

**Southeast Atlantic Coastal Ocean  
Observing System**

**NetCDF Standard: SEACOOS CDL v2.0**

**Manual by:**

Sara Haines

**Standard Drafted by:**

(in alphabetical order)

Tom Cook, Jeremy Cothran, Jeff Donovan, Ed Kearns, Sara Haines, Trent Moore, Charlton Purvis, Vembu Subramanian, Elizabeth Williams

**December 9, 2004**

# Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. General Background</b>	<b>1</b>
2.1 Data Management Coordinating Committee (DMCC)	1
2.2 Information Management within SEACOOS	2
2.3 Data Scope	3
<b>3. File Format Background</b>	<b>4</b>
3.1 Network Common Data Form (netCDF)	4
3.2 Common Data Language (CDL)	5
3.3 NetCDF Conventions	6
<b>4. Publishing and Sharing Data</b>	<b>7</b>
<b>5. Data Format Categories</b>	<b>7</b>
<b>6. Required Variables</b>	<b>10</b>
6.1 Time Variable	10
6.2 Longitude Variable	10
6.3 Latitude Variable	10
6.4 Vertical Variable	11
<b>7. Data Variables</b>	<b>11</b>
7.1 Wind	11
7.2 Water Temperature	12
7.3 Water Level	11
7.4 Ocean Currents	13
<b>8. Filename</b>	<b>14</b>
<b>9. Global Attributes</b>	<b>14</b>
9.1 Required Global Attributes	14
9.2 Recommended Global Attributes	16
<b>10. Variable Attributes</b>	<b>18</b>
10.1 Required Variable Attributes	18
10.2 Other Variable Attributes	18
<b>11. Standard Names</b>	<b>20</b>
<b>12. Units</b>	<b>20</b>
<b>13. Future Directions</b>	<b>20</b>
<b>14. References</b>	<b>21</b>
<b>Appendix A</b>	<b>22</b>
A.1 fixed-point	22
A.2 fixed-profiler	24
A.3 fixed-map	26
A.4 moving-point-2D	28
A.5 moving-point-3D	30
A.6 moving-profiler	32

## 1. Introduction

This document describes a set of conventions adopted by the SouthEast Atlantic Coastal Ocean Observing System (SEACOOS) to promote sharing and exchange of data from disparate ocean observing and remote-sensing data sources. These data include observations from buoys, offshore towers, ships, tide- and stream-gauging stations, acoustic profilers, radar, aircraft, satellites and other remote mapping sensors. SEACOOS is a regional partnership that has initiated an integrated coastal ocean observing system for a four-state (North Carolina, South Carolina, Georgia and Florida) region of the southeast coastal U.S. (Seim, et al., 2003, Seim, et al, 2002). SEACOOS partners publish near real-time data in netCDF (network Common Data Form) format and make it available on the Internet through OPeNDAP (Open source Project for a Network Data Access Protocol) Servers.

This document details the agreed upon netCDF format categories, required variables, and required and recommended attributes. The name of this standard is “SEACOOS CDL”. CDL stands for Common Data Language. The current accepted version is 2.0.

SEACOOS CDL provides conformity to develop automated search and aggregation tools. It is also flexible to allow SEACOOS to coordinate many different sources of data into a merged dataset and provide unique graphical displays of these merged data in near real-time. SEACOOS CDL provides an unambiguous output format for SEACOOS partners. It allows anyone to incorporate their observational data into powerful displays with similar data.

## 2. General Background

### 2.1 Data Management Coordinating Committee (DMCC)

The Data Management Coordinating Committee (DMCC) of SEACOOS establishes hardware, software, data format, and metadata conventions in order to help partners make data distribution systems operational. This committee is made up of data management personnel from each of the SEACOOS partner institutions. The DMCC is responsible for developing and documenting SEACOOS CDL. One main goal of the DMCC is “to provide access via a web interface to SEACOOS-supported, quality controlled data and associated metadata and derived products.” Another goal of this committee is “to implement the OPeNDAP software solution as a form of data sharing.” The OPeNDAP protocol has been designated by Ocean.US as a component for the delivery of data in a sustainable Integrated Ocean Observing System (IOOS).

The main obstacles that the DMCC overcame in drafting the SEACOOS CDL standard were how to unambiguously describe the time and vertical coordinates. A number of choices had to be made to provide consistency but remain flexible for the different sources of data that observational data presents. The first choice was that each partner would provide data in netCDF files and share their data in a distributed forum by using OPeNDAP. NetCDF is the most tested

and supported file formats under OPeNDAP. Other data formats such as XML, HDF, and RDBMS will probably be investigated by the DMCC in the future.

## 2.2 Information Management within SEACOOS

Ocean data and information management within SEACOOS (<http://www.seacoos.org>) occurs in four steps: aggregation and storage, normalization, visualization and dissemination (Purvis, et al., 2004). Figure 2.1 shows how these four-steps are interrelated and how data flow from one step to the next.

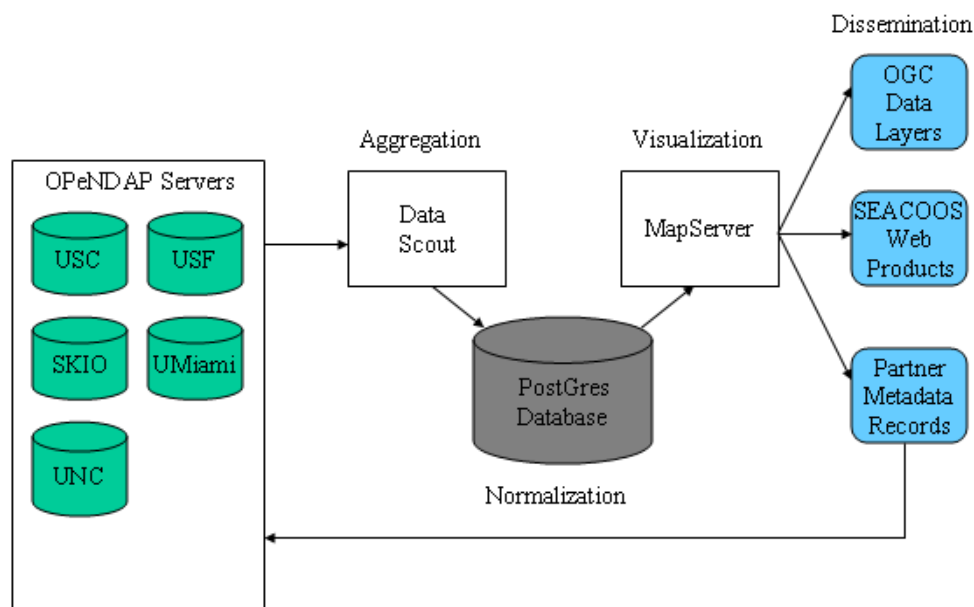


Figure 2.1. SEACOOS information management components are illustrated in this flow chart. It shows how the data flow from posting data on OPeNDAP Servers, to aggregation and database normalization, to visualization and then to the many levels of disseminated information.

SEACOOS partners post data to the Internet and their own OPeNDAP Server in netCDF format using the SEACOOS CDL standard. In addition to data in netCDF, many remote-sensing satellite products are provided and incorporated using Portable Network Graphics (PNG) raster image format. Model products are posted in netCDF with an agreed upon netCDF model standard (Blanton, et al. 2004).

All these data are then aggregated by the SEACOOS portal. Since many sources report on different time scales and have varied spatial coverage, the data are normalized to a central database that is coordinated in space and time. The database is then accessed by an online Geographic Information System (GIS) application. The SEACOOS Internet visualization tool provides traditional mapping, animation, and plotting capabilities (<http://www.seacoos.org>). Finally, data are disseminated at many levels. Metadata records maintain access to the original

data and original OPeNDAP Server. Also, Open GIS Consortium (OGC) data layers are made available using emerging technologies for such services as Web Map Service (WMS) and Web Feature Service (WFS). Customized access and interface approaches are being developed by SEACOOS to target specific audiences such as fisheries, emergency management, and recreation (Purvis, et al., 2004). Figure 2.2 shows the SEACOOS Internet interactive map display.

