

Encoding CoastWatch Satellite Data in NetCDF using the CF Metadata Conventions

Prepared by:

Peter Hollemans

Terrenus Earth Sciences / SP Systems Consultant for NOAA/NESDIS

Last revised: March 2010

Changes

Author	Changes Made	Pages/Sections	Date
Peter Hollemans	Initial revision	All	Feb 4, 2010
Peter Hollemans	Corrected DCS attribute names	Section 4.2	Mar 3, 2010

Table of Contents

Section 1: [Overview](#)

Section 2: [CF Attributes](#)

2.1 [Global Attributes](#)

2.2 [Data Variable Attributes](#)

2.3 [Time Coordinate Variable Attributes](#)

2.4 [Spatial Coordinate Variable Attributes](#)

2.5 [Grid Mapping Variable Attributes](#)

Section 3: [Examples](#)

3.1 [Swath Projection Data](#)

3.2 [Map Projection Data](#)

3.3 [Composite Data](#)

Section 4: [Non-CF Attributes](#)

4.1 [CoastWatch Attributes](#)

4.2 [DCS Attributes](#)

4.3 [Unidata Dataset Discovery Attributes](#)

Section 5: [References](#)

Section 1: Overview

This document is intended to be used by CoastWatch data providers to create NetCDF datasets that conform to the Climate Forecast (CF) 1.4 metadata specifications and contain equivalent information to CoastWatch HDF datasets. In the Section 2, we list all NetCDF CF attribute names:

- The CF attribute names suggested for the core CoastWatch file metadata.
- The CF attribute names less useful to CoastWatch.

This document only gives a brief description of the attributes for reference. The CF-1.4 documentation, COARDS 1.0 (upon which CF is based) and NetCDF User's Guide should be consulted for the complete description, rules, and usage of each attribute:

CF documentation:

<http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.4/cf-conventions.html>

COARDS documentation:

http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html

NetCDF User's Guide:

<http://www.unidata.ucar.edu/software/netcdf/docs/netcdf.html#Attribute-Conventions>

In Section 4, we also give a list of useful non-CF attributes of interest to CoastWatch data providers.

Section 2: CF Attributes

2.1 Global Attributes

Attribute name	NetCDF type	Description
Conventions	String	Comma-separated list of dataset conventions, eg: 'CF-1.4'.
source	String	Comma-separated list of source data, eg: 'NOAA16_AVHRR_HRPT_WI', 'NOAA18_AVHRR_HRPT_WI'.
institution	String	Where data was produced, eg: 'NOAA/NESDIS/CoastWatch'.
history	String	Audit trail of data modification.
title	String	Short description of dataset contents.
references	String	Published or web-based references describing the data or methods used to produce it, eg: 'http://coastwatch.noaa.gov/cwn/cw_publications.html'.
comment	String	Miscellaneous information about data or methods.

2.2 Data Variable Attributes

Attribute name	NetCDF type	Description
scale_factor	float, double	For packed data, factor to multiply after reading.
add_offset	float, double	For packed data, offset to add after reading, possibly after multiplying by factor.
_FillValue	<var>	Value used to represent missing or undefined data.
valid_min	<var>, float, double	Minimum valid data value in data.
valid_max	<var>, float, double	Maximum valid data value in data.
valid_range	<var>, float, double	Minimum and maximum values as an array.

long_name	String	Descriptive name for variable.
standard_name	String	Standard variable name referencing a entry in the CF standard name table.
units	String	Scientific units in Unidata UDUNITS-compliant format.
coordinates	String	Blank-separated list of references to coordinate variables.
grid_mapping	String	Reference to map projection variable.
cell_methods	String	Extra information about the cell data values and how they were computed.
ancillary_variables	String	Blank-separated list of references to related variables such as quality flags.
flag_masks	<var>	List of bit field values for boolean conditions.
flag_meanings	String	Blank-separated list of words or phrases for each flag.
flag_values	<var>	List of possible flag values.
reference	String	References that describe the data or methods used to produce it (in addition to global attribute).
source	String	Method of production of the original data (in addition to global attribute).
comment	String	Miscellaneous information about data or methods (in addition to global attribute).
cell_measures	String	Reference to variable that contains cell areas or volumes.
institution	String	Where the original data was produced (in addition to global attribute).
standard_err_multiplier	float, double	Standard error multiplier for error data.
missing_value	<var>	Same as _FillValue (deprecated).

