argo data management

Argo data management

09 July 2002

ar-um-02-01

USER'S MANUAL

Version 1.0

Argo data management

User's manual

Ref : ar-um-02-01

Version : 1.0

Date : 09/07/2002

Author : Thierry Carval / Ifremer

Reviewed by :

- Bob Keeley / MEDS
- Yasushi Takatsuki / JAMSTEC
- Stephen Loch / BODC
- Claudia Schmid / AOML
- Roger Goldsmith / WHOI

Table of contents

History

Version	Date	Comment
0.9	29/12/2001	Thierry Carval : creation of the document
0.9a	18/01/2001	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. Introduction

This document is the Argo data user's manual.

It contains the description of the formats and files produced by the data management group.

1.1. Argo program, data management context

Within a few years (2005), the Argo program will 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.

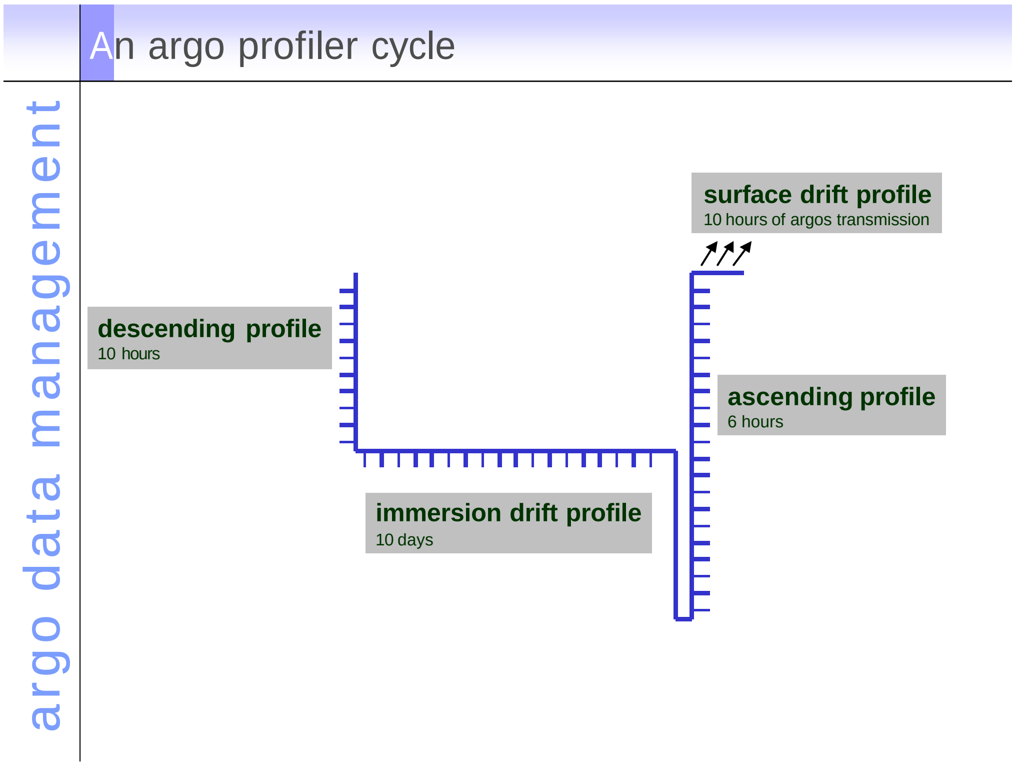
The Argo data formats are based on NetCDF because :

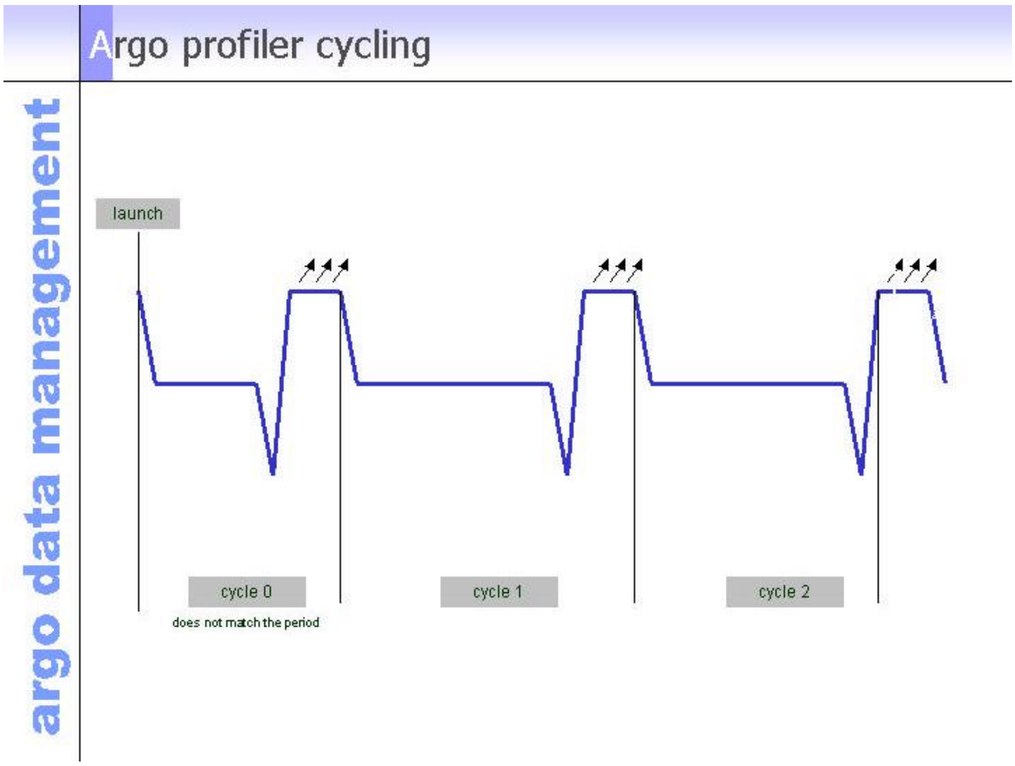
- It is a widely accepted data format by the user community,
- It is a self-describing format for which tools are widely available,
- It is a reliable and efficient format for data exchange.

