

direction de la technologie marine et des systèmes d'information  
département informatique et données marines

Thierry Carval, Valérie Harscoat, Gilbert Maudire

avril 01 – cor-do/dti-die/00-056

The logo for Ifremer, featuring a stylized grey fish silhouette with a halftone dot pattern, positioned to the left of the vertical yellow bar.

**ifremer**

---

# Coriolis data-center

External interfaces  
Volume 2/2  
Version 1.2

Coriolis data-center

External interfaces

<b><u>1. DOCUMENT HISTORY</u></b>	<b>4</b>
<b><u>2. PURPOSE OF THIS DOCUMENT</u></b>	<b>5</b>
<b><u>3. APPLICABLES DOCUMENTS AND REFERENCES</u></b>	<b>6</b>
3.1. APPLICABLE DOCUMENTS	6
3.2. REFERENCE DOCUMENTS	6
<b><u>4. EXTERNAL INTERFACES</u></b>	<b>7</b>
4.1. CO-IF0012 : ASCII MEDATLAS FORMAT	8
4.2. CO-IF0013 : MULTI-PROFILE CORIOLIS NETCDF FORMAT	14
4.2.1. POLICY OF CO-IF0013 DISSEMINATION FOR MERCATOR	20
4.3. CO-IF0024 : GRIDDED DATA CORIOLIS NETCDF FORMAT	22
4.4. CO-IF0022 : ARGO NETCDF MONO-PROFILE FORMAT	26
4.5. CO-IF0031 : ARGO META-DATA FORMAT	34
4.6. CO-IF0032 : ARGO NETCDF TRAJECTORY DATA	37

## 1. Document history

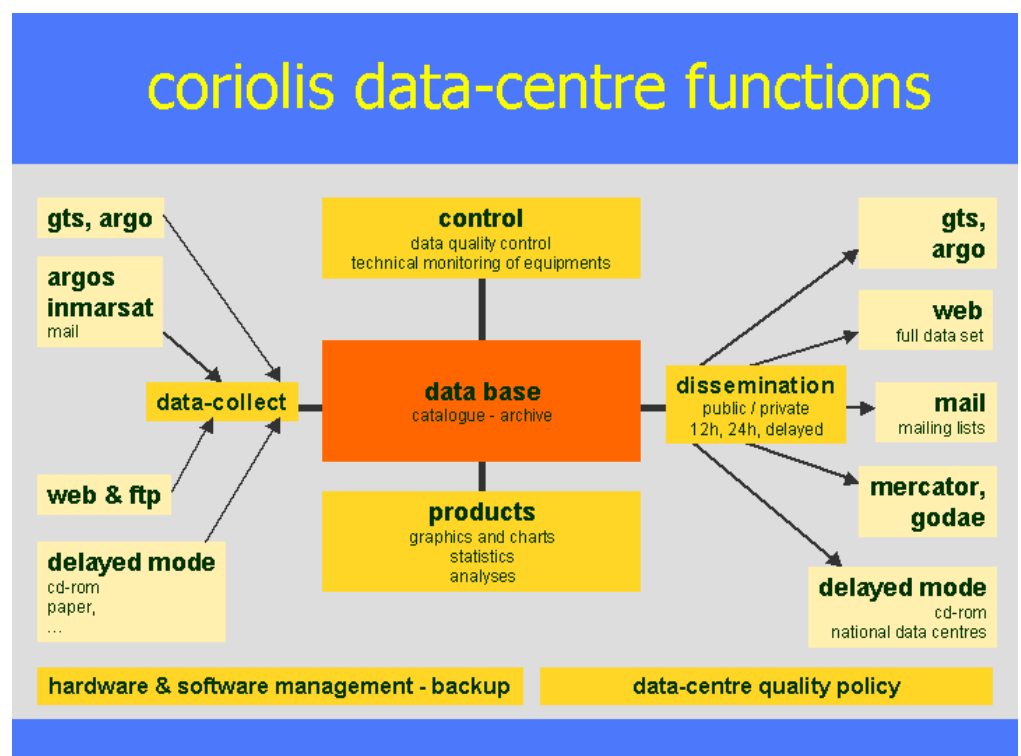
date	Updated interface	Update description
25/08/00	CO-IF0013	Description of field REFERENCE
25/08/00	CO-IF0013	Convention on field REC_TYPE
31/08/00	CO-IF0013	Latitudes and longitudes formats for global attributes
24/10/00	All interfaces	English translation of the document
09/11/00	CO-IF0013	Policy of CO-IF0013 dissemination for Mercator
19/01/01	CO-IF0013	Fill value of PRES set to 99999 instead of 9999
09/03/01	CO-IF0024	Update of the grided netcdf format
03/04/01	CO-IF0032	Creation of argo netcdf trajectory format
03/04/01	CO-IF0031	Creation of argo ascii meta-data format

## 2. Purpose of this document

This document describes the external interfaces of the Coriolis data-center. It supplies information to the teams in charge of system requiring external interfaces :

- Data-collect
- Data base
- Control
- Products
- Dissemination.

This chart describes the functions of Coriolis data-center.



## 3. Applicables documents and references

### 3.1. Applicable documents

Liste des documents du projet coriolis centre de données

Réf. cor-do/dti-doc/00-007

Dossier de spécifications systèmes Coriolis

Réf : cor-do/dti-dss/00-001

Dossier des interfaces internes Coriolis

Réf : cor-do/dti-dss/00-050

Plan qualité logiciel Coriolis-données

Réf. cor-do/dti-pql/00-008

Dossier de spécifications fonctionnelles Coriolis-provor

Réf. cor-do/dti-dsf/99-005

### 3.2. Reference documents

Dictionnaire des sigles Coriolis-données

Réf cor-do/dti-dis/00-023

« MFSPP VOS, data quality control procedures »

Réf . : MAS3-CT98-0171 (DG12-EHKN).

OMM n° 306 - Vol 1.1 Part A: Alphanumeric codes

ICES/IOC 1995. GETADE Formatting Guidelines for Oceanographic Data Exchange (Preliminary report)

UNESCO 1989. A General Formatting System for Geo-reference Data. IOC. Manuals and Guides N0. 17

UNESCO 1991. Manual on International Oceanographic Data Exchange. Manuals and Guides N0. 9

« Dictionnaire de données du centre d'archivage WOCE »

Réf. : sismer/woce/r3/000/version4.0

MEDAR/MEDATLAS protocol – Part 1

Exchange format and quality checks for observed profikes – MEDAR Group 1999

---

## 4. External interfaces

#### 4.1. CO-IF0012 : ascii medatlas format

Interface name	Médatlas format
Reference	CO-IF0012
Source	Function «medatlas dissemination» CO-FO-05-04
Destination	Function « web dissemination» CO-FO-05-01
Operating mode	-
Frequence	-
Planned volume	-
Last update	01/03/2000

This format has been defined by a group of data-centers in conformity with the ICES/IOC GETADE recommendations. Medatlas is designed :

1) To facilitate the reading of the data, (and not to optimise the data archiving on the magnetic medium, neither to speed up the data processings).

2) To be independent of the computer.

The consequence of these two points is that an auto-descriptive ASCII format will be preferred.

3) To keep track of the history of the data including the data collection and the processing. Then each cruise must be documented.

4) To allow the processing of profile independently. Therefore the date, time and geographical co-ordinate must be reported on each profile header.

5) To be flexible and accept (almost) any number of different parameters.

6) The real numbers (floating numbers must remain in the same way as they have been transmitted, not transcoded into integer numbers). The number of decimals must implicitly indicate the accuracy of the measurements.

The profiles measured with the same instrument (eg. bottles, CTD, XBT..) during the same cruise, are grouped within the same file which includes :

- a cruise header based on the international ROSCOP information;

- a profile (station) header including the cruise reference, the originator station reference within the cruise and the time and location .
- the data points of the profile.

The sequence 'profile header + data records ' is repeated for each profile.  
The main fields are described further on.

## CRUISE SUMMARY FORMAT

BP = Beginning Position, SL = string length, NDT= Number of data types (ROSCOP)  
NCO = Number of comment lines (textual information on the cruise)

LINE	FIELD	DESCRIPTION	BP	SL	TYPE
1	1st character *		1	1	char
	cruise	MEDATLAS reference	2	13	char
	name	originator cruise name/ref.	16	32	char
	ship code	standardised WDCA/ICES/IOC code	49	4	char
	ship name	full ship name	54	25	char
2	start date	DD/MM/YYYY	1	10	char
	end date	DD/MM/YYYY	12	10	char
	region	GF3 table	23	35	char
3	country	Source laboratory country code	1	2	char
	address	Laboratory, institution, town	4	75	char
4	name	chief scientist full name	1	40	char
	key word	'Project='	42	8	char
	project	name of the project	50	28	char
5	key	'Regional Archiving='	1	19	char
	data centre	regional archiving centre code	21	2	char
	key word	'Availability='	42	13	char
	availability	Data Availability code (P/L/C)	55	1	char
5+1	key word	'Data Type='	1	10	char
	data type	ROSCOP code	11	3	char
TO	key word	'n='	15	2	char
	number	number of profiles for the type	17	4	num
5+NDT	key word	'QC='	22	3	char
	QC	Y/N (Yes or No)	25	1	char
6+NDT	key word	'COMMENT'	1	7	char
.....	.....	.....			
5+NDT+NCO	key word	any other cruise information	1	80	char