2.3 Time Coordinate Variable Attributes

Attribute name	NetCDF type	Description
units	String	Time units in Unidata UDUNITS-compliant format, eg: 'seconds since 1970-01-01 00:00:00 UTC'.

long_name	String	Descriptive name for variable.
standard_name	String	Standard variable name, referencing an entry in the CF standard name table, eg: 'time'.
bounds	String	Reference to variable that contains vertices of time cell boundaries.
valid_min	<var>, float, double	Minimum valid data value in time axis data.
valid_max	<var>, float, double	Maximum valid data value in time axis data.
valid_range	<var>, float, double	Minimum and maximum time axis values as an array.
calendar	String	Calendar used for encoding time axes, eg: 'gregorian'.
climatology	String	Reference to variable that contains extents of climatology, similar to bounds but for climatology data.
compress	String	Reference to dimensions compressed by gathering.
leap_month	int	Specifies month lengthened by one day for user defined calendar.
leap_year	int	Specifies leap year for user defined calendar.
month_lengths	int	Specifies month lengths in non-leap year for user defined calendar.
axis	String	Identifies longitude, latitude, vertical, or time axes, eg: 'T' for time axis.

2.4 Spatial Coordinate Variable Attributes

Attribute name	NetCDF type	Description
units	String	Spatial units in Unidata UDUNITS-compliant format, eg: 'degrees_north', 'degrees_east', 'meters', 'km'.
long_name	String	Descriptive name for variable.
standard_name	String	Standard variable name, referencing an entry in the CF standard name table, eg: 'latitude', 'longitude',

		'projection_x_coordinate', 'projection_y_coordinate', etc.
bounds	String	Reference to variable that contains vertices of spatial cell boundaries.
valid_min	<var>, float, double	Minimum valid data value in spatial axis data.
valid_max	<var>, float, double	Maximum valid data value in spatial axis data.
valid_range	<var>, float, double	Minimum and maximum spatial axis values as an array.
positive	String	Direction of increasing vertical coordinate value: 'up' or 'down'.
axis	String	Identifies longitude, latitude, vertical, or time axes, eg: 'X' for longitude axis, 'Y' for latitude axis, 'Z' for vertical axis.
compress	String	Reference to dimensions compressed by gathering.
formula_terms	String	Reference to variables defined in formula for vertical coordinate systems.

2.5 Grid Mapping Variable Attributes

Attribute name	NetCDF type	Description
grid_mapping_name	String	FGDC-based map name: 'latitude_longitude', 'albers_conical_equal_area', 'mercator', 'polar_stereographic', etc.
earth_radius	float, double	For all projections: radius in meters of spherical Earth model.
semi_major_axis	float, double	For all projections: length in meters of semi-major axis of ellipsoid.
inverse_flattening	float, double	For all projections: inverse flattening of ellipsoid.
longitude_of_projection_origin	float, double	Mercator projection: longitude of origin of rectangular coordinates.
scale_factor_at_projection_origin	float, double	Mercator projection: multiplier for reducing a distance obtained from a map by computation or scaling to actual

		distance at projection origin.
straight_vertical_longitude_from_pole	float, double	Polar stereographic projection: longitude to be oriented straight up from north or south pole.
latitude_of_projection_origin	float, double	Polar stereographic, Albers equal area projections: latitude of origin of rectangular coordinates.
standard_parallel	float, double	Polar stereographic projection: line of intersection between map projection surface and sphere/ellipsoid, or latitude of true scale.
longitude_of_central_meridian	float, double	Albers equal area projection: line of longitude at center of map projection.
false_easting	float, double	Value added to all abscissa values in rectangular coordinates.
false_northing	float, double	Value added to all ordinate values in rectangular coordinates.
grid_north_pole_latitude	float, double	True latitude of north pole of rotated grid.
grid_north_pole_longitude	float, double	True longitude of north pole of rotated grid.
longitude_of_prime_meridian	float, double	Longitude, with respect to Greenwich, of prime meridian associated with geodetic datum.
north_pole_grid_longitude	float, double	Longitude of true north pole in rotated grid.
perspective_point_height	float, double	Height in meters of map projection perspective point above ellipsoid.
scale_factor_at_central_meridian	float, double	Multiplier for reducing a distance obtained from a map by computation or scaling to the actual distance along central meridian
semi_minor_axis		Length in metres of semi-minor axis of ellipsoid.