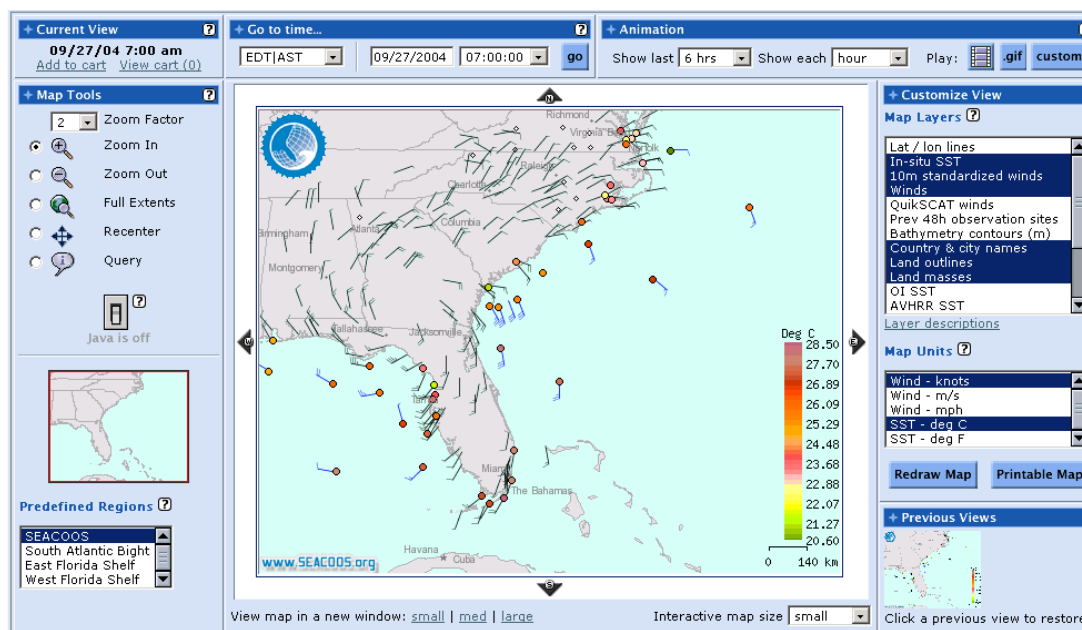


Figure 2.2. This figure is a “screen-shot” of the SEACOOS interactive map display and interface (<http://www.seacoos.org>). The data displayed are wind and sea-surface temperature observations from September 27, 2004 showing the remnants of Hurricane Ivan.

## 2.3 Data Scope

The scope of data that has been tested thus far with SEACOOS CDL is data from near real-time observing stations and platforms from a number of data sources and data variables. Table 2.1 lists the SEACOOS partners contributing data in netCDF files conforming to SEACOOS CDL. Also listed in Table 2.1 are federal data sources that have real-time Internet accessible data that are automatically “screen-scraped” and reformatted using SEACOOS CDL standard for netCDF. Table 2.2 lists the data variables that have been tested.

Since the purpose of SEACOOS CDL is to present real-time data, only the last 48 hours of observations should be stored in the netCDF file and then updated as new data are available. However, this format could also be used to disseminate and store data for historical archives.

**Table 2.1. Data Sources****Regional:**

**University of South Florida/COMPS**  
**University of Miami/RSMAS-Explorer of the Seas**  
**Skidaway Institute of Oceanography/SABSOON**  
**University of South Carolina/Caro-COOPS**  
**University of North Carolina at Chapel Hill/NCCOOS**

**National:**

**National Data Buoy Center/National Weather Service(NWS)/NOAA**  
**Center for Operational Oceanographic Products and Services/National**  
**Ocean Service/National Oceanographic and Atmospheric Administration**  
**(NOAA)**  
**United States Geological Survey (USGS)/Water Quality and River Levels**  
**National Weather Service/Federal Aviation Administration (FAA)**

**Table 2.2. Data Variables**

**Wind speed and direction**  
**Wind gust**  
**Horizontal and vertical wind components**  
**Sea-surface temperature**  
**Water-level**  
**Ocean current speed and direction**  
**Horizontal and vertical ocean current components**

### 3. File Format Background

#### 3.1 Network Common Data Form (netCDF)

NetCDF is one of many available file formats for storing real-time observational data. It is a relatively simple file format that is self-describing and portable. “Self-describing” means it contains data about the data, such as units and variable descriptions. “Portable” means that the data in a dataset is represented in a form that can be accessed by computers with different methods of storing integers, characters, and floating-point numbers. It has a strong set of functional libraries that can compress, subset and transform data. The netCDF software libraries and documentation are available from Unidata (<http://my.unidata.ucar.edu/content/software/netcdf/>). Many manipulation and display software utilities are available for netCDF (<http://my.unidata.ucar.edu/content/software/netcdf/software.html>).

NetCDF files contain the data in variables, which can be single numbers, vectors, or multi-dimensional arrays. Variables can be of data type `char`, `byte`, `short`, `int`, `float` or `real` and `double`. netCDF files can contain data that can be organized into a collection of named array variables with named attributes which is widely applicable to many situations. However, netCDF has some limitations. For instance, a netCDF file size cannot exceed two Gigabytes. Also, netCDF data model does not support nested data structures such as trees, nested arrays, or other recursive structures. The NetCDF User's Guide thoroughly documents the many assets and few limitations of netCDF (<http://my.unidata.ucar.edu/content/software/netcdf/docs.html>).

### 3.2 Common Data Language (CDL)

Common Data Language (CDL) is textual notation that describes the netCDF object and it is human readable. The netCDF utility `ncdump` converts netCDF object binary to CDL text. The netCDF utility `ncgen` creates netCDF binary file from well-formed CDL text file.

A CDL description of a netCDF dataset takes the form

```
netCDF name {  
    dimensions: ...  
    variables: ...  
    data: ...  
}
```

where the *name* is used only as a default in constructing file names by the `ncgen` utility. The CDL description consists of three optional parts, introduced by the keywords `dimensions`, `variables`, and `data`. NetCDF dimension declarations appear after the `dimensions` keyword, netCDF variables and attributes are defined after the `variables` keyword, and variable data assignments appear after the `data` keyword. CDL statements are terminated by a semicolon. Spaces, tabs, and newlines can be used freely for readability. Comments in CDL follow the characters `/**` on any line (Rew, et al, 1997). A simple CDL example is provided in Figure 3.1.

```
netcdf example { // example CDL for netCDF notation  
dimensions: // dimension names are declared first  
    time = 5 ;
```

```

    lat = 3 ;
    lon = 2 ;
variables:          // variable types, names, shapes, attributes
    double time(time) ;
        time:units = "hours since 1994-01-01 00:00:00" ;
        time:long_name = "time" ;
    float lat(lat) ;
        lat:units = "degrees_north" ;
        lat:long_name = "latitude" ;
    float lon(lon) ;
        lon:units = "degrees_east" ;
        lon:long_name = "longitude" ;
    float temp(time, lat, lon) ;
        temp:long_name = "Surface temperature in degrees C" ;
        temp:units = "deg_C" ;
// global attributes
    :source = "NetCDF Users Guide";
data:              // data assignments
    time = 0.5, 1.5, 2.5, 3.5, 4.5 ;
    lat = 54.2, 54.4, 54.6 ;
    lon = 2.0, 2.5 ;
    temp = 34.5, 31.2, 23.7, 19.6, 35.8, 29.2, 24.4, 5.6, 7.2,
8.1, 18.6, 15.2, 13.1, 4.6, 3.7, 8.2, 9.7, 34.2, 26.7, 28.7, 2.1, 3.4,
5.6, 7.8, 9.0, 10.2, 11.2, 11.6, 11.7, 11.8 ;

// Fill arrays by outermost dimension in CDL (other programming
// languages might take a different index order). Indexing in CDL for
// this example is as follows:
//     temp = temp(1,1,1), temp(1,1,2), temp(1,2,1), temp(1,2,2),
//           temp(1,3,1), temp(1,3,2), temp(2,1,1), etc.