1.2. Argo float data

An Argo float drifts for a number of years 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 defined pressure (eg : 1500 decibars),
- An subsurface drift (eg : 10 days),
- An ascending profile with measurements (eg : pressure, temperature, salinity),
- A surface drift with data transmission to a communication satellite.


An argo profiler cycle
argo data management
surface drift profile
10 hours of argos transmission
descending profile
10 hours
ascending profile
6 hours
immersion drift profile
10 days


Argo profiler cycling
argo data management
launch
cycle 0
does not match the period
cycle 1
cycle 2

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 3 months. These constitute the delayed mode data.

2. Data 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 explained in UTC time, universal time coordinates.

2.2. Argo profile file format 2.0

The current version of format is 2.0 .

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.

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

		• YYYY : year • MM : month • DD : day • HH : hour of the day (as 0 to 23) • MI : minutes (as 0 to 59) • SS : seconds (as 0 to 59) Date and time values are always in universal time coordinates (UTC). Examples : 20010105172834 : January 5th 2001 17:28:34 19971217000000 : December 17th 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";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference tables 4.1. Example : Argo profile
FORMAT_VERSION	char FORMAT_VERSION(STRING4); FORMAT_VERSION:comment = "File format version ";	File format version Example : «2.0»
HANDBOOK_VERSION	float HANDBOOK_VERSION; HANDBOOK_VERSION:comment = "Data handbook version";	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";	Date of reference for julian days. The recommended reference date time is "19500101000000" : January 1st 1950 00:00:00

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 : QA9IIIII";	WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : Q6900045
PROJECT_NAME	char PROJECT_NAME(N_PROF, STRING64); PROJECT_NAME:comment = "Name of the project";	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";	Name of the principal investigator in charge of the profiling float. Example : Yves Desaubies
STATION_PARAMETERS	char STATION_PARAMETERS(N_PROF, N_PARAM,STRING4); STATION_PARAMETERS:long_name = "List of available parameters for the station"; STATION_PARAMETERS:conventions = "GF3 code";	List of parameters contained in this profile. The parameter names are in the GF3 code list (see reference table 4.3). Examples : TEMP, PSAL, CNDC TEMP : temperature in degrees celsius PSAL : practical salinity in psu

		CNDC : conductvity in mhos/m
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 ";	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 = "GTSPP table";	Code for the data centre in charge of the float data management. The data centre codes are described in the reference table 4.4. Example : ME for MEDS
DATE_CREATION	char DATE_CREATION(DATE_TIME); DATE_ CREATION:comment = "Date of file creation "; DATE_ CREATION:conventions = "YYYYMMDDHHMISS";	Date and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS Example : 20011229161700 : December 29th 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 and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30th 2001 09 :05 :00
DC_REFERENCE	char DC_REFERENCE(N_PROF, STRING16); DC_REFERENCE:long_name = "Station unique identifier in data centre"; DC_REFERENCE:conventions = "Data centre convention";	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 = "OOPC table";	Degree of processing the data has passed through. The data state indicator is described in the reference table 4.6.
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";	Indicates if the profile contains real time or delayed mode data. R : real time data D : delayed mode data
INST_REFERENCE	char INST_REFERENCE(N_PROF, STRING64); INST_REFERENCE:long_name = "Instrument type"; INST_REFERENCE:conventions = "Brand, type, serial number";	References of the instrument : brand, type, serial number Example : APEX-SBE 259

WMO_INST_TYPE	char WMO_INST_TYPE(N_PROF, STRING4); WMO_INST_TYPE:long_name = "Coded instrument type"; WMO_INST_TYPE:conventions = "WMO code table 1770 – instrument type";	Instrument type from WMO code table 1770. A subset of WMO table 1770 is documented in the reference table 4.8. Example :
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[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
JULD_QC	char JULD_QC(N_PROF); JULD_QC:long_name = "Quality on Date and Time"; JULD_QC:conventions = "Q where Q =[0-9]"; JULD_QC:_FillValue = "0";	Quality flag on JULD date and time. The flag scale is described in the reference table 4.2. 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. The location of the profile is generally estimated after the end of the profile. The date of the profile and the date of the location of the profile can be therefore slightly different. 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.

[1] 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.