Section 3: Examples

To illustrate how CoastWatch datasets can be encoded in NetCDF with CF-1.4 conventions, we give examples of swath (satellite geometry), map projected, and composited datasets. These examples follow NetCDF CDL syntax, an ASCII syntax for specifying NetCDF binary file contents described here:

<http://www.unidata.ucar.edu/software/netcdf/docs/netcdf.html#CDL-Syntax>

3.1 Swath Projection Data

The following example shows a satellite data file containing NOAA AVHRR sensor data, as it would appear in NetCDF after being converted from NOAA 1b format.

```
netcdf 2010_007_0818_n19_wi {  
    // File dimensions  
    // -----  
    dimensions:  
        line = 4851;  
        sample = 2048;  
        level = 1;  
  
    // Variables  
    // -----  
    variables:  
  
        short avhrr_ch1 (level, line, sample);  
            avhrr_ch1:scale_factor = 0.01;  
            avhrr_ch1:add_offset = 0.0;  
            avhrr_ch1:FillValue = -32768s;  
            avhrr_ch1:standard_name = "surface_albedo";  
            avhrr_ch1:units = "percent";  
            avhrr_ch1:coordinates = "time lat lon";  
            avhrr_ch1:cell_methods = "area: mean";  
            avhrr_ch1:grid_mapping = "coord_ref";  
  
        short avhrr_ch2 (level, line, sample);  
            avhrr_ch2:scale_factor = 0.01;  
            avhrr_ch2:add_offset = 0.0;  
            avhrr_ch2:FillValue = -32768s;  
            avhrr_ch2:standard_name = "surface_albedo";  
            avhrr_ch2:units = "percent";  
            avhrr_ch2:coordinates = "time lat lon";  
            avhrr_ch2:cell_methods = "area: mean";  
            avhrr_ch2:grid_mapping = "coord_ref";  
  
        short avhrr_ch3 (level, line, sample);  
            avhrr_ch3:scale_factor = 0.01;  
            avhrr_ch3:add_offset = 0.0;  
            avhrr_ch3:FillValue = -32768s;  
            avhrr_ch3:standard_name = "brightness_temperature";  
            avhrr_ch3:units = "celsius";
```

```
    avhrr_ch3:coordinates = "time lat lon";
    avhrr_ch3:cell_methods = "area: mean";
    avhrr_ch3:grid_mapping = "coord_ref";

short avhrr_ch4 (level, line, sample);
    avhrr_ch4:scale_factor = 0.01;
    avhrr_ch4:add_offset = 0.0;
    avhrr_ch4:_FillValue = -32768s;
    avhrr_ch4:standard_name = "brightness_temperature";
    avhrr_ch4:units = "celsius";
    avhrr_ch4:coordinates = "time lat lon";
    avhrr_ch4:cell_methods = "area: mean";
    avhrr_ch4:grid_mapping = "coord_ref";

short avhrr_ch5 (level, line, sample);
    avhrr_ch5:scale_factor = 0.01;
    avhrr_ch5:add_offset = 0.0;
    avhrr_ch5:_FillValue = -32768s;
    avhrr_ch5:standard_name = "brightness_temperature";
    avhrr_ch5:units = "celsius";
    avhrr_ch5:coordinates = "time lat lon";
    avhrr_ch5:cell_methods = "area: mean";
    avhrr_ch5:grid_mapping = "coord_ref";

double lat (line, sample);
    lat:standard_name = "latitude";
    lat:units = "degrees_north";

double lon (line, sample);
    lon:standard_name = "longitude";
    lon:units = "degrees_east";

double time (line);
    time:standard_name = "time";
    time:units = "seconds since 1970-01-01 00:00:00 UTC";

double level (level);
    level:standard_name = "height";
    level:units = "m";
    level:positive = "up";

int coord_ref;
    coord_ref:grid_mapping_name = "latitude_longitude";
    coord_ref:semi_major_axis = 6378137.0;
    coord_ref:inverse_flattening = 298.257223563;
    coord_ref:longitude_of_prime_meridian = 0.0;

// Global attributes
// -----
:Conventions = "CF-1.4";
:source = "NOAA19 AVHRR HRPT WI";
:institution = "USDOC/NOAA/NESDIS CoastWatch Central Operations";
:history = "[2010-01-07 04:12:55 EST cwf-3.2.4-pre-build338 cwatch] cwexport
NSS.HRPT.NP.D10007.S0818.E0832.B0472222.WI 2010_007_0818_n19_wi.nc";
:title = "Level 2 NOAA AVHRR sensor data";
:references = "http://www.nsof.class.noaa.gov";

// Variable data
// -----
data:
    level = 0;
```

```
}
```

Notes:

- The 'source' attribute is in the style of GHR SST NetCDF data, where the platform, sensor, and type of data stream are separated by underscores. We've added the capture station here as well (WI = Wallops Island) because CoastWatch swath data often contains the station code to help with file tracking. Composite values are traditionally blank-separated in CF, so the underscore format allows for multiple sources to be named and blank-separated.
- The 'cell_methods' attribute should always be used with satellite image data, because each pixel represents an integration of energy over the pixel area. If not specified (for example wind data), then the data is taken to be point values taken at the pixel center.
- The 'grid_mapping' in CF is normally reserved to describe map projection data, but can also be used to specify the ellipsoid used for lat/lon data. We use it here to specify the WGS-84 ellipsoid geometry and prime meridian.
- The time axis is unusual, because each scan line of a NOAA polar orbiter is taken at a different time. The swath start time, end time, or center time are often used as the definitive data reference time, but this is not strictly accurate. We include a time axis that contains a unique time for each scan line, so that any subsequent map projection process can make the most appropriate decision as to what time the data was taken.