```

Figure 3.1 displays a simple CDL example with basic components: dimensions, variables and data.

### 3.3 NetCDF Conventions

SEACOOS CDL uses the NetCDF Climate and Forecast (CF) Metadata Conventions v1.0 wherever possible. However, this does not limit SEACOOS CDL to using just one convention where an attribute or concept from another convention might prove useful. After the DMCC conducted a survey of existing conventions and standards for netCDF, it was decided to adopt CF as the starting point.

The purpose of the CF conventions is to require conforming datasets to contain sufficient metadata so that they are self-describing in the sense that each variable in the file has an associated description of what it represents, including physical units if appropriate, and that each value can be located in space (relative to earth-based coordinates) and time (Eaton, et al, 2001). The main digression from CF that SEACOOS took is that SEACOOS CDL gets standard names from the SEACOOS Data Dictionary instead of the CF Standard Name Table. Section 11 on Standard Names contains more information about the SEACOOS Data Dictionary.

## 4. Publishing and Sharing Data

SEACOOS is dedicated to implement, test, and evaluate the OPeNDAP software solution as a form of data sharing. Each SEACOOS partner makes their data accessible to the Internet through an OPeNDAP-server (Seim, et al., 2002). This provides access to distributed data in two ways: simple file download or client-access. Both methods have their advantages and disadvantages. Another important aspect of data sharing is cataloging the OPeNDAP-servers and URL locations with SEACOOS and other metadata clearing-houses.

Simple file download is available because the OPeNDAP-server resides on an existing web-server. Basic web-service allows file download capability. A user can download the published netCDF file but must have the appropriate utility and libraries to open and manipulate the stored data.

The client-access method is achieved through the use of OPeNDAP. OPeNDAP is a client-server based distributed system for access to oceanographic data over the Internet that supports many underlying file formats one of which is netCDF. OPeNDAP supports use of the netCDF interface for clients. The server supports use of URL notation for accessing netCDF data from remote sites running the OPeNDAP server. OPeNDAP also allows subsetting, searching and seamless data access into higher level tools such as MATLAB. The server, client and library source-code and binaries are available from <http://www.opendap.org>.

By cataloging the OPeNDAP-server or URL with regional and national catalogs, it is ensured that the data are found and utilized. By listing an OPeNDAP URL with SEACOOS, the data can be found by the SEACOOS data scout and uploaded on a regular basis to the SEACOOS data display. Contact SEACOOS by email ([data@seacoos.org](mailto:data@seacoos.org)) to add a new OPeNDAP URL. Furthermore, by filing metadata records through catalog facilities such as the Global Change Master Directory (GCMD) or Federal Geographic Data Committee (FGDC) compliant tools, such as Meta-door (<http://carocoops.org/metadoor>), ensures that the data are found and utilized through FGDC search tools. The more ways to search and locate distributed data the better.

## 5. Data Format Categories

In order to represent many different data sources, a canonical set of data models or format categories are used to represent *in situ* and remote ocean observations. These data may come from *in situ* instruments located on many different types of platforms such as buoys, moorings, off-shore towers, tide- and stream-gauging stations, ships, towed-bodies, weather balloons, gliders, drifters, and airplanes. The data may also come from many remote sensing instruments (active or passive, optic, acoustic or microwave) located on different platforms like satellites, aircraft, ships, radar stations, and moored frames.

The format categories are based on how each of these platforms move (fixed or moving) and how the data are collected (point or profiler or map). By viewing each measurement or field

(temperature, salinity, velocity, etc) as a function of time and location(s) in the same coordinate system, we are able to merge data from different sources.

The coordinate system used by SEACOOS CDL is the Cartesian coordinate system. The three spatial dimensions are longitude (E-W), latitude (N-S), and depth (or height); denoted as  $x$ ,  $y$  and  $z$ , respectively. Usually a right-handed coordinate system is used where  $x$  is positive towards the east,  $y$  is positive towards the north and  $z$  is positive upwards; but sometimes in the ocean  $z$  is taken to be positive downward and a left-handed coordinate system is used.

Time ( $t$ ) and the three coordinates ( $x$ ,  $y$  and  $z$ ) are the independent variables. The field or measurement is the dependent variable.

Six categories emerge by viewing each measurement, for example temperature ( $T$ ) and velocity components ( $u,v$ ) as a function of the independent variables. These six categories are *fixed-point*, *fixed-profiler*, *fixed-map*, *moving-point-2D*, *moving-point-3D*, and *moving-profiler*. (A *moving-map* category is omitted. Not many oceanographic sensors fit this category.) These are summarized in Table 5.1.

Table 5.1. SEACOOS CDL Format Categories				
Format Category	Independent Variables		Dependent Variables	
	Cartesian	netCDF	Cartesian	netCDF
fixed-point (buoy, tower)	$t = t_1, t_2, t_3$ $x_0, \text{constant}$ $y_0, \text{constant}$ $z_0, \text{constant(s)}$	time(time) lon(lon) lat(lat) z(z)	$T(x_0, y_0, z_0, t)$ $u(x_0, y_0, z_0, t)$ $v(x_0, y_0, z_0, t)$	$T(\text{time})$ $u(\text{time})$ $v(\text{time})$
fixed-profiler (wind profiler, ADCP)	$t = t_1, t_2, t_3$ $x_0, \text{constant}$ $y_0, \text{constant}$ $z = z_1, z_2, z_3$	time(time) lon(lon) lat(lat) z(z)	$T(x_0, y_0, z, t)$ $u(x_0, y_0, z, t)$ $v(x_0, y_0, z, t)$	$T(\text{time}, z)$ $u(\text{time}, z)$ $v(\text{time}, z)$
fixed-map (HF Radar, Satellite Imagery)	$t = t_1, t_2, t_3$ $x = x_1, x_2, x_3$ $y = y_1, y_2, y_3$ $z_0, \text{constant}$	time(time) lon(lon) lat(lat) z(z)	$T(x, y, z_0, t)$ $u(x, y, z_0, t)$ $v(x, y, z_0, t)$	$T(\text{time}, x, y)$ $u(\text{time}, x, y)$ $v(\text{time}, x, y)$
moving-point-2D (ship, floating drifter)	$t = t_1, t_2, t_3$ $x = x(t)$ $y = y(t)$ $z_0, \text{constant(s)}$	time(time) lon(time) lat(time) z(z)	$T(x(t), y(t), z_0, t)$ $u(x(t), y(t), z_0, t)$ $v(x(t), y(t), z_0, t)$	$T(\text{time})$ $u(\text{time})$ $v(\text{time})$
moving-point-3D (aircraft, towed undulating vehicle, sea glider, lagrangian drifter)	$t = t_1, t_2, t_3$ $x = x(t)$ $y = y(t)$ $z = z(t)$	time(time) lon(time) lat(time) z(time)	$T(x(t), y(t), z(t), t)$ $u(x(t), y(t), z(t), t)$ $v(x(t), y(t), z(t), t)$	$T(\text{time})$ $u(\text{time})$ $v(\text{time})$
moving-profiler (ship-mounted ADCP, CTD surveys)	$t = t_1, t_2, t_3$ $x = x(t)$ $y = y(t)$ $z = z_1, z_2, z_3$	time(time) lon(time) lat(time) z(z)	$T(x(t), y(t), z, t)$ $u(x(t), y(t), z, t)$ $v(x(t), y(t), z, t)$	$T(\text{time}, z)$ $u(\text{time}, z)$ $v(\text{time}, z)$

Data providers should provide their data in file structures that fall into one of the six formatting categories listed in Table 5.1. This also helps the data provider determine if they need to create multiple files while limiting the number of possible configurations. CDL examples for each format category are given in Appendix A. Each file should contain variables that describe and provide data for the four independent variables whether they are used as a coordinate variable or are a function of time (t).