### Example

```
*FI35199406006 SUIVILION 09          35EU EUROPE
26/06/1994 05/07/1994 MED. WESTERN BASIN
35 CNRS, Universites de Perpignan, Paris VI et Aix-Marseille II
DURRIEU de MADRON Xavier             Project=Prog National Oceano Cotiere
Regional Archiving= FI                 Availability=P
Data Type=H10 n= 45 QC=Y
Data Type=H17 n= 45 QC=N
COMMENT
CTD casts were completed using the SEABIRD 9/11 plus CTD probe of the
Observatoire Oceanologique de Banyuls. The CTD data were recorded into 24 HZ
time-series during data acquisition.
The Pressure, temperature and conductivity channels were calibrated in
laboratory. Ligth transmission was measured with a 25 cm optical pathlength
Seatech transmissiometer.
No calibration in situ.
```

### Profile header format

NP = number of observed parameters incl. the reference (pressure); BP =Beginning Position  
NCO = number of comment lines in the header; SL = string length

LINE	FIELD	DESCRIPTION	BP	SL	TYPE
1	start character	*	1	1	char
	reference	MEDATLAS ref. (cruise+station+cast)	2	18	char
	key word	'Data Type='	21	10	char
	data type	ROSCOP code	31	3	char
2	date	'*DATE=', DDDMMYYYY	1	14	char
	time	'TIME=', HHMM	16	9	char
	latitude N/S	'LAT=', N or S	26	5	char
	lat. degrees	latitude degrees (0 to 90)	31	2	char
	lat. minutes	latitude minutes.hundredth	34	5	char
	longitude E/W	'LON=', E or W	40	5	char
	long. degrees	longitude degrees (0 to 180)	45	3	char
	long. minutes	minutes.hundredth	49	5	char
	key word	'DEPTH='	55	6	char
	bottom depth	bottom depth in metres	61	6	num
	key word	'QC='	68	3	char
	time flag	QC flag on date and time	71	1	num
	lat. flag	QC flag on latitude	72	1	num
	long. flag	QC flag on longitude	73	1	num
depth flag	QC flag on bottom depth	74	1	num	
3	key word	*NB PARAMETERS= '	1	15	char
	nb. of columns	number of measured parameters (NP)	16	2	num
	key word	'RECORD LINES='	19	13	char
	number of lines	number of observations records (NL)	32	5	num
4	1st character	*	1	1	char
	title column p	GP3 code for parameter p	2	4	char
	parameter p	parameter full name (IOC def.)	7	30	char
	unit p	( 'IS/IOC unit' )	37	30	char
3+NP	key word	'def.='	68	5	char
	default val.	string of '9'.string of '9' same format as the parameter p	73	nlen(np)	num
4+NP	key word	'*GLOBAL PROFILE QUALITY FLAG='	1	29	char
	profile QC	global quality flag for the profile	30	1	num
	key word	'GLOBAL PARAMETERS QC FLAGS='	32	27	char
	profile QC	global quality flags for the NP parameters	50	NP	num
5+NP	collect hist.1	'*DC HISTORY='method, instrum. etc	1	80	char
6+NP	collect hist.2	'*', continuation	1	80	char
7+NP	archv. hist.1	'*DM HISTORY', data mangt. history	1	80	char
8+NP	archv. hist.2	'*', end of data management	1	80	char
9+NP	comment 1	'*COMMENT ', free comment	1	80	char
TO	comments n	'*', end of comments	1	80	char
8+NP+NCO					
9+NP+NCO	surface obs. 1	'*SURFACE SAMPLES=', samples	1	80	char
10+NP+NCO	surface obs. 2	'*', end of surface samples	1	80	char

### Example

```

*FI351994060060B020 Data Type=H10
*DATE=30061994 TIME=0900 LAT=N42 34.00 LON=E003 17.00 DEPTH=88      QC=1111
*NB PARAMETERS=04 RECORD LINES=00090
*PRES SEA PRESSURE sea surface=0      (DECIBAR=10000 PASCALS)      def.= -999.9
*TEMP SEA TEMPERATURE                  (CELSIUS DEGREE)          def.= 99.999
*PSAL PRACTICAL SALINITY                (P.S.U.)                  def.= 99.999
*TUR2 TURBIDITY-attenuation coeff      (m-1)                    def.= 99.999
*GLOBAL PROFILE QUALITY FLAG=1 GLOBAL PARAMETERS QC FLAGS=1110
*DC HISTORY=
*
*DM HISTORY
*
*COMMENT
* Depth Above Bottom      :      0.000000
*SURFACE SAMPLES=
*

```

### Profile format (continuation)

Each parameter p can have any length nlen(p), but this length must be constant in the profile, and the decimal points at a constant position. The separators are blanks of one character (or more), after each parameter value. The quality flags are grouped after the last separator, with no blank between them.

The position of the first character of parameter p in the record line is then :  
 $bp(p) = nlen(1) + \dots + nlen(p-1) + p$   
 and the position of the first quality flag :  
 $bp(q1) = bp(NP) + nlen(NP) + 1$   
 $nlen(p)$  = number of digits of parameter p;  $NP$  = number of measured parameters (columns)  
 $NL$  = number of observations records (lines)  
 $hheader = NP + NCO + 11$  = total number of header lines;  $NCO$  = number of comment lines  
 $BP$  = Beginning Position,  $SL$  = string length

LINE	FIELD	DESCRIPTION	BP	SL	TYPE
<b>nheader</b>	start character	*	1	1	char
	title 1	'PRES' or ref. parameter	2	4	char
	separator	blank	6	$nlen(1)-4$	blank
	title 2	GF3 code parameter 2	$bp(2)$	4	char
	separator	blank	$bp(p-1)+4$	$nlen(p-1)-3$	
	title par. P	GF3 code parameter p	$bp(p)$	4	char
	title NP	GF3 code parameter NP	$bp(np)$	4	char
<b>nheader</b>	parameter 1	parameter 1 in IS units	1	$nlen(1)$	num
<b>+1</b>	separator	blank	$bp(2)-1$	1	
<b>to</b>	parameter 2	parameter 2 in IS units	$bp(2)$	$nlen(2)$	num
<b>+NL</b>	separator	blank	$bp(np)-1$	1	
	parameter NP	parameter NP in IS units	$bp(np)$	$nlen(np)$	num
	separator	blank	$bp(q1)-1$	1	
	QC Flag 1	Quality Flag parameter 1	$bp(q1)$	1	num
	QC Flag NP	Quality Flag parameter NP	$bp(q1)+np$	1	num
<b>nheader</b>	def. val. par.1	'-999.9' or string of '9'	1	$nlen(1)$	num
<b>+NL+1</b>	separator	blank	$bp(2)-1$	1	
<b>(Last</b>	def. val. par.2	string of '9' as in header	$bp(2)$	$nlen(2)$	num
<b>line</b>	def. val. par.NP	string of '9' as in header	$bp(NP)$	$nlen(NP)$	num
<b>of the</b>	separator	blank	$bp(q1)-1$	1	
<b>profile)</b>	QC Flag 1	'9' (Def. val. Quality Flag)	$bp(q1)$	1	num
	QC Flag NP	'9' (Def. val. Quality Flag)	$bp(q1)+np$	1	num

## Example

```
*PRES TEMP PSAL TUR2
0.0 19.441 37.596 0.525 1110
1.0 19.441 37.596 0.525 1110
2.0 19.441 37.596 0.525 1110
3.0 19.306 37.595 0.526 1110
4.0 19.234 37.592 0.530 1110
5.0 19.208 37.591 0.532 1110
6.0 19.188 37.591 0.531 1110
7.0 19.169 37.593 0.530 1110
8.0 19.067 37.599 0.525 1110
9.0 18.848 37.610 0.525 1110
10.0 18.621 37.603 0.530 1110
11.0 18.501 37.595 0.537 1110
.....
-999.9 99.999 99.999 99.999 9999
```

## DESCRIPTION OF THE MAIN FIELDS

### Files

All the profiles measured with the same instrument (eg. bottles, CTD, XBT..) is grouped within the same file. The data file begins with a short cruise descriptor based on the ROSCOP information forms. The profiles are attached behind an each of them begins with a profile header giving all the necessary environmental information on the observations. Each observed parameter represent a separate column. There is no limitation to the number of parameters (columns) but the number of parameters within the same cruise must be constant. If a parameter is missing in one station, the corresponding column must be fulfilled by default values.

### **Records**

The records consists in data collected at the same level. The record (line) length is not limited for observed data but reasonable (<120) number of characters in the lines is recommended.

### **Missing data**

For missing data, the default characters are :

- in character strings : 'X' as many times as the missing string length or 'UNKNOWN'
- in numbers : 9 or -9 at the same format as the expected numbers.

### **Latitude and Longitude**

In order to avoid ambiguity, N/S and E/W will be specified for latitude and longitude instead of signs.

### **Pressure/depth vertical reference parameter**

- Pressure is the reference co-ordinate (first column) for the vertical profiles.
- When the pressure is not explicitly defined (mainly the oldest historical data sets), the vertical co-ordinate is supposed to be pressure in decibars (which makes no significant differences on historical data).
- The first column parameter which is the reference (the pressure for vertical profiles) must be recorded in increasing order.

### **Temperature and Salinity and other parameters**

- Temperature, salinity and any archived parameters are observed parameters. The calculated parameters like density or potential temperature are not archived.
- Other measured parameters like nutrients are archived with the GF3 parameter code as the column title. The parameter code is related to the same unit, normally the International System (IS). A different unit corresponds to a different parameter code.

### **History and information on the data processing**

In order to maintain some flexibility with the format and not to loose existing complementary information on the data processing, meteorological observations etc. not taken into account in the format, three specific fields terminate each header profile :

\*DC HISTORY for the information linked to the data collection at sea (like instrument, calibration)

\*DM HISTORY for the information linked to the data management and archiving (like the source latitude if the sign has been changed)

\*COMMENT for all optional data and meta data like time and location at the end of the profile, meteorological observations. Avoid to repeat character strings already used (ex use LATEND= and not LAT=). The number of \*COMMENT lines is not limited.