3.2 Map Projection Data

This example shows a typical CoastWatch sea surface temperature mapped dataset converted to NetCDF.

```
netcdf 2010_007_0818_n19_ne {  
    // File dimensions  
    // -----  
    dimensions:  
        row = 1024;  
        column = 1024;  
        time = 1;  
        level = 1;  
        n_vals = 2;  
  
    // Variables  
    // -----  
    variables:  
  
        short sst (time, level, row, column);  
        sst:scale_factor = 0.01;  
        sst:add_offset = 0.0;  
        sst:_FillValue = -32768s;  
        sst:standard_name = "sea_surface_skin_temperature";  
        sst:units = "celsius";  
        sst:coordinates = "lat lon";  
        sst:cell_methods = "area: mean";  
        sst:ancillary_variables = "cloud";
```

```

sst:grid_mapping = "mercator_proj";
sst:source = "MCSST_atriple_night_2009_04_10";

byte cloud (time, level, row, column);
cloud:standard_name = "sea_surface_skin_temperature status_flag";
cloud:coordinates = "lat lon";
cloud:cell_methods = "area: mean";
cloud:grid_mapping = "mercator_proj";
cloud:flag_masks = 1b, 2b, 4b, 8b, 16b, 32b, 64b;
cloud:flag_meanings = "thermal_gross thermal_uniform uniform_low_stratus
four_minus_five cirrus ch3_albedo ch3_uniform";
cloud:source = "CWCLAVR_2009_04_10";

double lat (row, column);
lat:standard_name = "latitude";
lat:units = "degrees_north";

double lon (row, column);
lon:standard_name = "longitude";
lon:units = "degrees_east";

double x (column);
x:standard_name = "projection_x_coordinate";
x:units = "km";

double y (row);
y:standard_name = "projection_y_coordinate";
y:units = "km";

double time (time);
time:standard_name = "time";
time:units = "seconds since 1970-01-01 00:00:00 UTC";
time:bounds = "time_bounds";

double time_bounds (time, n_vals);

double level (level);
level:standard_name = "height";
level:units = "m";
level:positive = "up";

int mercator_proj;
mercator_proj:grid_mapping_name = "mercator";
mercator_proj:semi_major_axis = 6378137.0;
mercator_proj:inverse_flattening = 298.257223563;
mercator_proj:longitude_of_projection_origin = 0.0;
mercator_proj:scale_factor_at_projection_origin = 1.0;

// Global attributes
// -----
:Conventions = "CF-1.4";
:source = "NOAA19 AVHRR HRPT WI";
:institution = "USDOC/NOAA/NESDIS CoastWatch Central Operations";
:history = "[2010-01-07 04:21:48 EST cwf-3.2.4-pre-build338 cwatch] cwregister -
m ne.nc --include '(sst|cloud)' 2010_007_0818_n19_wi.nc 2010_007_0818_n19_ne.nc";
:title = "CoastWatch SST data from NOAA AVHRR";
:references = "http://coastwatch.noaa.gov";
:comment = "Created using the CoastWatch TeraScan System (CTera) v1.0";

// Variable data
// -----

```

```
data:
  time = 1262852734;
  time_bounds = 1262852330, 1262853138;
  level = 0;
}
```

Notes:

- The 'sst' variable has an associated quality variable 'cloud' that has much more specific metadata than CoastWatch HDF format, namely the full set of cloud mask test names and their byte values (a standard part of CF).
- We have chosen in this example to reference the time from the center of the AVHRR pass that captured the data, and have named the start and endpoints of the data capture as the time cell bounds.
- This example uses a Mercator projection. CoastWatch HDF supports a total of 31 map projections using the GCTP library, where as CF-1.4 supports only 13 of the most common projections as follows:
 - Albers equal area = 'albers_conical_equal_area'
 - Azimuthal equidistant = 'azimuthal_equidistant'
 - Lambert azimuthal equal area = 'lambert_azimuthal_equal_area'
 - Lambert conformal = 'lambert_conformal_conic'
 - Lambert cylindrical equal area = 'lambert_cylindrical_equal_area' (not in GCTP)
 - Latitude-longitude = 'latitude_longitude'
 - Mercator = 'mercator'
 - Orthographic = 'orthographic'
 - Polar stereographic = 'polar_stereographic'
 - Rotated pole = 'rotated_latitude_longitude' (not in GCTP)
 - Stereographic = 'stereographic'
 - Transverse mercator = 'transverse_mercator'
 - Vertical perspective = 'vertical_perspective'