There is a special provision for `fixed-point` and `moving-point-2D`, to accommodate data from sensors mounted at different heights in one netCDF file for the platform. For example, some buoy stations have wind sensors mounted above at 5 or more meters above the buoy while other atmospheric sensors are located on the buoy frame only 1 meter above the sea surface and in-water packages measure at 1 or 2 meters below the sea surface. For these cases, in the netCDF, a

fully-described vertical (z) independent variable is provided with the three different instrument heights, but the constant z value is also provided with the specific dependent variable (e.g. `wspd: z=5;`).

## 6. Required Variables

As described in the previous section on Format Categories, SEACOOS CDL files must always contain the four independent variables (time, longitude, latitude and vertical) that describe the spatial and temporal setting of the data. Each independent variable must be described in netCDF at the beginning of the variable section. Each independent variable must always explicitly include the `units` and `standard_name` attributes.

### 6.1 Time Variable

For real-time data, only the last 48 hours of observations should be retained in the netCDF file and should be updated as new data are available. Only one time frame for SEACOOS CDL is allowed and that is the number of seconds since January 1, 1970 at 00Z. All time data must be converted to this time frame. The DMCC may consider allowing more flexible time values in future versions of this standard.

```
long time(time);
  time: standard_name = "time";
  time: units = "sec since 1970-1-1 00:00:00";
  time: axis = "T";
```

### 6.2 Longitude Variable

Longitude may be represented as a coordinate variable, (e.g. `lon(lon)`), or a function of time, e.g. `lon(time)`.

```
float lon(lon); //or lon(time)
  lon: standard_name = "longitude";
  lon: units = "degrees_east" ;
  lon: reference = "geographical coordinates" ;
  lon: axis = "X";
```

### 6.3 Latitude Variable

Latitude may be represented as a coordinate variable (e.g. `lat(lat)`) or a function of time (e.g. `lat(time)`).

```
float lat(lat); //or lat(time)
  lat: standard_name = "latitude";
  lat: units = "degrees_north";
  lat: reference = "geographical coordinates";
  lat: axis = "Y";
```

## 6.4 Vertical Variable

The vertical variable represents the height or depth of the data measurement. The `positive` variable attribute provides the direction of positive. The vertical variable should be described unambiguously by providing the additional attribute of `axis` with a value of `z`. The vertical variable may be represented as a coordinate variable (e.g. `z(z)`) or a function of time (e.g. `z(time)`).

```
float z(z); //or z(time)
    z: standard_name = "height";
    z: units = "m";
    z: reference = "mean sea level (MSL)";
    z: positive = "up";
    z: axis = "Z";
```

## 7. Data Variables

Requirements for representing winds, water temperature, water level, and ocean currents are described in this section. Each dependent variable must always explicitly include the `units` and `standard_name` attributes.

### 7.1 Winds

Horizontal wind requires two dependent variables to describe the speed and direction of the wind or the eastward and northward components relative to True North. The variable attribute `wdir: reference` is used to define the reference direction such as True North or Magnetic North. The variable attribute (`wspd: z` and `wdir: z`) gives the height of the wind meter in relation to the description and units used by the vertical variable. The examples given are for a platform and sensor where the vertical variable is constant. The first example shows how to represent wind magnitude and direction (`wspd` and `wdir`).

```
float wspd(time);
    wspd: standard_name = "wind_speed";
    wspd: units = "m s-1";
    wspd: z = 45;
float wdir(time);
    wdir: standard_name = "wind_from_direction";
    wdir: units = "degrees";
    wdir: reference = "clockwise from true north";
    wdir: valid_range = 0.,360.;
    wdir: z = 45;
```

Data for wind gusts are provided with a separate variable (`wgust`). Wind direction data and metadata must be provided if not already provided with wind speed.

```
float wgust(time);
  wgust: standard_name = "wind_gust";
  wgust: units = "m s-1";
  wgust: z = 45;
```

The following example uses eastward and northward components (u and v).

```
float u(time) ;
  u:standard_name = "eastward_wind" ;
  u:units = "m s-1" ;
  u:z = 10. ;
float v(time) ;
  v:standard_name = "northward_wind" ;
  v:units = "m s-1" ;
  v:z = 10. ;
```

If a vertical wind component is measured then this can be provided with an upward wind component variable (w).

```
float w(time) ;
  w:standard_name = "upward_wind" ;
  w:units = "m s-1" ;
  w:z = 10. ;
```

## 7.2 Water Temperature

Ocean water temperature is a scalar. The variable attribute (*sst: z*) gives the depth of the temperature measurement in relation to the description and units used by the vertical variable. The example given is for a platform and sensor where the vertical variable is constant.

```
float SST(time);
  SST: standard_name = "water_temperature";
  SST: units = "degrees Celsius";
  SST: z = -5;
```

## 7.3 Water Level

Water level is a scalar. The sea surface is defined mathematically by statistical averaging of the observed values and is affected by local topography (Gill and Schultz, 2001). Water level data in the netCDF file is required to have the attribute *reference* value set to one of the particular phases of the tidal cycle (e.g. MLLW, MLW, etc). It is in the best interest of the data provider to also provide the statistically determined local offset to other tide phases and local bench mark if available and the averaging period and tidal epoch. The more offsets that the data provider can provide the more likely data users can transform water-level data to their reference level and make use of it.

The variable attribute (*wl: z*) gives the height of the water level measurement in relation to the description and units used by the vertical variable. The example given is for a platform and sensor where the vertical variable is constant. The first example shows how to provide all local reference differences so others may transform the data provided to other reference levels.

```

float wl(time) ;
  wl:standard_name = "water_level" ;
  wl:units = "m" ;
  wl:reference = "MLLW" ;
  wl:reference_to_MLLW = 0. ;
  wl:reference_to_MLW = 0.05899999999999999 ;
  wl:reference_to_MSL = 0.785 ;
  wl:reference_to_MTL = 0.775 ;
  wl:reference_to_DTL = 0.796 ;
  wl:reference_to_MHW = 1.492 ;
  wl:reference_to_MHHW = 1.593 ;
  wl:reference_to_NAVD88 = 0.858 ;
  wl:reference_to_NAVD88_benchmark_id = "KV3519" ;
  wl:reference_tide_datum_time_period = "January 1983 - December 2001" ;
  wl:reference_tide_datum_epoch = "1983-2001" ;
  wl:z = 0. ;

```

The second example shows the minimum variable attributes to provide for water level data.

```

float wl(time) ;
  wl:standard_name = "water_level" ;
  wl:units = "m" ;
  wl:reference = "MLLW" ;
  wl:reference_to_MLLW = 0. ;
  wl:z = 0. ;