### **REFERENCES**

ICES/IOC 1995. GETADE Formatting Guidelines for Oceanographic Data Exchange (Preliminary report)

---

UNESCO 1989. A General Formatting System for Geo-reference Data. IOC.  
Manuals and Guides N0. 17

UNESCO 1991. Manual on International Oceanographic Data Exchange. Manuals  
and Guides N0. 9

## 4.2. CO-IF0013 : multi-profile Coriolis NetCdf format

<b>Interface name</b>	<b>multi-profile Coriolis NetCdf format</b>
Reference	CO-IF0013
Source	Function «NetCdf dissemination» CO-FO-05-05
Destination	Function «web dissemination» CO-FO-05-01
Operating mode	-
Frequence	-
Estimated volume	-
Last update	31/08/2000

### Dimensions definitions

```

N_CODE = 4 ;
N_DATE_TIME = 19 ;

// Null terminated string
N_STRING64 = 64 ;
N_STRING16 = 16 ;
N_STRING4 = 4 ;

// These dimensions depend on data set
N_PARAM = 4 ;
mN_PROF = 80 ;
mN_ZLEV = 547 ;

```

## Metadata

Metadata are coded as global attributes

```
// global attributes (given with their fillValue)
// Date and comments on updates of this file (char*80)
:Last_update = "";
// File version (referencing the version of the format)
:Version = "V1.0";
// Data provider
:Data_provider = "IFREMER/CORIOLIS";
// References of experiment (char*80)
:Experiment_name = "";
:Project_name = "";
// Description of experiment (char*256)
:Experiment_description = "";
// Description of the geographical area (char*80)
:Geographical_area = "";
// Dates (convention : DD/MM/YYYY HH24:MI:SS)
// 01/01/1950 00:00 recommended
:Reference_date_time = "";
:Start_date = "";
:Stop_date = "";
// Limits of the experiment (decimal degrees ,positive =North,East)
:South_latitude = -999. ; // [-90,90] for latitudes
:North_latitude = -999. ;
:West_longitude = -999. ; // ]-180,180] for longitudes
:East_longitude = -999. ;
// Coordinate system (GMT convention)
:Coord_system = "GEOGRAPHICAL";
// GF3 code of the immersion reference parameter for the file (DEPH, PRES,
TIME)
:Reference_parameter = "";
// Data interpolated on fixed levels of reference parameterü ("YES" or "NO")
:Normalized_data = "";
// Data type = list of Roscop codes for the file (char*80)
:Data_type = "";
// Description of file constitution : data, user request... (char*1024)
:File_contents = "";
// Comments on file contents/modification (char*256)
:comments = "";
```

### General information on each profile

```
// General information on each profiles
char PARAMETERS(N_PARAM,N_CODE) ;
    PARAMETERS:long_name = "List of parameters" ;
    PARAMETERS:Conventions = "GF3 code of the indexed parameter among
(DEPH, PRES, PSAL, TEMP)";
    PARAMETERS:_FillValue = " " ;
char VOYAGE_NAME(mN_PROF,N_STRING16) ;
    VOYAGE_NAME:long_name = "Voyage name" ;
    VOYAGE_NAME:Conventions = "To Be Defined" ;
    VOYAGE_NAME:_FillValue = " " ;
char PLATFORM_NUMBER(mN_PROF,N_STRING16) ;
    PLATFORM_NUMBER:long_name = "Platform number" ;
    PLATFORM_NUMBER:Conventions = "To Be Defined" ;
    PLATFORM_NUMBER:_FillValue = " " ;
int STATION_NUMBER(mN_PROF) ;
    STATION_NUMBER:long_name = "Station or cycle number" ;
    STATION_NUMBER:Conventions = "To Be Defined" ;
    STATION_NUMBER:_FillValue = 99999 ;
char DIRECTION(mN_PROF) ;
    DIRECTION:long_name = "Direction of the station : A, D" ;
    DIRECTION:Conventions = "To Be Defined" ;
    DIRECTION:_FillValue = " " ;
char REFERENCE(mN_PROF,N_STRING16) ;
    REFERENCE:long_name = "Reference for each profile in Coriolis data base" ;
    REFERENCE:Conventions = "To Be Defined" ;
    REFERENCE:_FillValue = " " ;
char INST_TYPE(mN_PROF,N_STRING4) ;
    INST_TYPE:long_name = "Instrument type for profile" ;
    INST_TYPE:Conventions = "OMM probe code" ;
    INST_TYPE:_FillValue = " " ;
char REC_TYPE(mN_PROF,N_STRING4) ;
    REC_TYPE:long_name = "Recorder type for profiles" ;
    REC_TYPE:Conventions = "OMM recorder code" ;
    REC_TYPE:_FillValue = " " ;
char DATE(mN_PROF,N_DATE_TIME) ;
    DATE:long_name = "Dates of each profile" ;
    DATE:Conventions = "DD/MM/YYYY HH24:MI:SS" ;
    DATE:_FillValue = " " ;
float BOTTOM_DEPTH(mN_PROF) ;
    BOTTOM_DEPTH:long_name = "Bottom depth of each profile" ;
    BOTTOM_DEPTH:unit= "meters" ;
    BOTTOM_DEPTH:Conventions = "not mandatory, always positive" ;
    BOTTOM_DEPTH:_FillValue = -99999.f ;
    BOTTOM_DEPTH:valid_min = 0 ;
    BOTTOM_DEPTH:valid_max = 15000 ;
double JULD(mN_PROF) ;
    JULD:long_name = "Julian day of each profile relative to REFERENCE_DATE" ;
    JULD:units = "julian days" ;
    JULD:Conventions = "Relative julian days" ;
    JULD:_FillValue = -99999. ;
double LATITUDE(mN_PROF) ;
    LATITUDE:long_name = "Latitude of each profile" ;
    LATITUDE:units = "degrees_north" ;
    LATITUDE:_FillValue = -99999.f ;
    LATITUDE:valid_min = -90.f ;
    LATITUDE:valid_max = 90.f ;
double LONGITUDE(mN_PROF) ;
    LONGITUDE:long_name = "Longitude of each profile" ;
    LONGITUDE:units = "degrees_east" ;
    LONGITUDE:_FillValue = -99999.f ;
```

```

LONGITUDE:valid_min = -180.f ;
LONGITUDE:valid_max = 180.f ;
char Q_DATE(mN_PROF) ;
    GCQF:long_name = "Quality on Date and Time" ;
    GCQF:Conventions = "Q where Q =[0,9]" ;
    GCQF:_FillValue = "9" ;
char Q_POSITION(mN_PROF) ;
    GCQF:long_name = "Quality on position (latitude and longitude)" ;
    GCQF:Conventions = "Q where Q =[0,9]" ;
    GCQF:_FillValue = "9" ;
char Q_BOTTOM(mN_PROF) ;
    GCQF:long_name = "Quality on bottom depth" ;
    GCQF:Conventions = "Q where Q =[0,9], not mandatory" ;
    GCQF:_FillValue = "9" ;
char Q_DEPTH(mN_PROF) ;
    GCQF:long_name = "Quality on bottom depth" ;
    GCQF:Conventions = "Q where Q =[0,9], not mandatory" ;
    GCQF:_FillValue = "9" ;
char Q_PROFILE_PRES(mN_PROF) ;
    GCQF:long_name = "Global quality flag of pressure profile" ;
    GCQF:Conventions = "Q where Q =[0,9]" ;
    GCQF:_FillValue = "9" ;
char Q_PROFILE_DEPH(mN_PROF) ;
    GCQF:long_name = "Global quality flag of depth profile" ;
    GCQF:Conventions = "Q where Q =[0,9]" ;
    GCQF:_FillValue = "9" ;
char Q_PROFILE_TEMP(mN_PROF) ;
    GCQF:long_name = "Global quality flag of temperature profile" ;
    GCQF:Conventions = "Q where Q =[0,9]" ;
    GCQF:_FillValue = "9" ;
char Q_PROFILE_PSal(mN_PROF) ;
    GCQF:long_name = "Global quality flag of practical salinity profile" ;
    GCQF:Conventions = "Q where Q =[0,9]" ;
    GCQF:_FillValue = "9" ;

```

**Data**

```

// Data for profiles
// Variable naming conventions:
// Measured value : GF3 code (mandatory if measured and mentioned in PARAMETERS),
// Associated QC flag : "QC_" + GF3 (mandatory if measured and mentioned in
PARAMETERS),
// Error estimation : "Error_" + GF3 (not mandatory)
float PRES(mN_ZLEV, mN_PROF) ;
    PRES:long_name = "Pressure" ;
    PRES:_FillValue = 99999.f ;
    PRES:units = "decibar = 1000 Pa" ;
    PRES:valid_min = 0. ;
    PRES:valid_max = 15000. ;
char QC_PRES(mN_ZLEV, mN_PROF) ;
    Q_PRES:long_name = "Global quality on pressure" ;
    Q_PRES:Conventions = "Q where Q =[0,9]" ;
    Q_PRES:_FillValue = "9" ;
float Error_PRES(mN_ZLEV, mN_PROF) ;
    Error_PRES:long_name = "Error on Pressure" ;
    Error_PRES:_FillValue = 9999.f ;
    Error_PRES:units = "decibar = 1000 Pa" ;
    Error_PRES:valid_min = 0. ;
    Error_PRES:valid_max = 15000. ;
    Error_PRES:comment = "description of error calculation (convention TDB)";
float DEPH(mN_ZLEV, mN_PROF) ;
    DEPH:long_name = "Depth below sea surface" ;
    DEPH:_FillValue = 9999.f ;
    DEPH:units = "metres" ;
    DEPH:valid_min = 0. ;
    DEPH:valid_max = 15000. ;
char QC_DEPH(mN_ZLEV, mN_PROF) ;
    Q_DEPH:long_name = "Global quality on depth" ;
    Q_DEPH:Conventions = "Q where Q =[0,9]" ;
    Q_DEPH:_FillValue = "9" ;
float Error_DEPH(mN_ZLEV, mN_PROF) ;
    Error_DEPH:long_name = "Error on depth below sea surface" ;
    Error_DEPH:_FillValue = 9999.f ;
    Error_DEPH:units = "metres" ;
    Error_DEPH:valid_min = 0. ;
    Error_DEPH:valid_max = 15000. ;
    Error_DEPH:comment = "description of error calculation (convention TDB)";
float TEMP(mN_ZLEV, mN_PROF) ;
    TEMP:long_name = "Temperature in situ T90 scale" ;
    TEMP:_FillValue = 9999.f ;
    TEMP:units = "degres Celsius" ;
    TEMP:valid_min = -3. ;
    TEMP:valid_max = 40. ;
char QC_TEMP(mN_ZLEV, mN_PROF) ;
    Q_TEMP:long_name = "Global quality on temperature" ;
    Q_TEMP:Conventions = "Q where Q =[0,9]" ;
    Q_TEMP:_FillValue = "9" ;
float Error_TEMP(mN_ZLEV, mN_PROF) ;
    Error_TEMP:long_name = "Error Temperature (in situ)" ;
    Error_TEMP:_FillValue = 9999.f ;
    Error_TEMP:units = "degres Celsius" ;
    Error_TEMP:valid_min = 0. ;
    Error_TEMP:valid_max = 40. ;
    Error_TEMP:comment = "description of error calculation (convention TDB)";
float PSAL(mN_ZLEV, mN_PROF) ;
    PSAL:long_name = "Practical salinity, sal78" ;
    PSAL:_FillValue = 9999.f ;