3.3 Composite Data

This example shows a sea surface temperature dataset that has been created by compositing data from multiple datasets to form a mean composite.

```
netcdf composite {
    // File dimensions
    // -----
    dimensions:
      row = 1024;
      column = 1024;
      time = 1;
      level = 1;
      n_vals = 2;
```

```
// Variables
// -----
variables:

    short sst (time, level, row, column);
        sst:scale_factor = 0.01;
        sst:add_offset = 0.0;
        sst:_FillValue = -32768s;
        sst:standard_name = "sea_surface_skin_temperature";
        sst:units = "celsius";
        sst:coordinates = "lat lon";
        sst:cell_methods = "area: mean time: mean within days time: mean over days";
        sst:grid_mapping = "mercator_proj";
        sst:source = "MCSST_atriple_night_2009_04_10 NLSST_split_day_2009_04_10";

    double lat (row, column);
        lat:standard_name = "latitude";
        lat:units = "degrees_north";

    double lon (row, column);
        lon:standard_name = "longitude";
        lon:units = "degrees_east";

    double x (column);
        x:standard_name = "projection_x_coordinate";
        x:units = "km";

    double y (row);
        y:standard_name = "projection_y_coordinate";
        y:units = "km";

    double time (time);
        time:standard_name = "time";
        time:units = "seconds since 1970-01-01 00:00:00 UTC";
        time:climatology = "climatology_bounds";

    double climatology_bounds (time, n_vals);

    double level (level);
        level:standard_name = "height";
        level:units = "m";
        level:positive = "up";

    int mercator_proj;
        mercator_proj:grid_mapping_name = "mercator";
        mercator_proj:semi_major_axis = 6378137.0;
        mercator_proj:inverse_flattening = 298.257223563;
        mercator_proj:longitude_of_projection_origin = 0.0;
        mercator_proj:scale_factor_at_projection_origin = 1.0;

// Global attributes
// -----
:Conventions = "CF-1.4";
:source = "NOAA17_AVHRR_HRPT_WI NOAA18_AVHRR_HRPT_WI NOAA19_AVHRR_HRPT_WI";
:institution = "USDOC/NOAA/NESDIS CoastWatch Central Operations";
:history = "[2010-01-07 04:21:48 EST cwf-3.2.4-pre-build338 cwatch] cwcomposite
mapped1.nc mapped2.nc mapped3.nc composite.nc";
:title = "CoastWatch SST data from NOAA AVHRR";
:references = "http://coastwatch.noaa.gov";

// Variable data
```

```
// -----  
data:  
  time = 1263357360;  
  level = 0;  
  climatology_bounds = 1263347760, 1263366960;  
}
```

Notes:

- The global 'source' attribute contains a blank-separated list of the satellite sensors used, plus the data stream type and ground capture station in a GHR SST style format. This follows the CF attribute conventions of blanks as separators and underscores within names.
- The 'climatology' attribute is used on the time axis and the 'cell_methods' attribute on the 'sst' variable to indicate that the data has been averaged over a period of time, and how the data averaging was done. The CF documentation has an in-depth discussion of different climatology types and how to document them.

Section 4: Non-CF Attributes

In some cases it may be desirable to add other non-CF attributes to NetCDF datasets. For example, a number of attributes are present in CoastWatch HDF datasets that do not have a simple mapping to CF-1.4. We recommend that these be inserted into NetCDF datasets conforming to CF using an XML namespace-like syntax that has been proposed for future versions of CF:

<https://cf-pcmdi.llnl.gov/trac/ticket/27>

The proposal involves prefixing attribute names with a short namespace code to identify which convention they belong to. We recommend using the proposal to avoid attribute name clashes in the future, and to make it easier for humans to read NetCDF dataset metadata. For now, we recommend *not* prefixing CF attributes with 'cf' suggested by the proposal, for compatibility with existing applications.

4.1 CoastWatch Attributes

CoastWatch-specific attributes that are not mapped to CF in the examples of Section 3 are as follows (prefixed with 'cw' to indicate use by CoastWatch):

Attribute name	NetCDF type	Description
<code>cw:orbit_type</code>	String	Polar satellite direction, 'ascending', or 'descending'.
<code>cw:station_name</code>	String	Full name of ground capture station.
<code>cw:pass_type</code>	String	Pass scene type as 'day', 'night', or 'day/night' for mixed scenes.
<code>cw:region_code</code>	String	Short code for data processing geographic region.
<code>cw:region_name</code>	String	Full name for data processing geographic region.
<code><var>:cw:nav_affine</code>	double	Six coefficients for the 3x3 navigation affine transform in homogenous coordinates as a_{00} , a_{10} , a_{01} , a_{11} , a_{02} , a_{12} .
<code>cw:polygon_latitude</code>	double	Bounding polygon latitudes.
<code>cw:polygon_longitude</code>	double	Bounding polygon longitudes.

