

# USER'S MANUAL

Version 2.2

August 21<sup>th</sup> 2009

**ARGO**

*part of the integrated global observation strategy*



**ARGO**

*part of the integrated global observation strategy*



Argo data management

User's manual

Ref : ar-um-02-01

Ref ifremer : cor-do/dti-mut/02-084

Version : 2.2

Date : 21/08/2009

Authors : Thierry Carval / Ifremer, Bob Keeley / MEDS, Yasushi Takatsuki / JAMSTEC, Takashi Yoshida / JMA, Stephen Loch / BODC, Claudia Schmid / AOML, Roger Goldsmith / WHOI, Annie Wong / UW, Rebecca McCreadie / BODC, Ann Thresher / CSIRO, Ahn Tran / MEDS

# Table of contents

<b>HISTORY OF THE DOCUMENT</b>	<b>6</b>
<b>1. INTRODUCTION</b>	<b>8</b>
1.1. ARGO PROGRAM, DATA MANAGEMENT CONTEXT	8
1.2. ARGO FLOAT CYCLES	8
1.3. REAL-TIME AND DELAYED MODE DATA	10
<b>2. FORMATS DESCRIPTION</b>	<b>11</b>
2.1. OVERVIEW OF THE FORMATS	11
2.2. ARGO PROFILE FILE FORMAT	12
2.2.1. DIMENSIONS AND DEFINITIONS	12
2.2.2. GENERAL INFORMATION ON THE PROFILE FILE	13
2.2.3. GENERAL INFORMATION FOR EACH PROFILE	13
2.2.4. MEASUREMENTS FOR EACH PROFILE	17
2.2.5. CALIBRATION INFORMATION FOR EACH PROFILE	19
2.2.6. HISTORY INFORMATION FOR EACH PROFILE	20
2.3. TRAJECTORY FORMAT	22
2.3.1. DIMENSIONS AND DEFINITIONS	22
2.3.2. GENERAL INFORMATION ON THE TRAJECTORY FILE	23
2.3.3. GENERAL INFORMATION ON THE FLOAT	23
2.3.4. LOCATIONS AND MEASUREMENTS FROM THE FLOAT	25
2.3.5. CYCLE INFORMATION FROM THE FLOAT	27
2.3.6. HISTORY INFORMATION	30
2.4. META-DATA FORMAT VERSION 2.2	32
2.4.1. DIMENSIONS AND DEFINITIONS	32
2.4.2. GENERAL INFORMATION ON THE META-DATA FILE	33
2.4.3. FLOAT CHARACTERISTICS	34
2.4.4. FLOAT DEPLOYMENT AND MISSION INFORMATION	36
2.4.5. FLOAT SENSOR INFORMATION	37
2.4.6. FLOAT CALIBRATION INFORMATION	37
2.4.7. FLOAT CYCLE INFORMATION	39
2.4.8. HIGHLY DESIRABLE META-DATA PARAMETERS	40
2.5. META-DATA FORMAT VERSION 2.3	41
2.5.1. DIMENSIONS AND DEFINITIONS	41
2.5.2. GENERAL INFORMATION ON THE META-DATA FILE	42
2.5.3. FLOAT CHARACTERISTICS	43
2.5.4. CONFIGURATION PARAMETERS	44
2.5.5. FLOAT SENSOR INFORMATION	46
2.5.6. FLOAT CALIBRATION INFORMATION	46
2.5.7. FLOAT CYCLE INFORMATION	48
2.5.8. HIGHLY DESIRABLE META-DATA PARAMETERS	49
2.6. TECHNICAL INFORMATION FORMAT VERSION 2.2	50
2.6.1. DIMENSIONS AND DEFINITIONS	50
2.6.2. GENERAL INFORMATION ON THE TECHNICAL DATA FILE	51

2.6.3.	TECHNICAL DATA	52
<b>2.7.</b>	<b>TECHNICAL INFORMATION FORMAT VERSION 2.3</b>	<b>53</b>
2.7.1.	DIMENSIONS AND DEFINITIONS	53
2.7.2.	GENERAL INFORMATION ON THE TECHNICAL DATA FILE	54
<b>2.7.3.</b>	<b>TECHNICAL DATA</b>	<b>55</b>
<b>2.8.</b>	<b>GDAC FTP DIRECTORY FILE FORMAT</b>	<b>56</b>
2.8.1.	PROFILE DIRECTORY FILE FORMAT	56
<b>2.8.2.</b>	<b>PROFILE DIRECTORY FILE FORMAT VERSION 2.1</b>	<b>57</b>
2.8.3.	TRAJECTORY DIRECTORY FORMAT	58
2.8.4.	META-DATA DIRECTORY FORMAT	59
<b>3.</b>	<b>REFERENCE TABLES</b>	<b>61</b>
<b>3.1.</b>	<b>REFERENCE TABLE 1 : DATA TYPE</b>	<b>61</b>
<b>3.2.</b>	<b>REFERENCE TABLE 2 : ARGO QUALITY CONTROL FLAG SCALE</b>	<b>61</b>
3.2.1.	REFERENCE TABLE 2 : MEASUREMENT FLAG SCALE	61
3.2.2.	REFERENCE TABLE 2A : PROFILE QUALITY FLAG	62
<b>3.3.</b>	<b>REFERENCE TABLE 3: PARAMETER CODE TABLE</b>	<b>63</b>
<b>3.4.</b>	<b>REFERENCE TABLE 4 : DATA CENTRES AND INSTITUTIONS CODES</b>	<b>64</b>
<b>3.5.</b>	<b>REFERENCE TABLE 5 : LOCATION CLASSES (ARGOS)</b>	<b>64</b>
<b>3.6.</b>	<b>REFERENCE TABLE 6 : DATA STATE INDICATORS</b>	<b>65</b>
<b>3.7.</b>	<b>REFERENCE TABLE 7 : HISTORY ACTION CODES</b>	<b>66</b>
<b>3.8.</b>	<b>REFERENCE TABLE 8 : INSTRUMENT TYPES</b>	<b>67</b>
<b>3.9.</b>	<b>REFERENCE TABLE 9 : POSITIONING SYSTEM</b>	<b>67</b>
<b>3.10.</b>	<b>REFERENCE TABLE 10 : TRANSMISSION SYSTEM</b>	<b>67</b>
<b>3.11.</b>	<b>REFERENCE TABLE 11 : QC TEST IDS</b>	<b>67</b>
<b>3.12.</b>	<b>REFERENCE TABLE 12 : HISTORY STEPS CODES</b>	<b>69</b>
<b>3.13.</b>	<b>REFERENCE TABLE 13 : OCEAN CODES</b>	<b>70</b>
<b>3.14.</b>	<b>REFERENCE TABLE 14 : TECHNICAL PARAMETER NAMES</b>	<b>71</b>
<b>3.15.</b>	<b>REFERENCE TABLE 15 : CONVENTION FOR DESCRIBING PROGRESS WITHIN A CYCLE</b>	<b>72</b>
<b>4.</b>	<b>DATA ACCESS</b>	<b>73</b>
4.1.	FILE NAMING CONVENTION ON GDACS	73
4.2.	OTHER DATA SOURCES	74
<b>5.</b>	<b>USING THE HISTORY SECTION OF THE ARGO NETCDF STRUCTURE</b>	<b>75</b>
5.1.	RECORDING INFORMATION ABOUT THE DELAYED MODE QC PROCESS	75
5.2.	RECORDING PROCESSING STAGES	76
5.3.	RECORDING QC TESTS PERFORMED AND FAILED	77
5.4.	RECORDING CHANGES IN VALUES	78
<b>6.</b>	<b>DAC-GDAC DATA-MANAGEMENT</b>	<b>80</b>
6.1.	GREYLIST FILES COLLECTION	80
6.2.	GDAC FILES REMOVAL	80



## History of the document

Version	Date	Comment
0.9	29/12/2001	Thierry Carval : creation of the document
0.9a	18/01/2002	Bob Keeley : general comments and updates
0.9a	24/01/2002	Valérie Harscoat : general comments and updates
0.9a	25/01/2002	Claudia Schmid : general comments and updates
0.9a	24/01/2002	Roger Goldsmith : general comments and updates
0.9b	05/03/2002	Roger Goldsmith, Yasushi Takatsuki and Claudia Schmid comments implemented.
0.9c	24/04/2002	Comments from version 0.9b are implemented
1.0	09/07/2002	Comments from version 0.9c are implemented
1.0a	31/12/2002	Missing values in trajectory and calibration
1.0a	17/01/2003	Description of directory file format
1.0a	24/01/2003	Update of reference tables
1.0a	24/01/2003	Update of "measurements of each profile" to handle corrected values
1.0a	24/01/2003	Increase the size of DC_REFERENCE from STRING16 to STRING32
1.0b	17/03/2003	Replace corrected values with adjusted values
1.0b	29/04/2003	DC_REFERENCE removed from trajectory format general information of the float section
1.0b	30/04/2003	Use blank fill values for character variables
1.0c	30/04/2003	Proposal submitted on 30/04/2003
1.0d	14/08/2003	Proposal submitted on 14/08/2003 (green font)
1.0e	23/10/2003	Proposal submitted on 12/11/2003 (green font)
2.0	12/11/2003	All comments from "Argo user's manual comments" ref ar-dm-02-02 implemented. General agreement from Argo data management meeting in Monterey (Nov. 5-7, 2003)
2.01	15/12/2003	History section updated.
2.01	01/10/2004	Meta-data section : WMO_INST_TYPE added to history section INSTRUMENT_TYPE renamed INST_REFERENCE
2.01	10/11/2004	Reference table 2 quality control flag scale updated by Annie Wong
2.01	10/11/2004	Updates in reference table 3, parameter codes table DOXY, TEMP_DOXY, TEMP (use ITS-90 scale)
2.01	23/11/2004	Reference table 14 : instrument failure mode added by Annie Wong
2.01	25/02/2005	Table 11 updated for frozen profile and deepest pressure tests from Rebecca Macreadie
2.01	28/02/2005	Table 4 updated : CSIO, China Second Institute of Oceanography
2.01	12/04/2005	Mathieu Belbeoch : table 5 updated : argos location classes
2.01	12/06/2005	Change lengths of all parameter name variables to accomodate longer parameter names. Affects: STATION_PARAMETERS (section 2.2.3), PARAMETER (section 2.2.5), and HISTORY_PARAMETER (section 2.2.6) in the profile format; TRAJECTORY_PARAMETERS (section 2.3.3) and HISTORY_PARAMETER (section 2.3.6) in the trajectory format; SENSOR (section 2.4.5) and PARAMETER (section 2.4.6) in the meta-data format
2.01	12/06/2005	Change ":"conventions" attribute and description of PROFILE_<PARAM>_QC in section 2.2.3.
2.01	12/06/2005	Add reference table 2a for the redefined PROFILE_<PARAM>_QC variables
2.01	20/06/2005	New long name for TEMP_DOXY in section 3.3
2.01	22/06/2005	Claudia Schmid : general update of trajectory file history section (N_MEASUREMENT dimension removed)
2.01	07/11/2005	Claudia Schmid : create reference table 14 for technical parameter names. Minor typo corrections.
2.01	07/11/2005	Thierry Carval : add a GPS code for position accuracy in ref. Table 5.
2.01	08/11/2005	Ann Thresher : exemple of sensor type in meta-data
2.01	09/11/2005	Annie Wong : §3.2.2 usage of <PARAM_ADJUSTED_QC> and <PARAM_QC> Reference table 2 updated (qc 3 and 4)
2.01	11/11/2005	Thierry Carval : §2.2.4, §2.3.4 accept adjusted parameters in real time files
2.01	11/11/2005	Thierry Carval : §2.2.6 history section for multi-profile files is empty
2.01	11/11/2005	Thierry Carval : §1.3, §2.2.3, §2.3.4 real-time adjusted data
2.01	11/11/2005	Thierry Carval : §2.4.8 highly desirable meta-data description
2.1	30/11/2005	Annie Wong : §3.2.1 update on flag 4 real time comment
2.1	20/12/2005	Thierry Carval : remove erroneous blanks (ex : "Argo reference table 3")
2.1	01/03/2006	Mark Ignaszewski: §2.3.6 Change HISTORY_*_INDEX to "int", Change HISTORY_REFERENCE to STRING64. Change to "dependent" in all sections. Remove PLATFORM_SERIAL_NO from desirable parameter table. Add "No QC performed" to Table 2a. Change FORMAT_VERSION to 2.2 in all sections.

2.1	26/09/2006	Thierry Carval §2.4.3 : TRANS_SYSTEM_ID : use N/A when not applicable (eg : Iridium or Orbcomm)
2.1	27/11/2006	Thierry Carval §2.4.8 : highly desirable metadata; PARKING_PRESSURE may be empty for floats drifting along a selected density level.
2.1	09/06/2008	Claudia Schmid §3.3: use DOXY2 for floats equipped with 2 oxygen sensors.
2.2	12/02/2009	Claudia Schmid §4.1 : file naming convention, multi-profiles cycle
2.2	03/03/2009	Thierry Carval §6.1 : greylist file collection §2.2.2 : move date_creation and date_update to "general information on profile file section".
2.2	21/08/2009	§2.6 : technical data format 2.3 §1.2 : new graphic for float cycles description §2.2.3 : add a firmware version to general information for profile §2.3.4 : add a "cycle_progress" in trajectory file §2.3.5 : add "cycle_phase" and "cycle" in trajectory file §2.4.3 : general review of float characteristics §2.4.5 : configuration parameters §2.4.8. : metadata file version 2.3 §2.8.2 : profile directory file format version 2.1 §3.3 : add BPHASE_DOXY §3.3 : remark on unit conversion of oxygen §6.2 : GDAC files removal

# 1. Introduction

This document is the Argo data user's manual.

It contains the description of the formats and files produced by the Argo DACs.

## 1.1. Argo program, data management context

The objective of Argo program is to operate and manage a set of 3000 floats distributed in all oceans, with the vision that the network will be a permanent and operational system.

The Argo data management group is creating a unique data format for internet distribution to users and for data exchange between national data centres (DACs) and global data centres (GDACs).

Profile data, metadata, trajectories and technical data are included in this standardization effort.

The Argo data formats are based on NetCDF because :

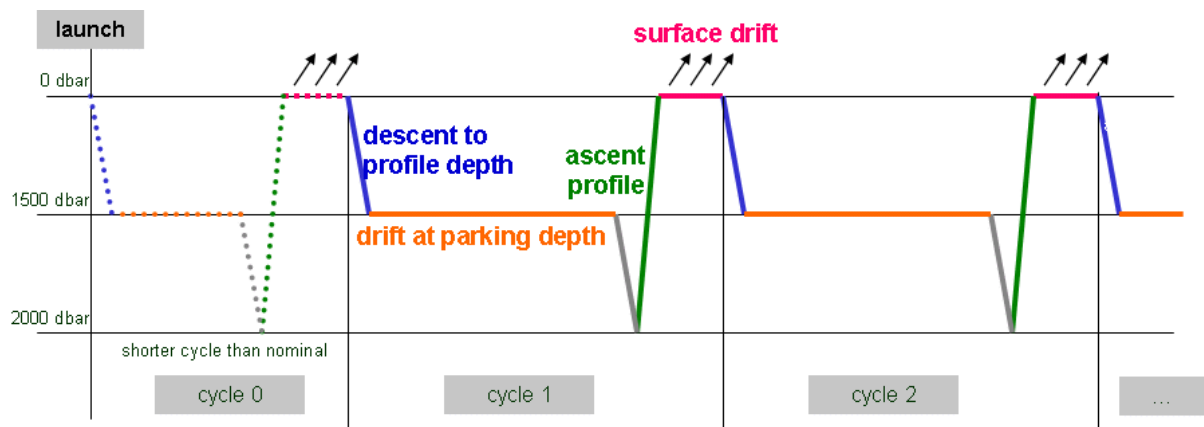
- It is a widely accepted data format by the user community,
- It is a self-describing format for which tools are widely available,
- It is a reliable and efficient format for data exchange.

## 1.2. Argo float cycles

A typical Argo float drifts for three years or more in the ocean. It continuously performs measurement cycles. Each cycle lasts about 10 days and can be divided into 4 phases:

- A descent from surface to a parking pressure (e.g. 1500 decibars),
- A subsurface drift at the parking pressure (e.g. 10 days),
- An ascent from a fixed pressure to surface (e.g. 2000 decibars),
- A surface drift with positioning and data transmission to a communication satellite (e.g. 8 hours).

Profile measurements (e.g. pressure, temperature, salinity) are performed during ascent, occasionally during descent. Subsurface measurements during parking are sometime performed (e.g. every 12 hours).



A typical Argo float performs continuously measurement cycle during 3 years or more in the ocean. A more detailed cycle description is available in reference table 15, chapter 3.15.

### Cycle naming convention

Float cycle numbers usually start at 1. The next cycles are increasing numbers (e.g. 2, 3,...N). If the float reports cycle number, this is what should be used in all Argo files.

Very conveniently some floats transmit their configuration during the transmissions before they descent for profile 1. Cycle 0 contains the first surface drift with technical data transmission or configuration information. This data is reported in the technical data files. Cycle 0 may contain subsurface measurements if a descending/ascending profile is performed before any data transmission. The time length of this cycle is usually shorter than the next nominal cycles. The cycle time is therefore regular only for later profiles and may be variable if the float is reprogrammed during its mission.

### 1.3. Real-time and Delayed mode data

Data from Argo floats are transmitted from the float, passed through processing and automatic quality control procedures as quickly as possible after the float begins reporting at the surface. The target is to issue the data to the GTS and Global Data servers within 24 hours of surfacing, or as quickly thereafter as possible. These are called real-time data.

The data are also issued to the Principle Investigators on the same schedule as they are sent to the Global servers. These scientists apply other procedures to check data quality and the target is for these data to be returned to the global data centres within 6 to 12 months. These constitute the delayed mode data.

The adjustments applied to delayed-data may also be applied to real-time data, to correct sensor drifts for real-time users. However, these real-time adjustments will be recalculated by the delayed mode quality control.

## 2. Formats description

### 2.1. Overview of the formats

Argo data formats are based on NetCDF from Ucar.

NetCDF (network Common Data Form) is an interface for array-oriented data access and a library that provides an implementation of the interface. The NetCDF library also defines a machine-independent format for representing scientific data. Together, the interface, library, and format support the creation, access, and sharing of scientific data. The NetCDF software was developed at the Unidata Program Centre in Boulder, Colorado. The [freely available](#) source can be obtained as [a compressed tar file](#) or [a zip file](#) from Unidata or from other [mirror sites](#).

- Ucar web site address : <http://www.ucar.edu/ucar>
- NetCDF documentation : <http://www.unidata.ucar.edu/packages/netcdf/index.html>

Argo formats are divided in 4 sections :

- Dimensions and definitions
- General information
- Data section
- History section

The Argo NetCDF formats do not contain any global attribute.

Argo date and times : all date and time have to be given in UTC time, universal time coordinates.

## 2.2. Argo profile file format

An Argo profile file contains a set of profiles. The minimum number is one profile. There is no defined maximum number of profiles.

A profile contains measurements performed at different pressures by an Argo float.

A profile contains typically 100 pressures, from 0 decibar (surface) to 2000 decibars (approximately 2000 meters depth).

For each pressure sample, there is a fixed number of parameters measured or calculated such as temperature, salinity or conductivity.

For file naming conventions, see §4.1 .

### 2.2.1. Dimensions and definitions

Name	Value	Definition
DATE_TIME	DATE_TIME = 14;	This dimension is the length of an ASCII date and time value. Date_time convention is : YYYYMMDDHHMISS <ul style="list-style-type: none"> <li>• YYYY : year</li> <li>• MM : month</li> <li>• DD : day</li> <li>• HH : hour of the day (as 0 to 23)</li> <li>• MI : minutes (as 0 to 59)</li> <li>• SS : seconds (as 0 to 59)</li> </ul> Date and time values are always in universal time coordinates (UTC). Examples : 20010105172834 : January 5 <sup>th</sup> 2001 17:28:34 19971217000000 : December 17 <sup>th</sup> 1997 00:00:00
STRING256 STRING64 STRING32 STRING16 STRING8 STRING4 STRING2	STRING256 = 256; STRING64 = 64; STRING32 = 32; STRING16 = 16; STRING8 = 8; STRING4 = 4; STRING2 = 2;	String dimensions from 2 to 256.
N_PROF	N_PROF = <int value>;	Number of profiles contained in the file. This dimension depends on the data set. A file contains at least one profile. There is no defined limit on the maximum number of profiles in a file. Example : N_PROF = 100
N_PARAM	N_PARAM = <int value> ;	Maximum number of parameters measured or calculated for a pressure sample. This dimension depends on the data set. Examples : (pressure, temperature) : N_PARAM = 2 (pressure, temperature, salinity) : N_PARAM = 3 (pressure, temperature, conductivity, salinity) : N_PARAM = 4
N_LEVELS	N_LEVELS = <int value> ;	Maximum number of pressure levels contained in a profile. This dimension depends on the data set. Example : N_LEVELS = 100
N_CALIB	N_CALIB = <int value> ;	Maximum number of calibrations performed on a profile. This dimension depends on the data set. Example : N_CALIB = 10
N_HISTORY	N_HISTORY = UNLIMITED;	Number of history records.