```

```

        PSAL:units = "P.S.U." ;
        PSAL:valid_min = 0. ;
        PSAL:valid_max = 60. ;
char QC_PSA(mN_ZLEV, mN_PROF) ;
        Q_PSA:long_name = "Global quality on practical salinity" ;
        Q_PSA:Conventions = "Q where Q = [0,9]" ;
        Q_PSA:_FillValue = "9" ;
float Error_PSA(mN_ZLEV, mN_PROF) ;
        Error_PSA:long_name = "Error on Practical salinity" ;
        Error_PSA:_FillValue = 9999.f ;
        Error_PSA:units = "P.S.U." ;
        Error_PSA:valid_min = 0. ;
        Error_PSA:valid_max = 60. ;
        Error_PSA:comment = "description of error calculation (convention TDB)";

```

Quality flags	
Value	Description
0	Unqualified
1	Correct value (All checks passed)
2	Value inconsistent with statistics (Differ from climatology)
3	Dubious value (spike, gradient, ... if other tests passed)
4	Impossible value (out of scale, vertical instability, constant profile, ...)
5	Value modified during Quality Control
6-7	Not used (available)
8	Profile interpolated at standard depth
9	Missing value

#### 4.2.1. Policy of CO-IF0013 dissemination for Mercator

##### **Input :**

Base de données Coriolis, interface CO-IF0005

##### **Description :**

Les données du centre Coriolis sont diffusées vers le projet Mercator au format Coriolis NetCdf CO-IF0013.

Les données sont déposées une fois par semaine, le mardi à 20 :00 sur le site ftp de l’Ifremer. Le site PDO (prototype de données) de Mercator vient les collecter le mardi à 21 :00.

Un fichier de données est généré par type de capteur.

Le fichier de données contient les profils qui répondent aux critères suivants :

- profils contrôlés par le centre Coriolis entre [J et J-7[.  
J représente la date et heure de génération du fichier (le mardi à 20 :00).
- la date des profils est comprise entre [J et J-30 [.

Nommage du fichier : CO\_TYPE\_DATE\_HEURE\_DATEMIN\_DATEMAX.nc

Exemple : CO\_XB\_20001001\_1715\_20000925\_20000930.nc

Types de capteurs concernés :

Code	Meaning
BA	BATHY
BO	Bottle
CD	CTD down cast
CU	CTD up cast
CT	CTD up or down cast
DB	DRIFTER
MB	MBT
PF	Profiling float
TE	TESAC
TK	TRACKOB
TR	THERMISTOR CHAIN (delayed mode)
XB	XBT

**Output :**

Un fichier Netcdf Coriolis, interface CO-IF0013 par type de capteur.

**Implémentation :**

15/11/2000

### 4.3. CO-IF0024 : grided data coriolis netcdf format

<b>Interface name</b>	Grided data coriolis netcdf format
Reference	CO-IF0024
<b>Source</b>	Function «coriolis products» CO-04
Destination	Function « dissemination » CO-05
<b>Operating mode</b>	-
Frequency	-
<b>Estimated volume</b>	-
Last update	09/03/2001

#### Dimension definitions

dimensions:

```

Level = 51 ;
Latitude = 33 ;
Longitude = 34 ;
N_DATA = 275 ;
N_PARAM = 1 ;
N_COOR = 3 ;
N_CODE = 4 ;
N_DATE_TIME = 19 ;
N_STRING4 = 4 ;
N_STRING16 = 16 ;

```

#### Variables

variables:

```

float Estimated_date ;
    Estimated_date:Conventions = "Julian day" ;
    Estimated_date:_FillValue = -9999.f ;
float South_latitude ;
    South_latitude:long_name = "South limit of the experiment" ;
    South_latitude:units = "degrees_north" ;
    South_latitude:_FillValue = -9999.f ;
    South_latitude:Conventions = "Decimal degrees, positive north" ;
    South_latitude:valid_min = -90. ;
    South_latitude:valid_max = 90. ;
float North_latitude ;
    North_latitude:long_name = "North limit of the experiment" ;
    North_latitude:units = "degrees_north" ;
    North_latitude:_FillValue = -9999.f ;
    North_latitude:Conventions = "Decimal degrees, positive north" ;
    North_latitude:valid_min = -90. ;
    North_latitude:valid_max = 90. ;
float West_longitude ;
    West_longitude:long_name = "West limit of the experiment" ;

```

```

        West_longitude:units = "degrees_east" ;
        West_longitude:_FillValue = -9999.f ;
        West_longitude:Conventions = "Decimal degrees, positive east" ;
        West_longitude:valid_min = -180. ;
        West_longitude:valid_max = 180. ;
    float East_longitude ;
        East_longitude:long_name = "East limit of the experiment" ;
        East_longitude:units = "degrees_east" ;
        East_longitude:_FillValue = -9999.f ;
        East_longitude:Conventions = "Decimal degrees, positive east, less than
West_longitude if the experiment crosses the 180" ;
        East_longitude:valid_min = -180. ;
        East_longitude:valid_max = 180. ;
    float Level(Level) ;
        Level:long_name = "Level" ;
        Level:units = "meters" ;
        Level:_FillValue = -9999.f ;
        Level:valid_min = 0.f ;
        Level:valid_max = 7000.f ;
    float Latitude(Latitude) ;
        Latitude:long_name = "Latitude of zone" ;
        Latitude:units = "degrees_north" ;
        Latitude:_FillValue = -9999.f ;
        Latitude:valid_min = -90.f ;
        Latitude:valid_max = 90.f ;
    float Longitude(Longitude) ;
        Longitude:long_name = "Longitude of zone" ;
        Longitude:units = "degrees_east" ;
        Longitude:_FillValue = -9999.f ;
        Longitude:valid_min = -180.f ;
        Longitude:valid_max = 180.f ;
    float Fact_error(Level) ;
        Fact_error:long_name = "Covariance Error Factor " ;
        Fact_error:_FillValue = -9999.f ;
    int Flag_180 ;
        Flag_180:long_name = "Does the area cross the 180 meridian? (e.g. Pacific Ocean)" ;
        Flag_180:Conventions = "1 : 180 crossed, 0 : 180 not crossed" ;
    char Parameters(N_PARAM, N_CODE) ;
        Parameters:long_name = "List of Parameters" ;
        Parameters:Conventions = "enhanced GF3 code" ;
    float TEMP(Level, Latitude, Longitude) ;
        TEMP:long_name = "Temperature in situ" ;
        TEMP:Conventions = "enhanced GF3 code" ;
        TEMP:Units = "degree Celsius" ;
        TEMP:_FillValue = -9999.f ;
    float TEMP_err(Level, Latitude, Longitude) ;
        TEMP_err:long_name = "estimated error on Temperature in situ" ;
        TEMP_err:Units = "degree Celsius" ;
        TEMP_err:_FillValue = -9999.f ;
        TEMP_err:valid_min = 0.f ;
    float Data_position(N_DATA, N_COOR) ;
        Data_position:long_name = "Position-Dates of the data" ;
        Data_position:Conventions = "lat long date" ;
        Data_position:_FillValue = -9999.f ;
    float Data_value(N_DATA, Level) ;
        Data_value:long_name = "Values of Data used by the analysis" ;
        Data_value:Conventions = "depending on data type" ;
        Data_value:Units = "depending on data type" ;
        Data_value:_FillValue = -9999.f ;
    char Data_type(N_DATA, N_CODE) ;
        Data_type:long_name = "Data Type of every profile" ;
        Data_type:Conventions = "enhanced GF3 code" ;
    char PLATFORM_NUMBER(N_DATA, N_STRING16) ;
        PLATFORM_NUMBER:long_name = "Platform number" ;

```

```

PLATFORM_NUMBER:Conventions = "To be defined" ;
PLATFORM_NUMBER:_FillValue = "" ;
char INST_TYPE(N_DATA, N_STRING4) ;
INST_TYPE:long_name = "Instrument type for each profile" ;
INST_TYPE:Conventions = "OMM probe code" ;
INST_TYPE:_FillValue = "" ;