```

## 7.4 Ocean Currents

Horizontal ocean currents require two dependent variables to describe the speed and direction of the current or the eastward and northward components. The variable attribute `water_dir:reference` is used to define the reference direction such as True North or Magnetic North. The variable attribute (`water_spd:z` and `water_dir:z`) gives the height of the current measurement in relation to the description and units used by the vertical variable. The examples given are for a platform and sensor where the vertical variable is constant. The first example shows how to represent current magnitude and direction (`water_spd` and `water_dir`).

```

float water_spd(time);
  water_spd: standard_name = "current_speed";
  water_spd: units = "cm s-1";
  water_spd: z = 0;
float water_dir(time);
  water_dir: standard_name = "current_to_direction";
  water_dir: units = "degrees";
  water_dir: reference = "clockwise from true north";
  water_dir: valid_range = 0.,360.;
  water_dir: z = -5;

```

The following example uses eastward and northward components (`water_u` and `water_v`).

```

float water_u(time, lon, lat) ;
  water_u:standard_name = "eastward_current" ;
  water_u:units = "cm s-1" ;
  water_u:z = -5. ;
float water_v(time, lon, lat) ;
  water_v:standard_name = "northward_current" ;
  water_v:units = "cm s-1" ;
  water_v:z = -5. ;

```

If a vertical current component is measured then this can be provided with an upward current component variable (`water_w`).

```
float water_w(time) ;
    water_w:standard_name = "upward_current" ;
    water_w:units = "m s-1" ;
    water_w:z = -5. ;
```

## 8. Filename

The file name extension of each netCDF file should be `.nc`.

To promote easy real-time aggregation, netCDF files with real-time observations should have `latest` as part of the filename proceeding the suffix `.nc`. The most general filename form is `latest.nc`.

In addition, while it is not mandatory, it is recommended that a partner acronym, platform identification, and instrument package type be added to the filename to distinguish the source of the data by virtue of just a directory listing. The general form that is recommended is `partner-platform-package-latest.nc`. For example, the most recent 48 hours of data from University of South Florida meteorological buoy (CMP4) at Pasco County, Florida, should be stored in the netCDF file named `usf-CMP4-met-latest.nc`.

## 9. Global Attributes

In this section the required and recommended global attributes are described. An example entry is provided for each description. These attributes provide information about where the data came from and what has been done to them. Whenever an attribute is present as both a global and variable attribute, the variable attribute takes precedence. The CDL notation for defining a global attribute is

```
:attribute_name = list_of_values;
```

### 9.1 Required Global Attributes

The following required global attributes must be present in each netCDF file. If an attribute does not have a value, still include it in the file but with some “not applicable” statement.

```
:title
    Text. A succinct description for what is in the file.
    Example: "SABSOON data for US Navy Tower R2"
:institution
    Text. The name of the institution responsible for creating the file.
    Example: "Skidaway Institute of Oceanography (SkIO)"
:institution_url
```

Text. The URL of the institution that created the file.

Example: "http://comps.marine.usf.edu/"

:institution\_dods\_url

Text. The OPENDAP data access URL where the file was originally stored.

Example: "http://comps.marine.usf.edu/DODS/"

:contact

Text. Contact information for the file.

Example: "Data Manager (data\_manager@institution.edu)"

:Conventions

Text. Name of the conventions used for the file. If there is another convention followed, then add this to the character array. For example, use "CF-v1.0;COARDS" if following the COARDS and CF-v1.0. Note the capitalization of the attribute name.

Example: "CF-1.0".

The following global attributes are required by the SEACOOS data scout.

:format\_category\_code

Text. The specified SEACOOS format or data model of the netCDF file. These formatting categories are described in more detail in Section 5. Only one of the options below is a valid entry for this attribute.

Example: "fixed-point"

fixed-point	(buoy, tower)
moving-point-2D	(ship, floating drifter)
moving-point-3D	(sea glider, towed undulating vehicle, aircraft)
fixed-profiler	(wind profiler, ADCP)
moving-profiler	(ship-mounted ADCP, CTD surveys)
fixed-map	(HF Radar surface, satellite imagery)

:institution\_code

Text. A short distinctive name (with no spaces) that is used by the institution responsible for creating the data file.

Examples: "usf", "skio", "carocoops", "ndbc", "nws", "nos", "usgs"

usf	= University of South Florida
skio	= Skidaway Institution of Oceanography
umiami	= University of Miami
nccoos	= North Carolina Coastal Ocean Observing System
carocoops	= Carolina Ocean Observing and Prediction System

vos	= Voluntary Observing Ship
sk	= SeaKeepers
ndbc	= National Data Buoy Center
nws	= NWS Airport Data
nos	= National Ocean Service
usgs	= United States Geological Survey

:platform\_code

Text. A short distinctive name or code (with no spaces) that identifies the site or platform that the sensor or sensor package is mounted. The code will be issued by the institution responsible for that platform data.

Examples: "R2", "KJAX", "KS011", "ELWX5", "PAS", "C10", "41009", "FBIS1"

KJAX = Jacksonville Florida Airport  
 KOBX = Outerbanks North Carolina Airport  
 KS011 = a VOS SeaKeeper Ship  
 ELWX5 = Explorer of the Seas  
 R2 = one of the SABSOON towers  
 SPAG1 = R2 but is NDBC identifier  
 PAS = USF/COMPS buoy  
 C10 = another USF buoy  
 41009 = NDBC buoy  
 FBIS1 = NDBC CMAN station

:package\_code

Text. A short distinctive name (with no spaces) for data from sensor or the package of sensors contained within the file.

Examples: "met", "metar", "ASIMET", "NGWLMS", "SK", "weatherpak".

The following global attributes are required for SEACOOS data format version tracking and verification.

:format

Text. The netCDF data format for real-time data posted for the SEACOOS data scout.

This attribute may be deprecated by the more widely used and accepted " :Conventions" attribute.

Example: "SEACOOS RT" or "seacoos\_rt"

:seacoos\_rt\_version

Text. The version number of the netCDF data format for real-time data posted for the SEACOOS data scout. This attribute may be deprecated by the more widely used

" :Conventions" attribute.

Example: "2.0"

## 9.2 Recommended Global Attributes

The following global attributes are recommended global attributes often used by the netCDF user community. These are more fully documented in the netCDF User's Guide and CF-v1.0.

:\_FillValue

Scalar value of the same type as the variable. This value denotes valid data that is not present or outside the valid range for the variable. It is also used to pre-fill a variable and is treated specially by the netCDF library function if "nofill" is not specified. This value will be returned by the netCDF library if the data are never written. Note the underscore character and capitalization. If the \_FillValue attribute is presented with the global attributes then each variable inherits this value unless specified with a variable attribute.

Example: -9999

:missing\_value

Scalar value of the same type as the variable. This attribute is not treated in any special way by the netCDF library or conforming generic applications, but is often useful documentation and may be used by specific applications.

Example: -9999

:source  
Text. Succinct characterization of the source of the data.  
Example: "fixed-platform observation"

:references  
Text. Article, book, or website reference used to describe the dataset and/or collection program.  
Example: "http://comps.marine.usf.edu/"

:history  
Text. An audit trail with a line for each invocation of a program that modifies the dataset.  
Example: "Created by Data Manager"

The following global attributes are recommended. They were discussed by the DMCC but never approved or fleshed out in great detail to be required by any SEACOOS processing. All of these global attributes are useful but not required.

:process\_level  
Text. Simple explanation of level of processing done to the data contained in the netCDF file.  
Example: "raw data"

:metadata\_url  
Text. URL of metadata record where more supporting information about the data can be obtained.  
Example: "http://comps.marine.usf.edu/metadata/usfseacoos.htm"

:start\_date  
Text. Date and time stamp of first sample of data.  
Example: "2004-10-16 13:52:20 UTC"

:end\_date  
Text. Date and time stamp of last sample of data.  
Example: "2004-10-18 13:28:19 UTC"

:release\_date  
Text. Date and time stamp for the release date of the data.  
Example: "2004-10-18 13:00:00 UTC"

:creation\_date  
Text. Date and time stamp for creation date of the data.  
Example: "2004-10-18 13:49:22 UTC"

:modification\_date  
Text. Date and time stamp for most recent modification date of the data.  
Example: "2004-10-18 13:49:22 UTC"

:instrument\_type  
Text. This attribute was initially intended to help describe the type of instrument or package of instruments used to collect the data in the dataset but can ultimately be deprecated by use of "platform\_code" and "package\_code" attributes.  
Example: "met"

:data\_type

Text. This attribute was initially intended to help describe the type of instrument or package of instruments used to collect the data in the dataset but can ultimately be deprecated by use of “:platform\_code” and “:package\_code” attributes.