## 2.2.2. General information on the profile file

This section contains information about the whole file.

Name	Definition	Comment
DATA_TYPE	char DATA_TYPE(String16); DATA_TYPE:comment = "Data type"; DATA_TYPE:_FillValue = " ";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference table 1. Example : Argo profile
FORMAT_VERSION	char FORMAT_VERSION(String4); FORMAT_VERSION:comment = "File format version"; FORMAT_VERSION:_FillValue = " ";	File format version Example : «2.2»
HANDBOOK_VERSION	char HANDBOOK_VERSION(String4); HANDBOOK_VERSION:comment = "Data handbook version"; HANDBOOK_VERSION:_FillValue = " ";	Version number of the data handbook. This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook. Example : «1.0»
REFERENCE_DATE_TIME	char REFERENCE_DATE_TIME(Date_Time); REFERENCE_DATE_TIME:comment = "Date of reference for Julian days"; REFERENCE_DATE_TIME:conventions = "YYYYMMDDHHMISS"; REFERENCE_DATE_TIME:_FillValue = " ";	Date of reference for Julian days. The recommended reference date time is "19500101000000" : January 1 <sup>st</sup> 1950 00:00:00
DATE_CREATION	char DATE_CREATION(Date_Time); DATE_CREATION:comment = "Date of file creation "; DATE_CREATION:conventions = "YYYYMMDDHHMISS"; DATE_CREATION:_FillValue = " ";	Date and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS Example : 20011229161700 : December 29 <sup>th</sup> 2001 16 :17 :00
DATE_UPDATE	char DATE_UPDATE(Date_Time); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " ";	Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 <sup>th</sup> 2001 09 :05 :00

## 2.2.3. General information for each profile

This section contains general information on each profile.

Each item of this section has a N\_PROF (number of profiles) dimension.

Name	Definition	Comment
PLATFORM_NUMBER	char PLATFORM_NUMBER(N_PROF, String8); PLATFORM_NUMBER:long_name = "Float unique identifier"; PLATFORM_NUMBER:conventions = "WMO float identifier : A911111"; PLATFORM_NUMBER:_FillValue = " ";	WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : 6900045
PROJECT_NAME	char PROJECT_NAME(N_PROF, String64); PROJECT_NAME:comment = "Name of the project"; PROJECT_NAME:_FillValue = " ";	Name of the project which operates the profiling float that performed the profile. Example : GYROSCOPE (EU project for ARGO program)
PI_NAME	char PI_NAME (N_PROF, String64); PI_NAME:comment = "Name of the principal investigator"; PI_NAME:_FillValue = " ";	Name of the principal investigator in charge of the profiling float. Example : Yves Desaubies
STATION_PARAMETERS	char STATION_PARAMETERS(N_PROF, N_PARAM,String16);	List of parameters contained in this profile. The parameter names are listed in reference table

	STATION_PARAMETERS:long_name = "List of available parameters for the station"; STATION_PARAMETERS:conventions = "Argo reference table 3"; STATION_PARAMETERS:_FillValue = " ";	3. Examples : TEMP, PSAL, CNDC TEMP : temperature PSAL : practical salinity CNDC : conductivity
CYCLE_NUMBER	int CYCLE_NUMBER(N_PROF); 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;	Float cycle number. A profiling float performs cycles. In each cycle, it performs an ascending vertical profile, a subsurface drift and a surface drift. In some cases, it also performs a descending vertical profile. 0 is the number of the launch cycle. The subsurface drift of the cycle 0 may not be complete. 1 is the number of the first complete cycle. Example : 10 : cycle number 10
DIRECTION	char DIRECTION(N_PROF); DIRECTION:long_name = "Direction of the station profiles"; DIRECTION:conventions = "A: ascending profiles, D: descending profiles "; DIRECTION:_FillValue = " ";	Type of profile on which measurement occurs. A : ascending profile D : descending profile
DATA_CENTRE	char DATA_CENTRE(N_PROF, STRING2); DATA_CENTRE:long_name = "Data centre in charge of float data processing"; DATA_CENTRE:conventions = "Argo reference table 4"; DATA_CENTRE:_FillValue = " ";	Code for the data centre in charge of the float data management. The data centre codes are described in the reference table 4. Example : ME for MEDS
DC_REFERENCE	char DC_REFERENCE(N_PROF, STRING32); DC_REFERENCE:long_name = "Station unique identifier in data centre"; DC_REFERENCE:conventions = "Data centre convention"; DC_REFERENCE:_FillValue = " ";	Unique identifier of the profile in the data centre. Data centres may have different identifier schemes. DC_REFERENCE is therefore not unique across data centres.
DATA_STATE_INDICATOR	char DATA_STATE_INDICATOR(N_PROF, STRING4); DATA_STATE_INDICATOR:long_name = "Degree of processing the data have passed through"; DATA_STATE_INDICATOR:conventions = "Argo reference table 6"; DATA_STATE_INDICATOR:_FillValue = " ";	Degree of processing the data has passed through. The data state indicator is described in the reference table 6.
DATA_MODE	char DATA_MODE(N_PROF); DATA_MODE:long_name = "Delayed mode or real time data"; DATA_MODE:conventions = "R : real time; D : delayed mode; A : real time with adjustment"; DATA_MODE:_FillValue = " ";	Indicates if the profile contains real time, delayed mode or adjusted data. R : real time data D : delayed mode data A : real time data with adjusted values
INST_REFERENCE	char INST_REFERENCE(N_PROF, STRING64); INST_REFERENCE:long_name = "Instrument type"; INST_REFERENCE:conventions = "Brand, type, serial number"; INST_REFERENCE:_FillValue = " ";	References of the instrument : brand, type, serial number Example : APEX-SBE 259
FIRMWARE_VERSION	char FIRMWARE_VERSION(N_PROF, STRING10); FIRMWARE_VERSION:long_name = "Instrument version"; FIRMWARE_VERSION:conventions = ""; FIRMWARE_VERSION:_FillValue = " ";	Firmware version of the float. Example : "013108"
WMO_INST_TYPE	char WMO_INST_TYPE(N_PROF, STRING4); WMO_INST_TYPE:long_name = "Coded instrument type"; WMO_INST_TYPE:conventions = "Argo reference table 8"; WMO_INST_TYPE:_FillValue = " ";	Instrument type from WMO code table 1770. A subset of WMO table 1770 is documented in the reference table 8. Example : 846 : Webb Research float, Seabird sensor

JULD	double JULD(N_PROF); JULD:long_name = "Julian day (UTC) of the station relative to REFERENCE_DATE_TIME"; JULD:units = "days since 1950-01-01 00:00:00 UTC"; JULD:conventions = "Relative julian days with decimal part (as parts of day)"; JULD:_FillValue = 999999.;	Julian day of the profile <sup>1</sup> . The integer part represents the day, the decimal part represents the time of the profile. Date and time are in universal time coordinates. The julian day is relative to REFERENCE_DATE_TIME. Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_QC	char JULD_QC(N_PROF); JULD_QC:long_name = "Quality on Date and Time"; JULD_QC:conventions = "Argo reference table 2"; JULD_QC:_FillValue = " ";	Quality flag on JULD date and time. The flag scale is described in the reference table 2. Example : 1 : the date and time seems correct.
JULD_LOCATION	double JULD_LOCATION(N_PROF); JULD:long_name = "Julian day (UTC) of the location relative to REFERENCE_DATE_TIME "; JULD:units = "days since 1950-01-01 00:00:00 UTC"; JULD:conventions = "Relative julian days with decimal part (as parts of day)"; JULD:_FillValue = 999999.;	Julian day of the location of the profile (1). The integer part represents the day, the decimal part represents the time of the profile. Date and time are in universal time coordinates. The julian day is relative to REFERENCE_DATE_TIME. Example : 18833.8013889885 : July 25 2001 19:14:00
LATITUDE	double LATITUDE(N_PROF); LATITUDE:long_name = "Latitude of the station, best estimate"; LATITUDE:units = "degree_north"; LATITUDE:_FillValue = 99999.;; LATITUDE:valid_min = -90.;; LATITUDE:valid_max = 90.;	Latitude of the profile. Unit : degree north This field contains the best estimated latitude. The latitude value may be improved in delayed mode. The measured locations of the float are located in the trajectory file. Example : 44.4991 : 44° 29' 56.76" N
LONGITUDE	double LONGITUDE(N_PROF); LONGITUDE:long_name = "Longitude of the station, best estimate"; LONGITUDE:units = "degree_east"; LONGITUDE:_FillValue = 99999.;; LONGITUDE:valid_min = -180.;; LONGITUDE:valid_max = 180.;	Longitude of the profile. Unit : degree east This field contains the best estimated longitude. The longitude value may be improved in delayed mode. The measured locations of the float are located in the trajectory file. Example : 16.7222 : 16° 43' 19.92" E
POSITION_QC	char POSITION_QC(N_PROF); POSITION_QC:long_name = "Quality on position (latitude and longitude)"; POSITION_QC:conventions = "Argo reference table 2"; POSITION_QC:_FillValue = " ";	Quality flag on position. The flag on position is set according to (LATITUDE, LONGITUDE) quality. The flag scale is described in the reference table 2. Example : 1 : position seems correct.
POSITIONING_SYSTEM	char POSITIONING_SYSTEM(N_PROF, STRING8); POSITIONING_SYSTEM:long_name = "Positioning system"; POSITIONING_SYSTEM:_FillValue = " ";	Name of the system in charge of positioning the float locations from reference table 9. Examples : ARGOS
PROFILE_<PARAM>_QC	char PROFILE_<PARAM>_QC(N_PROF); PROFILE_<PARAM>_QC:long_name = "Global quality flag of <PARAM> profile"; PROFILE_<PARAM>_QC:conventions = "Argo reference table 2a"; PROFILE_<PARAM>_QC:_FillValue = " ";	Global quality flag on the PARAM profile. PARAM is among the STATION_PARAMETERS. The overall flag is set to indicate the percentage of good data in the profile as described in reference table 2a. Example : PROFILE_TEMP_QC = A : the temperature profile contains only good values PROFILE_Psal_QC = C : the salinity profile contains 50% to 75% good values

<sup>1</sup> Assume that a float profiles on its ascent. When the float first comes to the surface, it begins to transmit data. Each data transmission has a time attached to it and the earliest time is what is recorded in JULD. It is possible that the first transmission from a float cannot be used to derive its location. In this case, the time of location, JULD\_LOCATION, is different and later than the time of the profile.



## 2.2.4. Measurements for each profile

This section contains information on each level of each profile.

Each variable in this section has a N\_PROF (number of profiles), N\_LEVELS (number of pressure levels) dimension.

The original data received from the float and examined by real-time quality control should be placed in the <PARAM> and the QC flags set by the real-time process should be placed in the <PARAM>\_QC field. The values and flags in the <PARAM> fields should never be altered.

Each parameter can be adjusted (in delayed-mode, but also in real-time if appropriate). In that case, <PARAM>\_ADJUSTED contains the adjusted values, <PARAM>\_ADJUSTED\_QC contains the QC flags set by the delayed-mode process, and <PARAM>\_ADJUSTED\_ERROR contains the adjustment uncertainties.

A real-time data file with no adjusted data has an adjusted section with fill values (<PARAM>\_ADJUSTED, <PARAM>\_ADJUSTED\_QC and <PARAM>\_ADJUSTED\_ERROR).

The Argo profile delayed mode QC is described in "Argo quality control manual" by Annie Wong et Al.

Name	Definition	Comment
<PARAM>	float <PARAM>(N_PROF, N_LEVELS); <PARAM>:long_name = "<X>"; <PARAM>:_FillValue = <X>; <PARAM>:units = "<X>"; <PARAM>:valid_min = <X>; <PARAM>:valid_max = <X>; <PARAM>:comment = "<X>"; <PARAM>:C_format = "<X>"; <PARAM>:FORTRAN_format = "<X>"; <PARAM>:resolution = <X>;	<PARAM> contains the original values of a parameter listed in reference table 3. <X> : this field is specified in the reference table 3.
<PARAM>_QC	char <PARAM>_QC(N_PROF, N_LEVELS); <PARAM>_QC:long_name = "quality flag"; <PARAM>_QC:conventions = "Argo reference table 2"; <PARAM>_QC:_FillValue = " ";	Quality flag applied on each <PARAM> values. The flag scale is specified in table 2.
<PARAM>_ADJUSTED	float <PARAM>_ADJUSTED(N_PROF, N_LEVELS); <PARAM>_ADJUSTED:long_name = "<X>"; <PARAM>_ADJUSTED:_FillValue = <X>; <PARAM>_ADJUSTED:units = "<X>"; <PARAM>_ADJUSTED:valid_min = <X>; <PARAM>_ADJUSTED:valid_max = <X>; <PARAM>_ADJUSTED:comment = "<X>"; <PARAM>_ADJUSTED:C_format = "<X>"; <PARAM>_ADJUSTED:FORTRAN_format = "<X>"; <PARAM>_ADJUSTED:resolution = <X>;	<PARAM>_ADJUSTED contains the adjusted values derived from the original values of the parameter. <X> : this field is specified in the reference table 3. <PARAM>_ADJUSTED is mandatory. When no adjustment is performed, the FillValue is inserted.
<PARAM>_ADJUSTED_QC	char <PARAM>_ADJUSTED_QC(N_PROF, N_LEVELS); <PARAM>_ADJUSTED_QC:long_name = "quality flag"; <PARAM>_ADJUSTED_QC:conventions = "Argo reference table 2"; <PARAM>_ADJUSTED_QC:_FillValue = " ";	Quality flag applied on each <PARAM>_ADJUSTED values. The flag scale is specified in reference table 2. <PARAM>_ADJUSTED_QC is mandatory. When no adjustment is performed, the FillValue is inserted.
<PARAM>_ADJUSTED_ERROR	float <PARAM>_ADJUSTED_ERROR(N_PROF, N_LEVELS); <PARAM>_ADJUSTED_ERROR:long_name = "<X>"; <PARAM>_ADJUSTED_ERROR:_FillValue = <X>; <PARAM>_ADJUSTED_ERROR:units = "<X>"; <PARAM>_ADJUSTED_ERROR:comment = "Contains the	<PARAM>_ADJUSTED_ERROR contains the error on the adjusted values of the parameter. <X> : this field is specified in the reference table 3. <PARAM>_ADJUSTED_ERROR is

	error on the adjusted values as determined by the delayed mode QC process."; <PARAM>_ADJUSTED_ERROR:C_format = "<X>"; <PARAM>_ADJUSTED_ERROR:FORTTRAN_format = "<X>"; <PARAM>_ADJUSTED_ERROR:resolution= <X>;	mandatory. When no adjustment is performed, the FillValue is inserted.
--	--	--

*Example of a profiling float performing temperature measurements with adjusted values of temperature :*

Parameter definition : PRES, TEMP, TEMP_ADJUSTED
<pre> float PRES(N_PROF, N_LEVELS); PRES:long_name = "SEA PRESSURE (sea surface = 0)"; PRES:_FillValue = 99999.f; PRES:units = "decibar"; PRES:valid_min = 0.f; PRES:valid_max = 1200.f; PRES:comment = "In situ measurement, sea surface = 0"; PRES:C_format = "7.1f" ; PRES:FORTTRAN_format= "F7.1"; PRES:resolution= 0.1f;  char PRES_QC(N_PROF, N_LEVELS); PRES_QC:long_name = "quality flag"; PRES_QC:conventions = "Argo reference table 2"; PRES_QC:_FillValue = " ";  float TEMP(N_PROF, N_LEVELS); TEMP:long_name = "SEA TEMPERATURE"; TEMP:_FillValue = 99999.f; TEMP:units = "degree_Celsius"; TEMP:valid_min = -2.f; TEMP:valid_max = 40.f; TEMP:comment = "In situ measurement"; TEMP:C_format = "%9.3f"; TEMP:FORTTRAN_format = "F9.3"; TEMP:resolution = 0.001f;  char TEMP_QC(N_PROF, N_LEVELS); TEMP_QC:long_name = "quality flag"; TEMP_QC:conventions = "Argo reference table 2"; TEMP_QC:_FillValue = " ";  float TEMP_ADJUSTED(N_PROF, N_LEVELS); TEMP_ADJUSTED:long_name = "ADJUSTED SEA TEMPERATURE"; TEMP_ADJUSTED:_FillValue = 99999.f; TEMP_ADJUSTED:units = "degree_Celsius"; TEMP_ADJUSTED:valid_min = -2.f; TEMP_ADJUSTED:valid_max = 40.f; TEMP_ADJUSTED:comment = "Adjusted parameter"; TEMP_ADJUSTED:C_format = "%9.3f"; TEMP_ADJUSTED:FORTTRAN_format= "F9.3"; TEMP_ADJUSTED:resolution= 0.001f;  char TEMP_ADJUSTED_QC(N_PROF, N_LEVELS); TEMP_ADJUSTED_QC:long_name = "quality flag"; TEMP_ADJUSTED_QC:conventions = "Argo reference table 2"; TEMP_ADJUSTED_QC:_FillValue = " ";  float TEMP_ADJUSTED_ERROR(N_PROF, N_LEVELS); TEMP_ADJUSTED_ERROR:long_name = "ERROR ON ADJUSTED SEA TEMPERATURE"; TEMP_ADJUSTED_ERROR:_FillValue = 99999.f; TEMP_ADJUSTED_ERROR:units = "degree_Celsius"; TEMP_ADJUSTED_ERROR:comment = "Contains the error on the adjusted values as determined by the delayed mode QC process."; TEMP_ADJUSTED_ERROR :C_format = "%9.3f"; TEMP_ADJUSTED_ERROR :FORTTRAN_format= "F9.3"; TEMP_ADJUSTED_ERROR:resolution= 0.001f; </pre>

### 2.2.5. Calibration information for each profile

Calibrations are applied to parameters to create adjusted parameters. Different calibration methods will be used by groups processing Argo data. When a method is applied, its description is stored in the following fields.

This section contains calibration information for each parameter of each profile.

Each item of this section has a N\_PROF (number of profiles), N\_CALIB (number of calibrations), N\_PARAM (number of parameters) dimension.

If no calibration is available, N\_CALIB is set to 1, all values of calibration section are set to fill values.

Name	Definition	Comment
PARAMETER	char PARAMETER(N_PROF, N_CALIB, N_PARAM, STRING16); PARAMETER:long_name = "List of parameters with calibration information"; PARAMETER:conventions = "Argo reference table 3"; PARAMETER:_FillValue = " ";	Name of the calibrated parameter. The list of parameters is in reference table 3. Example : PSAL
SCIENTIFIC_CALIB_EQUATION	Char SCIENTIFIC_CALIB_EQUATION(N_PROF, N_CALIB, N_PARAM, STRING256); SCIENTIFIC_CALIB_EQUATION:long_name = "Calibration equation for this parameter"; SCIENTIFIC_CALIB_EQUATION:_FillValue = " ";	Calibration equation applied to the parameter. Example : $T_c = a_1 * T + a_0$
SCIENTIFIC_CALIB_COEFFICIENT	Char SCIENTIFIC_CALIB_COEFFICIENT(N_PROF, N_CALIB, N_PARAM, STRING256); SCIENTIFIC_CALIB_COEFFICIENT:long_name = "Calibration coefficients for this equation"; SCIENTIFIC_CALIB_COEFFICIENT:_FillValue = " ";	Calibration coefficients for this equation. Example : $a_1=0.99997$ , $a_0=0.0021$
SCIENTIFIC_CALIB_COMMENT	Char SCIENTIFIC_CALIB_COMMENT(N_PROF, N_CALIB, N_PARAM, STRING256); SCIENTIFIC_CALIB_COMMENT:long_name = "Comment applying to this parameter calibration"; SCIENTIFIC_CALIB_COMMENT:_FillValue = " ";	Comment about this calibration Example : The sensor is not stable
CALIBRATION_DATE	Char CALIBRATION_DATE (N_PROF, N_CALIB, N_PARAM, DATE_TIME) CALIBRATION_DATE:_FillValue = " ";	Date of the calibration. Example : 20011217161700

## 2.2.6. History information for each profile

This section contains history information for each action performed on each profile by a data centre.

Each item of this section has a N\_HISTORY (number of history records), N\_PROF (number of profiles) dimension.

A history record is created whenever an action is performed on a profile.

The recorded actions are coded and described in the history code table from the reference table 7.

On the GDAC, multi-profile history section is empty to reduce the size of the file. History section is available on mono-profile files, or in multi-profile files distributed from the web data selection.