		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 = "Q where Q =[0-9]"; POSITION_QC:_FillValue = "0";	Quality flag on position. The flag on position is set according to (LATITUDE, LONGITUDE, JULD_LOCATION) quality. The flag scale is described in the reference table 4.2. Example : 1 : position seems correct.
POSITIONING_SYSTEM	char POSITIONING_SYSTEM(N_PROF, STRING8); POSITIONING_SYSTEM:long_name = "Positioning system"; POSITIONING_SYSTEM:conventions = "ARGOS or GPS";	Name of the system in charge of positioning the float locations. Examples : ARGOS, GPS
PROFILE_PRES_QC	char PROFILE_PRES_QC(N_PROF); PROFILE_PRES_QC:long_name = "Global quality flag of pressure profile"; PROFILE_PRES_QC:conventions = "Q where Q =[0-9]"; PROFILE_PRES_QC:_FillValue = "0";	Global quality flag on the pressure profile. The flag scale is described in the reference table 4.2. The flagging policy is described in the Argo data handbook. Example : 1 : the pressure profile seems correct.
PROFILE_<PARAM>_QC	char PROFILE_<PARAM>_QC(N_PROF); PROFILE_<PARAM>_QC:long_name = "Global quality flag of <PARAM> profile"; PROFILE_<PARAM>_QC:conventions = "Q where Q =[0-9]"; PROFILE_<PARAM>_QC:_FillValue = "0";	Global quality flag on the PARAM profile. PARAM is among the STATION_PARAMETERS. The flag scale is described in the reference table 4.2. The flagging policy is described in the Argo data handbook. Example : PROFILE_TEMP_QC = 1 : the temperature profile seems correct. PROFILE_PSAL_QC = 1 : the salinity profile seems correct.

2.2.4. Measurements for each profile

This section contains information on each level of each profile.

Each NetCDF variable in this section has a N_PROF (number of profiles), N_LEVELS (number of pressure levels) dimension.

Name	Definition	Comment
PRES	float PRES(N_PROF, N_LEVELS); PRES:long_name = "Pressure"; PRES:_FillValue = 99999.f; PRES:units = "decibar"; PRES:valid_min = 0.f; PRES:valid_max = 15000.f; PRES:C_format = "%7.1f"; PRES:FORTRAN_format="F7.1"; PRES:resolution=0.1f;	PRES contains N_LEVELS values of measured pressure. Unit : decibar Example : 17.0, 25.0, 35.0, ... ,1950.0, 1990.0, 2030.0
PRES_CORRECTED	float PRES_CORRECTED(N_PROF, N_LEVELS); PRES_CORRECTED:long_name = "Corrected pressure";	PRES_CORRECTED contains N_LEVELS values of pressure. PRES_CORRECTED is the best estimated value of the pressure.

	PRES_CORRECTED:_FillValue = 99999.f; PRES_CORRECTED:units = "decibar"; PRES_CORRECTED:valid_min = 0.f; PRES_CORRECTED:valid_max = 15000.f; PRES_CORRECTED:C_format = “%7.1f”; PRES_CORRECTED:FORTRAN_format=”F7.1”; PRES_CORRECTED:resolution=0.1f;	PRES_CORRECTED values comes from PRES values after calibration and after any corrections resulting from quality control procedures. If no calibration is available or if no other corrections are made, PRES_CORRECTED contains the same values that PRES. Unit : decibar Example : 17.0, 25.0, 35.0, ... ,1950.0, 1990.0, 2030.0
PRES_CORRECTED_QC	char PRES _CORRECTED_QC(N_PROF, N_LEVELS); PRES _CORRECTED_QC:long_name = "Quality on pressure"; PRES _CORRECTED_QC:conventions = "Q where Q =[0-9]"; PRES _CORRECTED_QC:_FillValue = "0";	Quality flag on corrected pressure. The flag scale is described in the reference table 4.2.
TEMP	float TEMP(N_PROF, N_LEVELS); TEMP:long_name = "Temperature in situ T90 scale"; TEMP:_FillValue = 99999.f; TEMP:units = "degree_Celsius"; TEMP:valid_min = -3.f; TEMP:valid_max = 40.f; TEMP:comment = "In situ measurement"; TEMP:C_format = “%9.3f”; TEMP:FORTRAN_format=”F9.3”; TEMP:resolution=0.001f;	TEMP contains N_LEVELS values of measured temperature. Each temperature has been measured at the corresponding pressure of PRES. Unit : degree celsius Example : 18.790, 18.490, 17.630, ... , 4.200, 3.960, 3.810
TEMP_CORRECTED	float TEMP_CORRECTED(N_PROF, N_LEVELS); TEMP_CORRECTED:long_name = "Corrected temperature"; TEMP_CORRECTED:_FillValue = 99999.f; TEMP_CORRECTED:units = "degree_Celsius"; TEMP_CORRECTED:valid_min = -3. f; TEMP_CORRECTED:valid_max = 40. f; TEMP_CORRECTED:comment = "Corrected value”; TEMP_CORRECTED:C_format = “%9.3f”; TEMP_CORRECTED:FORTRAN_format=”F9.3”; TEMP_CORRECTED:resolution=0.001f;	TEMP_CORRECTED contains N_LEVELS values of temperature. TEMP_CORRECTED is the best estimated value of the temperature. TEMP_CORRECTED values comes from TEMP values after calibration and after any corrections resulting from quality control procedures. If no calibration is available or if no other corrections are made, TEMP_CORRECTED contains the same values that TEMP. Unit : degree celsius Example : 18.792, 18.492, 17.632, ... , 4.203, 3.963, 3.813
TEMP_CORRECTED_QC	char TEMP_CORRECTED_QC(N_PROF, N_LEVELS); TEMP_CORRECTED_QC:long_name = "quality on temperature"; TEMP_QC:conventions = "Q where Q =[0-9]"; TEMP_CORRECTED_QC:_FillValue = "0";	Quality flag on corrected temperature. The flag scale is described in the reference table 4.2.
<PARAM>	float <PARAM>(N_PROF, N_LEVELS); <PARAM>:long_name = "parameter name from GF3 table"; <PARAM>:_FillValue = 99999.f; <PARAM>:units = "psu"; <PARAM>:valid_min = 0. f; <PARAM>:valid_max = 60. f; <PARAM>:comment = "In situ measurement, mandatory if salinity measured”; <PARAM>:C_format = “%9.3f”; <PARAM>:FORTRAN_format=”F9.3”; <PARAM>:resolution=0.001f;	PARAM represents another variable selected from the GF3 code list (see reference table 4.3). This field contains the analogous information as for TEMP above. Example of PARAM : • PSAL for practical salinity • CNDC for conductivity
<PARAM>_CORRECTED	float <PARAM>_CORRECTED(N_PROF, N_LEVELS); <PARAM>_CORRECTED:long_name = "Corrected parameter name from GF3 table”; <PARAM>_CORRECTED:_FillValue = 99999.f; <PARAM>_CORRECTED:units = "psu"; <PARAM>_CORRECTED:valid_min = 0. f;	PARAM represents another variable selected from the GF3 code list (see reference table 4.3). This field contains the analogous information as for TEMP_CORRECTED above. Example of PARAM: • PSAL for practical salinity • CNDC for conductivity