```

### Global attributes

```

// global attributes:
:Title = "Essai NetCDF" ;
>Last_update = "26-Feb-2001 17:15:07" ;
:Description = "" ;
:Software = "CORIOLIS" ;
:Contents_type = "C" ;
:Data_types = "C" ;
:Analysis_period = "C" ;
:Experiment_name = "C" ;
:Project_name = "C" ;
:Data_manager = "C" ;
:Geographical_area = "C" ;
:Covariance_field = "ech_ls_x (m)= 150000 ; ech_ls_y (m)= 150000 ; ech_ls_t (j)=
60 ; ampli_ls = 1" ;
:Covariance_error = "ech_ms_x (m)= 50000 ; ech_ms_y (m)= 50000 ; ech_ms_t (j)=
10 ; ampli_ms = 1" ;
:Input_data_files =
"dat_mltnc_std/weekly/PF/CO_PF_20001017_2010_20000927_20001016_zzz.nc
...
dat_mltnc_std/weekly/PF/CO_PF_20001017_2010_20000927_20001016_zzz.nc" ;
:Reference_date_time = "01/01/1950 00:00:00" ;
:Estimated_date_time = "15/12/2000 00:00:00" ;
:Start_date_time = "15/12/2000 00:00:00" ;
:Stop_date_time = "15/12/2000 00:00:00" ;
:Reference_parameter = "DEPH" ;

```

### Data

```

data:
Estimated_date = 18611 ;
South_latitude = 33 ;
North_latitude = 49 ;
West_longitude = -25 ;
East_longitude = -8.5 ;
Level = 5, 10, 20, ..., 1900, 1950 ;
Latitude = 33, 33.5, 34, ..., 48.5, 49 ;
Longitude = -25, -24.5, ..., -9.5, -9, -8.5 ;
Fact_error = 1.168235, 1.14998, ..., 0.1444564, 0.1375847 ;
Flag_180 = 0 ;
Parameters =
"TEMP" ;
TEMP =
20.15765, 20.13289, ... .., 17.47281, 17.47281,
...
3.419777, 3.422966, ..., -NaN, -NaN ;
TEMP_err =
1.168235, 1.168235, ... 1.168235, 1.168235,
...
0.1375847, ..., 0.1375847, 0.1375847 ;

Data_position =
-16.36, 41.09, 18551.01,

```

```
...  
-39.406, 55.788, 18652.38 ;  
Data_value =  
2.30343, 2.309472... NaN, NaN,  
...  
-3.211526, -2.990488...NaN, NaN ;  
Data_type =  
"TEMP",  
...  
"TEMP" ;  
PLATFORM_NUMBER =  
"69031 " ,  
...  
"62690 " ;  
INST_TYPE =  
"840 " ,  
...  
"847 " ;
```

#### 4.4. CO-IF0022 : argo netcdf mono-profile format

<b>Interface name</b>	<b>ARGO NETCDF MONO-PROFILE FORMAT</b>
Reference	CO-IF0022
Version	0.99
Comment	This format is in a request for comments status. It has not yet been approved.
Contact	Thierry.Carval@ifremer.fr
Last update	13/04/01

The differences between version 0.99 and the previous proposal are typed in orange letters.

##### Dimension definitions

```
// Date and time dimension (YYYYMMDDHH24MISS)
DATE_TIME = 14 ;

// Null terminated string dimensions
STRING64 = 64 ;
STRING32 = 32 ;
STRING16 = 16 ;
STRING8 = 8 ;
STRING4 = 4 ;
STRING2 = 2 ;

// These dimensions depends on profiles
N_PARAM = 4 ;
N_LEVELS = 270 ;

// Quality control history
N_HISTORY = UNLIMITED;

// Number of technical parameters
N_TECH_PARAM = 25;
```

**Global attributes**

```
// global attributes
```

```
:DATA_TYPE = "Argo float";  
// Data type, EPIC requirement
```

```
:FORMAT_VERSION = "1.0";  
// File format version
```

```
:ARGO_VERSION = "1.0";  
// Argo data processing version
```

```
:CREATION_DATE = "YYYYMMDDHH24MI";  
// Date of file creation, Epic requirement
```

```
:REFERENCE_DATE_TIME = "";  
// Date of reference (convention : YYYYMMDDHH24MI) for Julian days  
// 199501010000 recommended as the first day (day one)
```

```
:PI_NAME = ""  
// PI's identifier (char*80 max.)
```

```
:PROJECT_NAME = "";  
// Name of the project (char*80 max.)
```

### General information on profile

#### // General information on profile

```

char STATION_PARAMETERS(N_PARAM,STRING4) ;
    STATION_PARAMETERS:long_name = "List of available parameters for the station" ;
    STATION_PARAMETERS:Conventions = "GF3 code list in (PRES, TEMP, PSAL, CNDC)";
    STATION_PARAMETERS:_FillValue = " " ;
char PLATFORM_NUMBER(STRING8) ;
    PLATFORM_NUMBER:long_name = "Float unique identifier" ;
    PLATFORM_NUMBER:Conventions = "Extended WMO float identifier : QA911111" ;
    PLATFORM_NUMBER:_FillValue = " " ;
int CYCLE_NUMBER;
    CYCLE_NUMBER:long_name = "Float cycle number" ;
    CYCLE_NUMBER:Conventions = "0..N, 0 : launch cycle (if exists), 1 : first complete cycle" ;
    CYCLE_NUMBER:_FillValue = 99999 ;
char DIRECTION;
    DIRECTION:long_name = "Direction of the station profiles" ;
    DIRECTION:Conventions = "A: ascending profiles, D: descending profiles " ;
    DIRECTION:_FillValue = " " ;
char DATA_CENTRE(STRING2) ;
    DATA_CENTRE:long_name = "Data centre in charge of float real-time processing" ;
    DATA_CENTRE:Conventions = "GTSP table" ;
    DATA_CENTRE:_FillValue = " " ;
char DATE_UPDATE(DATE_TIME) ;
    DATE:long_name = "Date of update of this file" ;
    DATE:Conventions = "YYYYMMDDHH24MISS" ;
    DATE:_FillValue = " " ;
char DC_REFERENCE(STRING8) ;
    DC_REFERENCE:long_name = "Station unique identifier in data centre" ;
    DC_REFERENCE:Conventions = "Data centre convention" ;
    DC_REFERENCE:_FillValue = " " ;
char DATA_TYPE ;
    DATA_TYPE:long_name = "Delayed mode or real time data" ;
    DATA_TYPE = "R : real time ; D : delayed mode" ;
    DATA_TYPE:_FillValue = " " ;
char INST_REFERENCE(STRING64) ;
    INST_REFERENCE:long_name = "Instrument type" ;
    INST_REFERENCE:Conventions = "Brand/type/serial number" ;
    INST_REFERENCE:_FillValue = " " ;
char INST_TYPE(STRING4) ;
    INST_TYPE:long_name = "Coded instrument type" ;
    INST_TYPE:Conventions = "WMO code table 1770 – instrument type" ;
    INST_TYPE:_FillValue = " " ;
char REC_TYPE(STRING2) ;
    REC_TYPE:long_name = "Recorder type" ;
    REC_TYPE:Conventions = "WMO code table 4770 – recorder type" ;
    REC_TYPE:_FillValue = " " ;
char DATE(DATE_TIME) ;
    DATE:long_name = "Date of the station" ;
    DATE:Conventions = "YYYYMMDDHH24MISS" ;
    DATE:_FillValue = " " ;
double JULD;
    JULD:long_name = "Julian day of the station relative to REFERENCE_DATE" ;
    JULD:units = "julian days" ;
    JULD:Conventions = "Relative julian days with decimal part (as parts of day)" ;
    JULD:_FillValue = 99999.f ;
    JULD:epic_code = 627 ;
double LATITUDE;
    LATITUDE:long_name = "Latitude of the station, best estimated value" ;
    LATITUDE:units = "degrees_north" ;
    LATITUDE:_FillValue = 99999.f ;
    LATITUDE:valid_min = -90.f ;

```

```

        LATITUDE:valid_max = 90.f ;
        LATITUDE:epic_code = 500;
double LONGITUDE;
        LONGITUDE:long_name = "Longitude of the station, best estimated value" ;
        LONGITUDE:units = "degrees_east" ;
        LONGITUDE:_FillValue = 99999.f ;
        LONGITUDE:valid_min = -180.f ;
        LONGITUDE:valid_max = 180.f ;
        LONGITUDE:epic_code = 502;
char Q_DATE;
        Q_DATE:long_name = "Quality on Date and Time" ;
        Q_DATE:Conventions = "Q where Q =[0,9]" ;
        Q_DATE:_FillValue = "9" ;
char Q_POSITION;
        Q_POSITION:long_name = "Quality on position (latitude and longitude)" ;
        Q_POSITION:Conventions = "Q where Q =[0,9]" ;
        Q_POSITION:_FillValue = "9" ;
char POSITIONNING_SYSTEM(STRING8) ;
        POSITIONNING_SYSTEM:long_name = "Positionning system" ;
        POSITIONNING_SYSTEM:Conventions = "ARGOS or GPS" ;
        POSITIONNING_SYSTEM:_FillValue = " " ;

// Following QC Variables may be discarded if the matching parameter is not measured
char Q_PROFILE_PRES;
        Q_PROFILE_PRES:long_name = "Global quality flag of pressure profile" ;
        Q_PROFILE_PRES:Conventions = "Q where Q =[0,9]" ;
        Q_PROFILE_PRES:_FillValue = "9" ;
char Q_PROFILE_TEMP;
        Q_PROFILE_TEMP:long_name = "Global quality flag of temperature profile" ;
        Q_PROFILE_TEMP:Conventions = "Q where Q =[0,9]" ;
        Q_PROFILE_TEMP:_FillValue = "9" ;
char Q_PROFILE_PSAI;
        Q_PROFILE_PSAI:long_name = "Global quality flag of practical salinity profile" ;
        Q_PROFILE_PSAI:Conventions = "Q where Q =[0,9]" ;
        Q_PROFILE_PSAI:_FillValue = "9" ;
char Q_PROFILE_CNDC;
        Q_PROFILE_CNDC:long_name = "Global quality flag of practical salinity profile" ;
        Q_PROFILE_CNDC:Conventions = "Q where Q =[0,9]" ;
        Q_PROFILE_CNDC:_FillValue = "9" ;
int Q_CHECK_SUMMARY;
        Q_CHECK_SUMMARY:long_name = "Bit string identifying passed quality checks" ;
        Q_CHECK_SUMMARY:Conventions = "GTSPP check ident : 1 bit per check" ;
        Q_CHECK_SUMMARY:_FillValue = 0x000000 ;