Name	Definition	Comment
HISTORY_INSTITUTION	char HISTORY_INSTITUTION ( N_HISTORY, N_PROF, STRING4); HISTORY_INSTITUTION:long_name = "Institution which performed action"; HISTORY_INSTITUTION:conventions = "Argo reference table 4"; HISTORY_INSTITUTION:_FillValue = " ";	Institution that performed the action. Institution codes are described in reference table 4. Example : ME for MEDS
HISTORY_STEP	char HISTORY_STEP ( N_HISTORY, N_PROF, STRING4); HISTORY_STEP:long_name = "Step in data processing"; HISTORY_STEP:conventions = "Argo reference table 12"; HISTORY_STEP:_FillValue = " ";	Code of the step in data processing for this history record. The step codes are described in reference table 12. Example : ARGQ : Automatic QC of data reported in real-time has been performed
HISTORY_SOFTWARE	Char HISTORY_SOFTWARE ( N_HISTORY, N_PROF, STRING4); HISTORY_SOFTWARE:long_name = "Name of software which performed action"; HISTORY_SOFTWARE:conventions = "Institution dependent"; HISTORY_SOFTWARE:_FillValue = " ";	Name of the software that performed the action. This code is institution dependent. Example : WJO
HISTORY_SOFTWARE_RELEASE	Char HISTORY_SOFTWARE_RELEASE ( N_HISTORY, N_PROF, STRING4); HISTORY_SOFTWARE_RELEASE:long_name = "Version/release of software which performed action"; HISTORY_SOFTWARE_RELEASE:conventions = "Institution dependent"; HISTORY_SOFTWARE_RELEASE:_FillValue = " ";	Version of the software. This name is institution dependent. Example : «1.0»
HISTORY_REFERENCE	char HISTORY_REFERENCE ( N_HISTORY, N_PROF, STRING64); HISTORY_REFERENCE:long_name = "Reference of database"; HISTORY_REFERENCE:conventions = "Institution dependent"; HISTORY_REFERENCE:_FillValue = " ";	Code of the reference database used for quality control in conjunction with the software. This code is institution dependent. Example : WOD2001
HISTORY_DATE	char HISTORY_DATE( N_HISTORY, N_PROF, DATE_TIME); HISTORY_DATE:long_name = "Date the history record was created"; HISTORY_DATE:conventions = "YYYYMMDDHHMISS";	Date of the action. Example : 20011217160057

	<code>HISTORY_DATE:_FillValue = " ";</code>	
HISTORY_ACTION	<code>char HISTORY_ACTION( N_HISTORY, N_PROF, STRING4); HISTORY_ACTION:long_name = "Action performed on data"; HISTORY_ACTION:conventions = "Argo reference table 7"; HISTORY_ACTION:_FillValue = " ";</code>	Name of the action. The action codes are described in reference table 7. Example : QCF\$ for QC failed
HISTORY_PARAMETER	<code>char HISTORY_PARAMETER( N_HISTORY, N_PROF, STRING16); HISTORY_PARAMETER:long_name = "Station parameter action is performed on"; HISTORY_PARAMETER:conventions = "Argo reference table 3"; HISTORY_PARAMETER:_FillValue = " ";</code>	Name of the parameter on which the action is performed. Example : PSAL
HISTORY_START_PRES	<code>float HISTORY_START_PRES( N_HISTORY, N_PROF); HISTORY_START_PRES:long_name = "Start pressure action applied on"; HISTORY_START_PRES:_FillValue = 99999.f; HISTORY_START_PRES:units = "decibar";</code>	Start pressure the action is applied to. Example : 1500.0
HISTORY_STOP_PRES	<code>float HISTORY_STOP_PRES( N_HISTORY, N_PROF); HISTORY_STOP_PRES:long_name = "Stop pressure action applied on"; HISTORY_STOP_PRES:_FillValue = 99999.f; HISTORY_STOP_PRES:units = "decibar";</code>	Stop pressure the action is applied to. This should be greater than START_PRES. Example : 1757.0
HISTORY_PREVIOUS_VALUE	<code>float HISTORY_PREVIOUS_VALUE( N_HISTORY, N_PROF); HISTORY_PREVIOUS_VALUE:long_na me = "Parameter/Flag previous value before action"; HISTORY_PREVIOUS_VALUE:_FillVal ue = 99999.f;</code>	Parameter or flag of the previous value before action. Example : 2 (probably good) for a flag that was changed to 1 (good)
HISTORY_QCTEST	<code>char HISTORY_QCTEST( N_HISTORY, N_PROF, STRING16); HISTORY_QCTEST:long_name = "Documentation of tests performed, tests failed (in hex form)"; HISTORY_QCTEST:conventions = "Write tests performed when ACTION=QCP\$; tests failed when ACTION=QCF\$"; HISTORY_QCTEST:_FillValue = " ";</code>	This field records the tests performed when ACTION is set to QCP\$ (qc performed), the test failed when ACTION is set to QCF\$ (qc failed). The QCTEST codes are describe in reference table 11.  Example : 0A (in hexadecimal form)

The usage of the History section is described in §5 "Using the History section of the Argo netCDF Structure".

## 2.3. Trajectory format

An Argo trajectory file contains all received locations of an Argo float. There is one trajectory file per float.

In addition to locations, a trajectory file may contain measurements such as temperature, salinity or conductivity performed at some or all locations.

For file naming conventions, see §4.1 .

### 2.3.1. Dimensions and definitions

Name	Definition	Comment
DATE_TIME	DATE_TIME = 14;	This dimension is the length of an ASCII date and time value. Date_time convention is : YYYYMMDDHHMISS YYYY : year MM : month DD : day HH : hour of the day MI : minutes SS : seconds Date and time values are always in universal time coordinates (UTC). Examples : 20010105172834 : January 5 <sup>th</sup> 2001 17:28:34 19971217000000 : December 17 <sup>th</sup> 1997 00:00:00
STRING256 STRING64 STRING32 STRING16 STRING8 STRING4 STRING2	STRING256 = 256; STRING64 = 64; STRING32 = 32; STRING16 = 16; STRING8 = 8; STRING4 = 4; STRING2 = 2;	String dimensions from 2 to 256.
N_PARAM	N_PARAM = <int value> ;	Maximum number of parameters measured or calculated for a pressure sample. Examples : (pressure, temperature) : N_PARAM = 2 (pressure, temperature, salinity) : N_PARAM = 3 (pressure, temperature, conductivity, salinity) : N_PARAM = 4
N_MEASUREMENT	N_MEASUREMENT = unlimited;	This dimension is the number of recorded locations and measurements of the file.
N_CYCLE	N_CYCLE = <int value> ;	Maximum number of cycles performed by the float. This dimension depends on the data set. Example : N_CYCLE = 100
N_HISTORY	N_HISTORY = <int value> ;	Maximum number of history records for a location. This dimension depends on the data set Exemple : N_HISTORY = 10
N_HISTORY2	N_HISTORY2 = <int value> ;	Maximum dimension of the history record with respect to the time axis. This dimension depends on the data set. Example : N_HISTORY2 = 2 if only two values were corrected

### 2.3.2. General information on the trajectory file

This section contains information about the whole file.

Name	Definition	Comment
DATA_TYPE	char DATA_TYPE(String16); DATA_TYPE:comment = "Data type"; DATA_TYPE:_FillValue = " ";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference table 1. Example : Argo trajectory
FORMAT_VERSION	char FORMAT_VERSION(String4); FORMAT_VERSION:comment = "File format version "; FORMAT_VERSION:_FillValue = " ";	File format version Example : «2.2»
HANDBOOK_VERSION	char HANDBOOK_VERSION(String4); HANDBOOK_VERSION:comment = "Data handbook version"; HANDBOOK_VERSION:_FillValue = " ";	Version number of the data handbook. This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook. Example : «1.0»
REFERENCE_DATE_TIME	char REFERENCE_DATE_TIME(Date_Time); REFERENCE_DATE_TIME:comment = "Date of reference for Julian days"; REFERENCE_DATE_TIME:conventions = "YYYYMMDDHHMISS"; REFERENCE_DATE_TIME:_FillValue = " ";	Date of reference for julian days. The recommended reference date time is «19500101000000» : January 1 <sup>st</sup> 1950 00:00:00

### 2.3.3. General information on the float

This section contains general information on the float.

Name	Definition	Comment
PLATFORM_NUMBER	char PLATFORM_NUMBER(String8); PLATFORM_NUMBER:long_name = "Float unique identifier"; PLATFORM_NUMBER:conventions = "WMO float identifier : A911111"; PLATFORM_NUMBER:_FillValue = " ";	WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : 6900045
PROJECT_NAME	char PROJECT_NAME(String64); PROJECT_NAME:comment = "Name of the project"; PROJECT_NAME:_FillValue = " ";	Name of the project which operates the float that performed the trajectory. Example : GYROSCOPE (EU project for ARGO program)
PI_NAME	char PI_NAME (String64); PI_NAME:comment = "Name of the principal investigator"; PI_NAME:_FillValue = " ";	Name of the principal investigator in charge of the float. Example : Yves Desaubies
TRAJECTORY_PARAMETERS	char TRAJECTORY_PARAMETERS(N_PARAM,String16); TRAJECTORY_PARAMETERS:long_name = "List of available parameters for the station"; TRAJECTORY_PARAMETERS:conventions = "Argo reference table 3"; TRAJECTORY_PARAMETERS:_FillValue = " ";	List of parameters contained in this trajectory file. The parameter names are listed in reference table 3. Examples : TEMP, PSAL, CNDC TEMP : temperature PSAL : practical salinity CNDC : conductivity
DATA_CENTRE	char DATA_CENTRE(String2); DATA_CENTRE:long_name = "Data centre in charge of float data processing"; DATA_CENTRE:conventions = "Argo reference table 4"; DATA_CENTRE:_FillValue = " ";	Code for the data centre in charge of the float data management. The data centre codes are described in the reference table 4. Example : ME for MEDS
DATE_CREATION	char DATE_CREATION(Date_Time); DATE_CREATION:comment = "Date of	Date and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS

	file creation "; DATE_CREATION:conventions = "YYYYMMDDHHMISS"; DATE_CREATION:_FillValue = " ";	Example : 20011229161700 : December 29 <sup>th</sup> 2001 16 :17 :00
DATE_UPDATE	char DATE_UPDATE(DATE_TIME); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " ";	Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 <sup>th</sup> 2001 09 :05 :00
DATA_STATE_INDICATOR	char DATA_STATE_INDICATOR(STRING4); DATA_STATE_INDICATOR:long_name = "Degree of processing the data have passed through"; DATA_STATE_INDICATOR:conventions = "Argo reference table 6"; DATA_STATE_INDICATOR:_FillValue = " ";	Degree of processing the data has passed through. The data state indicator is described in the reference table 6.
INST_REFERENCE	char INST_REFERENCE(STRING64); INST_REFERENCE:long_name = "Instrument type"; INST_REFERENCE:conventions = "Brand, type, serial number"; INST_REFERENCE:_FillValue = " ";	Information about instrument : brand, type, serial number Example : APEX-SBE 259
WMO_INST_TYPE	char WMO_INST_TYPE(STRING4); WMO_INST_TYPE:long_name = "Coded instrument type"; WMO_INST_TYPE:conventions = "Argo reference table 8"; WMO_INST_TYPE:_FillValue = " ";	Instrument type from WMO code table 1770. A subset of WMO table 1770 is documented in the reference table 8. Example : 831
POSITIONING_SYSTEM	char POSITIONING_SYSTEM(STRING8); POSITIONING_SYSTEM:long_name = "Positioning system"; POSITIONING_SYSTEM:_FillValue = " ";	Name of the system used to derive the float locations, see reference table 9. Examples : ARGOS

### 2.3.4. Locations and measurements from the float

This section contains locations for one Argo float. It may also contain measurements performed along the trajectory.

Each field in this section has a `N_MEASUREMENT` dimension.

`N_MEASUREMENT` is the number of locations (or measurement) received from the float.

When no parameter is measured along the trajectory, `N_PARAM` (number of parameters) and any field with a `N_PARAM` dimension are removed from the file : `PARAM`, `PARAM_QC`, `PARAM_ADJUSTED`, `PARAM_ADJUSTED_QC`, `PARAM_ADJUSTED_ERROR` and `TRAJECTORY_PARAMETERS`.

The original data received from the float and examined by real-time quality control should be placed in the `<PARAM>` and the QC flags set by the real-time process should be placed in the `<PARAM>_QC` field. The values and flags in the `<PARAM>` fields should never be altered.

Each parameter can be adjusted. In that case, `<PARAM>_ADJUSTED` contains the adjusted values, `<PARAM>_ADJUSTED_QC` contains the QC flags set by the delayed-mode process, and `<PARAM>_ADJUSTED_ERROR` contains the adjustment uncertainties.

A file with no adjusted data contains adjusted sections with fill values (`<PARAM>_ADJUSTED`, `<PARAM>_ADJUSTED_QC` and `<PARAM>_ADJUSTED_ERROR`).

Name	Definition	Comment
DATA_MODE	char DATA_MODE(N_MEASUREMENT); DATA_MODE:long_name = "Delayed mode or real time data"; DATA_MODE:conventions = "R : real time; D : delayed mode; A : real time with adjustment"; DATA_MODE:_FillValue = " ";	Indicates if the profile contains real time or delayed mode data. R : real time data D : delayed mode data A : real time data with adjusted values
DC_REFERENCE	char DC_REFERENCE(N_MEASUREMENT STRING32); DC_REFERENCE:long_name = "Location unique identifier in data centre"; DC_REFERENCE:conventions = "Data centre convention"; DC_REFERENCE:_FillValue = " ";	Unique identifier of the location in the data centre. Data centres may have different identifier schemes. DC_REFERENCE is therefore not always unique across data centres.
JULD	double JULD(N_MEASUREMENT); JULD:long_name = "Julian day (UTC) of each measurement relative to REFERENCE_DATE_TIME"; JULD:units = "days since 1950-01-01 00:00:00 UTC"; JULD:conventions = "Relative julian days with decimal part (as parts of the day)"; JULD:_FillValue = 999999.;	Julian day of the location (or measurement). The integer part represents the day, the decimal part represents the time of the measurement. Date and time are in universal time coordinates. The julian day is relative to REFERENCE_DATE_TIME. Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_QC	char JULD_QC(N_MEASUREMENT); JULD_QC:long_name = "Quality on date and time"; JULD_QC:conventions = "Argo reference table 2"; JULD_QC:_FillValue = " ";	Quality flag on JULD date and time. The flag scale is described in the reference table 2. Example : 1 : the date and time seems correct.
LATITUDE	double LATITUDE(N_MEASUREMENT); LATITUDE:long_name = "Latitude of each location";	Latitude of the location (or measurement). Unit : degree north Example : 44.4991 for 44° 29' 56.76" N

	LATITUDE:units = "degree_north"; LATITUDE:_FillValue = 99999.; LATITUDE:valid_min = -90.; LATITUDE:valid_max = 90.;	
LONGITUDE	double LONGITUDE(N_MEASUREMENT); LONGITUDE:long_name = "Longitude of each location"; LONGITUDE:units = "degree_east"; LONGITUDE:_FillValue = 99999.; LONGITUDE:valid_min = -180.; LONGITUDE:valid_max = 180.;	Longitude of the location (or measurement). Unit : degree east Example : 16.7222 for 16° 43' 19.92" E
POSITION_ACCURACY	char POSITION_ACCURACY(N_MEASUREMENT); POSITION_ACCURACY:long_name = "Estimated accuracy in latitude and longitude"; POSITION_ACCURACY:conventions = "Argo reference table 5"; POSITION_ACCURACY:_FillValue = " ";	Position accuracy received from the positioning system. The location classes from ARGOS are described in the reference table 5. Example : 3 for a latitude and longitude accuracy < 150 m.
POSITION_QC	char POSITION_QC(N_MEASUREMENT); POSITION_QC:long_name = "Quality on position"; POSITION_QC:conventions = "Argo reference table 2"; POSITION_QC:_FillValue = " ";	Quality flag on position. The flag on position is set according to (LATITUDE, LONGITUDE, JULD) quality. The flag scale is described in the reference table 2. Example : 1 : position seems correct.
CYCLE_NUMBER	int CYCLE_NUMBER(N_MEASUREMENT); CYCLE_NUMBER:long_name = "Float cycle number of the measurement"; CYCLE_NUMBER:conventions = "0..N, 0 : launch cycle, 1 : first complete cycle"; CYCLE_NUMBER:_FillValue = 99999;	Cycle number of the float for this measurement. For one cycle number, there are usually several locations/measurement received. Example : 17 for measurements performed during the 17 <sup>th</sup> cycle of the float.
CYCLE_PROGRESS	int CYCLE_PROGRESS (N_MEASUREMENT); CYCLE_PROGRESS:long_name = "Cycle progress"; CYCLE_PROGRESS:conventions = "Argo reference table 15"; CYCLE_PROGRESS:_FillValue = 99999;	Cycle progress, see reference table 15 for conventions on cycle progress within a cycle. Examples: 1 : descent start 2 : descent end 200 : parking pressure phase 400 : ascent to surface 500 : surface drift phase
<PARAM>	float <PARAM>(N_MEASUREMENT); <PARAM>:long_name = "<X>"; <PARAM>:_FillValue = <X>; <PARAM>:units = "<X>"; <PARAM>:valid_min = <X>; <PARAM>:valid_max = <X>; <PARAM>:comment = "<X>"; <PARAM>:C_format = "<X>"; <PARAM>:FORTRAN_format = "<X>"; <PARAM>:resolution = <X>;	<PARAM> contains the original values of a parameter listed in reference table 3. <X> : this field is specified in the reference table 3.
<PARAM>_QC	char <PARAM>_QC(N_MEASUREMENT); <PARAM>_QC:long_name = "quality flag"; <PARAM>_QC:conventions = "Argo reference table 2"; <PARAM>_QC:_FillValue = " ";	Quality flag applied on each <PARAM> values. The flag scale is specified in table 2.
<PARAM>_ADJUSTED	float <PARAM>_ADJUSTED(N_MEASUREMENT); <PARAM>_ADJUSTED:long_name = "<X>"; <PARAM>_ADJUSTED:_FillValue = <X>; <PARAM>_ADJUSTED:units = "<X>"; <PARAM>_ADJUSTED:valid_min = <X>; <PARAM>_ADJUSTED:valid_max = <X>; <PARAM>_ADJUSTED:comment = "<X>"; <PARAM>_ADJUSTED:C_format = "<X>"; <PARAM>_ADJUSTED:FORTRAN_format = "<X>"; <PARAM>_ADJUSTED:resolution= <X>;	<PARAM>_ADJUSTED contains the adjusted values derived from the original values of the parameter. <X> : this field is specified in the reference table 3. <PARAM>_ADJUSTED is mandatory. When no adjustment is performed, the FillValue is inserted.
<PARAM>_ADJUSTED_QC	char <PARAM>_ADJUSTED_QC(N_MEASUREMENT); <PARAM>_ADJUSTED_QC:long_name = "quality flag"; <PARAM>_ADJUSTED_QC:conventions =	Quality flag applied on each <PARAM>_ADJUSTED values. The flag scale is specified in reference table 2. <PARAM>_ADJUSTED_QC is mandatory. When no adjustment is performed, the FillValue is inserted.

	"Argo reference table 2"; <PARAM>_ADJUSTED_QC:_FillValue = " ";	
<PARAM>_ADJUSTED_ERROR	float <PARAM>_ADJUSTED_ERROR(N_MEASUREMENT); <PARAM>_ADJUSTED_ERROR:long_name = "<X>"; <PARAM>_ADJUSTED_ERROR:_FillValue = <X>; <PARAM>_ADJUSTED_ERROR:units = "<X>"; <PARAM>_ADJUSTED_ERROR:comment = "Contains the error on the adjusted values as determined by the delayed mode QC process."; <PARAM>_ADJUSTED_ERROR:C_format = "<X>"; <PARAM>_ADJUSTED_ERROR:FORTRAN_format = "<X>"; <PARAM>_ADJUSTED_ERROR:resolution=<X>;	<PARAM>_ADJUSTED_ERROR contains the error on the adjusted values of the parameter. <X> : this field is specified in the reference table 3. <PARAM>_ADJUSTED_ERROR is mandatory. When no adjustment is performed, the FillValue is inserted.

### 2.3.5. Cycle information from the float

This section contains information on the cycles performed by the float.

Each field in this section has a N\_CYCLE dimension.

N\_CYCLE is the number of cycles performed by the float.

Name	Definition	Comment
JULD_ASCENT_START	double JULD_ASCENT_START(N_CYCLE); JULD_ASCENT_START:long_name = "Start date of the ascending profile"; JULD_ASCENT_START:units = "days since 1950-01-01 00:00:00 UTC"; JULD_ASCENT_START:conventions = "Relative julian days with decimal part (as part of day)"; JULD_ASCENT_START:_FillValue=999999;	Julian day (UTC) of the beginning of the ascending profile. Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_ASCENT_START_STATUS	Char JULD_ASCENT_START_STATUS(N_CYCLE); JULD_ASCENT_START_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 : Transmitted"; JULD_ASCENT_START_STATUS:_FillValue = " ";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float 9 : date is unknown
JULD_ASCENT_END	double JULD_ASCENT_END(N_CYCLE); JULD_ASCENT_END:long_name = "End date of the ascending profile"; JULD_ASCENT_END:units = "days since 1950-01-01 00:00:00 UTC"; JULD_ASCENT_END:conventions = "Relative julian days with decimal part (as part of day)"; JULD_ASCENT_END:_FillValue=999999.;	Julian day (UTC) of the end of the ascending profile. Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_ASCENT_END_STATUS	Char JULD_ASCENT_END_STATUS(N_CYCLE); JULD_ASCENT_END_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 : Transmitted"; JULD_ASCENT_END_STATUS:_FillValue = " ";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float 9 : date is unknown
JULD_DESCENT_START	double	Julian day (UTC) of the beginning of the descending profile.