	<PARAM>_CORRECTED:valid_max = 60. f; <PARAM>_CORRECTED:comment =" Corrected value"; <PARAM>_CORRECTED :C_format = "%9.3f"; <PARAM>_CORRECTED :FORTRAN_format="F9.3"; <PARAM>_CORRECTED:resolution=0.001 f;	
<PARAM>_CORRECTED_QC	char <PARAM>_CORRECTED_QC(N_PROF, N_LEVELS); <PARAM>_CORRECTED_QC:long_name = "quality on parameter name from GF3 table "; <PARAM>_CORRECTED_QC:conventions = "Q where Q =[0-9]"; <PARAM>_CORRECTED_QC:_FillValue = "0";	PARAM represents another variable selected from the GF3 code list (see reference table 4.3). This field contains the analogous information as for TEMP_QC above. Example of PARAM: • PSAL for practical salinity • CNDC for conductivity

2.2.5. Calibration information for each profile

Calibrations are applied to parameters to create corrected 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. The best scientifically corrected parameter are stored in the corrected profile.

Example : in a temperature profile (TEMP), the calibrated values are stored in a corrected profile (TEMP_CORRECTED).

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.

Name	Definition	Comment
PARAMETER	char PARAMETER(N_PROF, N_CALIB, N_PARAM,STRING4); PARAMETER:long_name = "List of parameters with calibration information"; PARAMETER:conventions = "GF3 code";	Name of the calibrated parameter. 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";	Calibration equation applied to the parameter. Example : Tc = a1 * T + a0
SCIENTIFIC_CALIB_COEFFICIENT	char SCIENTIFIC_CALIB_COEFFICIENT(N_PROF, N_CALIB, N_PARAM,STRING256); SCIENTIFIC_CALIB_ COEFFICIENT:long_name = "Calibration coefficients for this equation";	Calibration coefficients for this equation. Example : a1=0.99997 , a0=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";	Comment about this calibration Example : The sensor is not stable
CALIBRATION_DATE	DATE_TIME(N_PROF N_CALIB, , N_PARAM, DATE_TIME)	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 4.7.

Name	Definition	Comment
HISTORY_INSTITUTION	char HISTORY_ INSTITUTION (N_HISTORY, N_PROF, STRING4); HISTORY_INSTITUTION:long_name = "Institution which performed action"; HISTORY_INSTITUTION:conventions = "GTSPP institution code";	Institution that performed the action. Institution codes are described in reference tables. Example : ME for MEDS
HISTORY_SOFTWARE	char HISTORY_ SOFTWARE(N_HISTORY, N_PROF, STRING4); HISTORY_ SOFTWARE:long_name = "Software which performed action"; HISTORY_ SOFTWARE:conventions = "Institution dependent";	Name of the software that performed the action. Example : QCA1
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";	Version of the software. Example : «1.0»
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
HISTORY_ACTION	char HISTORY_ACTION(N_HISTORY, N_PROF, STRING4); HISTORY_ACTION:long_name = "Action performed on data"; HISTORY_ACTION:conventions = "GTSPP (MEDS) action code";	Name of the action. The action codes are described in reference table 4.7. Example : QCF$ for QC failed
HISTORY_PARAMETER	char HISTORY_PARAMETER(N_HISTORY, N_PROF, STRING4); HISTORY_PARAMETER:long_name = "Station parameter action is performed on"; HISTORY_PARAMETER:conventions = "GF3 parameter code";	Name of the parameter on which the action is performed. Example : PSAL
HISTORY_START_PRES	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";	Start pressure the action is applied to. Example : 1500.0
HISTORY_STOP_PRES	float HISTORY_STOP_PRES(N_HISTORY, N_PROF); HISTORY_STOP_PRES:long_name =	Stop pressure the action is applied to. This should be greater than START_PRES.