```

## Profile data

```
// Data for profiles
// Variable naming conventions:
// Measured value : GF3 code (mandatory if measured and mentioned in PARAMETERS),
// Associated QC flag : "QC_" + GF3 (mandatory if measured and mentioned in PARAMETERS),
// Error estimation : "Error_" + GF3 (not mandatory)
float PRES(N_LEVELS) ;
    PRES:long_name = "Pressure";
    PRES:_FillValue = 99999.f ;
    PRES:units = "decibar = 1000 Pa" ;
    PRES:valid_min = 0. ;
    PRES:valid_max = 15000. ;
    PRES:epic_code = 1 ;
    PRES:c_format = "%5.1f";
    PRES:fortran_format="F5.1";
    PRES:resolution=0.1;
float PRES_CORRECTED(N_LEVELS) ;
    PRES_CORRECTED:long_name = "Pressure" ;
    PRES_CORRECTED:_FillValue = 99999.f ;
    PRES_CORRECTED:units = "decibar = 1000 Pa" ;
    PRES_CORRECTED:valid_min = 0. ;
    PRES_CORRECTED:valid_max = 15000. ;
    PRES_CORRECTED:epic_code = 1 ;
    PRES_CORRECTED:c_format = "%5.1f";
    PRES_CORRECTED:fortran_format="F5.1";
    PRES_CORRECTED:resolution=0.1;
char PRES_QC(N_LEVELS) ;
    PRES_QC:long_name = "Global quality on pressure" ;
    PRES_QC:Conventions = "Q where Q =[0,9]" ;
    PRES_QC:_FillValue = "9" ;
float TEMP(N_LEVELS) ;
    TEMP:long_name = "Temperature in situ T90 scale" ;
    TEMP:_FillValue = 99999.f ;
    TEMP:units = "degres Celsius" ;
    TEMP:valid_min = -3. ;
    TEMP:valid_max = 40. ;
    TEMP:comment = "In situ measurement";
    TEMP:epic_code = 28 ;
    TEMP:c_format = "%5.3f";
    TEMP:fortran_format="F5.3";
    TEMP:resolution=0.001;
float TEMP_CORRECTED(N_LEVELS) ;
    TEMP_CORRECTED:long_name = "Temperature in situ T90 scale" ;
    TEMP_CORRECTED:_FillValue = 99999.f ;
    TEMP_CORRECTED:units = "degres Celsius" ;
    TEMP_CORRECTED:valid_min = -3. ;
    TEMP_CORRECTED:valid_max = 40. ;
    TEMP_CORRECTED:comment = "Corrected value";
    TEMP_CORRECTED:epic_code = 28 ;
    TEMP_CORRECTED:c_format = "%5.3f";
    TEMP_CORRECTED:fortran_format="F5.3";
    TEMP_CORRECTED:resolution=0.001;
char TEMP_QC(N_LEVELS) ;
    TEMP_QC:long_name = "temperature quality flag" ;
    TEMP_QC:Conventions = "Q where Q =[0,9]" ;
    TEMP_QC:_FillValue = "9" ;
float PSAL(N_LEVELS) ;
    PSAL:long_name = "Practical salinity, sal78" ;
    PSAL:_FillValue = 99999.f ;
    PSAL:units = "P.S.U." ;
    PSAL:valid_min = 0. ;
```

```

PSAL:valid_max = 60. ;
PSAL:comment = "In situ measurement, mandatory if salinity measured";
PSAL:epic_code = 41 ;
PSAL :c_format = "%5.3f";
PSAL :fortran_format="F5.3";
PSAL:resolution=0.001;
float PSAL_CORRECTED(N_LEVELS) ;
PSAL_CORRECTED:long_name = "Practical salinity, sal78" ;
PSAL_CORRECTED:_FillValue = 99999.f ;
PSAL_CORRECTED:units = "P.S.U." ;
PSAL_CORRECTED:valid_min = 0. ;
PSAL_CORRECTED:valid_max = 60. ;
PSAL_CORRECTED:comment = " Corrected value";
PSAL_CORRECTED:epic_code = 41 ;
PSAL_CORRECTED :c_format = "%5.3f";
PSAL_CORRECTED :fortran_format="F5.3";
PSAL_CORRECTED:resolution=0.001;
char PSAL_QC(N_LEVELS) ;
PSAL_QC:long_name = "practical salinity quality flag" ;
PSAL_QC:Conventions = "Q where Q =[0,9]" ;
PSAL_QC:_FillValue = "9" ;
float CNDC(N_LEVELS) ;
CNDC:long_name = "Electrical conductivity" ;
CNDC:_FillValue = 99999.f ;
CNDC:units = "mmho/cm " ;
CNDC:valid_min = 0. ;
CNDC:valid_max = 60. ;
CNDC:comment = "Not mandatory, available only if measured";
CNDC:epic_code = 50;
CNDC:c_format = "%5.3f";
CNDC:fortran_format="F5.3";
CNDC:resolution=0.001;
float CNDC_CORRECTED(N_LEVELS) ;
CNDC_CORRECTED:long_name = "Electrical conductivity" ;
CNDC_CORRECTED:_FillValue = 99999.f ;
CNDC_CORRECTED:units = "mmho/cm " ;
CNDC_CORRECTED:valid_min = 0. ;
CNDC_CORRECTED:valid_max = 60. ;
CNDC_CORRECTED:comment = "Not mandatory, available only if measured, corrected value";
CNDC_CORRECTED:epic_code = 50 ;
CNDC_CORRECTED:c_format = "%5.3f";
CNDC_CORRECTED:fortran_format="F5.3";
CNDC_CORRECTED:resolution=0.001;
char CNDC_QC(N_LEVELS) ;
CNDC_QC:long_name = "electrical conductivity quality flag" ;
CNDC_QC:Conventions = "Q where Q =[0,9]" ;
CNDC_QC:_FillValue = "9" ;
CNDC_QC:comment = " Not mandatory, available only if measured,";

```

### History data

```
// History on profiles QC
// Based on GTSP format
// One history row = same index for (INSTITUTION, SOFTWARE, SOFTWARE_RELEASE,
// ACTION, PARAMETER, IMMERSION, PREVIOUS_VALUE)
// N_HISTORY is unlimited to allow new history to be added after file creation
char CALIBRATION_INFO (STRING256) ;
    CALIBRATION_INFO:long_name = "Information about sensor calibration and/or sensor drift estimation";
    CALIBRATION_INFO:Conventions = "Text" ;
    CALIBRATION_INFO:_FillValue = " " ;

char HISTORY_INSTITUTION (N_HISTORY, STRING4) ;
    HISTORY_INSTITUTION:long_name = "Institution which performed QC or actions";
    HISTORY_INSTITUTION:Conventions = "GTSP institution code" ;
    HISTORY_INSTITUTION:_FillValue = " " ;
char HISTORY_SOFTWARE(N_HISTORY, STRING4) ;
    HISTORY_SOFTWARE:long_name = "Software which performed QC or actions " ;
    HISTORY_SOFTWARE:Conventions = "Institution dependent" ;
    HISTORY_SOFTWARE:_FillValue = " " ;
char HISTORY_SOFTWARE_RELEASE(N_HISTORY, STRING4) ;
    HISTORY_SOFTWARE_RELEASE:long_name = "Version/release of software which performed QC or actions " ;
    HISTORY_SOFTWARE_RELEASE:Conventions = "Institution dependent" ;
    HISTORY_SOFTWARE_RELEASE:_FillValue = " " ;
char HISTORY_ACTION(N_HISTORY, STRING4) ;
    HISTORY_ACTION:long_name = " QC or action performed on data" ;
    HISTORY_ACTION:Conventions = "GTSP (MEDS) action code" ;
    HISTORY_ACTION:_FillValue = " " ;
char HISTORY_PARAMETER(N_HISTORY, STRING4) ;
    HISTORY_PARAMETER:long_name = "Parameter QC action is performed on" ;
    HISTORY_PARAMETER:Conventions = "GF3 parameter code" ;
    HISTORY_PARAMETER:_FillValue = " " ;
float HISTORY_START_PRES(N_HISTORY) ;
    HISTORY_START_PRES:long_name = "Start pressure QC action applied on" ;
    HISTORY_START_PRES:_FillValue = 99999.f ;
    HISTORY_START_PRES:units = "dbars" ;
float HISTORY_STOP_PRES(N_HISTORY) ;
    HISTORY_STOP_PRES:long_name = "Stop pressure QC action applied on" ;
    HISTORY_STOP_PRES:_FillValue = 99999.f ;
    HISTORY_STOP_PRES:units = "dbars" ;
float HISTORY_PREVIOUS_VALUE(mN_HISTORY) ;
    HISTORY_PREVIOUS_VALUE:long_name = "Parameter/Flag previous value before QC action" ;
    HISTORY_PREVIOUS_VALUE:_FillValue = 99999.f ;
```