RT	JULD_DESCENT_START(N_CYCLE); JULD_DESCENT_START:long_name = "Descent start date of the cycle"; JULD_DESCENT_START:units = "days since 1950-01-01 00:00:00 UTC"; JULD_DESCENT_START:conventions = "Relative julian days with decimal part (as part of day)"; JULD_DESCENT_START:_FillValue=999999.;	Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_DESCENT_START_STATUS	Char JULD_DESCENT_START_STATUS(N_CYCLE); JULD_DESCENT_START_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 : Transmitted"; JULD_DESCENT_START_STATUS:_FillValue = " ";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float 9 : date is unknown
JULD_DESCENT_END	double JULD_DESCENT_END(N_CYCLE); JULD_DESCENT_END:long_name = "Descent end date of the cycle"; JULD_DESCENT_END:units = "days since 1950-01-01 00:00:00 UTC"; JULD_DESCENT_END:conventions = "Relative julian days with decimal part (as part of day)"; JULD_DESCENT_END:_FillValue=999999.;	Julian day (UTC) of the end of the descending profile. Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_DESCENT_END_STATUS	char JULD_DESCENT_END_STATUS(N_CYCLE); JULD_DESCENT_END_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 : Transmitted"; JULD_DESCENT_END_STATUS:_FillValue = " ";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float 9 : date is unknown
JULD_START_TRANSMISSION	double JULD_START_TRANSMISSION(N_CYCLE); JULD_START_TRANSMISSION:long_name = "Start date of transmission"; JULD_START_TRANSMISSION:units = "days since 1950-01-01 00:00:00 UTC"; JULD_START_TRANSMISSION:conventions = "Relative julian days with decimal part (as part of day)"; JULD_START_TRANSMISSION:_FillValue=999999.;	Julian day (UTC) of the beginning of data transmission. Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_START_TRANSMISSION_STATUS	char JULD_START_TRANSMISSION_STATUS(N_CYCLE); JULD_START_TRANSMISSION_STATUS:conventions = "0 : Nominal, 1 : Estimated, 2 : Transmitted"; JULD_START_TRANSMISSION_STATUS:_FillValue = " ";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float 9 : date is unknown
GROUNDED	char GROUNDED(N_CYCLE); GROUNDED:long_name = "Did the profiler touch the ground for that cycle"; GROUNDED:conventions = "Y,N,U"; GROUNDED:_FillValue = " ";	GROUNDED indicates if the float touched the ground for that cycle. Format : Y, N, U Examples : Y : yes the float touched the ground N : no U : unknown
CYCLE_PHASE	int CYCLE_PHASE (N_CYCLE); CYCLE_PHASE:long_name = "Phase of the cycle described in metadata"; CYCLE_PHASE:conventions = "Y,N,U"; CYCLE_PHASE:_FillValue = " ";	Phase of that cycle related to CONFIGURATION_PHASE in the float metadata (see Configuration parameters of metadata files description).
CYCLE	int CYCLE(N_CYCLE); CYCLE:long_name = "Float cycle number of the measurement"; CYCLE:conventions = "0..N, 0 : launch cycle, 1 : first complete cycle";	Cycle number of the float. For one cycle number, there is a collection of useful information recorded (e.g. grounded or not). Example : 17 for measurements performed during the 17 <sup>th</sup> cycle of the float.

	CYCLE_FillValue = 99999	
--	-------------------------	--

### 2.3.6. History information

This section contains history information for each action performed on each measurement.

Each item of this section has a N\_MEASUREMENT (number of locations or measurements), N\_HISTORY (number of history records) dimension.

Name	Definition	Comment
HISTORY_INSTITUTION	char HISTORY_INSTITUTION (N_HISTORY, STRING4); HISTORY_INSTITUTION:long_name = "Institution which performed action"; HISTORY_INSTITUTION:conventions = "Argo reference table 4"; HISTORY_INSTITUTION:_FillValue = " ";	Institution that performed the action. Institution codes are described in reference table 4. Example : ME for MEDS
HISTORY_STEP	char HISTORY_STEP (N_HISTORY, STRING4); HISTORY_STEP:long_name = "Step in data processing"; HISTORY_STEP:conventions = "Argo reference table 12"; HISTORY_STEP:_FillValue = " ";	Code of the step in data processing for this history record. The step codes are described in reference table 12. Example : ARGQ : Automatic QC of data reported in real-time has been performed
HISTORY_SOFTWARE	Char HISTORY_SOFTWARE (N_HISTORY, STRING4); HISTORY_SOFTWARE:long_name = "Name of software which performed action"; HISTORY_SOFTWARE:conventions = "Institution dependent"; HISTORY_SOFTWARE:_FillValue = " ";	Name of the software that performed the action. This code is institution dependent. Example : WJO
HISTORY_SOFTWARE_RELEASE	Char HISTORY_SOFTWARE_RELEASE (N_HISTORY, STRING4); HISTORY_SOFTWARE_RELEASE:long_name = "Version/release of software which performed action"; HISTORY_SOFTWARE_RELEASE:conventions = "Institution dependent"; HISTORY_SOFTWARE_RELEASE:_FillValue = " ";	Version of the software. This name is institution dependent. Example : «1.0»
HISTORY_REFERENCE	char HISTORY_REFERENCE (N_HISTORY, STRING64); HISTORY_REFERENCE:long_name = "Reference of database"; HISTORY_REFERENCE:conventions = "Institution dependent"; HISTORY_REFERENCE:_FillValue = " ";	Code of the reference database used for quality control in conjunction with the software. This code is institution dependent. Example : WOD2001
HISTORY_DATE	char HISTORY_DATE(N_HISTORY, DATE_TIME); HISTORY_DATE:long_name = "Date the history record was created"; HISTORY_DATE:conventions = "YYYYMMDDHHMISS"; HISTORY_DATE:_FillValue = " ";	Date of the action. Example : 20011217160057
HISTORY_ACTION	char HISTORY_ACTION (N_HISTORY, STRING64); HISTORY_ACTION:long_name = "Action performed on data"; HISTORY_ACTION:conventions = "Argo reference table 7"; HISTORY_ACTION:_FillValue = " ";	Name of the action. The action codes are described in reference table 7. Example : QCF\$ for QC failed

HISTORY_PARAMETER	Char HISTORY_PARAMETER(N_HISTORY, STRING16); HISTORY_PARAMETER:long_name = "Station parameter action is performed on"; HISTORY_PARAMETER:conventions = "Argo reference table 3"; HISTORY_PARAMETER:_FillValue = " ";	Name of the parameter on which the action is performed. Example : PSAL
HISTORY_PREVIOUS_VALUE	Float HISTORY_PREVIOUS_VALUE(N_HISTORY, N_HISTORY2); HISTORY_PREVIOUS_VALUE:long_name = "Parameter/Flag previous value before action"; HISTORY_PREVIOUS_VALUE:_FillValue = 99999.f;	Parameter or flag of the previous value before action. Example : 2 (probably good) for a flag that was changed to 1 (good)
HISTORY_INDEX_DIMENSION	char HISTORY_INDEX_DIMENSION(N_HISTORY);	Name of dimension to which HISTORY_START_INDEX and HISTORY_STOP_INDEX correspond. C: N_CYCLE M: N_MEASUREMENT
HISTORY_START_INDEX	int HISTORY_START_INDEX (N_HISTORY); HISTORY_START_INDEX:long_name = "Start index action applied on"; HISTORY_START_INDEX:_FillValue = 99999;	Start index the action is applied to. This index corresponds to N_MEASUREMENT or N_CYCLE, depending on the corrected parameter Example : 100
HISTORY_STOP_INDEX	int HISTORY_STOP_INDEX (N_HISTORY); HISTORY_STOP_INDEX:long_name = "Stop index action applied on"; HISTORY_STOP_INDEX:_FillValue = 99999;	Stop index the action is applied to. This index corresponds to N_MEASUREMENT or N_CYCLE, depending on the corrected parameter Example : 150
HISTORY_QCTEST	char HISTORY_QCTEST(N_HISTORY, N_HISTORY2, STRING16); HISTORY_QCTEST:long_name = "Documentation of tests performed, tests failed (in hex form)"; HISTORY_QCTEST:conventions = "Write tests performed when ACTION=QCP\$; tests failed when ACTION=QCF\$"; HISTORY_QCTEST:_FillValue = " ";	This field records the tests performed when ACTION is set to QCP\$ (qc performed), the test failed when ACTION is set to QCF\$ (qc failed). The QCTEST codes are describe in reference table 11.  Example : 0A (in hexadecimal form)

The usage of history section is described in §5 "Using the History section of the Argo netCDF Structure".

## 2.4. Meta-data format version 2.2

An Argo meta-data file contains information about an Argo float.

For file naming conventions, see §4.1 .

### 2.4.1. Dimensions and definitions

Name	Definition	Comment
DATE_TIME	DATE_TIME = 14;	This dimension is the length of an ASCII date and time value. Date_time convention is : YYYYMMDDHHMISS YYYY : year MM : month DD : day HH : hour of the day MI : minutes SS : seconds Date and time values are always in universal time coordinates (UTC). Examples : 20010105172834 : January 5 <sup>th</sup> 2001 17:28:34 19971217000000 : December 17 <sup>th</sup> 1997 00:00:00
STRING256 STRING64 STRING32 STRING16 STRING8 STRING4 STRING2	STRING256 = 256; STRING64 = 64; STRING32 = 32; STRING16 = 16; STRING8 = 8; STRING4 = 4; STRING2 = 2;	String dimensions from 2 to 256.
N_CYCLES	N_CYCLES = <int value> ;	Number of different nominal cycles. This value is usually set to 1 : all the cycles are programmed to be the same. However, some floats may perform cycles with different programming. Example : a float is programmed to perform regularly 4 cycles with 400 decibar profiles and the 5 <sup>th</sup> cycle with a 2000 decibar profile. In that case, N_CYCLE is set to 2. N_CYCLES = 2 The first N_CYCLE has a REPETITION_RATE of 4 and the second has a REPETITION_RATE of 1.
N_PARAM	N_PARAM=<int value> ;	Number of parameters measured or calculated for a pressure sample. Examples : (pressure, temperature) : N_PARAM = 2 (pressure, temperature, salinity) : N_PARAM = 3 (pressure, temperature, conductivity, salinity) : N_PARAM = 4

## 2.4.2. General information on the meta-data file

This section contains information about the whole file.

Name	Definition	Comment
DATA_TYPE	char DATA_TYPE(String16); DATA_TYPE:comment = "Data type"; DATA_TYPE:_FillValue = " ";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference table 1. Example : Argo meta-data
FORMAT_VERSION	char FORMAT_VERSION(String4); FORMAT_VERSION:comment = "File format version "; FORMAT_VERSION:_FillValue = " ";	File format version Example : «2.2»
HANDBOOK_VERSION	char HANDBOOK_VERSION(String4); HANDBOOK_VERSION:comment = "Data handbook version"; HANDBOOK_VERSION:_FillValue = " ";	Version number of the data handbook. This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook. Example : «1.0»
DATE_CREATION	char DATE_CREATION(Date_Time); DATE_CREATION:comment = "Date of file creation "; DATE_CREATION:conventions = "YYYYMMDDHHMISS"; DATE_CREATION:_FillValue = " ";	Date and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS Example : 20011229161700 : December 29 <sup>th</sup> 2001 16:17:00
DATE_UPDATE	char DATE_UPDATE(Date_Time); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " ";	Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 <sup>th</sup> 2001 09:05:00

### 2.4.3. Float characteristics

This section contains the main characteristics of the float.

Name	Definition	Comment
PLATFORM_NUMBER	char PLATFORM_NUMBER(String8); PLATFORM_NUMBER:long_name = "Float unique identifier"; PLATFORM_NUMBER:conventions = "WMO float identifier : A9IIIII"; PLATFORM_NUMBER:_FillValue = " ";	WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : 6900045
PTT	char PTT (String256); PTT:long_name = "Transmission identifier (ARGOS, ORBCOMM, etc.)"; PTT:_FillValue = " ";	Transmission identifier of the float. Comma separated list for multi-beacon transmission. Example : 22507 : the float is equipped with one ARGOS beacon. 22598,22768 : the float is equipped with 2 ARGOS beacons.
TRANS_SYSTEM	char TRANS_SYSTEM(String16); TRANS_SYSTEM:long_name = "The telecommunications system used"; TRANS_SYSTEM:_FillValue = " ";	Name of the telecommunication system from reference table 10. Example : ARGOS
TRANS_SYSTEM_ID	char TRANS_SYSTEM_ID(String32); TRANS_SYSTEM_ID:long_name = "The program identifier used by the transmission system"; TRANS_SYSTEM_ID:_FillValue = " ";	Program identifier of the telecommunication subscription. Use N/A when not applicable (eg : Iridium or Orbcmm) Example : 38511 is a program number for all the beacons of an ARGOS customer.
TRANS_FREQUENCY	char TRANS_FREQUENCY(String16); TRANS_FREQUENCY:long_name = "The frequency of transmission from the float"; TRANS_FREQUENCY:units = "hertz"; TRANS_FREQUENCY:_FillValue = " ";	Frequency of transmission from the float. Unit : hertz Example : 1/44
TRANS_REPETITION	float TRANS_REPETITION; TRANS_REPETITION:long_name = "The repetition rate of transmission from the float"; TRANS_REPETITION:units = "second"; TRANS_REPETITION:_FillValue = 99999.f;	Repetition rate of the transmission system. Unit : second Example : 40 for a repetition of messages every 40 seconds.
POSITIONING_SYSTEM	char POSITIONING_SYSTEM(String8); POSITIONING_SYSTEM:long_name = "Positioning system"; POSITIONING_SYSTEM:_FillValue = " ";	Position system from reference table 9. ARGOS or GPS are 2 positioning systems. Example : ARGOS
CLOCK_DRIFT	float CLOCK_DRIFT; CLOCK_DRIFT:long_name = "The rate of drift of the float clock"; CLOCK_DRIFT:units = "decisecond/day"; CLOCK_DRIFT:_FillValue = "99999.f";	Rate of drift of the float internal clock. Unit : decisecond/day Example : 1.57
PLATFORM_MODEL	char PLATFORM_MODEL (String16); PLATFORM_MODEL:long_name = "Model of the float "; PLATFORM_MODEL:_FillValue = " ";	Model of the float. Example : APEX-SBE
PLATFORM_MAKER	char PLATFORM_MAKER (String256); PLATFORM_MAKER:long_name = "The name of the manufacturer "; PLATFORM_MAKER:_FillValue = " ";	Name of the manufacturer. Example : Webb research
INST_REFERENCE	char INST_REFERENCE(String64); INST_REFERENCE:long_name = "Instrument type"; INST_REFERENCE:conventions = "Brand, type, serial number"; INST_REFERENCE:_FillValue = " ";	References of the instrument : brand, type, serial number Example : APEX-SBE 259
WMO_INST_TYPE	char WMO_INST_TYPE(String4); WMO_INST_TYPE:long_name = "Coded instrument type"; WMO_INST_TYPE:conventions = "Argo reference table 8";	Instrument type from WMO code table 1770. A subset of WMO table 1770 is documented in the reference table 8. Example :

	WMO_INST_TYPE:_FillValue = " ";	846 : Webb Research float, Seabird sensor
DIRECTION	char DIRECTION; DIRECTION:long_name = "Direction of the profiles"; DIRECTION:conventions = "A: ascending profiles, B: descending and ascending profiles"; DIRECTION:_FillValue = " ";	Direction of the profiles of the float. A : ascending profiles only B : descending and ascending profiles
PROJECT_NAME	char PROJECT_NAME (STRING64); PROJECT_NAME:long_name = "The program under which the float was deployed"; PROJECT_NAME:_FillValue = " ";	Name of the project which operates the profiling float that performed the profile. Example : GYROSCOPE (EU project for Argo program)
DATA_CENTRE	char DATA_CENTRE (STRING2); DATA_CENTRE:long_name = "Data centre in charge of float real-time processing"; DATA_CENTRE:conventions = "Argo reference table 4"; DATA_CENTRE:_FillValue = " ";	Code of the data centre in charge of the float data management. The data centre codes are described in the reference table 4. Example : ME for MEDS
PI_NAME	char PI_NAME (STRING64); PI_NAME:comment = "Name of the principal investigator"; PI_NAME:_FillValue = " ";	Name of the principal investigator in charge of the profiling float. Example : Yves Desaubies
ANOMALY	char ANOMALY (STRING256); ANOMALY:long_name = "Describe any anomalies or problems the float may have had."; ANOMALY:_FillValue = " ";	This field describes any anomaly or problem the float may have had. Example : "the immersion drift is not stable."

## 2.4.4. Float deployment and mission information

Name	Definition	Comment
LAUNCH_DATE	char LAUNCH_DATE (DATE_TIME); LAUNCH_DATE:long_name = "Date (UTC) of the deployment"; LAUNCH_DATE:conventions = "YYYYMMDDHHMISS"; LAUNCH_DATE:_FillValue = " ";	Date and time (UTC) of launch of the float. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 <sup>th</sup> 2001 03:05:00
LAUNCH_LATITUDE	double LAUNCH_LATITUDE; LAUNCH_LATITUDE:long_name = "Latitude of the float when deployed"; LAUNCH_LATITUDE:units = "degrees_north"; LAUNCH_LATITUDE:_FillValue = 99999.; LAUNCH_LATITUDE:valid_min = -90.; LAUNCH_LATITUDE:valid_max = 90.;	Latitude of the launch. Unit : degree north Example : 44.4991 : 44° 29' 56.76" N
LAUNCH_LONGITUDE	double LAUNCH_LONGITUDE; LAUNCH_LONGITUDE:long_name = "Longitude of the float when deployed"; LAUNCH_LONGITUDE:units = "degrees_east"; LAUNCH_LONGITUDE:_FillValue = 99999.; LAUNCH_LONGITUDE:valid_min = -180.; LAUNCH_LONGITUDE:valid_max = 180.;	Longitude of the launch. Unit : degree east Example : 16.7222 : 16° 43' 19.92" E
LAUNCH_QC	char LAUNCH_QC; LAUNCH_QC:long_name = "Quality on launch date, time and location"; LAUNCH_QC:conventions = "Argo reference table 2"; LAUNCH_QC:_FillValue = " ";	Quality flag on launch date, time and location. The flag scale is described in the reference table 2. Example : 1 : launch location seems correct.
START_DATE	char START_DATE (DATE_TIME); START_DATE:long_name = "Date (UTC) of the first descent of the float."; START_DATE:conventions = "YYYYMMDDHHMISS"; START_DATE:_FillValue = " ";	Date and time (UTC) of the first descent of the float. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 <sup>th</sup> 2001 06 :05 :00
START_DATE_QC	char START_DATE_QC; START_DATE_QC:long_name = "Quality on start date"; START_DATE_QC:conventions = "Argo reference table 2"; START_DATE_QC:_FillValue = " ";	Quality flag on start date. The flag scale is described in the reference table 2. Example : 1 : start date seems correct.
DEPLOY_PLATFORM	char DEPLOY_PLATFORM (STRING32); DEPLOY_PLATFORM:long_name = "Identifier of the deployment platform"; DEPLOY_PLATFORM:_FillValue = " ";	Identifier of the deployment platform. Example : L'ATALANTE
DEPLOY_MISSION	char DEPLOY_MISSION (STRING32); DEPLOY_MISSION:long_name = "Identifier of the mission used to deploy the float"; DEPLOY_MISSION:_FillValue = " ";	Identifier of the mission used to deploy the platform. Example : POMME2
DEPLOY_AVAILABLE_PROFILE_ID	char DEPLOY_AVAILABLE_PROFILE_ID (STRING256); DEPLOY_AVAILABLE_PROFILE_ID:long_name = "Identifier of stations used to verify the first profile"; DEPLOY_AVAILABLE_PROFILE_ID:_FillValue = " ";	Identifier of CTD or XBT stations used to verify the first profile. Example : 58776, 58777
END_MISSION_DATE	char END_MISSION_DATE (DATE_TIME); END_MISSION_DATE:long_name = "Date (UTC) of the end of mission of the float"; END_MISSION_DATE:conventions = "YYYYMMDDHHMISS"; END_MISSION_DATE:_FillValue = " ";	Date (UTC) of the end of mission of the float. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 <sup>th</sup> 2001 03:05:00
END_MISSION_STATUS	char END_MISSION_STATUS; END_MISSION_STATUS:long_name = "Status of the end of mission of the float"; END_MISSION_STATUS:conventions = "T:No more transmission received, R:Retrieved"; END_MISSION_STATUS:_FillValue = " ";	Status of the end of mission of the float.