	"Stop pressure action applied on"; HISTORY_STOP_PRES:_FillValue = 99999.f; HISTORY_STOP_PRES:units = "decibar";	Example : 1757.0
HISTORY_PREVIOUS_VALUE	float HISTORY_PREVIOUS_VALUE(N_HISTORY, N_PROF); 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_QCTEST	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$; tests failed when ACTION=QCF$";	This field records • the tests performed when ACTION is set to QCP$ (qc passed), • the test failed when ACTION is set to QCF$ (qc failed). Example : 0A (in hexadecimal form)

Information on HISTORY_QCTEST handling

Every time a record passes through a QC test, a record is created in HISTORY. The nature of QC test is stored in HISTORY_QCTEST.

The HISTORY_ACTION is set to QCP$ meaning QC performed.

QC performed only means that the test was performed, whether it failed or not.

If a test failed a second record is created in HISTORY. The nature of the failed test is stored in HISTORY_QCTEST.

The HISTORY_ACTION is set to QCF$ meaning QC failed.

For example, assume the case of a profile with a test failure. The history structure would be as follows.

Note that we have used a parameter code of RCRD to indicate that the result applies to the entire record, not just to any particular parameter.

History information Example	
HISTORY_INSTITUTION	MEDS
HISTORY_SOFTWARE	QCA1
HISTORY_SOFTWARE_RELEASE	1.0
HISTORY_ACTION	QCP$
HISTORY_PARAMETER	RCRD
HISTORY_START_PRES	99999.
HISTORY_STOP_PRES	99999.
HISTORY_PREVIOUS_VALUE	99999.
HISTORY_QCTEST	6A34CFE
HISTORY_INSTITUTION	MEDS
HISTORY_SOFTWARE	QCA1
HISTORY_SOFTWARE_RELEASE	1.0
HISTORY_ACTION	QCF$

HISTORY_PARAMETER	RCRD
HISTORY_START_PRES	99999.
HISTORY_STOP_PRES	99999.
HISTORY_PREVIOUS_VALUE	99999.
HISTORY_QCTEST	0A

2.3. Trajectory format 2.0

The current version of format is 2.0 .

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.

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 5th 2001 17:28:34 19971217000000 : December 17th 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

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";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference table 4.1. Example : Argo trajectory
FORMAT_VERSION	char FORMAT_VERSION(STRING4); FORMAT_VERSION:comment = "File format version ";	File format version Example : «2.0»
HANDBOOK_VERSION	float HANDBOOK_VERSION; HANDBOOK_VERSION:comment = "Data handbook version";	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";	Date of reference for julian days. The recommended reference date time is «19500101000000» : January 1st 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 : QA9IIIII";	WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : Q6900045
PROJECT_NAME	char PROJECT_NAME(STRING64); PROJECT_NAME:comment = "Name of the project";	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";	Name of the principal investigator in charge of the float. Example : Yves Desaubies
TRAJECTORY_PARAMETERS	char TRAJECTORY_PARAMETERS(N_PARAM,STRING4); TRAJECTORY _PARAMETERS:long_name = "List of available parameters for the station"; TRAJECTORY _PARAMETERS:conventions = "GF3 code";	List of parameters contained in this trajectory file. The parameter names are in the GF3 code list (see reference table 4.3). Examples : TEMP, PSAL, CNDC TEMP : temperature in degrees celsius PSAL : practical salinity in psu CNDC : conductvity in mhos/m

DATA_CENTRE	char DATA_CENTRE(STRING2); DATA_CENTRE:long_name = "Data centre in charge of float data processing"; DATA_CENTRE:conventions = "GTSPP table";	Code for the data centre in charge of the float data management. The data centre codes are described in the reference table 4.4. Example : ME for MEDS
DATE_CREATION	char DATE_CREATION(DATE_TIME); DATE_ CREATION:comment = "Date of file creation "; DATE_ CREATION:conventions = "YYYYMMDDHHMISS";	Date and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS Example : 20011229161700 : December 29th 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 and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30th 2001 09 :05 :00
DC_REFERENCE	char DC_REFERENCE(STRING16); DC_REFERENCE:long_name = "Trajectory unique identifier in data centre"; DC_REFERENCE:conventions = "Data centre convention";	Unique identifier of the trajectory in the data centre. Data centres may have different identifier schemes. DC_REFERENCE is therefore not always unique across data centres.
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 = "OOPC table";	Degree of processing the data has passed through. The data state indicator is described in the reference table 4.6.
INST_REFERENCE	char INST_REFERENCE(STRING64); INST_REFERENCE:long_name = "Instrument type"; INST_REFERENCE:conventions = "Brand, type, serial number";	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 = "WMO code table 1770 – instrument type";	Instrument type from WMO code table 1770. A subset of WMO table 1770 is documented in the reference table 4.8. Example : 831
POSITIONING_SYSTEM	char POSITIONING_SYSTEM(STRING8); POSITIONING_SYSTEM:long_name = "Positioning system"; POSITIONING_SYSTEM:conventions = "ARGOS or GPS";	Name of the system used to derive the float locations. Examples : ARGOS, GPS

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.