It is useful to consult the document *Comparison of Metadata Standards* listed in the References, in order to compare which parts of CoastWatch HDF are handled by CF, how they are handled, and which parts are not explicitly handled.

4.2 DCS Attributes

Ocean color DCS (Data Content Standard) metadata defined by the NOAA IOOS Data Integration Framework can be inserted into NetCDF using a 'dcs' namespace as follows:

Attribute name	NetCDF type	Description
dcs:createInstitution	String	Name of the organization or agency that developed the dataset.
dcs:createDateTime	String	Time and date that the dataset was created, eg: '2010-01-21T11:29:27Z'.
dcs:acquisitionStartDateTime	String	Date and time that the image acquisition began.
dcs:acquisitionEndDateTime	String	Date and time that the image acquisition ended.
dcs:sensor	String	The complete name of the sensor employed for the observation, collection, or modeled event.
dcs:sensorPlatform	String	The complete name of the satellite platform.
dcs:mapProjection	String	The method used to represent the curved surface of the earth on a plane.
dcs:geodeticDatum	String	Defines the reference coordinate system for remotely sensed observation.
dcs:northernLatitude	double	Northern-most coordinate of the limit of coverage expressed in latitude.
dcs:southernLatitude	double	Southern-most coordinate of the limit of coverage expressed in latitude.
dcs:westernLongitude	double	Western-most coordinate of the limit of coverage expressed in longitude.
dcs:easternLongitude	double	Eastern-most coordinate of the limit of coverage expressed in longitude.
dcs:observedProperty	String	Defines the parameter or variable measured or calculated.
dcs:observedPropertyAlgorithm	String	Algorithm used to compute observed property.
dcs:processingLevel	String	Defines the status of the result as defined by data provider classifications.

These are the 15 mandatory DCS attributes, out of 32 attributes in total (see the DCS documentation for the remaining attributes). The following is an example of CoastWatch chlorophyll data encoded as NetCDF with CF, CoastWatch, and DCS mandatory attributes:

```
netcdf MODSCW_P2010015_0110_PB05_closest_chlora {  
    // File dimensions  
    // -----  
    dimensions:  
        row = 4616;  
        column = 3204;  
        time = 1;  
        level = 1;  
        n_vals = 2;  
  
    // Variables  
    // -----  
    variables:  
  
        float chlora (time, level, row, column);  
        chlora:FillValue = -1.0f;  
        chlora:standard_name = "mass_concentration_of_chlorophyll_in_sea_water";  
        chlora:units = "milligram m^-3";  
        chlora:coordinates = "lat lon";  
        chlora:cell_methods = "area: mean";  
        chlora:grid_mapping = "mercator_proj";  
        chlora:source = "OC3 NASA OBPG";  
        chlora:references = "NASA OBPG SeaWiFS/MODIS Correction (Gordon and Wang,  
1994)";  
  
        double lat (row, column);  
        lat:standard_name = "latitude";  
        lat:units = "degrees_north";  
  
        double lon (row, column);  
        lon:standard_name = "longitude";  
        lon:units = "degrees_east";  
  
        double x (column);  
        x:standard_name = "projection_x_coordinate";  
        x:units = "km";  
  
        double y (row);  
        y:standard_name = "projection_y_coordinate";  
        y:units = "km";  
  
        double time (time);  
        time:standard_name = "time";  
        time:units = "seconds since 1970-01-01 00:00:00 UTC";  
        time:bounds = "time_bounds";  
  
        double time_bounds (time, n_vals);  
  
        double level (level);  
        level:standard_name = "height";  
        level:units = "m";  
        level:positive = "up";  
  
        int mercator_proj;
```

```
mercator_proj:grid_mapping_name = "mercator";
mercator_proj:semi_major_axis = 6378137.0;
mercator_proj:inverse_flattening = 298.257223563;
mercator_proj:longitude_of_projection_origin = 0.0;
mercator_proj:scale_factor_at_projection_origin = 1.0;

// Global attributes
// -----
:Conventions = "CF-1.4";
:source = "Aqua MODIS";
:institution = "USDOC/NOAA/NESDIS CoastWatch Central Operations";
:history = "PGE01:5.0.39;PGE02:5.0.37.4;SeaDAS Version ?., MS112 4.0.2, Linux
2.6.9-89.ELsmp\n[cwf 3.2.4-pre-build28] cwregister --method=mixed ...";
:title = "CoastWatch chlorophyll data from Aqua MODIS";
:references = "http://coastwatch.noaa.gov";

:cw\:orbit_type = "ascending";
:cw\:pass_type = "day";
:cw\:region_code = "PB";
:cw\:region_name = "Pacific Basin";
:cw\:polygon_latitude = 50.0036, 50.0036, 50.0036, 50.0036, 50.0036, 39.8060,
27.8229, 14.3429, -0.0062, -0.0062, -0.0062, -0.0062, -0.0062, 14.3429, 27.8229,
39.8060, 50.0036;
:cw\:polygon_longitude = -179.9878, -169.9892, -159.9905, -149.9919, -139.9933,
-139.9933, -139.9933, -139.9933, -139.9933, -149.9919, -159.9905, -169.9892,
-179.9878, -179.9878, -179.9878, -179.9878, -179.9878;

:dcs\:createInstitution = "USDOC/NOAA/NESDIS CoastWatch Central Operations";
:dcs\:createDateTime = "2010-01-15T03:00:28Z";
:dcs\:acquisitionStartDateTime = "2010-01-15T01:10:07Z";
:dcs\:acquisitionEndDateTime = "2010-01-15T01:15:07Z";
:dcs\:sensor = "MODIS";
:dcs\:sensorPlatform = "Aqua";
:dcs\:mapProjection = "Mercator";
:dcs\:geodeticDatum = "WGS84";
:dcs\:northernLatitude = 50.0036;
:dcs\:southernLatitude = 0.0062;
:dcs\:westernLongitude = 179.9878;
:dcs\:easternLongitude = 159.9906;
:dcs\:observedProperty = "Chlorophyll concentration";
:dcs\:observedPropertyAlgorithm = "OC3";
:dcs\:processingLevel = "Level 3";

// Variable data
// -----
data:
  time = 1263517957;
  time_bounds = 1263517807, 1263518107;
  level = 0;
}
```