### 2.4.5. Float sensor information

This section contains information about the sensors of the profiler.

Name	Definition	Comment
SENSOR	char SENSOR(N_PARAM,STRING16); SENSOR:long_name = "List of sensors on the float "; SENSOR:conventions = "Argo reference table 3"; SENSOR:_FillValue = " ";	Parameters measured by sensors of the float. The parameter names are listed in reference table 3. Examples : TEMP, PSAL, CNDC TEMP : temperature in celsius PSAL : practical salinity in psu CNDC : conductivity in mhos/m
SENSOR_MAKER	char SENSOR_MAKER(N_PARAM,STRING256); SENSOR_MAKER:long_name = "The name of the manufacturer "; SENSOR_MAKER:_FillValue = " ";	Name of the manufacturer of the sensor. Example : SEABIRD
SENSOR_MODEL	char SENSOR_MODEL (N_PARAM,STRING256); SENSOR_MODEL:long_name = "Type of sensor"; SENSOR_MODEL:_FillValue = " ";	Model of sensor. Example : SBE41
SENSOR_SERIAL_NO	char SENSOR_SERIAL_NO(N_PARAM,STRING16); SENSOR_SERIAL_NO:long_name = "The serial number of the sensor"; SENSOR_SERIAL_NO:_FillValue = " ";	Serial number of the sensor. Example : 2646 036 073
SENSOR_UNITS	char SENSOR_UNITS(N_PARAM, STRING16); SENSOR_UNITS:long_name = "The units of accuracy and resolution of the sensor"; SENSOR_UNITS:_FillValue = " ";	Units of accuracy of the sensor. Example : psu
SENSOR_ACCURACY	float SENSOR_ACCURACY(N_PARAM); SENSOR_ACCURACY:long_name = "The accuracy of the sensor"; SENSOR_ACCURACY:_FillValue = 99999.f;	Accuracy of the sensor. Example : 0.005
SENSOR_RESOLUTION	float SENSOR_RESOLUTION(N_PARAM); SENSOR_RESOLUTION:long_name = "The resolution of the sensor"; SENSOR_RESOLUTION:_FillValue = 99999.f;	Resolution of the sensor. Example : 0.001

### 2.4.6. Float calibration information

This section contains information about the calibration of the profiler. The calibration described in this section is an instrumental calibration. The delayed mode calibration, based on a data analysis is described in the profile format.

Name	Definition	Comment
PARAMETER	char PARAMETER(N_PARAM,STRING16); PARAMETER:long_name = "List of parameters with calibration information"; PARAMETER:conventions = "Argo reference table 3"; PARAMETER:_FillValue = " ";	Parameters measured on this float. The parameter names are listed in reference table 3. Examples : TEMP, PSAL, CNDC TEMP : temperature in celsius PSAL : practical salinity in psu CNDC : conductivity in mhos/m
PREDEPLOYMENT_CALIB_EQUATION	char PREDEPLOYMENT_CALIB_EQUATION(N_PARAM,STRING256); PREDEPLOYMENT_CALIB_EQUATION:long_name = "Calibration equation for this parameter"; PREDEPLOYMENT_CALIB_EQUATION:_FillValue = " ";	Calibration equation for this parameter. Example : $T_c = a_1 * T + a_0$
PREDEPLOYMENT_CALIB_COEFFICIENT	char PREDEPLOYMENT_CALIB_COEFFICIENT(N_PARAM,STRING256); PREDEPLOYMENT_CALIB_COEFFICIENT:long_name = "Calibration coefficients for this equation";	Calibration coefficients for this equation. Example : $a_1=0.99997$ , $a_0=0.0021$

	<code>PREDEPLOYMENT_CALIB_COEFFICIENT:_FillValue = " ";</code>	
<code>PREDEPLOYMENT_CALIB_COMMENT</code>	<code>char PREDEPLOYMENT_CALIB_COMMENT(N_PARAM,STRING2 56); PREDEPLOYMENT_CALIB_COMMENT:long_name = "Comment applying to this parameter calibration"; PREDEPLOYMENT_CALIB_COMMENT:_FillValue = " ";</code>	Comments applying to this parameter calibration. Example : The sensor is not stable

## 2.4.7. Float cycle information

This section contains information on the cycle characteristics of the float. The values included in this section are programmed or estimated. They are not measured.

Each value has a N\_CYCLES dimension. Each N\_CYCLE describes a cycle configuration.

Name	Definition	Comment
REPETITION_RATE	int REPETITION_RATE(N_CYCLES); REPETITION_RATE:long_name = "The number of times this cycle repeats"; REPETITION_RATE:units = "number"; REPETITION_RATE:_FillValue = 99999;	Number of times this cycle repeats. Usually, REPETITION_RATE and N_CYCLE are set to 1 : all the cycles are programmed to be the same. However, some floats may perform cycles with different programming. Example : a float is programmed to perform regularly 4 cycles with 400 decibar profiles and the 5 <sup>th</sup> cycle with a 2000 decibar profile. In that case, N_CYCLE is set to 2. The first N_CYCLE has a REPETITION_RATE of 4 and the second has a REPETITION_RATE of 1.
CYCLE_TIME	float CYCLE_TIME(N_CYCLES); CYCLE_TIME:long_name = "The total time of a cycle : descent + parking + ascent + surface"; CYCLE_TIME:units = "decimal hour"; CYCLE_TIME:_FillValue = 99999.f;	Total time of a cycle. This time includes the descending time, the parking time, the ascending time and the surface time. Unit : decimal hour Example : 240 hours for a ten day cycle.
PARKING_TIME	float PARKING_TIME(N_CYCLES); PARKING_TIME:long_name = "The time spent at the parking pressure"; PARKING_TIME:units = "decimal hour"; PARKING_TIME:_FillValue = 99999.f;	Time spent at the parking pressure. This time does not include the descending and ascending times. Unit : decimal day Example : 222 for 9 days and 6 hours at parking pressure.
DESCENDING_PROFILING_TIME	float DESCENDING_PROFILING_TIME(N_CYCLES); DESCENDING_PROFILING_TIME:long_name = "The time spent sampling the descending profile"; DESCENDING_PROFILING_TIME:units = "decimal hour"; DESCENDING_PROFILING_TIME:_FillValue = 99999.f;	Time spent in descent. Unit : decimal hour Example : 8.5 for 8 hours 30 minutes of descending
ASCENDING_PROFILING_TIME	float ASCENDING_PROFILING_TIME(N_CYCLES); ASCENDING_PROFILING_TIME:long_name = "The time spent sampling the ascending profile"; ASCENDING_PROFILING_TIME:units = "decimal hour"; ASCENDING_PROFILING_TIME:_FillValue = 99999.f;	Time spent in ascent. Unit : decimal hour Example : 7.5 for 7 hours 30 minutes of descending
SURFACE_TIME	float SURFACE_TIME(N_CYCLES); SURFACE_TIME:long_name = "The time spent at the surface."; SURFACE_TIME:units = "decimal hour"; SURFACE_TIME:_FillValue = 99999.f;	Time spent on the surface (surface drift). Unit : decimal hour Example : 10 for a 10 hours surface drift.
PARKING_PRESSURE	float PARKING_PRESSURE(N_CYCLES); PARKING_PRESSURE:long_name = "The pressure of subsurface drifts"; PARKING_PRESSURE:units = "decibar"; PARKING_PRESSURE:_FillValue = 99999.f;	Pressure of the subsurface drift. Unit : decibar Example : 1500.0 for a subsurface drift at 1500.0 decibars.
DEEPEST_PRESSURE	float DEEPEST_PRESSURE(N_CYCLES); DEEPEST_PRESSURE:long_name = "The deepest pressure sampled in the ascending profile"; DEEPEST_PRESSURE:units = "decibar"; DEEPEST_PRESSURE:_FillValue = 99999.f;	Deepest pressure sampled in the ascending profile. Unit : decibar Example : 2000.0 for an ascending profile starting at 2000.0 decibar.
DEEPEST_PRESSURE_DESCENDING	float DEEPEST_PRESSURE_DESCENDING(N_CYCLES); DEEPEST_PRESSURE_DESCENDING:long_name = "The deepest pressure sampled in the descending profile"; DEEPEST_PRESSURE_DESCENDING:units = "decibar"; DEEPEST_PRESSURE_DESCENDING:_FillValue = 99999.f;	Deepest pressure sampled in the descending profile. Unit : decibar Example : 500.0 for a descending profile ending at 500.0 decibar.

### 2.4.8. Highly desirable meta-data parameters

A highly desirable meta-data parameter should be correctly filled according to the following table.

Highly desirable meta-data	mandatory format	example
DATA_TYPE	"Argo meta-data";	DATA_TYPE = "Argo meta-data";
FORMAT_VERSION	"2.2 ";	FORMAT_VERSION = "2.2 ";
HANDBOOK_VERSION	"1.2 ";	HANDBOOK_VERSION = "1.2 ";
DATE_CREATION	YYYYMMDDHHMISS	DATE_CREATION = "20040210124422";
DATE_UPDATE	YYYYMMDDHHMISS	DATE_UPDATE = "20040210124422";
PLATFORM_NUMBER	XXXXX or XXXXXXX	PLATFORM_NUMBER = "5900077 ";
PTT	not empty	PTT = "23978 ";
TRANS_SYSTEM	see reference table 10	TRANS_SYSTEM = "ARGOS ";
TRANS_SYSTEM_ID	not empty	TRANS_SYSTEM_ID = "14281";
POSITIONING_SYSTEM	see reference table 9	POSITIONING_SYSTEM = "ARGOS";
PLATFORM_MODEL	not empty	PLATFORM_MODEL = "SOLO";
DIRECTION	"A" or "D"	DIRECTION = "A";
DATA_CENTRE	see reference table 4	DATA_CENTRE = "AO ;
LAUNCH_DATE	YYYYMMDDHHMISS	LAUNCH_DATE = "20010717000100";
LAUNCH_LATITUDE	not empty, -90 <= real <= 90	LAUNCH_LATITUDE = -7.91400003433228;
LAUNCH_LONGITUDE	not empty, -180 <= real <= 180	LAUNCH_LONGITUDE = -179.828338623047;
LAUNCH_QC	see reference table 2	LAUNCH_QC = "1";
START_DATE	YYYYMMDDHHMISS	START_DATE = "20010702000000";
START_DATE_QC	see reference table 2	START_DATE_QC = "2";
PARAMETER	see reference table 3	PARAMETER = "PRES", "TEMP", "PSAL";
CYCLE_TIME	not empty	CYCLE_TIME = 10;
DEEPEST_PRESSURE	not empty	DEEPEST_PRESSURE = 1092;
PARKING_PRESSURE	not empty	PARKING_PRESSURE = 1000;

## 2.5. Meta-data format version 2.3

The format version 2.3 of Argo metadata will replace version 2.2 gradually. During the transition period, both formats will be valid. However, when a Data Assembly Center (DAC) produces metadata files with the new 2.3 format, all its metadata files must be provided in version 2.3

An Argo meta-data file contains information about an Argo float.

For file naming conventions, see §4.1 .

### 2.5.1. Dimensions and definitions

Name	Definition	Comment
DATE_TIME	DATE_TIME = 14;	This dimension is the length of an ASCII date and time value. Date_time convention is : YYYYMMDDHHMISS YYYY : year MM : month DD : day HH : hour of the day MI : minutes SS : seconds Date and time values are always in universal time coordinates (UTC). Examples : 20010105172834 : January 5 <sup>th</sup> 2001 17:28:34 19971217000000 : December 17 <sup>th</sup> 1997 00:00:00
STRING256 STRING64 STRING32 STRING16 STRING8 STRING4 STRING2	STRING256 = 256; STRING64 = 64; STRING32 = 32; STRING16 = 16; STRING8 = 8; STRING4 = 4; STRING2 = 2;	String dimensions from 2 to 256.
N_CYCLES	N_CYCLES = <int value> ;	Number of different nominal cycles. This value is usually set to 1 : all the cycles are programmed to be the same. However, some floats may perform cycles with different programming. Example : a float is programmed to perform regularly 4 cycles with 400 decibar profiles and the 5 <sup>th</sup> cycle with a 2000 decibar profile. In that case, N_CYCLE is set to 2. N_CYCLES = 2 The first N_CYCLE has a REPETITION_RATE of 4 and the second has a REPETITION_RATE of 1.
N_PARAM	N_PARAM=<int value> ;	Number of parameters measured or calculated for a pressure sample. Examples : (pressure, temperature) : N_PARAM = 2 (pressure, temperature, salinity) : N_PARAM = 3 (pressure, temperature, conductivity, salinity) : N_PARAM = 4

## 2.5.2. General information on the meta-data file

This section contains information about the whole file.

Name	Definition	Comment
DATA_TYPE	char DATA_TYPE(String16); DATA_TYPE:comment = "Data type"; DATA_TYPE:_FillValue = " ";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference table 1. Example : Argo meta-data
FORMAT_VERSION	char FORMAT_VERSION(String4); FORMAT_VERSION:comment = "File format version "; FORMAT_VERSION:_FillValue = " ";	File format version Example : «2.3»
HANDBOOK_VERSION	char HANDBOOK_VERSION(String4); HANDBOOK_VERSION:comment = "Data handbook version"; HANDBOOK_VERSION:_FillValue = " ";	Version number of the data handbook. This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook. Example : «1.0»
DATE_CREATION	char DATE_CREATION(Date_Time); DATE_CREATION:comment = "Date of file creation "; DATE_CREATION:conventions = "YYYYMMDDHHMISS"; DATE_CREATION:_FillValue = " ";	Date and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS Example : 20011229161700 : December 29 <sup>th</sup> 2001 16:17:00
DATE_UPDATE	char DATE_UPDATE(Date_Time); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " ";	Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 <sup>th</sup> 2001 09:05:00

### 2.5.3. Float characteristics

This section contains the main characteristics of the float.

Name	Definition	Comment
PLATFORM_NUMBER	<pre>char PLATFORM_NUMBER(String8); PLATFORM_NUMBER:long_name = "Float unique identifier"; PLATFORM_NUMBER:conventions = "WMO float identifier : A911111"; PLATFORM_NUMBER:_FillValue = " ";</pre>	<p>WMO float identifier. WMO is the World Meteorological Organization.</p> <p>This platform number is unique. Example : 6900045</p>

## 2.5.4. Configuration parameters

This section contains the configuration parameters for a float.

For each configuration parameter, the name of the parameter and the value of the parameter are recorded.

The parameter name and its value are recorded as strings of 128 characters.

Most parameter names are standardized and available in reference table 14b.

The phase is used to record informations that change from cycles to cycles.

The configuration phase repetition allows to repeat a cycle configuration a certain time.

The phase 0 contains the usefull meta-data that should not change during the float lifetime.

Name	Definition	Comment
<b>CONFIGURATION_PARAMETER_NAME</b>	char CONFIGURATON_PARAMETER_NAME(N_CONF_PARAM, STRING128) CONFIGURATON_PARAMETER_NAME:long_name="Name of configuration parameter"; CONFIGURATON_PARAMETER_NAME:_FillValue = " ";	Name of the configuration parameter. Example : "PRES_ParkPressure_dBAR" See reference table 14 for standard configuraton parameter names.
<b>CONFIGURATION_PARAMETER_VALUE</b>	char CONFIGURATON_PARAMETER_VALUE(N_CONF_PARAM, STRING128) CONFIGURATON_PARAMETER_VALUE:long_name="Value of configuraton parameter"; CONFIGURATON_PARAMETER_VALUE:_FillValue = " ";	Value of the configuration parameter. Example : "1500"
<b>CONFIGURATION_PHASE_NUMBER</b>	int PHASE_NUMBER(N_CONF_PARAM); PHASE_NUMBER:long_name = "Float cycle number"; PHASE_NUMBER:conventions = "0..N, 0 : launch phase (if exists), 1 : first complete phase"; PHASE_NUMBER:_FillValue = 99999;	Phase number of the configuration parameter. Example : 1  See note on floats with multiple configurations.
<b>CONFIGURATION_PHASE_REPETITION</b>	int CONFIGURATION_PHASE_REPETITION (N_CONF_PARAM); CONFIGURATION_PHASE_REPETITION:long_name = "The number of times this cycle repeats"; CONFIGURATION_PHASE_REPETITION:units = "number"; CONFIGURATION_PHASE_REPETITION:_FillValue = 99999;	Number of cycle repetition for this phase. Example : 1  See note on floats with multiple configurations.

### Note on floats with multiple configurations

Usually, an Argo float configuration is valid for the whole life of the float. Each cycle is repeated with the same behaviour (basic configuration).

However, some floats may be configured to change their behaviour from cycle to cycle (changing configuration).

For the basic configuration, CONFIGURATION\_PHASE\_NUMBER is set to 1 and CONFIGURATION\_PHASE\_REPETITION is set to 99999: all the cycles are programmed to be the same.

For changing configuration, CONFIGURATION\_PHASE\_NUMBER is used to describe a configuration. CONFIGURATION\_PHASE\_REPETITION is used for the number of consecutive cycles configured in a given phase.

Example 1: a float is programmed to perform regularly 4 cycles with 400 decibar profiles and the 5th cycle with a 2000 decibar profile.

```
CONFIGURATION_PARAMETER_NAME = "PRES_ParkPressure_dBAR"
CONFIGURATION_PARAMETER_VALUE = "400"
PHASE_NUMBER = 1
REPETITION_RATE = 4
```

```
CONFIGURATION_PARAMETER_NAME = "PRES_ParkPressure_dBAR"
CONFIGURATION_PARAMETER_VALUE = "2000"
CONFIGURATION_PHASE_NUMBER = 2
CONFIGURATION_PHASE_REPETITION = 1
```

Example 2: a float is configured with telemetry. The float operator randomly changes the park pressure to : 200, 400, 500, 1500 decibars

```
CONFIGURATION_PARAMETER_NAME = "PRES_ParkPressure_dBAR"
CONFIGURATION_PARAMETER_VALUE = "200"
CONFIGURATION_PHASE_NUMBER = 1
CONFIGURATION_PHASE_REPETITION = 1
```

```
CONFIGURATION_PARAMETER_NAME = "PRES_ParkPressure_dBAR"
CONFIGURATION_PARAMETER_VALUE = "400"
CONFIGURATION_PHASE_NUMBER = 2
CONFIGURATION_PHASE_REPETITION = 1
```

```
CONFIGURATION_PARAMETER_NAME = "PRES_ParkPressure_dBAR"
CONFIGURATION_PARAMETER_VALUE = "500"
CONFIGURATION_PHASE_NUMBER = 3
CONFIGURATION_PHASE_REPETITION = 1
```

```
CONFIGURATION_PARAMETER_NAME = "PRES_ParkPressure_dBAR"
CONFIGURATION_PARAMETER_VALUE = "1500"
CONFIGURATION_PHASE_NUMBER = 4
CONFIGURATION_PHASE_REPETITION = 99999
```

The repetition of CONFIGURATION\_PHASE\_NUMBER 4 is 99999, the fill value. It means that the float will keep this configuration with a parking pressure equals to 1500 decibars.

### 2.5.5. Float sensor information

This section contains information about the sensors of the profiler.

Name	Definition	Comment
SENSOR	char SENSOR(N_PARAM,STRING16); SENSOR:long_name = "List of sensors on the float "; SENSOR:conventions = "Argo reference table 3"; SENSOR:_FillValue = " ";	Parameters measured by sensors of the float. The parameter names are listed in reference table 3. Examples : TEMP, PSAL, CNDC TEMP : temperature in celsius PSAL : practical salinity in psu CNDC : conductivity in mhos/m
SENSOR_MAKER	char SENSOR_MAKER(N_PARAM,STRING256); SENSOR_MAKER:long_name = "The name of the manufacturer "; SENSOR_MAKER:_FillValue = " ";	Name of the manufacturer of the sensor. Example : SEABIRD
SENSOR_MODEL	char SENSOR_MODEL (N_PARAM,STRING256); SENSOR_MODEL:long_name = "Type of sensor"; SENSOR_MODEL:_FillValue = " ";	Model of sensor. Example : SBE41
SENSOR_SERIAL_NO	char SENSOR_SERIAL_NO(N_PARAM,STRING16); SENSOR_SERIAL_NO:long_name = "The serial number of the sensor"; SENSOR_SERIAL_NO:_FillValue = " ";	Serial number of the sensor. Example : 2646 036 073
SENSOR_UNITS	char SENSOR_UNITS(N_PARAM, STRING16); SENSOR_UNITS:long_name = "The units of accuracy and resolution of the sensor"; SENSOR_UNITS:_FillValue = " ";	Units of accuracy of the sensor. Example : psu
SENSOR_ACCURACY	float SENSOR_ACCURACY(N_PARAM); SENSOR_ACCURACY:long_name = "The accuracy of the sensor"; SENSOR_ACCURACY:_FillValue = 99999.f;	Accuracy of the sensor. Example : 0.005
SENSOR_RESOLUTION	float SENSOR_RESOLUTION(N_PARAM); SENSOR_RESOLUTION:long_name = "The resolution of the sensor"; SENSOR_RESOLUTION:_FillValue = 99999.f;	Resolution of the sensor. Example : 0.001

### 2.5.6. Float calibration information

This section contains information about the calibration of the profiler. The calibration described in this section is an instrumental calibration. The delayed mode calibration, based on a data analysis is described in the profile format.