Example: “met”

## 10. Variable Attributes

In this section the required and recommended variable attributes are described. An example entry is provided for each description. They provide specific information about a variable. Whenever an attribute is present as both a global and variable attribute, the variable attribute takes precedence. The CDL notation for defining a variable attribute is

```
variable_name:attribute_name = list_of_values;
```

### 10.1 Required Variable Attributes

The following required variable attributes should be present with each netCDF variable so that either one can obtain more information about the dataset or automated software can process the attributes. If an attribute does not have a value, an option for each variable attribute will be the “not applicable” statement.

:standard\_name

Text. A unique string that unambiguously identifies the physical nature of the variable stored in the dataset. Only one valid standard name chosen from the SEACOOS Data Dictionary can be used. The SEACOOS Data Dictionary is discussed in Section 11. In case that no standard name is provided in the Data Dictionary, place “none” in the text field. No spaces or tabs are allowed in the value string.

Example: “water\_temperature” or “northward\_current” or “none”

:units

Text. The physical units of the variable stored in the dataset. In the case of dimensionless units, place “none” in the text field. The text string should conform to Unidata’s `udunits` package to maintain portability of the data units described. More information about Unidata’s `udunits` package is provided in Section 12.

Example: “m s-1” “meter second-1” or “none”

### 10.2 Other Variable Attributes

Some of the following variable attributes may be recommended while others may be required for a specific dataset variable, for example water-level or format category, for example fixed-point observation.

`:long_name`  
Text. Descriptive name that indicates the variable's content which, for example, may be used for labeling plots.  
Example: "Sea Surface Temperature";

`:short_name`  
Text. Descriptive acronym or short version of a longer name that indicates the variable's content. This does not have to match the variable name used by the netCDF file. No spaces or tabs are allowed in the value string.  
Example: "sst" or "wtemp"

`:_FillValue`  
Scalar value, same data type as the variable. This value denotes valid data that is not present or outside the valid range for the variable or pre-fill. Note the underscore character and capitalization. If the `_FillValue` attribute is presented with the global attributes then each variable inherits this value unless specified as different with a variable attribute.  
Example: -9999;

`:missing_value`  
Scalar value, same data type as the variable. This attribute is not treated in any special way by the netCDF library, but is often useful documentation and may be used by specific applications or for second fill value.  
Example: -9999;

`:valid_range`  
Scalar values, same data type as the variable. Values that describe the geophysical range of the variable.  
Example: 0., 360;

`:reference`  
Text. Description of the reference datum for the variable.  
Example: "clockwise from true North" or "Mean Low-low Water (MLLW)"

`:positive`  
Value. Description of direction of positive if it cannot be inferred from the units. Most often used for height or depth variables.  
Example: "up" or "down"

`:axis`  
Text. Indicates which axis if the variable is an independent variable.  
Example: "X" or "Y" or "Z" or "T"

`:z`  
Value. Depth or height of fixed instrument if dimension of the z-variable is more than one. If the dimension of the z-variable is only one, then that value is taken as the depth or height of the data variable.  
Example: 10.

`:sample_interval`  
Value. How often the data is sampled in units chosen for the time variable. For example, the instrument samples every 6 minutes or 360 seconds.  
Example: 360.

`:average_interval`  
Value. Period over which data are averaged. For example, wind data are sampled every 6 minutes, but data are collected and averaged over 2 minutes or 120 seconds.  
Example: 120.



## 11. Standard Names

The main purpose of CF is for the climate and forecast data. Many ocean observing and instrument-specific standard names are missing from the CF standard name set. SEACOOS CDL digresses from CF on this point. SEACOOS CDL gets standard name entries from the SEACOOS Data Dictionary. This data dictionary contains names and definitions that the variable attribute `:standard_name` can be given the appropriate unique identifier of the variable's content.

The SEACOOS Data Dictionary also provides other useful metadata about each variable that give suggested entries to include in variable attribute fields of a netCDF file. These are physical measurement units (`:units`), geophysical data ranges (`:valid_range`), reference datum (`:reference`), variable long name or description (`:long_name`) and variable abbreviation (`:short_name`).

In addition, the SEACOOS Data Dictionary functions to associate the given standard name with what other conventions and standards define as the same or similar variable. The ability to associate or correlate dictionaries is one of the unique aspects of the SEACOOS Data Dictionary.

The most recent version of the SEACOOS Data Dictionary can be accessed at <http://www.seacoos.org/documents/seacoos-data-dictionary-v2dot0> .

## 12. Units

The Unidata units library, `udunits`, supports conversion of unit specifications between formatted and binary forms, arithmetic manipulation of unit specifications, and conversion of values between compatible scales of measurement. Each variable in the netCDF file is required to have a units attribute. The units attribute is a string that is recognized by the `udunits` package as specified by the CF Conventions. The `udunits` package includes a file `udunits.dat` which lists its supported unit names. The library is available from <http://www.unidata.ucar.edu/packages/udunits/>.

## 13. Future Directions

The SEACOOS CDL was originally drafted to coordinate many SEACOOS data sources. Some aspects were overlooked while other aspects were purposefully simplified in the first and second round drafts so other processes could be developed. Future directions and topics to include in the SEACOOS CDL, but not limited to, are:

- Guidelines for QA/QC variables
- Relax time constraint to other time frames
- Expand format categories to include model output.

## 14. References

- Blanton, B., I. Bang, and R. He, 2004. Implementation of the SEACOOS Nowcast/Forecast Model System Version 1.0 Barotropic Models and Skill. SEACOOS Internal Document. November, 2004.
- Eaton, B., J. Gregory, B. Drach, K. Taylor, S. Hankin, PMEL, 2001. NetCDF Climate and Forecast (CF) Metadata Conventions. [Online]. Available: <http://www.cgd.ucar.edu/cms/eaton/cf-metadata/> [2004, November].
- Gill, S. and J. Schultz, 2001. Tidal Datums and Their Applications. NOAA Special Publication NOS CO-OPS 1, Silver Spring, MD. February 2001. [Online]. Available: [http://co-ops.nos.noaa.gov/publications/tidal\\_datums\\_and\\_their\\_applications.pdf](http://co-ops.nos.noaa.gov/publications/tidal_datums_and_their_applications.pdf) [2004, November].
- Purvis, C.; J. Cothran; J. Donovan; M. Fletcher; S. Haines; R. Helber; C. Hu; E. Kearns; F. Muller-Karger, 2004. Near Real-Time Ocean Observations Online: Data Management within the Southeast Atlantic Coastal Ocean Observation System (SEACOOS), SEACOOS Spring 2004 Workshop, Miami, Florida
- Rew, R., G. Davis, S. Emmerson, Ha. Davies, June 1997. NetCDF User's Guide for FORTRAN: An Access Interface for Self-Describing, Portable Data; version 3. [Online]. Available: <http://my.unidata.ucar.edu/content/software/netcdf/guidef/index.html> [2004, November].
- Seim, H., B. Bacon, C. Barans, M. Fletcher, K. Gates, R. Jahnke, E. Kearns, R. Lea, M. Luther, C. Mooers, J. Nelson, D. Porter, L. Shay, M. Spranger, J. Thigpen, R. Weisberg, and F. Werner, 2003. SEA-COOS - A Model for a Multi-State, Multi-Institutional Regional Observation System, MTS Journal, 37(3), 92-101.
- Seim, H., F. Werner, M. Fletcher, J. Nelson, R. Jahnke, L. Shay, R. Weisburg, M. Luther, 2002. SEA-COOS: Southeast Atlantic Coastal Ocean Observing System. Proceedings of the Oceans 2002 Conference, Biloxi, MS, October 2002, pg ??-??. MTS/IEEE.