4.3 Unidata Dataset Discovery Attributes

The following table contains non-CF attributes from by the Unidata Dataset Discovery (UDD) convention, found here (note that we only list the “recommended” attributes):

<http://www.unidata.ucar.edu/software/netcdf-java/formats/DataDiscoveryAttConvention.html>

This convention is for use in listing NetCDF datasets as part of a data discovery system such as a digital library or a THREDDS data server. Since the convention already exists and may have applications that expect the attribute names to appear without any XML-style namespace prefix, we recommend not using any prefix such as 'udd' for now.

Attribute name	NetCDF type	Description
summary	String	A paragraph describing the dataset.
keywords	String	A comma separated list of key words and phrases.
id	String	Combined with the naming authority, a globally unique identification for the dataset, eg: '2010_007_0818_n19_ne'
naming_authority	String	Combined with the ID, a globally unique identification for the dataset, eg: 'gov.noaa.coastwatch'.
keywords_vocabulary	String	If available, a guideline for the words/phrases in the keywords attribute.
cdm_data_type	String	The THREDDS data type appropriate for this dataset: 'Grid', 'Image', 'Station', 'Swath', 'Trajectory'.
date_created	String	The date on which the data was created, eg: '2010-01-15T03:00:28Z'.
creator_name	String	The data creator's name.
creator_url	String	The data creator's URL.
creator_email	String	The data creator's email.
project	String	The scientific project that produced the data.
processing_level	String	A textual description of the processing (or quality control) level of the data.
acknowledgement	String	A place to acknowledge various type of support for the project that produced this data.
geospatial_lat_min	double	Describes a simple latitude, longitude, and vertical bounding box.
geospatial_lat_max	double	
geospatial_lon_min	double	
geospatial_lon_max	double	

geospatial_vertical_min	double	Describes the temporal coverage of the data as a time range.
geospatial_vertical_max	double	
time_coverage_start	String	
time_coverage_end	String	
time_coverage_duration	String	
time_coverage_resolution	String	The name of the controlled vocabulary from which variable standard names are taken, eg: 'CF-1.4'.
standard_name_vocabulary	String	
license	String	Describe the restrictions to data access and distribution.