Name	Definition	Comment
PARAMETER	char PARAMETER(N_PARAM,STRING16); PARAMETER:long_name = "List of parameters with calibration information"; PARAMETER:conventions = "Argo reference table 3"; PARAMETER:_FillValue = " ";	Parameters measured on this float. The parameter names are listed in reference table 3. Examples : TEMP, PSAL, CNDC TEMP : temperature in celsius PSAL : practical salinity in psu CNDC : conductivity in mhos/m
PREDEPLOYMENT_CALIB_EQUATION	char PREDEPLOYMENT_CALIB_EQUATION(N_PARAM,STRING256); PREDEPLOYMENT_CALIB_EQUATION:long_name = "Calibration equation for this parameter"; PREDEPLOYMENT_CALIB_EQUATION:_FillValue = " ";	Calibration equation for this parameter. Example : $T_c = a_1 * T + a_0$
PREDEPLOYMENT_CALIB_COEFFICIENT	char PREDEPLOYMENT_CALIB_COEFFICIENT(N_PARAM,STRING256); PREDEPLOYMENT_CALIB_COEFFICIENT:long_name = "Calibration coefficients for this equation";	Calibration coefficients for this equation. Example : $a_1=0.99997, a_0=0.0021$

	<code>PREDEPLOYMENT_CALIB_COEFFICIENT:_FillValue = " ";</code>	
PREDEPLOYMENT_CALIB_COMMENT	<code>char PREDEPLOYMENT_CALIB_COMMENT(N_PARAM,STRING2 56); PREDEPLOYMENT_CALIB_COMMENT:long_name = "Comment applying to this parameter calibration"; PREDEPLOYMENT_CALIB_COMMENT:_FillValue = " ";</code>	Comments applying to this parameter calibration. Example : The sensor is not stable

## 2.5.7. Float cycle information

This section contains information on the cycle characteristics of the float. The values included in this section are programmed or estimated. They are not measured.

Each value has a N\_CYCLES dimension. Each N\_CYCLE describes a cycle configuration.

Name	Definition	Comment
REPETITION_RATE	int REPETITION_RATE(N_CYCLES); REPETITION_RATE:long_name = "The number of times this cycle repeats"; REPETITION_RATE:units = "number"; REPETITION_RATE:_FillValue = 99999;	Number of times this cycle repeats. Usually, REPETITION_RATE and N_CYCLE are set to 1 : all the cycles are programmed to be the same. However, some floats may perform cycles with different programming. Example : a float is programmed to perform regularly 4 cycles with 400 decibar profiles and the 5 <sup>th</sup> cycle with a 2000 decibar profile. In that case, N_CYCLE is set to 2. The first N_CYCLE has a REPETITION_RATE of 4 and the second has a REPETITION_RATE of 1.
CYCLE_TIME	float CYCLE_TIME(N_CYCLES); CYCLE_TIME:long_name = "The total time of a cycle : descent + parking + ascent + surface"; CYCLE_TIME:units = "decimal hour"; CYCLE_TIME:_FillValue = 99999.f;	Total time of a cycle. This time includes the descending time, the parking time, the ascending time and the surface time. Unit : decimal hour Example : 240 hours for a ten day cycle.
PARKING_TIME	float PARKING_TIME(N_CYCLES); PARKING_TIME:long_name = "The time spent at the parking pressure"; PARKING_TIME:units = "decimal hour"; PARKING_TIME:_FillValue = 99999.f;	Time spent at the parking pressure. This time does not include the descending and ascending times. Unit : decimal hour Example : 222 for 9 days and 6 hours at parking pressure.
DESCENDING_PROFILING_TIME	float DESCENDING_PROFILING_TIME(N_CYCLES); DESCENDING_PROFILING_TIME:long_name = "The time spent sampling the descending profile"; DESCENDING_PROFILING_TIME:units = "decimal hour"; DESCENDING_PROFILING_TIME:_FillValue = 99999.f;	Time spent in descent. Unit : decimal hour Example : 8.5 for 8 hours 30 minutes of descending
ASCENDING_PROFILING_TIME	float ASCENDING_PROFILING_TIME(N_CYCLES); ASCENDING_PROFILING_TIME:long_name = "The time spent sampling the ascending profile"; ASCENDING_PROFILING_TIME:units = "decimal hour"; ASCENDING_PROFILING_TIME:_FillValue = 99999.f;	Time spent in ascent. Unit : decimal hour Example : 7.5 for 7 hours 30 minutes of descending
SURFACE_TIME	float SURFACE_TIME(N_CYCLES); SURFACE_TIME:long_name = "The time spent at the surface."; SURFACE_TIME:units = "decimal hour"; SURFACE_TIME:_FillValue = 99999.f;	Time spent on the surface (surface drift). Unit : decimal hour Example : 10 for a 10 hours surface drift.
PARKING_PRESSURE	float PARKING_PRESSURE(N_CYCLES); PARKING_PRESSURE:long_name = "The pressure of subsurface drifts"; PARKING_PRESSURE:units = "decibar"; PARKING_PRESSURE:_FillValue = 99999.f;	Pressure of the subsurface drift. Unit : decibar Example : 1500.0 for a subsurface drift at 1500.0 decibars.
DEEPEST_PRESSURE	float DEEPEST_PRESSURE(N_CYCLES); DEEPEST_PRESSURE:long_name = "The deepest pressure sampled in the ascending profile"; DEEPEST_PRESSURE:units = "decibar"; DEEPEST_PRESSURE:_FillValue = 99999.f;	Deepest pressure sampled in the ascending profile. Unit : decibar Example : 2000.0 for an ascending profile starting at 2000.0 decibar.
DEEPEST_PRESSURE_DESCENDING	float DEEPEST_PRESSURE_DESCENDING(N_CYCLES); DEEPEST_PRESSURE_DESCENDING:long_name = "The deepest pressure sampled in the descending profile"; DEEPEST_PRESSURE_DESCENDING:units = "decibar"; DEEPEST_PRESSURE_DESCENDING:_FillValue = 99999.f;	Deepest pressure sampled in the descending profile. Unit : decibar Example : 500.0 for a descending profile ending at 500.0 decibar.
FLOAT_CYCLE_COMMENT	char FLOAT_CYCLE_COMMENT(N_CYCLES,STRING256); FLOAT_CYCLE_COMMENT:long_name = "Comment	Comments applying to this float cycle. Example :

	applying to this float cycle" FLOAT_CYCLE_COMMENT: FillValue = " ";	"Seasonal cycle duration : 72 hours between January and March, 120 hours between April and December, else 240 hours"
--	--	---

Claudia : See float cycle description about adding the UP\_TIME and DOWN\_TIME.

### 2.5.8. Highly desirable meta-data parameters

A highly desirable meta-data parameter should be correctly filled according to the following table.

Highly desirable meta-data	mandatory format	example
DATA_TYPE	"Argo meta-data";	DATA_TYPE = "Argo meta-data";
FORMAT_VERSION	"2.2 ";	FORMAT_VERSION = "2.2 ";
HANDBOOK_VERSION	"1.2 ";	HANDBOOK_VERSION = "1.2 ";
DATE_CREATION	YYYYMMDDHHMISS	DATE_CREATION = "20040210124422";
DATE_UPDATE	YYYYMMDDHHMISS	DATE_UPDATE = "20040210124422";
PLATFORM_NUMBER	XXXXX or XXXXXXX	PLATFORM_NUMBER = "5900077 ";
PTT	not empty	PTT = "23978 ";
TRANS_SYSTEM	see reference table 10	TRANS_SYSTEM = "ARGOS ";
TRANS_SYSTEM_ID	not empty	TRANS_SYSTEM_ID = "14281";
POSITIONING_SYSTEM	see reference table 9	POSITIONING_SYSTEM = "ARGOS";
PLATFORM_MODEL	not empty	PLATFORM_MODEL = "SOLO";
DIRECTION	"A" or "D"	DIRECTION = "A";
DATA_CENTRE	see reference table 4	DATA_CENTRE = "AO ";
LAUNCH_DATE	YYYYMMDDHHMISS	LAUNCH_DATE = "20010717000100";
LAUNCH_LATITUDE	not empty, -90 <= real <= 90	LAUNCH_LATITUDE = -7.91400003433228;
LAUNCH_LONGITUDE	not empty, -180 <= real <= 180	LAUNCH_LONGITUDE = -179.828338623047;
LAUNCH_QC	see reference table 2	LAUNCH_QC = "1";
START_DATE	YYYYMMDDHHMISS	START_DATE = "20010702000000";
START_DATE_QC	see reference table 2	START_DATE_QC = "2";
PARAMETER	see reference table 3	PARAMETER ="PRES","TEMP","PSAL";
CYCLE_TIME	not empty	CYCLE_TIME = 10;
DEEPEST_PRESSURE	not empty	DEEPEST_PRESSURE = 1092;
PARKING_PRESSURE	not empty	PARKING_PRESSURE = 1000;

## 2.6. Technical information format version 2.2

An Argo technical file contains technical information from an Argo float. This information is registered for each cycle performed by the float.

The number and the type of technical information is different from one float model to another. To be flexible, for each cycle, the name of the parameters and their values are recorded. The name of the parameters recorded may therefore change from one model of float to another.

For file naming conventions, see §4.1 .

### 2.6.1. Dimensions and definitions

Name	Definition	Comment
DATE_TIME	DATE_TIME = 14;	This dimension is the length of an ASCII date and time value. Date and time values are always in universal time coordinates (UTC). Date_time convention is : YYYYMMDDHHMISS <ul style="list-style-type: none"> <li>• YYYY : year</li> <li>• MM : month</li> <li>• DD : day</li> <li>• HH : hour of the day</li> <li>• MI : minutes</li> <li>• SS : seconds</li> </ul> Examples : 20010105172834 : January 5 <sup>th</sup> 2001 17:28:34 19971217000000 : December 17 <sup>th</sup> 1997 00:00:00
STRING256 STRING128, STRING64, STRING32 STRING16, STRING8 STRING4 STRING2	STRING256 = 256; STRING128 = 128; STRING64 = 64; STRING32 = 32; STRING16 = 16; STRING8 = 8; STRING4 = 4; STRING2 = 2;	String dimensions from 2 to 256.
N_TECH_PARAM	N_TECH_PARAM = <int value> ;	Number of technical parameters. Example : N_TECH_PARAM=25 Twenty five different parameters are recorded for each cycle.
N_CYCLE	N_CYCLE = UNLIMITED;	Number of cycles performed by the float.

## 2.6.2. General information on the technical data file

This section contains information about the technical data file itself.

Name	Definition	Comment
PLATFORM_NUMBER	char PLATFORM_NUMBER(String8); PLATFORM_NUMBER:long_name = "Float unique identifier"; PLATFORM_NUMBER:conventions = "WMO float identifier : A911111"; PLATFORM_NUMBER:_FillValue = " ";	WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : 6900045
DATA_TYPE	char DATA_TYPE(String32); DATA_TYPE:comment = "Data type"; DATA_TYPE:_FillValue = " ";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference table 1. Example : "Argo technical data"
FORMAT_VERSION	char FORMAT_VERSION(String4); FORMAT_VERSION:comment = "File format version "; FORMAT_VERSION:_FillValue = " ";	File format version Example : «2.2»
HANDBOOK_VERSION	char HANDBOOK_VERSION(String4); HANDBOOK_VERSION:comment = "Data handbook version"; HANDBOOK_VERSION:_FillValue = " ";	Version number of the data handbook. This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook. Example : «1.0»
DATA_CENTRE	char DATA_CENTRE(String2); DATA_CENTRE:long_name = "Data centre in charge of float data processing"; DATA_CENTRE:conventions = "Argo reference table 4"; DATA_CENTRE:_FillValue = " ";	Code of the data centre in charge of the float data management. The data centre codes are described in the reference table 4. Example : ME for MEDS
DATE_CREATION	char DATE_CREATION(Date_Time); DATE_CREATION:comment = "Date of file creation "; DATE_CREATION:conventions = "YYYYMMDDHHMISS"; DATE_CREATION:_FillValue = " ";	Date and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS Example : 20011229161700 : December 29 <sup>th</sup> 2001 16 :17 :00
DATE_UPDATE	char DATE_UPDATE(Date_Time); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " ";	Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 <sup>th</sup> 2001 09 :05 :00

### 2.6.3. Technical data

This section contains a set of technical data for each profile.

There are N\_TECH\_PARAM (eg : 25) technical parameters recorded for each cycle.

For each cycle, for each technical parameter, the name of the parameter and the value of the parameter are recorded.

The parameter name and its value are recorded as strings of 32 characters.

The naming convention for TECHNICAL\_PARAMETER\_NAME is :

- Upper case letters only ;
- No space in the name (use underscore "\_").

Name	Definition	Comment
TECHNICAL_PARAMETER_NAME	char TECHNICAL_PARAMETER_NAME( N_CYCLE, N_TECH_PARAM, STRING32) TECHNICAL_PARAMETER_NAME: long_name="Name of technical parameters for this cycle"; TECHNICAL_PARAMETER_NAME: _FillValue = " ";	Name of the technical parameter. Example : "BATTERY_VOLTAGE" See reference table 14 for standard technical parameter names.
TECHNICAL_PARAMETER_VALUE	char TECHNICAL_PARAMETER_VALUE( N_CYCLE, N_TECH_PARAM, STRING32) TECHNICAL_PARAMETER_VALUE: long_name="Value of technical parameters for this cycle"; TECHNICAL_PARAMETER_VALUE: _FillValue = " ";	Value of the technical parameter. Example : "11.5"

Most parameter names are not standardized. However, certain usual parameter names are available in reference table 14.

## 2.7. Technical information format version 2.3

The format version 2.3 of Argo technical data will replace version 2.2 gradually. During the transition period, both formats will be valid. However, when a Data Assembly Center (DAC) produces technical files with the new 2.3 format, all its technical files must be provided in version 2.3.

An Argo technical file contains technical information from an Argo float. This information is registered for each cycle performed by the float.

The number and the type of technical information is different from one float model to another. To be flexible, for each cycle, the name of the parameters and their values are recorded. The name of the parameters recorded may therefore change from one model of float to another.

For file naming conventions, see §4.1.

### 2.7.1. Dimensions and definitions

Name	Definition	Comment
DATE_TIME	DATE_TIME = 14;	This dimension is the length of an ASCII date and time value. Date and time values are always in universal time coordinates (UTC). Date_time convention is : YYYYMMDDHHMISS <ul style="list-style-type: none"> <li>• YYYY : year</li> <li>• MM : month</li> <li>• DD : day</li> <li>• HH : hour of the day</li> <li>• MI : minutes</li> <li>• SS : seconds</li> </ul> Examples : 20010105172834 : January 5 <sup>th</sup> 2001 17:28:34 19971217000000 : December 17 <sup>th</sup> 1997 00:00:00
STRING128, STRING32 STRING8 STRING4 STRING2	STRING128 = 128; STRING32 = 32; STRING8 = 8; STRING4 = 4; STRING2 = 2;	String dimensions from 2 to 256.
N_TECH_PARAM	N_TECH_PARAM = UNLIMITED;	Number of technical parameters.

## 2.7.2. General information on the technical data file

This section contains information about the technical data file itself.

Name	Definition	Comment
PLATFORM_NUMBER	char PLATFORM_NUMBER(String8); PLATFORM_NUMBER:long_name = "Float unique identifier"; PLATFORM_NUMBER:conventions = "WMO float identifier : A911111"; PLATFORM_NUMBER:_FillValue = " ";	WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : 6900045
DATA_TYPE	char DATA_TYPE(String32); DATA_TYPE:comment = "Data type"; DATA_TYPE:_FillValue = " ";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference table 1. Example : "Argo technical data"
FORMAT_VERSION	char FORMAT_VERSION(String4); FORMAT_VERSION:comment = "File format version "; FORMAT_VERSION:_FillValue = " ";	File format version Example : «2.3»
HANDBOOK_VERSION	char HANDBOOK_VERSION(String4); HANDBOOK_VERSION:comment = "Data handbook version"; HANDBOOK_VERSION:_FillValue = " ";	Version number of the data handbook. This field indicates that the data contained in this file are managed according to the policy described in the Argo data management handbook. Example : «1.0»
DATA_CENTRE	char DATA_CENTRE(String2); DATA_CENTRE:long_name = "Data centre in charge of float data processing"; DATA_CENTRE:conventions = "Argo reference table 4"; DATA_CENTRE:_FillValue = " ";	Code of the data centre in charge of the float data management. The data centre codes are described in the reference table 4. Example : ME for MEDS
DATE_CREATION	char DATE_CREATION(Date_Time); DATE_CREATION:comment = "Date of file creation "; DATE_CREATION:conventions = "YYYYMMDDHHMISS"; DATE_CREATION:_FillValue = " ";	Date and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS Example : 20011229161700 : December 29 <sup>th</sup> 2001 16 :17 :00
DATA_UPDATE	char DATE_UPDATE(Date_Time); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " ";	Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 <sup>th</sup> 2001 09 :05 :00

### 2.7.3. Technical data

This section contains a set of technical data for each profile.

For each cycle, for each technical parameter, the name of the parameter and the value of the parameter are recorded.

The parameter name and its value are recorded as strings of 128 characters.

Most parameter names are standardized and available in reference table 14.

Name	Definition	Comment
TECHNICAL_PARAMETER_NAME	char TECHNICAL_PARAMETER_NAME( N_TECH_PARAM, STRING128) TECHNICAL_PARAMETER_NAME: long_name="Name of technical parameter"; TECHNICAL_PARAMETER_NAME: _FillValue = " ";	Name of the technical parameter. Example : "CLOCK_FloatTime_HHMMSS" See reference table 14 for standard technical parameter names.
TECHNICAL_PARAMETER_VALUE	char TECHNICAL_PARAMETER_VALUE( N_TECH_PARAM, STRING128) TECHNICAL_PARAMETER_VALUE: long_name="Value of technical parameter"; TECHNICAL_PARAMETER_VALUE: _FillValue = " ";	Value of the technical parameter. Example : "125049"
CYCLE_NUMBER	int CYCLE_NUMBER(N_TECH_PARAM ); 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;	Cycle number of the technical parameter. Example : 157

## 2.8. GDAC FTP directory file format

### 2.8.1. Profile directory file format

The profile directory file describes all individual profile files of the GDAC ftp site. Its format is an autodescriptive Ascii with comma separated values.

The directory file contains :

- A header with a list of general informations : title, description, project name, format version, date of update, ftp root addresses, GDAC node
- A table with a description of each file of the GDAC ftp site. This table is a comma separated list.

#### Profile directory format definition

```
# Title : Profile directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all individual profile files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : YYYYMMDDHHMISS
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file,date,latitude,longitude,ocean,profiler_type,institution,date_update
```

- file : path and file name on the ftp site. The file name contain the float number and the cycle number.  
Fill value : none, this field is mandatory
- date : date of the profile, YYYYMMDDHHMISS  
Fill value : " " (blank)
- latitude, longitude : location of the profile  
Fill value : 99999.
- ocean : code of the ocean of the profile as described in reference table 13  
Fill value : " " (blank)
- profiler\_type : type of profiling float as described in reference table 8  
Fill value : " " (blank)
- institution : institution of the profiling float described in reference table 4  
Fill value : " " (blank)
- date\_update : : date of last update of the file, YYYYMMDDHHMISS  
Fill value : " " (blank)

Each line describes a file of the gdac ftp site.

#### Profile directory format example

```
# Title : Profile directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all profile files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : 20031028075500
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file,date,latitude,longitude,ocean,profiler_type,institution,date_update
aoml/13857/profiles/R13857_001.nc,199707292003,0.267,-16.032,A,0845,AO,20030214155117
aoml/13857/profiles/R13857_002.nc,199708091921,0.072,-17.659,A,0845,AO,20030214155354
aoml/13857/profiles/R13857_003.nc,199708201845,0.543,-19.622,A,0845,AO,20030214155619
...
```

```
jma/29051/profiles/R29051_025.nc,200110250010,30.280,143.238,P,846,JA,20030212125117
jma/29051/profiles/R29051_026.nc,200111040004,30.057,143.206,P,846,JA,20030212125117
```

## 2.8.2. Profile directory file format version 2.1

The profile directory file describes all individual profile files of the GDAC ftp site. Its format is an autodescription Ascii with comma separated values.