### Technical parameters

```
// Technical data from the float
// This information is not mandatory
CHAR TECHNICAL_PARAMETER_NAME(N_TECH_PARAM, STRING32)
    TECHNICAL_PARAMETER:long_name="Name of technical parameters for this cycle";
CHAR TECHNICAL_PARAMETER_VALUE(N_TECH_PARAM, STRING32)
    TECHNICAL_PARAMETER_VALUE:long_name="Value of technical parameters for this cycle";
```

Quality flags	
Value	Description
0	Unqualified
1	Correct value (all checks passed)
2	Probably good but value inconsistent with statistics (differ from climatology)
3	Probably bad (spike, gradient, ... if other tests passed)
4	Bad value, Impossible value (out of scale, vertical instability, constant profile, ...)
5	Value modified during quality control
6-7	Not used (available)
8	Interpolated value
9	Missing value

## 4.5. CO-IF0031 : argo meta-data format

<b>Interface name</b>	Argo meta-data
Reference	CO-IF0031
<b>Version</b>	0.2
Last update	24/04/01
<b>Comment</b>	This format is in a request for comments status. It is not yet validated

### Argo meta-data format version 0.2

```

#-----
#
#           meta file specifications
#           compiled by
#           Claudia Schmid
#
#-----
internal ID number :      currently an integer might become alphanumeric
                          (data center id + integer)
                          or obsolete (if replaced by extended WMO number)
transmission ID number : if ARGOS float: integer
                          if ORBCOMM float: I don't know
transmission type :      character string
                          ARGOS or ORBCOMM
WMO ID number :          integer
WMO instrument type (table 1770) : integer
WMO recorder type (table 4770) I : integer
ARGOS program number :   integer
                          Is this field obsolete for ORBCOMM floats?
                          If not we have to give it a new name ...
instrument type :        assigned by AOML for ARGO-USA
                          purpose: tells software what to do
                          Is there a requirement to coordinate this
                          between the data centers, or is the
                          detailed information below sufficient?
#-----
delay of first down time [hours] : real number
down time [days] :          real number
up time [hours] :            real number
transmission repetition rate [sec] : real number
clock drift (hours/hours) :  real number
                          obviously this allows only a linear clock drift correction
nominal drift depth [dbar] :  integer or real
max. departure from drift depth [dbar] : integer or real
                          for floats that control it actively
nominal profile depth [dbar] : integer or real
start time [yyyy mm dd hh mm (Z)] : UTC date and time the float was started
                                      (reset of float, not the setting of the internal clock)
                                      1997 07 23 04 30
launch time [yyyy mm dd hh mm (Z)] : UTC date and time 1997 07 23 10 41

```

launch position [lat lon] :	real numbers (longitude is -180 to 180)
deployment platform type :	RV, VOS, airplane (what else?)
deployment platform identification :	for research vessels: name for VOS: name or radio code? for airplanes: radio code?
deployment cruise id :	for ships only?
CTD station(s) :	give station number or NaN
#-----	
last cycle :	you will find the last cycle here if a float died otherwise the value is NaN
comment :	character string describe problems (reason for early death etc)
#-----	
temp. recalibration date [yyyy mm dd]	integers or empty
pres. recalibration date [yyyy mm dd]	integers or empty
#-----	
float manufacturer :	character string
float serial number :	alphanumeric
float provider :	first and last name
float provider institution :	use common abbreviation
originating country :	USA
float deployer :	first and last name
float deployer institution :	use common abbreviation
PI :	first and last name
PI institution :	use common abbreviation
PI email :	character string
#-----	
conductivity sensor type :	alphanumeric
conductivity sensor manufacturer :	name
conductivity sensor serial number :	alphanumeric
conductivity sensor accuracy :	real number
conductivity sensor resolution :	real number
temperature sensor type :	alphanumeric
temperature sensor manufacturer :	name
temperature sensor serial number :	alphanumeric
temperature sensor accuracy :	real number
temperature sensor resolution :	real number
pressure sensor type :	alphanumeric
pressure sensor manufacturer :	name
pressure sensor serial number :	alphanumeric
pressure sensor accuracy :	real number
pressure sensor resolution :	real number
battery type :	character string
initial battery voltage :	real number
Board type :	alphanumeric
Board serial number :	alphanumeric
CPU serial number :	alphanumeric
CPU date [yyyy mm dd] :	integers
AUX serial number :	alphanumeric
AUX date [yyyy mm dd] :	integers
software version :	alphanumeric
GPS (Y/N) :	if Y give GPS type, serial number etc
#-----	
calib eq 1 for pressure :	$OPB2 = PB1 + (PB2 - PB1) / AV1$
calib eq 2 for pressure :	$IF\ BIN \leq PB1\ P = (BIN - 0.5) * PGAIN * BLOK$
calib eq 3 for pressure :	$IF\ BIN > PB1\ P = (BIN - PB1 - 0.25) * AV1 * PGAIN * BLOK + P(PB1)$
calib eq 4 for pressure :	$IF\ BIN > OPB2\ P = (BIN - OPB2 - 0.25) * AV2 * PGAIN * BLOK + P(OPB2)$
calib coef for pressure :	blok=10; av1=2; av2=5; pb1=26; pb2=60; pgain=1.0446; poff=-104.42;
calib eq 1 for temperature :	$R = AT * (BT + TCNTS) / (CT + TCNTS)$
calib coef for temperature :	at=-27.39286; bt=-5863.026; ct=4366.056;
calib eq 2 for temperature :	$T = (H1 + H2 * \ln(R) + H3 * \ln(R) ** 3) ** -1 - 273.15$
calib coef for temperature :	h1=0.0011192; h2=0.000235665; h3=0.82431e-7;

---

```
calib eq 1 for conductivity : C = 32+CCNTS*cdiv*4+asal+TCNTS*(bsal*csal*TCNTS)
calib coef for conductivity : cdiv=0.002; asal=-0.68479e+01;
                               bsal=0.41177e02;csal=0.14414e-05;
calib eq 2 for conductivity : C = coff+cgain*C
calib coef for conductivity : coff=0.0; cgain=1.00;
calib eq 3 for conductivity : C = C*(1+(ctref-T)*7.5e-6+1.5E-8*P)
calib coef for conductivity : ctref=18.365;
calib eq 1 for cpu voltage : VCPU = CPUGAIN*CNTS+CPUOFF
calib coef for cpu voltage : cpugain=0.02703; cpuoff=0.18919;
calib eq 1 for pump voltage : VPMP = PMP_GAIN*CNTS+PMP_OFF
calib coef for pump voltage : pmp_gain=0.06061; pmp_off=1.0909;
#-----
#
#General remark:
#The calibration coefficients are part of each profile and they vary
#considerably between float types (the calibration information given
#here is hopefully much more complicated than than any future
#versions). Sylvie will work out a way to get them into netcdf ...
#
```

---

## 4.6. CO-IF0032 : argo netcdf trajectory data

<b>Interface name</b>	ARGO NETCDF TRAJECTORY DATA FORMAT
Reference	CO-IF0032
Version	0.1
Last update	03/04/01
Comment	This format is in a request for comments status. It is not yet validated

The differences with the previous version are typed in orange letters.

### Dimension definitions

```
// Date and time dimension (YYYYMMDDHH24MISS)
DATE_TIME = 14 ;

// Null terminated string dimensions
STRING2 = 2 ;
STRING4 = 4 ;
STRING8 = 8 ;
STRING64 = 64 ;

// These dimensions depends on data set
N_PARAM = 3;
N_MEASUREMENT = unlimited;
```

**Global attributes**

```
// global attributes

:DATA_TYPE = "Argo float";
// Data type, EPIC requirement

:FORMAT_VERSION = "1.0";
// File format version

:ARGO_VERSION = "1.0";
// Argo data processing version

:CREATION_DATE = "YYYYMMDDHH24MI";
// Date of file creation, Epic requirement

:REFERENCE_DATE_TIME = "";
// Date of reference (convention : YYYYMMDDHH24MI) for Julian days
// 199501010000 recommended as the first day (day one)

:PI_NAME = "";
// PI's identifier (char*80 max.)

:PROJECT_NAME = "";
// Name of the project (char*80 max.)

:FLOAT_PARKING_DEPTH = 1800;
// Pressure of the float parking depth
```