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";	Indicates if the profile contains real time or delayed mode data. R : real time data D : delayed mode data
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 = "Q where Q =[0-9]"; JULD_QC:_FillValue = "0";	Quality flag on JULD date and time. The flag scale is described in the reference table 4.2. Example : 1 : the date and time seems correct.
LATITUDE	double LATITUDE(N_MEASUREMENT); LATITUDE:long_name = "Latitude of each location"; LATITUDE:units = "degree_north"; LATITUDE:_FillValue = -99999.; LATITUDE:valid_min = -90.; LATITUDE:valid_max = 90.;	Latitude of the location (or measurement). Unit : degree north Example : 44.4991 for 44° 29' 56.76'' N
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 = "Argos location classes";	Position accuracy received from the positioning system. The location classes from Argos are described in the refrence tables section. 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 = "Q where Q =[0-9]"; POSITION_QC:_FillValue = "0";	Quality flag on position. The flag on position is set according to (LATITUDE, LONGITUDE, JULD_LOCATION) quality. The flag scale is described in the reference table 4.2. Example : 1 : position seems correct.
CYCLE_NUMBER	int CYCLE_NUMBER(N_MEASUREMENT);	Cycle number of the float for this 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;	For one cycle number, there are usually several locations/measurement received. Example : 17 for measurements performed during the 17th cycle of the float.
PRES	float PRES(N_MEASUREMENT); PRES:long_name = "Pressure"; PRES:_FillValue = 99999.f; PRES:units = "decibar"; PRES:valid_min = 0.; PRES:valid_max = 15000.f; PRES:C_format = “%7.1f”; PRES:FORTRAN_format=”F7.1”; PRES:resolution=0.1f;	Pressure value of a measurement. The measurements are usually performed during a surface drift where the pressure value is close to 0. But, some floats may also perform measurements during the subsurface drift where the pressure value is different from 0. Unit : decibar Example : 0.0
PRES_CORRECTED	float PRES_CORRECTED(N_MEASUREMENT); PRES_CORRECTED:long_name = "Corrected pressure"; PRES_CORRECTED:_FillValue = 99999.f; PRES_CORRECTED:units = "decibar"; PRES_CORRECTED:valid_min = 0.f; PRES_CORRECTED:valid_max = 15000.f; PRES_CORRECTED:C_format = “%7.1f”; PRES_CORRECTED:FORTRAN_format=”F 7.1”; PRES_CORRECTED:resolution=0.1f;	PRES_CORRECTED contains the best estimated value of the pressure for a measurement. PRES_CORRECTED values comes from PRES values after calibration and after any corrections resulting from quality control procedures. If no calibration is available or if no other corrections are made, PRES_CORRECTED contains the same values that PRES. Unit : decibar Example : 0.0
PRES_CORRECTED_QC	char PRES _CORRECTED_QC(N_ MEASUREMENT); PRES _CORRECTED_QC:long_name = "Quality on pressure"; PRES _CORRECTED_QC:conventions = "Q where Q =[0-9]"; PRES _CORRECTED_QC:_FillValue = "0";	Quality flag on corrected pressure. The flag scale is described in the reference table 4.2.
TEMP	float TEMP(N_ MEASUREMENT); TEMP:long_name = "Temperature in situ T90 scale"; TEMP:_FillValue = 99999.f; TEMP:units = "degree_Celsius"; TEMP:valid_min = -3.f; TEMP:valid_max = 40.f; TEMP:comment = "In situ measurement"; TEMP:C_format = “%9.3f”; TEMP:FORTRAN_format=”F9.3”; TEMP:resolution=0.001f;	TEMP contains a measured temperature . Unit : degree celsius Example : 18.790
TEMP_CORRECTED	float TEMP_CORRECTED(N_ MEASUREMENT); TEMP_CORRECTED:long_name = "Corrected temperature"; TEMP_CORRECTED:_FillValue = 99999.f; TEMP_CORRECTED:units = "degree_Celsius"; TEMP_CORRECTED:valid_min = -3. f; TEMP_CORRECTED:valid_max = 40. f; TEMP_CORRECTED:comment = "Corrected value”; TEMP_CORRECTED:C_format = “%9.3f”; TEMP_CORRECTED:FORTRAN_format=”F 9.3”; TEMP_CORRECTED:resolution=0.001f;	TEMP_CORRECTED contains the best estimated value of the temperature. TEMP_CORRECTED values comes from TEMP values after calibration and after any corrections resulting from quality control procedures. If no calibration is available or if no other corrections are made, TEMP_CORRECTED contains the same values that TEMP. Unit : degree celsius Example : 18.792
TEMP_CORRECTED_QC	char TEMP_CORRECTED_QC(N_ MEASUREMENT); TEMP_CORRECTED_QC:long_name = "quality on temperature"; TEMP_CORRECTED_QC:conventions = "Q where Q =[0-9]"; TEMP_CORRECTED_QC:_FillValue = "0";	Quality flag on corrected temperature. The flag scale is described in the reference table 4.2.
<PARAM>	float <PARAM>(N_ MEASUREMENT); <PARAM>:long_name = "Practical salinity, sal78"; <PARAM>:_FillValue = 99999.f;	Equivalent to TEMP for a PARAM among GF3 codes. Example of PARAM :