The following is an example of a CoastWatch dataset encoded in NetCDF using a mixture of CF, CoastWatch, and UDD attributes. Note that we leave out DCS attributes because the dataset presented is not ocean color, though strictly speaking the DCS attributes for ocean color are appropriate for most other products as well.

```
netcdf SSMI1002704ds_WW {
    // File dimensions
    // -----
    dimensions:
        lat = 540;
        lon = 1080;
        time = 1;
        level = 1;
        n_vals = 2;

    // Variables
    // -----
    variables:

        float windspeed (time, level, lat, lon);
        windspeed: FillValue = -3.4e38f;
        windspeed:standard_name = "wind_speed";
        windspeed:units = "m s^-1";
        windspeed:source = "Goodberlet";
        windspeed:references = "Goodberlet et al. 1989";

        double lat (lat);
        lat:standard_name = "latitude";
        lat:units = "degrees_north";

        double lon (lon);
        lon:standard_name = "longitude";
        lon:units = "degrees_east";

        double time (time);
        time:standard_name = "time";
        time:units = "seconds since 1970-01-01 00:00:00 UTC";
        time:bounds = "time_bounds";
}
```

```

double time_bounds (time, n_vals);

double level (level);
    level:standard_name = "height";
    level:units = "m";
    level:positive = "up";

// Global attributes
// -----
:Conventions = "CF-1.4";
:source = "DMSP_F15_SSMI";
:institution = "USDOC/NOAA/NESDIS CoastWatch Central Operations";
:history = "/home/coastwatch/winds/bin/ssmi_assemble_cwhdf -o /data/ssmi/winds/
hdf/ssmi/4hr/SSMI1002704ds_WW.hdf
/data/ssmi/winds/hdf/ssmi/4hr/SSMI1002704ds_s9.hdf ...";
:title = "CoastWatch wind data from DMSP F15 SSM/I";
:references = "http://coastwatch.noaa.gov";

:cw\:region_code = "WW";
:cw\:region_name = "Global";

:summary = "DMSP SSM/I ocean surface winds data provide wind speeds (no
direction) at an elevation of 19.5 m. The strictest rain flag has been used to
yield wind speeds within 2 m/s. The data coverage is global, at approximately 30
km resolution. Datasets are updated continuously every 4 hours as measurements
are taken, and assembled into daily coverage maps.";
:keywords = "Atmosphere > Atmospheric Winds > Surface Winds";
:id = "SSMI1002704ds_WW";
:naming_authority = "gov.noaa.coastwatch";
:keywords_vocabulary = "GCMD Science Keywords";
:cdm_data_type = "Grid";
:date_created = "2010-01-27T11:10:00Z";
:creator_name = "USDOC/NOAA/NESDIS CoastWatch Central Operations";
:creator_url = "http://coastwatch.noaa.gov";
:creator_email = "coastwatch.info@noaa.gov";
:processing_level = "Level 3";
:acknowledgement = "Fleet Numerical Meteorology and Oceanography Center
(FNMOC)";
:geospatial_lat_min = -89.8333;
:geospatial_lat_max = 89.8333;
:geospatial_lon_min = -179.8333;
:geospatial_lon_max = 179.8333;
:geospatial_vertical_min = 19.5;
:geospatial_vertical_max = 19.5;
:time_coverage_start = "2010-01-27T04:11:21Z";
:time_coverage_end = "2010-01-27T09:39:41Z";
:time_coverage_duration = "PT05:28:20";
:standard_name_vocabulary = "CF-1.4";
:license = "The data may be used and redistributed for free but is not intended
for legal use, since it may contain inaccuracies. The data is provided without
warranties of merchantability and fitness for a particular purpose. The provider
assumes no legal liability for the accuracy, completeness, or usefulness, of any
information contained in the data.";

// Variable data
// -----
data:
    time = 1264575331;
    time_bounds = 1264565481, 1264585181;
    level = 19.5;

```

```
}
```


Section 5: References

- *CoastWatch Software Library and Utilities User's Guide*, Version 3.2.4, Development Prerelease, Revised July 1, 2009.
- *Comparison of Metadata Standards*, Peter Hollemans, January, 2010 (Keynote slides).
- *NetCDF Climate and Forecast (CF) Metadata Conventions*, Version 1.4, February 27, 2009: <http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.4/cf-conventions.html>
- *COARDS netCDF profile: Conventions for the standardization of NetCDF files*, Accessed November 25, 2009: http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html
- *Data Content Standard for Remotely Sensed Ocean Color Data*, IOOS Data Integration Framework - 2008, Version 1.0, July 7, 2008.
- *Mapping NCDDS Ocean Color DCS v1.0 to CoastWatch netCDF v0.61*, Dave Foley et. al., 1998.
- *The Recommended GHRSSST Data Processing Specification Version 2.0*, Revision 0.6 L2P Product Specification, June 10, 2009.
- *Unidata Dataset Discovery Attribute Conventions*: <http://www.unidata.ucar.edu/software/netcdf-java/formats/DataDiscoveryAttConvention.html>
- *GCMD Science Keywords*: <http://gcmd.gsfc.nasa.gov/Resources/valids/>
- *CF Convention Compliance Checker*: <http://puma.nerc.ac.uk/cgi-bin/cf-checker.pl>