## General information on profile

### // General information on profile

```

char STATION_PARAMETERS(N_PARAM,STRING4) ;
    STATION_PARAMETERS:long_name = "List of available parameters for the station" ;
    STATION_PARAMETERS:Conventions = "GF3 code list in (PRES, TEMP, PSAL, CNDC)";
    STATION_PARAMETERS:_FillValue = " " ;
char PLATFORM_NUMBER(STRING8) ;
    PLATFORM_NUMBER:long_name = "Float unique identifier" ;
    PLATFORM_NUMBER:Conventions = "Extended WMO float identifier : QA911111" ;
    PLATFORM_NUMBER:_FillValue = " " ;
char DATA_CENTRE(STRING2) ;
    DATA_CENTRE:long_name = "Data centre in charge of float real-time processing" ;
    DATA_CENTRE:Conventions = "GTSP table" ;
    DATA_CENTRE:_FillValue = " " ;
char DATE_UPDATE(DATE_TIME) ;
    DATE_UPDATE:long_name = "Date of update of this file" ;
    DATE_UPDATE:Conventions = "YYYYMMDDHH24MISS" ;
    DATE_UPDATE:_FillValue = " " ;
char DC_REFERENCE(STRING8) ;
    DC_REFERENCE:long_name = "Station unique identifier in data centre" ;
    DC_REFERENCE:Conventions = "Data centre convention" ;
    DC_REFERENCE:_FillValue = " " ;
char DATA_TYPE ;
    DATA_TYPE:long_name = "Delayed mode or real time data" ;
    DATA_TYPE = "R : real time ; D : delayed mode" ;
    DATA_TYPE:_FillValue = " " ;
char INST_TYPE(STRING4) ;
    INST_TYPE:long_name = "Coded instrument type" ;
    INST_TYPE:Conventions = "WMO probe code table 1770" ;
    INST_TYPE:_FillValue = " " ;
char INST_REFERENCE(STRING64) ;
    INST_REFERENCE:long_name = "Instrument type" ;
    INST_REFERENCE:Conventions = "Brand/type/serial number" ;
    INST_REFERENCE:_FillValue = " " ;
char REC_TYPE(STRING2) ;
    REC_TYPE:long_name = "Recorder type" ;
    REC_TYPE:Conventions = "WMO recorder code table 4770" ;
    REC_TYPE:_FillValue = " " ;
char POSITIONNING_SYSTEM(STRING8) ;
    POSITIONNING_SYSTEM:long_name = "Positionning system" ;
    POSITIONNING_SYSTEM:Conventions = "ARGOS or GPS" ;
    POSITIONNING_SYSTEM:_FillValue = " " ;

// The following QC Variables may be discarded if the matching parameter is not measured
char Q_PROFILE_TEMP;
    Q_PROFILE_TEMP:long_name = "Global quality flag of temperature profile" ;
    Q_PROFILE_TEMP:Conventions = "Q where Q =[0,9]" ;
    Q_PROFILE_TEMP:_FillValue = "9" ;
char Q_PROFILE_PSAL;
    Q_PROFILE_PSAL:long_name = "Global quality flag of practical salinity profile" ;
    Q_PROFILE_PSAL:Conventions = "Q where Q =[0,9]" ;
    Q_PROFILE_PSAL:_FillValue = "9" ;
char Q_PROFILE_CNDC;
    Q_PROFILE_CNDC:long_name = "Global quality flag of practical salinity profile" ;
    Q_PROFILE_CNDC:Conventions = "Q where Q =[0,9]" ;
    Q_PROFILE_CNDC:_FillValue = "9" ;
int Q_CHECK_SUMMARY;
    Q_CHECK_SUMMARY:long_name = "Bit string identifying passed quality checks" ;
    Q_CHECK_SUMMARY:Conventions = "GTSP check ident : 1 bit per check" ;
    Q_CHECK_SUMMARY:_FillValue = 0x000000 ;

```

### Trajectory data

```
// Trajectory data
// Each row uses the same N_MEASUREMENT index for each parameter (date, latitude, longitude, temp...)
//
char DATE(N_MEASUREMENT, DATE_TIME) ;
    SURFACE_DATE:long_name = "Dates of each locations" ;
    SURFACE_DATE:Conventions = "YYYYMMDDHH24MISS" ;
    SURFACE_DATE:_FillValue = " " ;
char Q_DATE(N_MEASUREMENT);
    Q_DATE:long_name = "Quality on date and time" ;
    Q_DATE:Conventions = "Q where Q =[0,9]" ;
    Q_DATE:_FillValue = "9" ;
double JULD(N_MEASUREMENT) ;
    SURFACE_JULD:long_name = "Julian day of each locations relative to REFERENCE_DATE" ;
    SURFACE_JULD:units = "julian days" ;
    SURFACE_JULD:Conventions = "Relative julian days with decimal part (as parts of the day)" ;
    SURFACE_JULD:_FillValue = -99999.;
    SURFACE_JULD:epic_code = 627;
double LATITUDE(N_MEASUREMENT);
    LATITUDE:long_name = "Latitude of each locations" ;
    LATITUDE:units = "degrees_north" ;
    LATITUDE:_FillValue = -99999.f ;
    LATITUDE:valid_min = -90.f ;
    LATITUDE:valid_max = 90.f ;
    LATITUDE:epic_code = 500;
double LONGITUDE(N_MEASUREMENT);
    LONGITUDE:long_name = "Longitude of each locations " ;
    LONGITUDE:units = "degrees_east" ;
    LONGITUDE:_FillValue = -99999.f ;
    LONGITUDE:valid_min = -180.f ;
    LONGITUDE:valid_max = 180.f ;
    LONGITUDE:epic_code = 502;
char Q_POSITION(N_MEASUREMENT);
    Q_POSITION:long_name = "Estimated accuracy in latitude and longitude" ;
    Q_POSITION:Conventions = "Argos locations classes" ;
    Q_POSITION:_FillValue = "9" ;
float TEMP(N_MEASUREMENT);
    TEMP:long_name = "Temperature in situ T90 scale" ;
    TEMP:_FillValue = 9999.f ;
    TEMP:units = "degres Celsius" ;
    TEMP:valid_min = -3. ;
    TEMP:valid_max = 40. ;
    TEMP:comment = "In situ measurement";
    TEMP:epic_code = 28 ;
    TEMP:c_format = "%5.3f";
    TEMP:fortran_format="F5.3";
    TEMP:resolution=0.001;
float TEMP_CORRECTED(N_MEASUREMENT);
    TEMP_CORRECTED:long_name = "Temperature in situ T90 scale" ;
    TEMP_CORRECTED:_FillValue = 9999.f ;
    TEMP_CORRECTED:units = "degres Celsius" ;
    TEMP_CORRECTED:valid_min = -3. ;
    TEMP_CORRECTED:valid_max = 40. ;
    TEMP_CORRECTED:comment = "Corrected value";
    TEMP_CORRECTED:epic_code = 28 ;
    TEMP_CORRECTED:c_format = "%5.3f";
    TEMP_CORRECTED:fortran_format="F5.3";
    TEMP_CORRECTED:resolution=0.001;
char TEMP_QC(N_MEASUREMENT);
    TEMP_QC:long_name = "temperature quality flag" ;
    TEMP_QC:Conventions = "Q where Q =[0,9]" ;
    TEMP_QC:_FillValue = "9" ;
```

```

float PSAL(N_MEASUREMENT);
    PSAL:long_name = "Practical salinity, sal78" ;
    PSAL:_FillValue = 9999.f ;
    PSAL:units = "P.S.U." ;
    PSAL:valid_min = 0. ;
    PSAL:valid_max = 60. ;
    PSAL:comment = "In situ measurement, mandatory if salinity measured";
    PSAL:epic_code = 41 ;
    PSAL:c_format = "%5.3f";
    PSAL:fortran_format="F5.3";
    PSAL:resolution=0.001;

float PSAL_CORRECTED(N_MEASUREMENT);
    PSAL_CORRECTED:long_name = "Practical salinity, sal78" ;
    PSAL_CORRECTED:_FillValue = 9999.f ;
    PSAL_CORRECTED:units = "P.S.U." ;
    PSAL_CORRECTED:valid_min = 0. ;
    PSAL_CORRECTED:valid_max = 60. ;
    PSAL_CORRECTED:comment = " Corrected value";
    PSAL_CORRECTED:epic_code = 41 ;
    PSAL_CORRECTED:c_format = "%5.3f";
    PSAL_CORRECTED:fortran_format="F5.3";
    PSAL_CORRECTED:resolution=0.001;

char PSAL_QC(N_MEASUREMENT);
    PSAL_QC:long_name = "practical salinity quality flag" ;
    PSAL_QC:Conventions = "Q where Q =[0,9]" ;
    PSAL_QC:_FillValue = "9" ;

float CNDC(N_MEASUREMENT);
    CNDC:long_name = "Electrical conductivity" ;
    CNDC:_FillValue = 9999.f ;
    CNDC:units = "mmho/cm " ;
    CNDC:valid_min = 0. ;
    CNDC:valid_max = 60. ;
    CNDC:comment = "Not mandatory, available only if measured";
    CNDC:epic_code = 50;
    CNDC:c_format = "%5.3f";
    CNDC:fortran_format="F5.3";
    CNDC:resolution=0.001;

float CNDC_CORRECTED(N_MEASUREMENT);
    CNDC_CORRECTED:long_name = "Electrical conductivity" ;
    CNDC_CORRECTED:_FillValue = 9999.f ;
    CNDC_CORRECTED:units = "mmho/cm " ;
    CNDC_CORRECTED:valid_min = 0. ;
    CNDC_CORRECTED:valid_max = 60. ;
    CNDC_CORRECTED:comment = "Not mandatory, available only if measured, corrected value";
    CNDC_CORRECTED:epic_code = 50 ;
    CNDC_CORRECTED:c_format = "%5.3f";
    CNDC_CORRECTED:fortran_format="F5.3";
    CNDC_CORRECTED:resolution=0.001;

char CNDC_QC(N_MEASUREMENT) ;
    CNDC_QC:long_name = "electrical conductivity quality flag" ;
    CNDC_QC:Conventions = "Q where Q =[0,9]" ;
    CNDC_QC:_FillValue = "9" ;
    CNDC_QC:comment = " Not mandatory, available only if measured,";

integer FLOAT_CYCLE_NUMBER(N_MEASUREMENT);
char Q_PARKING_DEPTH(N_MEASUREMENT);
    Q_PARKING_DEPTH:long_name = "Quality on parking depth of the cycle" ;
    Q_PARKING_DEPTH:Conventions = "Q where Q =[0,9]" ;
    Q_PARKING_DEPTH:_FillValue = "9" ;

```

Quality flags	
Value	Description
0	Unqualified
1	Correct value (all checks passed)
2	Probably good but value inconsistent with statistics (differ from climatology)
3	Probably bad (spike, gradient, ... if other tests passed)
4	Bad value, Impossible value (out of scale, vertical instability, constant profile, ...)
5	Value modified during quality control
6-7	Not used (available)
8	Interpolated value
9	Missing value

Argos location classes	
Value	Estimated accuracy in latitude and longitude
3	< 150 m
2	150 m <= accuracy < 350 m
1	350 m <= accuracy < 1000 m
0	> 1000 m
A	No estimate of location accuracy
B	No estimate of location accuracy
Z	Invalid location