	<PARAM>:units = "psu"; <PARAM>:valid_min = 0. f; <PARAM>:valid_max = 60. f; <PARAM>:comment = "In situ measurement, mandatory if salinity measured"; <PARAM>:C_format = "%9.3f"; <PARAM>:FORTRAN_format="F9.3"; <PARAM>:resolution=0.001f;	• PSAL for practical salinity • CNDC for conductivity
<PARAM>_CORRECTED	float <PARAM>_CORRECTED(N_MEASUREMENT); <PARAM>_CORRECTED:long_name = "Corrected parameter name"; <PARAM>_CORRECTED:_FillValue = 99999.f; <PARAM>_CORRECTED:units = "psu"; <PARAM>_CORRECTED:valid_min = 0. f; <PARAM>_CORRECTED:valid_max = 60. f; <PARAM>_CORRECTED:comment =" Corrected value"; <PARAM>_CORRECTED :C_format = "%9.3f"; <PARAM>_CORRECTED :FORTRAN_format="F9.3"; <PARAM>_CORRECTED:resolution=0.001 f;	Equivalent to TEMP_CORRECTED for a PARAM among GF3 codes. Example of PARAM: • PSAL for practical salinity • CNDC for conductivity
<PARAM>_CORRECTED_QC	char <PARAM>_CORRECTED_QC(N_MEASUREMENT); <PARAM>_CORRECTED_QC:long_name = "quality on parameter name"; <PARAM>_CORRECTED_QC:conventions = "Q where Q =[0-9]"; <PARAM>_CORRECTED_QC:_FillValue = "0";	Equivalent to TEMP_QC for a PARAM among GF3 codes. Example of PARAM: • PSAL for practical salinity • CNDC for conductivity

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.f;	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";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float
JULD_ASCENT_END	double JULD_ASCENT_END(N_CYCLE); JULD_ASCENT_END:long_name = "End date of the ascending profile";	Julian day (UTC) of the end of the ascending profile. Example :

	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.f;	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";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float
JULD_DESCENT_STA RT	double 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.f;	Julian day (UTC) of the beginning of the descending profile. Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_DESCENT_STA RT _STATUS	char JULD_DESCENT_START_STATUS(N_CYCLE); JULD_DESCENT_START_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float
JULD_DESCENT_EN D	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.f;	Julian day (UTC) of the end of the descending profile. Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_DESCENT_EN D _STATUS	char JULD_DESCENT_END_STATUS(N_CYCLE); JULD_DESCENT_END_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float
JULD_START_TRAN SMISSION	double JULD_START_TRANSMISSION(N_CYCLE); JULD_START_TRANSMISSION:long_name = "Start date of transmssion"; 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.f;	Julian day (UTC) of the beginning of data transmission. Example : 18833.8013889885 : July 25 2001 19:14:00
JULD_ START_TRANSMISSI ON_STATUS	char JULD_ START_TRANSMISSION_STATUS(N_CYCLE); JULD_START_TRANSMISSION_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";	0 : date comes from the float meta data 1 : date is estimated 2 : date is transmitted by the float
GROUNDED	char GROUNDED(N_CYCLE); GROUNDED:long_name = "Did the profiler touch the ground for that cycle”; GROUNDED:conventions = "Y,N,U";	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

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_MEASUREMENT, N_HISTORY, STRING4); HISTORY_INSTITUTION:long_name = "Institution which performed action"; HISTORY_INSTITUTION:conventions = "GTSPP institution code";	Institution that performed the action. Institution codes are described in reference table 4.4. Example : ME for MEDS
HISTORY_SOFTWARE	char HISTORY_ SOFTWARE(N_MEASUREMENT, N_HISTORY, STRING4); HISTORY_ SOFTWARE:long_name = "Software which performed action"; HISTORY_ SOFTWARE:conventions = "Institution dependent";	Name of the software that performed the action. Example : QCA1
HISTORY_SOFTWARE_RELEASE	char HISTORY_ SOFTWARE_RELEASE(N_MEASUREMENT, N_HISTORY, STRING4); HISTORY_SOFTWARE_RELEASE:long_name = "Version/release of software which performed action"; HISTORY_SOFTWARE_RELEASE:conventions = "Institution dependent";	Version of the software. Example : «1.0»
HISTORY_DATE	char HISTORY_DATE(N_MEASUREMENT, N_HISTORY, DATE_TIME); HISTORY_DATE:long_name = "Date the history record was created"; HISTORY_DATE:conventions = "YYYYMMDDHHMISS";	Date and time (UTC) of the action. Example : 20011217160057
HISTORY_ACTION	char HISTORY_ACTION(N_MEASUREMENT, N_HISTORY, STRING4); HISTORY_ACTION:long_name = "Action performed on data"; HISTORY_ACTION:conventions = "GTSPP (MEDS) action code";	Name of the action. The action codes are described in reference table 4.7. Example : QCF$ for quality control failed
HISTORY_PARAMETER	char HISTORY_PARAMETER(N_MEASUREMENT, N_HISTORY, STRING4); HISTORY_PARAMETER:long_name = "Station parameter action is performed on"; HISTORY_PARAMETER:conventions = "GF3 parameter code";	Name of the parameter on which the action is performed. Example : PSAL
HISTORY_PREVIOUS_VALUE	float HISTORY_PREVIOUS_VALUE(N_MEASUREMENT, N_HISTORY); HISTORY_PREVIOUS_VALUE:long_na	Parameter or flag of the previous value before action.

	me = "Parameter/Flag previous value before action"; HISTORY_PREVIOUS_VALUE:_FillValue = 99999.f;	Example : 2 (probably good) for a flag that was changed to 1 (good)
HISTORY_QCTEST	char HISTORY_QCTEST(N_MEASUREMENT, N_HISTORY, STRING16); HISTORY_QCTEST:long_name = "Documentation of tests performed, tests failed (in hex form)"; HISTORY_QCTEST:conventions = "Write tests performed when ACTION=QCP$; tests failed when ACTION=QCF$";	This field records • the tests performed when ACTION is set to QCP$ (qc passed), • the test failed when ACTION is set to QCF$ (qc failed). Example : 0A (in hexadecimal form)

2.4. Meta-data format 2.0

The current version of format is 2.0 .