The directory file contains :

- A header with a list of general informations : title, description, project name, format version, date of update, ftp root addresses, GDAC node
- A table with a description of each file of the GDAC ftp site. This table is a comma separated list.

### Compression du fichier index

- Le fichier est compressé avec gzip (la version non compressée est supprimée).

### Signature MD5

A chaque génération de fichier index, une signature MD5 est produite dans un fichier.

La signature permet à l'utilisateur de vérifier que le fichier qu'il a collecté par ftp est identique au fichier original.

### Nomenclature

- etc/argo\_profile\_detailed\_index.txt.gz
- etc/argo\_profile\_detailed\_index.txt.gz.md5

## Detailed profile directory format definition

```
# Title : Profile directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all individual profile files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.1
# Date of update : YYYYMMDDHHMISS
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file,date,latitude,longitude,ocean,profiler_type,institution,date_update,profile_temp_qc,profile_psal_qc,profile_doxy_qc,ad_psal_adjustment_mean,ad_psal_adjustment_deviation
```

- file: path and file name on the ftp site. The file name contain the float number and the cycle number.  
Fill value : none, this field is mandatory
- date: date of the profile, YYYYMMDDHHMISS  
Fill value : " " (blank)
- latitude, longitude : location of the profile  
Fill value : 99999.
- ocean: code of the ocean of the profile as described in reference table 13  
Fill value : " " (blank)
- profiler\_type : type of profiling float as described in reference table 8  
Fill value : " " (blank)
- institution: institution of the profiling float described in reference table 4  
Fill value : " " (blank)
- date\_update: date of last update of the file, YYYYMMDDHHMISS  
Fill value: " " (blank)

- profile\_temp\_qc,profile\_psal\_qc,profile\_doxy\_qc : global quality flag on temperature, salinity and oxygene profile.  
Fill value: " " (blank)
- ad\_psal\_adjustment\_mean : for delayed mode or adjusted mode  
Mean of psal\_adjusted – psal on the deepest 500 meters with good psal\_adjusted\_qc (equal to 1)  
Fill value: " " (blank)
- ad\_psal\_adjustment\_deviation : for delayed mode or adjusted mode  
Standard deviation of psal\_adjusted – psal on the deepest 500 meters with good psal\_adjusted\_qc (equal to 1)  
Fill value: " " (blank)
- n\_levels : maximum number of pressure levels contained in a profile  
Fill value: " " (blank)

Each line describes a file of the gdac ftp site.

#### Profile directory format example

```
# Title : Profile directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all individual profile files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.1
# Date of update : 20081025220004
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file,date,latitude,longitude,ocean,profiler_type,institution,date_update,profile_temp_qc,profile_psal_qc,profile_doxy_qc,ad_psal_adjustment_mean,ad_psal_adjustment_deviation,n_levels
aoml/13857/profiles/R13857_001.nc,19970729200300,0.267,-16.032,A,845,AO,20080918131927,A, , ,
aoml/13857/profiles/R13857_002.nc,19970809192112,0.072,-17.659,A,845,AO,20080918131929,A, , ,
aoml/13857/profiles/R13857_003.nc,19970820184545,0.543,-19.622,A,845,AO,20080918131931,A, , ,
...
meds/3900084/profiles/D3900084_099.nc,20050830130800,-45.74,-58.67,A,846,ME,20060509152833,A,A, ,0.029,0.000
meds/3900084/profiles/D3900084_103.nc,20051009125300,-42.867,-56.903,A,846,ME,20060509152833,A,A, ,-0.003,0.000
...
```

### 2.8.3. Trajectory directory format

The trajectory directory file describes all trajectory files of the GDAC ftp site. Its format is an autodescriptive Ascii with comma separated values.

The directory file contains :

- A header with a list of general informations : title, description, project name, format version, date of update, ftp root addresses, GDAC node
- A table with a description of each file of the GDAC ftp site. This table is a comma separated list.

#### Trajectory directory format definition

```
# Title : Trajectory directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all trajectory files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : YYYYMMDDHHMISS
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file, latitude_max, latitude_min, longitude_max, longitude_min, profiler_type, institution, date_update
```

- file : path and file name on the ftp site

Fill value : none, this field is mandatory

- latitude\_max, latitude\_min, longitude\_max, longitude\_min : extreme locations of the float  
Fill values : 99999.
- profiler\_type : type of profiling float as described in reference table 8  
Fill value : " " (blank)
- institution : institution of the profiling float described in reference table 4  
Fill value : " " (blank)
- date\_update : date of last update of the file, YYYYMMDDHHMISS  
Fill value : " " (blank)

#### Trajectory directory format example

```
# Title : Trajectory directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all trajectory files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : 20031028075500
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file, latitude_max, latitude_min, longitude_max, longitude_min, profiler_type, institution, date_update
aoml/13857/13857_traj.nc,1.25,0.267,-16.032,-18.5,0845,AO,20030214155117
aoml/13857/13857_traj.nc,0.072,-17.659,A,0845,AO,20030214155354
aoml/13857/13857_traj.nc,0.543,-19.622,A,0845,AO,20030214155619
...
jma/29051/29051_traj.nc,32.280,30.280,143.238,140.238,846,JA,20030212125117
jma/29051/29051_traj.nc,32.352,30.057,143.206,140.115,846,JA,20030212125117
```

#### 2.8.4. Meta-data directory format

The metadata directory file describes all metadata files of the GDAC ftp site. Its format is an autodescription in Ascii with comma separated values.

The directory file contains :

- A header with a list of general informations : title, description, project name, format version, date of update, ftp root addresses, GDAC node
- A table with a description of each file of the GDAC ftp site. This table is a comma separated list.

#### Metadata directory format definition

```
# Title : Metadata directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all metadata files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : YYYYMMDDHHMISS
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgodae.usgodae.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file, profiler_type, institution, date_update
```

- file : path and file name on the ftp site  
Fill value : none, this field is mandatory
- profiler\_type : type of profiling float as described in reference table 8  
Fill value : " " (blank)

- institution : institution of the profiling float described in reference table 4  
Fill value : " " (blank)
- date\_update : date of last update of the file, YYYYMMDDHHMISS  
Fill value : " " (blank)

#### Metadata directory example

```
# Title : Metadata directory file of the Argo Global Data Assembly Center
# Description : The directory file describes all metadata files of the argo GDAC ftp site.
# Project : ARGO
# Format version : 2.0
# Date of update : 20031028075500
# FTP root number 1 : ftp://ftp.ifremer.fr/ifremer/argo/dac
# FTP root number 2 : ftp://usgoda.usgoda.org/pub/outgoing/argo/dac
# GDAC node : CORIOLIS
file, profiler_type, institution, date_update
aoml/13857/13857_meta.nc,0845,AO,20030214155117
aoml/13857/13857_meta.nc,0845,AO,20030214155354
aoml/13857/13857_meta.nc,0845,AO,20030214155619
...
jma/29051/29051_meta.nc,846,JA,20030212125117
jma/29051/29051_meta.nc,846,JA,20030212125117
```

### 3. Reference tables

#### 3.1. Reference table 1 : data type

The following table contains the list of acceptable contents for DATA\_TYPE field.

Name
Argo profile
Argo trajectory
Argo meta-data
Argo technical data

#### 3.2. Reference table 2 : Argo quality control flag scale

##### 3.2.1. Reference table 2 : measurement flag scale

n	Meaning	Real-time comment	Delayed-mode comment
0	No QC was performed	No QC was performed	No QC was performed
1	Good data	All Argo real-time QC tests passed.	The adjusted value is statistically consistent and a statistical error estimate is supplied.
2	Probably good data	Probably good data	Probably good data
3	Bad data that are potentially correctable	Test 15 or Test 16 or Test 17 failed and all other real-time QC tests passed. These data are not to be used without scientific correction. A flag '3' may be assigned by an operator during additional visual QC for bad data that may be corrected in delayed mode.	An adjustment has been applied, but the value may still be bad.
4	Bad data	Data have failed one or more of the real-time QC tests, excluding Test 16. A flag '4' may be assigned by an operator during additional visual QC for bad data that are not correctable.	Bad data. Not adjustable.
5	Value changed	Value changed	Value changed
6	Not used	Not used	Not used
7	Not used	Not used	Not used
8	Interpolated value	Interpolated value	Interpolated value
9	Missing value	Missing value	Missing value

### 3.2.2. Reference table 2a : profile quality flag

*N* is defined as the percentage of levels with good data where:

- o QC flag values of 1, 2, 5, or 8 are GOOD data
- o QC flag values of 9 (missing) are NOT USED in the computation
- o All other QC flag values are BAD data

The computation should be taken from <PARAM\_ADJUSTED\_QC> if available and from <PARAM\_QC> otherwise.

n	Meaning
“ “	No QC performed
A	<b>N</b> = 100%; All profile levels contain good data.
B	75% <= <b>N</b> < 100%
C	50% <= <b>N</b> < 75%
D	25% <= <b>N</b> < 50%
E	0% < <b>N</b> < 25%
F	<b>N</b> = 0%; No profile levels have good data.

Example:

A TEMP profile has 60 levels (3 levels contain missing values).

- o 45 levels are flagged as 1
- o 5 levels are flagged as 2
- o 7 levels are flagged as 4
- o 3 levels are flagged as 9 (missing)

Percentage of good levels =  $((45 + 5) / 57) * 100 = 87.7\%$

PROFILE\_TEMP\_QC = “B”;

### 3.3. Reference table 3: parameter code table

The following table describes the parameter codes used for Argo data management.

Code	Parameter long name	Comment	Unit	Valid min	Valid max	C_Format FORTRAN_ Format resolution	Fill value
CNDC	ELECTRICAL CONDUCTIVITY	In situ measurement	mhos/m	0.f	60.f	%10.4f F10.4 0.0001f	99999.f
DOXY	DISSOLVED OXYGEN	In situ measurement	micromole/kg	0.f	650.f	%9.3f F9.3 0.001f	99999.f
PRES	SEA PRESSURE	In situ measurement, sea surface = 0	decibar	0.f	12000.f	%7.1f F7.1 0.1f	99999.f
PSAL	PRACTICAL SALINITY	In situ measurement	psu	0.f	42.f	%9.3f F9.3 0.001f	99999.f
TEMP	SEA TEMPERATURE IN SITU ITS-90 SCALE	In situ measurement	degree_Celsius	-2.f	40.f	%9.3f F9.3 0.001f	99999.f
TEMP_DOXY	SEA TEMPERATURE FROM DOXY SENSOR (ITS-90 SCALE)	In situ measurement	degree_Celsius	-2.f	40.f	%9.3f F9.3 0.001f	99999.f
BPHASE_DOXY	BPHASE_DOXY	In situ measurement	degree	10.f	70.f	%8.2f F8.2 0.01f	99999.f

#### Remark on duplicate sensors

Some floats are equipped with 2 different sensors, measuring the same physical parameter. In that case, add the integer "2" at the end of the code of the duplicate parameter (e.g. DOXY2). If more sensors that measure the same physical parameter are added, then the integer will simply increase by 1 (i.e. DOXY3, DOXY4, and so on).

#### Example

If a float has one Optode and one SBE oxygen sensor:

- Use DOXY and TEMP\_DOXY for Optode
- Use DOXY2 for SBE

If a float has two Optode oxygen sensors:

- Use DOXY and TEMP\_DOXY, and DOXY2 and TEMP\_DOXY2

If a float has two SBE oxygen sensors:

- Use DOXY and DOXY2

#### Remark on unit conversion of oxygen

The unit of DOXY is micromole/kg in Argo data and the oxygen measurements are sent from Argo floats in another unit such as micromole/L for Optode and ml/L for SBE. Thus the unit conversion is carried out by DACs as follows:

- $O_2 [\text{micromole/kg}] = O_2 [\text{micromole/L}] / \rho$
- $O_2 [\text{micromole/L}] = 44.6596 \times O_2 [\text{ml/L}]$

Here,  $\rho$  is the potential density of water [kg/L] at zero pressure and at the potential temperature (e.g., 1.0269 kg/L; e.g., UNESCO, 1983). The value of 44.6596 is derived from the molar volume of the oxygen gas, 22.3916 L/mole, at standard temperature and pressure (0°C, 1 atmosphere; e.g., García and Gordon, 1992).

García, H.E. and L.I. Gordon (1992): Oxygen solubility in sea water: better fitting equations. *Limnol. Oceanogr.*, 37(6), 1307-1312.

UNESCO (1983): Algorithms for computation of fundamental properties of seawater. *Unesco technical papers in marine science*, 44, 53pp.

### 3.4. Reference table 4 : data centres and institutions codes

Data centres and institutions	
AO	AOML, USA
BO	BODC, United Kingdom
CI	Institute of Ocean Sciences, Canada
CS	CSIRO, Australia
GE	BSH, Germany
GT	GTS : used for data coming from WMO GTS network
HZ	CSIO, China Second Institute of Oceanography
IF	Ifremer, France
IN	INCOIS, India
JA	JMA, Japan
JM	Jamstec, Japan
KM	KMA, Korea
KO	KORDI, Korea
ME	MEDS, Canada
NA	NAVO, USA
PM	PMEL, USA
RU	Russia
SI	SIO, Scripps, USA
SP	Spain
UW	University of Washington, USA
VL	Far Eastern Regional Hydrometeorological Research Institute of Vladivostock, Russia
WH	Woods Hole Oceanographic Institution, USA

### 3.5. Reference table 5 : location classes (ARGOS)

ARGOS location classes	
Value	Estimated accuracy in latitude and longitude
0	accuracy estimation over 1500m radius
1	accuracy estimation better than 1500m radius
2	accuracy estimation better than 500 m radius
3	accuracy estimation better than 250 m radius
G	GPS positioning

### 3.6. Reference table 6 : data state indicators

Level	Descriptor
0	Data are the raw output from instruments, without calibration, and not necessarily converted to engineering units. These data are rarely exchanged
1	Data have been converted to values independent of detailed instrument knowledge. Automated calibrations may have been done. Data may not have full geospatial and temporal referencing, but have sufficient information to uniquely reference the data to the point of measurement.
2	Data have complete geospatial and temporal references. Information may have been compressed (e.g. subsampled, averaged, etc.) but no assumptions of scales of variability or thermodynamic relationships have been used in the processing.
3	The data have been processed with assumptions about the scales of variability or hermodynamic relationships. The data are normally reduced to regular space, time intervals with enhanced signal to noise.

Class	Descriptor	Subclass
A	No scrutiny, value judgements or intercomparisons are performed on the data. The records are derived directly from the input with no filtering, or subsampling.	<ul style="list-style-type: none"> <li>- Some reductions or subsampling has been performed, but the original record is available.</li> <li>+ Geospatial and temporal properties are checked. Geophysical values are validated. If not validated, this is clearly indicated.</li> </ul>
B	Data have been scrutinized and evaluated against a defined and documented set of measures. The process is often automated (i.e. has no human intervention) and the measures are published and widely available.	<ul style="list-style-type: none"> <li>- Measures are completely automated, or documentation is not widely available.</li> <li>+ The measures have been tested on independent data sets for completeness and robustness and are widely accepted.</li> </ul>
C	Data have been scrutinized fully including intra-record and intra-dataset comparison and consistency checks. Scientists have been involved in the evaluation and brought latest knowledge to bear. The procedures are published, widely available and widely accepted.	<ul style="list-style-type: none"> <li>- Procedures are not published or widely available. Procedures have not undergone full scrutiny and testing.</li> <li>+ Data are fully quality controlled, peer reviewed and are widely accepted as valid. Documentation is complete and widely available.</li> </ul>

#### Data state indicator recommended use

The following table describes the processing stage of data and the value to be assigned the data state indicator (DS Indicator). It is the concatenation of level and class described above.

Processing Stage	DS Indicator
1. Data pass through a communications system and arrive at a processing centre. The data resolution is the highest permitted by the technical constraints of the floats and communications system.	0A (note 1)
2. The national centre assembles all of the raw information into a complete profile located in space and time.	1A (note 2)
3. The national centre passes the data through automated QC procedures and prepares the data for distribution on the GTS, to global servers and to PIs.	2B
4. Real-time data are received at global data centres that apply QC including visual inspection of the data. These are then distributed to users in near real-time	2B+ (note 3)

5. Data are reviewed by PIs and returned to processing centres. The processing centres forward the data to the global Argo servers.	2C
6. Scientists accept data from various sources, combine them as they see fit with other data and generate a product. Results of the scientific analysis may be returned to regional centres or global servers. Incorporation of these results improves the quality of the data.	2C+
7. Scientists working as part of GODAE generate fields of gridded products delivered in near real-time for distribution from the global servers. Generally, these products mostly will be based on data having passed through automated QC procedures.	3B (note 4)
8. Scientists working as part of GODAE generate fields of gridded products delivered with some time delay for distribution from the global servers. Generally, these products mostly will be based on data having passed through manual or more sophisticated QC procedures than employed on the real-time data.	3C

## Notes

1. We need to have a pragmatic approach to what constitutes "original" or "raw" data. Despite the fact that an instrument may be capable of high sampling rates, what is reported from the instrument defines what is considered "raw". For example, Argo floats can certainly sample at finer scales than every 10 db, but because of communications, all we see for now is data at that (or worse) vertical resolution. Therefore the data "coming from the instrument" is "raw" output at 10db resolution.
2. The conversion of the raw data stream from the communications system into profiles of variables causes the data state indicator to switch from level 0 to 1.
3. Even though the data at global data centres use manual or semi-automated QC procedures, there is often not the intercomparisons to larger data collections and fields that would qualify the data state indicator to be set to class C. This is generally only provided by scientific scrutiny of the data.
4. The transition from class 2 to 3 occurs when assumptions of scales of variability are applied. During the course of normal data processing it is common to carry out some averaging and subsampling. This is usually done to exploit oversampling by the instrument, and to ensure good measurements are achieved. These are considered to be part of the geospatial and temporal referencing process.

### 3.7. Reference table 7 : history action codes

Code	Meaning
CF	Change a quality flag
CR	Create record
CV	Change value
DC	Station was checked by duplicate checking software
ED	Edit a parameter value
IP	This history group operates on the complete input record
NG	No good trace
PE	Position error. Profile position has been erroneously encoded. Corrected if possible.
QC	Quality Control
QCF\$	Tests failed
QCP\$	Test performed
SV	Set a value

TE	Time error. Profile date/time has been erroneously encoded. Corrected if possible.
UP	Station passed through the update program

### 3.8. Reference table 8 : instrument types

The instrument type codes comes from WMO table 1770. The WMO instrument types are available on the following web site :

[http://www.meds-sdmm.dfo-mpo.gc.ca/meds/Prog\\_Int/J-COMM/CODES/wmortable\\_e.htm#ct1770](http://www.meds-sdmm.dfo-mpo.gc.ca/meds/Prog_Int/J-COMM/CODES/wmortable_e.htm#ct1770)

Code Figure	Instrument
831	P-Alace float
840	Provor, no conductivity
841	Provor, Seabird conductivity sensor
842	Provor, FSI conductivity sensor
845	Webb Research, no conductivity
846	Webb Research, Seabird sensor
847	Webb Research, FSI sensor
850	Solo, no conductivity
851	Solo, Seabird conductivity sensor
852	Solo, FSI conductivity sensor
855	Ninja, no conductivity sensor
856	Ninja, SBE conductivity sensor
857	Ninja, FSI conductivity sensor
858	Ninja, TSK conductivity sensor

### 3.9. Reference table 9 : positioning system

Code	Description
ARGOS	ARGOS positioning system
GPS	GPS positioning system

### 3.10. Reference table 10 : transmission system

Code	Description
ARGOS	Argos transmission system
IRIDIUM	Iridium transmission system
ORBCOMM	Orbcomm transmission system

### 3.11. Reference table 11 : QC Test Ids

ID	Test
2	Platform Identification test
4	Impossible Date test
8	Impossible Location test
16	Position on Land test
32	Impossible Speed test
64	Global Range test
128	Regional Global Parameter test
256	Pressure Increasing test
512	Spike test
1024	Top and Bottom Spike test
2048	Gradient test
4096	Digit Rollover test
8192	Stuck Value test
16384	Density Inversion test
32768	Grey List test
65536	Gross Salinity or Temperature Sensor Drift test
131072	Visual QC test
261144	Frozen profile test
524288	Deepest pressure test

### 3.12. Reference table 12 : history steps codes

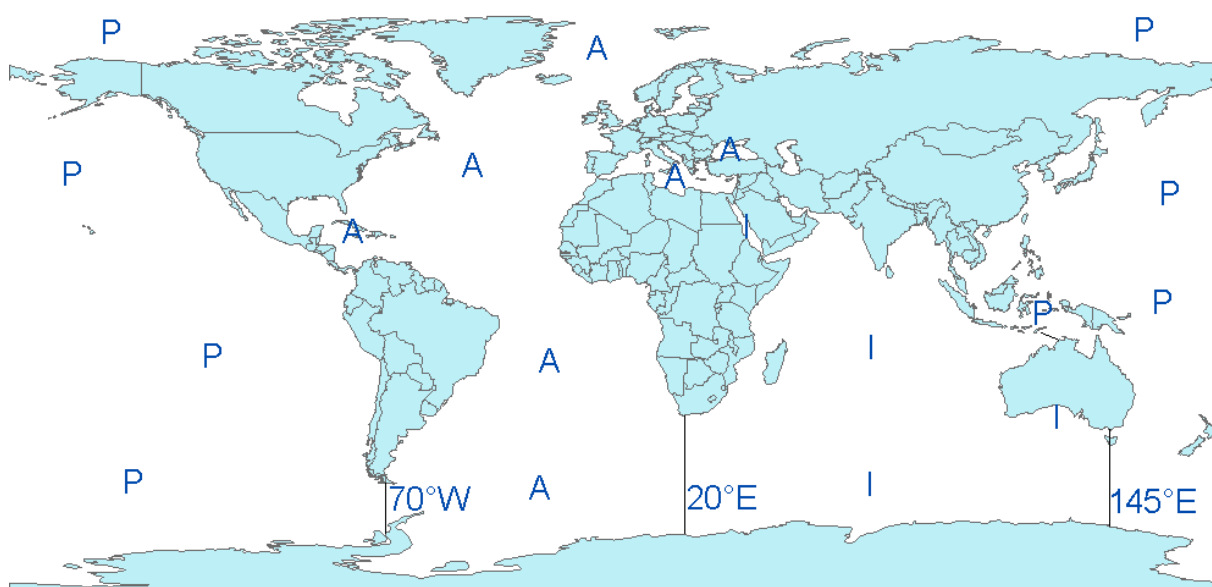
Code	Meaning
ARFM	Convert raw data from telecommunications system to a processing format
ARGQ	Automatic QC of data reported in real-time has been performed
IGO3	Checking for duplicates has been performed
ARSQ	Delayed mode QC has been performed
ARCA	Calibration has been performed
ARUP	Real-time data have been archived locally and sent to GDACs
ARDU	Delayed data have been archived locally and sent to GDACs
RFMT	Reformat software to convert hexadecimal format reported by the buoy to our standard format

If individual centres wish to record other codes, they may add to this list as they feel is appropriate.

