crs ;
      crs:grid_mapping_name = "latitude_longitude" ;
      crs:longitude_of_prime_meridian = 0.0 ;
```

```
crs:semi_major_axis = 6378137.0 ;
crs:inverse_flattening = 298.257223563 ;
```

ii. **semi_major_axis** and **semi_minor_axis**

```
int transverse_mercator ;
      transverse_mercator:grid_mapping_name =
      "transverse_mercator" ;
      transverse_mercator:longitude_of_central_meridian = -93. ;

      transverse_mercator:latitude_of_projection_origin = 0. ;

      transverse_mercator:scale_factor_at_central_meridian =
      0.9996 ;
      transverse_mercator:false_easting = 500000. ;
      transverse_mercator:false_northing = 0. ;
      transverse_mercator:semi_major_axis = 6378137. ;
      transverse_mercator:semi_minor_axis = 6356752.31424518 ;

      transverse_mercator:_CoordinateAxisTypes = "GeoY GeoX" ;
```

iii. Note: For Java development, the above attributes are only supported in version of the NetCDF Java libraries 4.2.20100726.1909 and higher.

2.5 Global attributes

a. Global attributes are used to describe the contents of an entire NetCDF file. A CERP compliant NetCDF file *should* include the following global attributes:
   i. **title** - a concise description of the file.
   ii. **author** - the name of the author or originator of the file.
   iii. **institution** - the name of the institution or organization that created the file.
   iv. **Conventions** - the CF version that the file complies with (currently "1.4").
   v. **source** - a description of the software that created the file, including the name, version number, and any configuration information that is needed to recreate the file if possible. If the file contains empirical data, then the method of collection and any other pertinent information should be listed here.
   vi. **history** - a history of all modifications performed to the file. Entries should include a date stamp followed by a description of the modification. The first entry in this attribute should ALWAYS be the initial file creation date stamp and a description stating that it was the initial creation of the file.
   vii. **cerp_version** - the CERP version that the file complies with (currently "1.2").
   viii. **comment** - a roughly 1 to 3 sentence explanation of what this file contains.
   ix. **qaqc** - a value describing the quality control/quality assurance status of the file. If the originator's institution has standard qaqc terms, then the appropriate term should be used here.

b. Optionally, the author, institution, source, qaqc, and comment attributes can be added to individual data variables.

Appendix A. Example NetCDF header for a file containing temporal data.

```
dimensions:
      time = UNLIMITED ; // (10 currently)
      y = 401 ;
      x = 301 ;
      cells = 120000 ;
      edges = 4 ;
      nodes = 480000 ;
      two = 2 ;
variables:
      int time(time) ;
            time:long_name = "time step" ;
            time:_CoordinateAxisType = "Time" ;
            time:units = "years since 2014-08-18T00:00:00 +0000" ;
      double y(y) ;
            y:long_name = "y coordinate of projection" ;
            y:standard_name = "projection_y_coordinate" ;
            y:_CoordinateAxisType = "GeoY" ;
            y:units = "Meter" ;
      double x(x) ;
            x:long_name = "x coordinate of projection" ;
            x:standard_name = "projection_x_coordinate" ;
            x:_CoordinateAxisType = "GeoX" ;
            x:units = "Meter" ;
      int locations(nodes, two) ;
            locations:long_name = "locations in x and y variables" ;
            locations:description = "Locations for each node entry two
            indices, one for use with the x variable and one for use with the
            y variable. These indices can be used to look up the actual node
            coordinates within the x and y variables." ;
      int connections(cells, edges) ;
            connections:long_name = "indices into the locations variable" ;
            connections:description = "Gives, for each cell entry, edges
            indices into the locations variable." ;
      int cell_map(cells, two) ;
            cell_map:long_name = "cell id to cell index into the connections
            variable" ;
            cell_map:description = "Gives each cell entry an (id, index). The
            index is into the connections variable." ;
      int transverse_mercator ;
            transverse_mercator:grid_mapping_name = "transverse_mercator" ;
            transverse_mercator:longitude_of_central_meridian = -81. ;
            transverse_mercator:latitude_of_projection_origin = 0. ;
            transverse_mercator:scale_factor_at_central_meridian = 0.9996 ;
            transverse_mercator:earth_radius = 6371229. ;
            transverse_mercator:false_easting = 500000. ;
            transverse_mercator:false_northing = 0. ;
            transverse_mercator:semi_major_axis = 6378137. ;
            transverse_mercator:semi_minor_axis = 6356752.31414036 ;
            transverse_mercator:_CoordinateAxisTypes = "GeoY GeoX" ;
      float example(time, cells) ;
            example:long_name = "example data variable" ;
            example:units = "n/a" ;
```

```
            example:mapping = "cell_map" ;
            example:connectivity = "connections" ;
            example:positions = "locations" ;
            example:coordinates = "time y x" ;
            example:min = 0.f ;
            example:max = 13.60787f ;
            example:_FillValue = 1.#QNAN0f ;
            example:esri_pe_string = "PROJCS[\"NAD83 / UTM zone 17N\",
            GEOGCS[\"NAD83\",     DATUM[\"North American Datum 1983\",
            SPHEROID[\"GRS 1980\", 6378137.0, 298.257222101,
            AUTHORITY[\"EPSG\",\"7019\"]],       TOWGS84[0.0, 0.0, 0.0, 0.0,
            0.0, 0.0, 0.0],       AUTHORITY[\"EPSG\",\"6269\"]],
            PRIMEM[\"Greenwich\", 0.0, AUTHORITY[\"EPSG\",\"8901\"]],
            UNIT[\"degree\", 0.017453292519943295],     AXIS[\"Geodetic
            longitude\", EAST],     AXIS[\"Geodetic latitude\", NORTH],
            AUTHORITY[\"EPSG\",\"4269\"]],
            PROJECTION[\"Transverse_Mercator\", AUTHORITY[\"EPSG\",\"9807\"]],
            PARAMETER[\"central_meridian\", -81.0],
            PARAMETER[\"latitude_of_origin\", 0.0],
            PARAMETER[\"scale_factor\", 0.9996],
            PARAMETER[\"false_easting\", 500000.0],
            PARAMETER[\"false_northing\", 0.0],   UNIT[\"m\", 1.0],
            AXIS[\"Easting\", EAST],   AXIS[\"Northing\", NORTH],
            AUTHORITY[\"EPSG\",\"26917\"]]" ;
            example:grid_mapping = "transverse_mercator" ;