## Appendix A

CDL examples for each format category or data model.

### Appendix A.1 fixed-point

```

netcdf fixed-point {

// This is a fixed-point station with dependent variables
// measured by sensors at different heights. Z constant for a given sensor
// and denoted with variable attribute "variable_name:z = value".

dimensions:
time = 4;
lon = 1;
lat = 1;
z = 3;

variables:

// INDEPENDENT VARIABLES
long time(time);
    time: long_name = "Sample Time";
    time: standard_name = "time";
    time: units = "sec since 1970-1-1 00:00:00";
    time: axis = "T";
float lon(lon);
    lon: long_name = "longitude in decimal degrees";
    lon: standard_name = "longitude";
    lon: units = "degrees_east" ;
    lon: reference = "geographical coordinates" ;
    lon: axis = "X";
    lon: valid_range = -90.,90.;
float lat(lat);
    lat: long_name = "latitude in decimal degrees";
    lat: standard_name = "latitude";
    lat: units = "degrees_east";
    lat: reference = "geographical coordinates";
    lat: axis = "Y";
    lat: valid_range = -180.,180.;
float z(z);
    z: long_name = "Height";
    z: standard_name = "height";
    z: units = "m";
    z: reference = "mean sea level (MSL)";
    z: positive = "up";
    z: axis = "Z";

// DEPENDENT VARIABLES
float wspd(time);
    wspd: long_name = "Wind Speed";
    wspd: standard_name = "wind_speed";
    wspd: units = "m s-1";
    wspd: z = 45;

```

```
float wdir(time);
    wdir: long_name= "Wind Direction (from)";
    wdir: standard_name = "wind_from_direction";
    wdir: units = "degrees";
    wdir: reference = "clockwise from true north";
    wdir: valid_range = 0.,360.;
    wdir: z = 45;
float atemp(time);
    atemp: long_name = "Air temperature";
    atemp: standard_name = "air_temperature";
    atemp: units = "degrees Celsius";
    atemp: z = 24;
float SST(time);
    SST: long_name = "Sea Surface Temperature";
    SST: standard_name = "sea_surface_temperature";
    SST: units = "degrees Celsius";
    SST: z = -5;

// global attributes:
// (use all for adopted SEACOOS CDL v2.0)
:format_category = "fixed-point";

data:

time = 1062804600,1062808200,1062810000,1062811800;
lon = -82.543;
lat = 18.462702;
z = 45, 24, -5;

wspd = 9.80, 12.63, 16.52, 18.97;
wdir = 88.8,103.6,112.7,105.0;
atemp = 27.00, 27.23, 27.05, 26.97;
SST = 28.11, 28.17, 28.17, 28.35;
}
```

**Appendix A.2 fixed-profiler**

```

netcdf fixed-profiler {

// This is a fixed-profiler station with dependent variables
// measured at multiple heights for a given time.

dimensions:
time = 4;
lon = 1;
lat = 1;
z = 3;

variables:

// INDEPENDENT VARIABLES
long time(time);
    time: long_name = "Sample Time";
    time: standard_name = "time";
    time: units = "sec since 1970-1-1 00:00:00";
    time: axis = "T";
float lon(lon);
    lon: long_name = "longitude in decimal degrees";
    lon: standard_name = "longitude";
    lon: units = "degrees_east" ;
    lon: reference = "geographical coordinates" ;
    lon: axis = "X";
    lon: valid_range = -90.,90.;
float lat(lat);
    lat: long_name = "latitude in decimal degrees";
    lat: standard_name = "latitude";
    lat: units = "degrees_east";
    lat: reference = "geographical coordinates";
    lat: axis = "Y";
    lat: valid_range = -180.,180.;
float z(z);
    z: long_name = "Height";
    z: standard_name = "height";
    z: units = "m";
    z: axis = "Z";

// DEPENDENT VARIABLES
float wspd(time,z);
    wspd: long_name = "Wind Speed";
    wspd: standard_name = "wind_speed";
    wspd: units = "m s-1";
float wdir(time,z);
    wdir: long_name= "Wind Direction (from)";
    wdir: standard_name = "wind_from_direction";
    wdir: units = "degrees";
    wdir: reference = "clockwise from true north";
    wdir: valid_range = 0.,360.;

// global attributes:
// (use all for adopted CDL v2.0)
:format_category = "fixed-profiler";

data:

```

```
time = 1062804600,1062808200,1062810000,1062811800;  
lat = 18.462702;  
lon = -82.543;  
z = 10, 20, 30;  
  
wspd =    9.80, 9.80,  9.80,  
        12.63, 12.63, 12.63,  
        16.52, 16.52, 16.52,  
        18.97, 18.97, 18.97;  
wdir =    88.8, 88.8,  88.8,  
        103.6, 103.6, 103.6,  
        112.7, 112.7, 112.7,  
        105.0, 105.0, 105.0;  
}
```

### Appendix A.3 fixed-map

```

netcdf fixed-map {
// This is a fixed-map dataset with dependent variables
// measured at multiple lat and lon and specific z at a given time.

dimensions:
time = 4;
lon = 3;
lat = 3;
z = 1;

variables:
// INDEPENDENT VARIABLES
long time(time);
    time: long_name = "Sample Time";
    time: standard_name = "time";
    time: units = "sec since 1970-1-1 00:00:00";
    time: axis = "T";
float lon(lon);
    lon: long_name = "longitude in decimal degrees";
    lon: standard_name = "longitude";
    lon: units = "degrees_east" ;
    lon: reference = "geographical coordinates" ;
    lon: axis = "X";
    lon: valid_range = -90.,90.;
float lat(lat);
    lat: long_name = "latitude in decimal degrees";
    lat: standard_name = "latitude";
    lat: units = "degrees_east";
    lat: reference = "geographical coordinates";
    lat: axis = "Y";
    lat: valid_range = -180.,180.;
float z(z);
    z: long_name = "Height";
    z: standard_name = "height";
    z: units = "m";
    z: reference = "mean sea level (MSL)";
    z: positive = "up";
    z: axis = "Z";

// DEPENDENT VARIABLES
float wspd(time,lon,lat);
    wspd: long_name = "Wind Speed";
    wspd: standard_name = "wind_speed";
    wspd: units = "m s-1";
float wdir(time,lon,lat);
    wdir: long_name= "Wind Direction (from)";
    wdir: standard_name = "wind_from_direction";
    wdir: units = "degrees";
    wdir: reference = "clockwise from true north";
    wdir: valid_range = 0.,360.;

// global attributes:
// (use all for adopted SEACOOS CDL v2.0)
:format_category = "fixed-map";

data:

```

```
time = 1062804600,1062808200,1062810000,1062811800;
lon = -82.50, -83.00, -83.50;
lat = 32.00, 32.25, 32.50;
z = 10;

// time in rows, x and y columns (3x3=9)
wspd = 9.80, 9.80, 9.80, 9.80, 9.80, 9.80, 9.80, 9.80, 9.80,
12.63, 12.63, 12.63, 12.63, 12.63, 12.63, 12.63, 12.63, 12.63,
16.52, 16.52, 16.52, 16.52, 16.52, 16.52, 16.52, 16.52, 16.52,
18.97, 18.97, 18.97, 18.97, 18.97, 18.97, 18.97, 18.97, 18.97;
wdir = 88.8, 88.8, 88.8, 88.8, 88.8, 88.8, 88.8, 88.8, 88.8,
103.6, 103.6, 103.6, 103.6, 103.6, 103.6, 103.6, 103.6, 103.6,
112.7, 112.7, 112.7, 112.7, 112.7, 112.7, 112.7, 112.7, 112.7,
105.0, 105.0, 105.0, 105.0, 105.0, 105.0, 105.0, 105.0, 105.0;
}
```