### 3.13. Reference table 13 : ocean codes

The ocean codes are used in the GDAC ftp directory files. The ocean code is not used in Argo NetCDF files.

Code	Meaning
A	Atlantic ocean area
I	Indian ocean area
P	Pacific ocean area



- The Pacific/Atlantic boundary is 70°W.
- The Pacific/Indian boundary is 145°E.
- The Atlantic/Indian boundary is 20°E.

### 3.14. Reference table 14 : technical parameter names

Most technical parameter codes are not standardized. However, certain usual parameter names are available.

Parameter name	Meaning
SURFACE_PRESSURE	Sea surface pressure
BATTERY_VOLTAGE	Battery voltage in volts

15/10/2008 : new proposal for technical parameters

A proposal for technical parameters naming convention is available at :

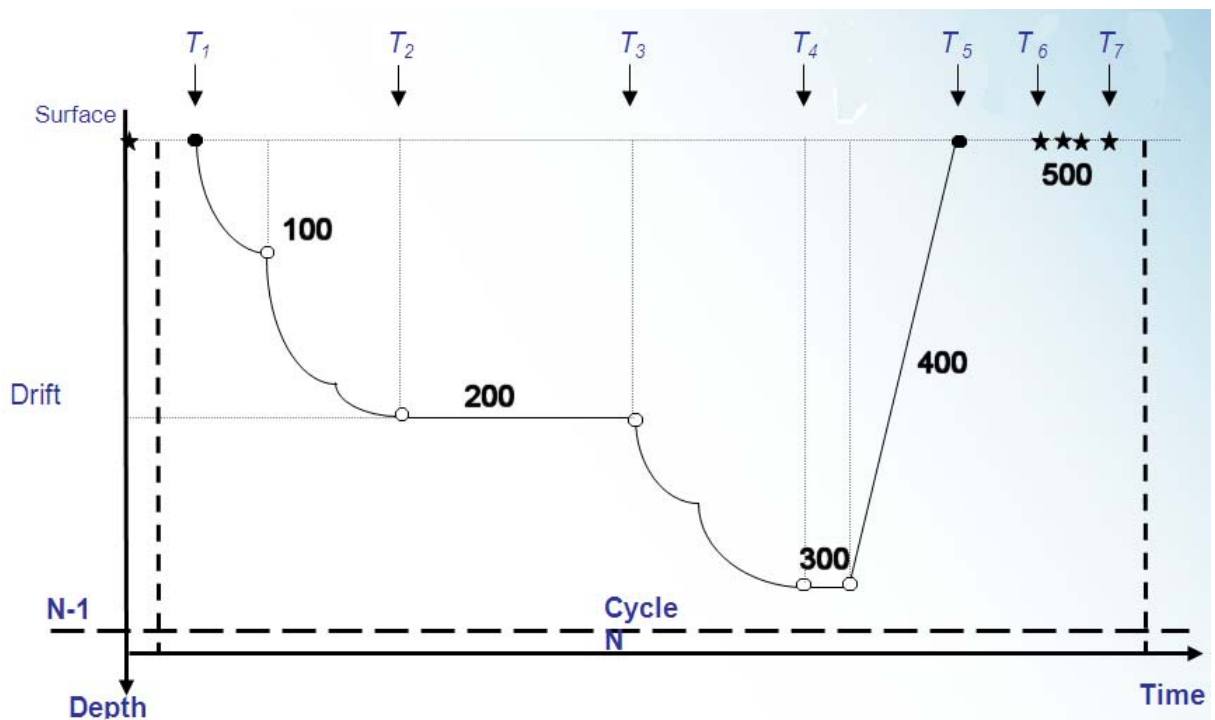
[http://www.coriolis.eu.org/cdc/argo/Technical\\_Naming\\_Convention\\_rules.doc](http://www.coriolis.eu.org/cdc/argo/Technical_Naming_Convention_rules.doc)

The new technical parameters are available at :

[http://www.coriolis.eu.org/cdc/argo/table\\_of\\_technical\\_name\\_equivalents\\_draft\\_final\\_v2.xls](http://www.coriolis.eu.org/cdc/argo/table_of_technical_name_equivalents_draft_final_v2.xls)

### 3.15. Reference table 15 : convention for describing progress within a cycle

The lookup table allows matching the part of the cycle when a located event is performed.



Progress code	Meaning
1	descent start (end of the surface drift)
2	beginning of drift at parking pressure
3	end of drift at parking pressure
4	start of ascent from maximal pressure
5	ascent end (start of the surface drift)
6	start of transmission
7	end of transmission
8	best estimate of parking pressure (can be either measured or from meta-data). More detail may be available in the metadata and technical files.
100	descending phase (between codes 1 and 2)
200	parking pressure phase (between codes 2 and 3)
300	descent to maximum pressure (between codes 3 and 4)
400	ascent to surface (between codes 4 and 5)
500	surface drift phase (between code 5 and code 1 of the following cycle)

Codes from 1 to 99 are specific events. They occur no more than once per cycle.

Codes 100 and above describe phases of the cycle. They can occur multiple times, some are measured events, some may be estimated.

## 4. Data access

The whole Argo data set is available in real time and delayed mode from the global data centres (GDACs).

The internet addresses are :

- <http://www.usgodae.org/argo/argo.html>
- <http://www.coriolis.eu.org/cdc>

The FTP addresses are :

- <ftp://usgodae1.fnmoc.navy.mil/pub/outgoing/argo>
- <ftp://ftp.ifremer.fr/ifremer/argo>

The 2 GDACs offer the same data set that is mirrored in real time.

### 4.1. File naming convention on GDACs

The GADC ftp sites comply with the following naming conventions :

#### Profile data

For floats that collect no more than 1 ascending and 1 descending profile per cycle the file names for individual profiles are <R/D><FloatID>\_<XXX><D>.nc where the initial R indicates Real-Time data the initial D indicates Delayed-Mode data XXX is the cycle number the second D indicates a descending profile (profiles without this D are collected during ascent).

For floats that collect 2 or more ascending or descending profiles per cycle the file names for individual profiles are <R/D><FloatID>\_<XXX><D><\_YY>.nc where the initial R indicates Real-Time data the initial D indicates Delayed-Mode data XXX is the cycle number the second D indicates a descending profile (profiles without this D are collected during ascent).

YY counts multiple ascending/descending profiles separately

Since floats can alternate between the two modes, they may have file names following both conventions.

Examples:

- R1900045\_003.nc, R1900045\_003D.nc
- R1900046\_007\_01.nc, R1900067\_007\_02.nc, R1900067\_007\_03.nc
- R1900046\_007D\_01.nc, R1900067\_007D\_02.nc, R1900067\_007D\_03.nc
- R1900045\_003.nc, R1900045\_004\_01.nc, R1900045\_004\_02.nc, R1900045\_004\_03.nc, R1900045\_004\_04.nc, R1900045\_005.nc

#### Trajectory Data

- <FloatID>\_traj.nc  
Example : 1900045\_traj.nc

#### Metadata

- <FloatID>\_meta.nc  
Example : 1900045\_meta.nc

### Technical Data

- <FloatID>\_tech.nc  
Example : 1900045\_tech.nc

## 4.2. Other data sources

All Argo data are available from Argo GDACs (Global data centres).

Some Argo data are also available from GTS (Global Telecommunication System), a network operated by WMO (World Meteorological Organization).

On GTS there are 2 formats for Argo profiles :

- TESAC : an Ascii format
- BUFR : a binary format under development.

The description of these format is available from the WMO web site :

- <http://www.wmo.ch>
- <http://www.wmo.ch/web/www/DPS/NewCodesTables/WMO306vol-I-1PartA.pdf>

## 5. Using the History section of the Argo netCDF Structure

Within the netCDF format are a number of fields that are used to track the progression of the data through the data system. This section records the processing stages, results of actions that may have altered the original values and information about QC tests performed and failed. The purpose of this document is to describe how to use this section of the format.

The creation of entries in the history section is the same for both profile and trajectory data. The next sections provide examples of what is expected. The information shown in the column labeled "Sample" is what would be written into the associated "Field" name in the netCDF format.

### 5.1. Recording information about the Delayed Mode QC process

The process of carrying out delayed mode QC may result in adjustments being made to observed variables. The table below shows how to record that the delayed mode QC has been done. Note that the fields HISTORY\_SOFTWARE, HISTORY\_SOFTWARE\_RELEASE and HISTORY\_REFERENCE are used together to document the name and version of software used to carry out the delayed QC, and the reference database used in the process. The contents of these three fields are defined locally by the person carrying out the QC.

Example: History entry to record that delayed mode QC has been carried out

Field	Sample	Explanation
HISTORY_INSTITUTION	CI	Selected from the list in reference table 4
HISTORY_STEP	ARSQ	Selected from the list in reference table 12.
HISTORY_SOFTWARE	WJO	This is a locally defined name for the delayed mode QC process employed.
HISTORY_SOFTWARE_RELEASE	1	This is a locally defined indicator that identifies what version of the QC software is being used.
HISTORY_REFERENCE	WOD2001	This is a locally defined name for the reference database used for the delayed mode QC process.
HISTORY_DATE	200308050000	The year, month, day, hour, minute, second that the process ran
HISTORY_ACTION	IP	Selected from the list in reference table 7
HISTORY_PARAMETER	FillValue	This field does not apply (1)
HISTORY_START PRES	FillValue	This field does not apply
HISTORY_STOP PRES	FillValue	This field does not apply
HISTORY_PREVIOUS_VALUE	FillValue	This field does not apply
HISTORY_QCTEST	FillValue	This field does not apply

#### Note :

(1) The present version of delayed mode QC only tests salinity and as such it is tempting to place "PSAL" in the \_PARAMETER field. In future, delayed mode QC tests may include tests for temperature, pressure and perhaps other parameters. For this reason, simply addressing the software and version number will tell users what parameters have been tested.

## 5.2. Recording processing stages

Each entry to record the processing stages has a similar form. An example is provided to show how this is done. Note that reference table 12 contains the present list of processing stages and there should be at least one entry for each of these through which the data have passed. If data pass through one of these steps more than once, an entry for each passage should be written and the variable N\_HISTORY updated appropriately.

Some institutions may wish to record more details of what they do. In this case, adding additional “local” entries to table 12 is permissible as long as the meaning is documented and is readily available. These individual additions can be recommended to the wider community for international adoption.

**Example:** History entry to record decoding of the data.

Field	Sample	Explanation
HISTORY_INSTITUTION	ME	Selected from the list in reference table 4
HISTORY_STEP	ARFM	Selected from the list in reference table 12.
HISTORY_SOFTWARE	FillValue	This field does not apply
HISTORY_SOFTWARE_RELEASE	FillValue	This field does not apply
HISTORY_REFERENCE	FillValue	This field does not apply
HISTORY_DATE	2003080500000	The year, month, day, hour, minute, second that the process ran
HISTORY_ACTION	IP	Selected from the list in reference table 7
HISTORY_PARAMETER	FillValue	This field does not apply
HISTORY_START_PRES	FillValue	This field does not apply
HISTORY_STOP_PRES	FillValue	This field does not apply
HISTORY_PREVIOUS_VALUE	FillValue	This field does not apply
HISTORY_QCTEST	FillValue	This field does not apply

### 5.3. Recording QC Tests Performed and Failed

The delayed mode QC process is recorded separately from the other QC tests that are performed because of the unique nature of the process and the requirement to record other information about the reference database used. When other tests are performed, such as the automated real-time QC, a group of tests are applied all at once. In this case, instead of recording that each individual test was performed and whether or not the test was failed, it is possible to document all of this in two history records.

The first documents what suite of tests was performed, and the second documents which tests in the suite were failed. A test is failed if the value is considered to be something other than good (i.e. the resulting QC flag is set to anything other than “1”). An example of each is provided. If data pass through QC more than once, an entry for each passage should be written and the variable N\_HISTORY updated appropriately.

Example: QC tests performed and failed.

The example shown here records that the data have passed through real-time QC and that two tests failed. The encoding of tests performed is done by adding the ID numbers provided in reference table 11 for all tests performed, then translating this to a hexadecimal number and recording this result.

**Record 1:** Documenting the tests performed

Field	Sample	Explanation
HISTORY_INSTITUTION	ME	Selected from the list in reference table 4
HISTORY_STEP	ARGQ	Selected from the list in reference table 12.
HISTORY_SOFTWARE	FillValue	This field does not apply
HISTORY_SOFTWARE_RELEASE	FillValue	This field does not apply
HISTORY_REFERENCE	FillValue	This field does not apply
HISTORY_DATE	20030805000000	The year, month, day, hour, minute, second that the process ran
HISTORY_ACTION	QCP\$	Selected from the list in reference table 7
HISTORY_PARAMETER	FillValue	This field does not apply
HISTORY_START_PRES	FillValue	This field does not apply
HISTORY_STOP_PRES	FillValue	This field does not apply
HISTORY_PREVIOUS_VALUE	FillValue	This field does not apply
HISTORY_QCTEST	1BE	This is the result of all tests with IDs from 2 to 256 having been applied (see reference table 11)

**Record 2:** Documenting the tests that were failed

Field	Sample	Explanation
HISTORY_INSTITUTION	ME	Selected from the list in reference table 4
HISTORY_STEP	ARGQ	Selected from the list in reference table 12.
HISTORY_SOFTWARE	FillValue	This field does not apply
HISTORY_SOFTWARE_RELEASE	FillValue	This field does not apply
HISTORY_REFERENCE	FillValue	This field does not apply
HISTORY_DATE	20030805000000	The year, month, day, hour, minute, second that the process ran
HISTORY_ACTION	QCF\$	Selected from the list in reference table 7
HISTORY_PARAMETER	FillValue	This field does not apply
HISTORY_START_PRES	FillValue	This field does not apply

HISTORY_STOP_PRES	FillValue	This field does not apply
HISTORY_PREVIOUS_VALUE	FillValue	This field does not apply
HISTORY_QCTEST	A0	This is the result when data fail tests with IDs of 32 and 128 (see reference table 11)

#### 5.4. Recording changes in values

The PIs have the final word on the content of the data files in the Argo data system. In comparing their data to others there may arise occasions when changes may be required in the data.

We will use the example of recomputation of where the float first surfaced as an example. This computation process can be carried out once all of the messages from a float have been received. Not all real-time processing centres make this computation, but it can be made later on and added to the delayed mode data. If this is the case, we would insert the new position of the profile into the latitude and longitude fields in the profile and we would record the previous values in two history entries. Recording these allows us to return to the original value if we have made an error in the newly computed position. The two history entries would look as follows.

##### Example: Changed latitude

Field	Sample	Explanation
HISTORY_INSTITUTION	CI	Selected from the list in reference table 4
HISTORY_STEP	ARGQ	Selected from the list in reference table 12.
HISTORY_SOFTWARE	FillValue	This field does not apply
HISTORY_SOFTWARE_RELEASE	FillValue	This field does not apply
HISTORY_REFERENCE	FillValue	This field does not apply
HISTORY_DATE	20030805000000	The year, month, day, hour, minute, second that the process ran
HISTORY_ACTION	CV	Selected from the list in reference table 7
HISTORY_PARAMETER	LAT\$	A new entry for reference table 3 created by institution CI to indicate changes have been made in the latitude.
HISTORY_START_PRES	FillValue	This field does not apply
HISTORY_STOP_PRES	FillValue	This field does not apply
HISTORY_PREVIOUS_VALUE	23.456	This is the value of the latitude before the change was made.
HISTORY_QCTEST	FillValue	This field does not apply

##### Notes :

1. Be sure that the new value is recorded in the latitude and longitude of the profile section.
2. Be sure that the POSITION\_QC flag is set to "5" to indicate to a user that the value now in the position has been changed from the original one that was there.
3. Be sure to record the previous value in history entries.

It is also sometimes desirable to record changes in quality flags that may arise from reprocessing data through some QC procedures. In this example, assume that whereas prior to the analysis, all temperature values from 75 to 105 dbars were considered correct, after the analysis, they are considered wrong. The history entry to record this would look as follows.

**Example:** Changed flags

Field	Sample	Explanation
HISTORY_INSTITUTION	CI	Selected from the list in reference table 4
HISTORY_STEP	ARGQ	Selected from the list in reference table 12.
HISTORY_SOFTWARE	FillValue	This field does not apply
HISTORY_SOFTWARE_RELEASE	FillValue	This field does not apply
HISTORY_REFERENCE	FillValue	This field does not apply
HISTORY_DATE	20030805000000	The year, month, day, hour, minute, second that the process ran
HISTORY_ACTION	CF	Selected from the list in reference table 7
HISTORY_PARAMETER	TEMP	Selected from the list in reference table 3
HISTORY_START_PRES	75	Shallowest pressure of action.
HISTORY_STOP_PRES	105	Deepest pressure of action.
HISTORY_PREVIOUS_VALUE	1	This is the value of the quality flag on temperature readings before the change was made.
HISTORY_QCTEST	FillValue	This field does not apply

**Notes :**

1. The new QC flag of “4” (to indicate wrong values) would appear in the <param>\_QC field.

## 6. DAC-GDAC data-management

This chapter describes the data management organization between Argo DACs and GDACS.

### 6.1. Greylist files collection

Each DAC maintains a greylist that is submitted to the GDAC for updates. The DACs greylist are collected by the GDAC and merged into a global Argo greylist.

Greylist file collection from DAC to GDAC :

1. Query xxx\_greylist.csv file in each DAC submit directory; xxx must be identical to the DAC (eg : aoml, coriolis); otherwise the file is rejected.
2. Check the format of xxx\_greylist.csv . The whole file is rejected if the format check fails.
  - Floatid : valid Argo float id; the corresponding meta-data file must exist
  - Parameter : PSAL, TEMP, PRES or DOXY
  - Start date : YYYYMMDD valid, mandatory
  - End date : YYYYMMDD valid, fill value : ','
  - Flag : valid argo flag
  - Comment : free
  - DAC : valid DAC, mandatory
3. Remove all the floats of the DAC from the GDAC grey list and add the content of the submitted xxx\_greylist.csv file

Note : after each submission, a copy of the Argo greylist is stored in etc/greylist/ar\_greylist.txt\_YYYYMMDD

The global Argo greylist is sorted by DAC, PLATFORM\_CODE and START\_DATE in alphabetical order.

### 6.2. GDAC files removal

A DAC can ask the GDAC to remove individual profile, trajectory, technical or meta-data files. A "removal file" is submitted to GDAC which will perform the removals .

The "removal file" contains one line per file to remove.

"Removal file" collection from DAC to GDAC :

- Query xxx\_removal.txt file in each DAC submit directory; xxx must be identical to the DAC (eg : aoml, coriolis); otherwise the file is rejected.

- 
- Check the format of xxx\_removal.txt . The whole file is rejected is the format check fails.
    - File name : valid Argo file name; the corresponding meta-data file must exist for this DAC
  - Move all the named files from GDAC into a etc/removed directory
  - The removes files are kept for 3 months in the etc/removed directory and erased after that delay.