// global attributes:
            :Conventions = "1.4" ;
            :cerp_version = "1.2" ;
            :history = "Created Mon Aug 18 13:46:09 EDT 2014" ;
            :source = "CERP NetCDF Library" ;
            :comment = "CERP NetCDF Library v1.2" ;
            :institution = "JEM" ;
            :author = "mckelvym on IGS-McKelvy" ;
```

Appendix B. Example NetCDF header for a file containing atemporal data.

```
dimensions:
      y = 401 ;
      x = 301 ;
      cells = 120000 ;
      edges = 4 ;
      nodes = 480000 ;
      two = 2 ;
variables:
      double y(y) ;
            y:long_name = "y coordinate of projection" ;
            y:standard_name = "projection_y_coordinate" ;
            y:_CoordinateAxisType = "GeoY" ;
            y:units = "Meter" ;
      double x(x) ;
            x:long_name = "x coordinate of projection" ;
            x:standard_name = "projection_x_coordinate" ;
            x:_CoordinateAxisType = "GeoX" ;
            x:units = "Meter" ;
      int locations(nodes, two) ;
            locations:long_name = "locations in x and y variables" ;
            locations:description = "Locations for each node entry two
            indices, one for use with the x variable and one for use with the
            y variable. These indices can be used to look up the actual node
            coordinates within the x and y variables." ;
      int connections(cells, edges) ;
            connections:long_name = "indices into the locations variable" ;
            connections:description = "Gives, for each cell entry, edges
            indices into the locations variable." ;
      int cell_map(cells, two) ;
            cell_map:long_name = "cell id to cell index into the connections
            variable" ;
            cell_map:description = "Gives each cell entry an (id, index). The
            index is into the connections variable." ;
      int transverse_mercator ;
            transverse_mercator:grid_mapping_name = "transverse_mercator" ;
            transverse_mercator:longitude_of_central_meridian = -81. ;
            transverse_mercator:latitude_of_projection_origin = 0. ;
            transverse_mercator:scale_factor_at_central_meridian = 0.9996 ;
            transverse_mercator:earth_radius = 6371229. ;
            transverse_mercator:false_easting = 500000. ;
            transverse_mercator:false_northing = 0. ;
            transverse_mercator:semi_major_axis = 6378137. ;
            transverse_mercator:semi_minor_axis = 6356752.31414036 ;
            transverse_mercator:_CoordinateAxisTypes = "GeoY GeoX" ;
      float example(cells) ;
            example:long_name = "example data variable" ;
            example:units = "n/a" ;
            example:mapping = "cell_map" ;
            example:connectivity = "connections" ;
            example:positions = "locations" ;
            example:coordinates = "y x" ;
            example:min = 0.f ;
```

```
                example:max = 13.60787f ;
                example:_FillValue = 1.#QNAN0f ;
                example:esri_pe_string = "PROJCS[\"NAD83 / UTM zone 17N\",
                GEOGCS[\"NAD83\",     DATUM[\"North American Datum 1983\",
                SPHEROID[\"GRS 1980\", 6378137.0, 298.257222101,
                AUTHORITY[\"EPSG\",\"7019\"]],       TOWGS84[0.0, 0.0, 0.0, 0.0,
                0.0, 0.0, 0.0],       AUTHORITY[\"EPSG\",\"6269\"]],
                PRIMEM[\"Greenwich\", 0.0, AUTHORITY[\"EPSG\",\"8901\"]],
                UNIT[\"degree\", 0.017453292519943295],     AXIS[\"Geodetic
                longitude\", EAST],     AXIS[\"Geodetic latitude\", NORTH],
                AUTHORITY[\"EPSG\",\"4269\"]],
                PROJECTION[\"Transverse_Mercator\", AUTHORITY[\"EPSG\",\"9807\"]],
                PARAMETER[\"central_meridian\", -81.0],
                PARAMETER[\"latitude_of_origin\", 0.0],
                PARAMETER[\"scale_factor\", 0.9996],
                PARAMETER[\"false_easting\", 500000.0],
                PARAMETER[\"false_northing\", 0.0],   UNIT[\"m\", 1.0],
                AXIS[\"Easting\", EAST],   AXIS[\"Northing\", NORTH],
                AUTHORITY[\"EPSG\",\"26917\"]]" ;
                example:grid_mapping = "transverse_mercator" ;

// global attributes:
                :Conventions = "1.4" ;
                :cerp_version = "1.2" ;
                :history = "Created Mon Aug 18 13:49:25 EDT 2014" ;
                :source = "CERP NetCDF Library" ;
                :comment = "CERP NetCDF Library v1.2" ;
                :institution = "JEM" ;
                :author = "mckelvym on IGS-McKelvy" ;
```