**Appendix A.4 moving-point-2D**

```

netcdf moving-point-2D {
// This is a moving-point station with dependent variables
// measured by sensors at different heights. Z constant for a given sensor.

dimensions:
time = 4;
z = 3;

variables:

// INDEPENDENT VARIABLES
long time(time);
    time: long_name = "Sample Time";
    time: standard_name = "time";
    time: units = "sec since 1970-1-1 00:00:00";
    time: axis = "T";
float lon(time);
    lon: long_name = "longitude in decimal degrees";
    lon: standard_name = "longitude";
    lon: units = "degrees_north" ;
    lon: reference = "geographical coordinates" ;
    lon: axis = "X";
    lon: valid_range = -90.,90.;
float lat(time);
    lat: long_name = "latitude in decimal degrees";
    lat: standard_name = "latitude";
    lat: units = "degrees_east";
    lat: reference = "geographical coordinates";
    lat: axis = "Y";
    lat: valid_range = -180.,180.;
float z(z);
    z: long_name = "Height";
    z: standard_name = "height";
    z: units = "m";
    z: reference = "mean sea level (MSL)";
    z: positive = "up";
    z: axis = "Z";

// DEPENDENT VARIABLES
float wspd(time);
    wspd: long_name = "Wind Speed";
    wspd: standard_name = "wind_speed";
    wspd: units = "m s-1";
    wspd: z = 45;
float wdir(time);
    wdir: long_name= "Wind Direction (from)";
    wdir: standard_name = "wind_from_direction";
    wdir: units = "degrees";
    wdir: reference = "clockwise from true north";
    wdir: valid_range = 0.,360.;
    wdir: z = 45;
float atemp(time);
    atemp: long_name = "Air temperature";
    atemp: standard_name = "air_temperature";
    atemp: units = "degrees Celsius";
    atemp: z = 24;
float SST(time);
    SST: long_name = "Sea Surface Temperature";

```

```
SST: standard_name = "sea_surface_temperature";
SST: units = "degrees Celsius";
SST: z = -5;

// global attributes:
// (use all for adopted CDL v2.0)
:format_category = "moving-point-2D";

data:

time = 1062804600,1062808200,1062810000,1062811800;
lat = 24.17, 24.29, 24.35, 24.41;
lon = -82.83, -82.90, -83.97, -84.04;
z = 45, 24, -5;

wspd = 9.80, 12.63, 16.52, 18.97;
wdir = 88.8,103.6,112.7,105.0;
atemp = 27.00, 27.23, 27.05, 26.97;
SST = 28.11, 28.17, 28.17, 28.35;
}
```

**Appendix A.5 moving-point-3D**

```

netcdf moving-point-3D {

// This is a moving-point platform that moves in 3D space
// and all sensors are sampling at the same time.

dimensions:

time = 4;

variables:

// INDEPENDENT VARIABLES
long time(time);
    time: long_name = "Sample Time";
    time: standard_name = "time";
    time: units = "sec since 1970-1-1 00:00:00";
    time: axis = "T";
float lon(time);
    lon: long_name = "longitude in decimal degrees";
    lon: standard_name = "longitude";
    lon: units = "degrees_north" ;
    lon: reference = "geographical coordinates" ;
    lon: axis = "X";
    lon: valid_range = -90.,90.;
float lat(time);
    lat: long_name = "latitude in decimal degrees";
    lat: standard_name = "latitude";
    lat: units = "degrees_east";
    lat: reference = "geographical coordinates";
    lat: axis = "Y";
    lat: valid_range = -180.,180.;
float z(time);
    z: long_name = "Height";
    z: standard_name = "height";
    z: units = "m";
    z: reference = "sea-surface";
    z: positive = "up";
    z: axis = "Z";

// DEPENDENT VARIABLES
//
float wspd(time);
    wspd: long_name = "Wind Speed";
    wspd: standard_name = "wind_speed";
    wspd: units = "m s-1";
float wdir(time);
    wdir: long_name= "Wind Direction (from)";
    wdir: standard_name = "wind_from_direction";
    wdir: units = "degrees";
    wdir: reference = "clockwise from true north";
    wdir: valid_range = 0.,360.;
float atemp(time);
    atemp: long_name = "Air temperature";
    atemp: standard_name = "air_temperature";
    atemp: units = "degrees Celsius";

// global attributes:
// (use all for adopted SEACOOS CDL v2.0)
:format_category = "moving-point-3D";

```

```
data:
time = 1062804600,1062808200,1062810000,1062811800;
lat = 24.17, 24.29, 24.35, 24.41;
lon = -82.83, -82.90, -83.97, -84.04;
z = 200, 210, 225, 245;

wspd = 9.80, 12.63, 16.52, 18.97;
wdir = 88.8,103.6,112.7,105.0;
atemp = 28.11, 28.17, 28.17, 28.35;
}
```

**Appendix A.6 moving-profiler**

```
netcdf moving-profiler {  
  
  // This is a moving-profiler station with dependent variables  
  // measured at multiple heights for a given time and varying in position.  
  
  dimensions:  
    time = 4;  
    z = 3;  
  
  variables:  
  
    // INDEPENDENT VARIABLES  
    long time(time);  
      time: long_name = "Sample Time";  
      time: standard_name = "time";  
      time: units = "sec since 1970-1-1 00:00:00";  
      time: axis = "T";  
    float lon(time);  
      lon: long_name = "longitude in decimal degrees";  
      lon: standard_name = "longitude";  
      lon: units = "degrees_north" ;  
      lon: reference = "geographical coordinates" ;  
      lon: axis = "X";  
      lon: valid_range = -90.,90.;  
    float lat(time);  
      lat: long_name = "latitude in decimal degrees";  
      lat: standard_name = "latitude";  
      lat: units = "degrees_east";  
      lat: reference = "geographical coordinates";  
      lat: axis = "Y";  
      lat: valid_range = -180.,180.;  
    float z(z);  
      z: long_name = "Height";  
      z: standard_name = "height";  
      z: units = "m";  
      z: reference = "mean sea level (MSL)";  
      z: positive = "up";  
      z: axis = "Z";  
  
    // DEPENDENT VARIABLES  
    float wspd(time,z);  
      wspd: long_name = "Wind Speed";  
      wspd: standard_name = "wind_speed";  
      wspd: units = "m s-1";  
    float wdir(time,z);  
      wdir: long_name= "Wind Direction (from)";  
      wdir: standard_name = "wind_from_direction";  
      wdir: units = "degrees";  
      wdir: reference = "clockwise from true north";  
      wdir: valid_range = 0.,360.;  
  
    // global attributes:  
    // (use all for adopted SEACOOS CDL v2.0)  
    :format_category = "moving-profiler";  
  
  data:
```

```
time = 1062804600,1062808200,1062810000,1062811800;
lat = 24.17, 24.29, 24.35, 24.41;
lon = -82.83, -82.90, -83.97, -84.04;
z = 10, 20, 30;

// time rows, z columns
wspd = 9.80, 9.80, 9.80,
       12.63, 12.63, 12.63,
       16.52, 16.52, 16.52,
       18.97, 18.97, 18.97;
wdir = 88.8, 88.8, 88.8,
       103.6, 103.6, 103.6,
       112.7, 112.7, 112.7,
       105.0, 105.0, 105.0;
}
```