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

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 5th 2001 17:28:34 19971217000000 : December 17th 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 5th 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";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference table 4.1. Example : Argo meta-data
FORMAT_VERSION	char FORMAT_VERSION(STRING4); FORMAT_VERSION:comment = "File format version ";	File format version Example : «2.0»
HANDBOOK_VERSION	float HANDBOOK_VERSION; HANDBOOK_VERSION:comment = "Data handbook version";	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 and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS Example : 20011229161700 : December 29th 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 and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30th 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 = "Extended WMO float identifier : QA9IIIII";	WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : Q6900045
PTT	char PTT (STRING256); PTT:long_name = "Transmission identifier (ARGOS, ORBCOMM, etc.)"; // Comma separated list for multi-beacon transmission	Transmission identifier of the float. 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:conventions = "ARGOS or ORBCOMM";	Name of the telecommunication system. Example : ARGOS, ORBCOMM, IRIDIUM
TRANS_SYSTEM_ID	char TRANS_SYSTEM_ID(STRING32); TRANS_SYSTEM_ID:long_name = "The program identifier used by the transmission system";	Program identifier of the telecommunication subscription. 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";	Frequency of transmission from the float. Unit : hertz Example : ...
TRANS_REPETITION	float TRANS_REPETITION; TRANS_REPETITION:long_name = "The repetition rate of transmission from the float"; TRANS_ REPETITION:units = "second";	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:conventions = "ARGOS or GPS";	Position system. 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 ";	Model of the float. Example : APEX-SBE
PLATFORM_MAKER	char PLATFORM_MAKER (STRING256); PLATFORM_MAKER:long_name = "The name of the manufacturer ";	Name of the manufacturer. Example : Webb research
PLATFORM_SERIAL_NO	char PLATFORM_SERIAL_NO(STRING16); PLATFORM_SERIAL_NO:long_name = "The serial number of the platform ";	Serial number of the float. Example : APEX-SBE 259

DIRECTION	char DIRECTION; DIRECTION:long_name = "Direction of the profiles"; DIRECTION:conventions = "A: ascending profiles, B: descending and ascending profiles ";	Direction of the profiles of the float. A : ascending profiles only B : descending and ascending profiles
PROJECT_NAME	char PROJECT_NAME(STRING32); PROJECT_NAME:long_name = "The program under which the float was deployed";	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 = "GTSPP table";	Code of the data centre in charge of the float data management. The data centre codes are described in the reference table 4.4. Example : ME for MEDS
PI_NAME	char PI_NAME (STRING64); PI_NAME:comment = "Name of the principal investigator";	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.";	This field describes any anomaly or problem the float may have had. Example : “the immersion drift is not stable.”

2.4.4. Float deployment information

Name	Definition	Comment
LAUNCH_DATE	char LAUNCH_DATE(DATE_TIME); LAUNCH_DATE:long_name = "Date (UTC) of the deployment"; LAUNCH_DATE:conventions = "YYYYMMDDHHMISS";	Date and time (UTC) of launch of the float. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30th 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 = "Q where Q =[0-9]"; LAUNCH_QC:_FillValue = "0";	Quality flag on launch date, time and location. The flag scale is described in the reference table 4.2. 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";	Date and time (UTC) of the first descent of the float. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30th 2001 06 :05 :00
START_DATE_QC	char START_DATE_QC; START_DATE_QC:long_name = "Quality on start date"; START_DATE_QC:conventions = "Q where Q =[0-9]"; START_DATE_QC:_FillValue = "0";	Quality flag on start date. The flag scale is described in the reference table 4.2. Example : 1 : start date seems correct.
DEPLOY_PLATFORM	char DEPLOY_PLATFORM(STRING32); DEPLOY_PLATFORM:long_name = "Identifier of the deployment platform";	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";	Identifier of the mission used to deploy the platform. Example : POMME2
DEPLOY_AVAILABLE_PROFILE_ID	char DEPLOY_AVAILABLE_PROFILE_ID(STRING256); DEPLOY_AVALAIBLE_PROFILE_ID:long_name = "Identifier of stations used to verify the first profile";	Identifier of CTD or XBT stations used to verify the first profile. Example : 58776, 58777

2.4.5. Float sensor information

This section contains information about the sensors of the profiler.

Name	Definition	Comment
SENSOR	char SENSOR(N_PARAM,STRING4); SENSOR:long_name = "List of sensors on the float "; SENSOR:conventions = "GF3 code list in (PRES, TEMP, CNDC)";	Parameters measured by sensors of the float. The parameter names are in GF3 code list (see reference table 4.3). Examples : TEMP, PSAL, CNDC TEMP : temperature in celsius PSAL : practical salinity in psu CNDC : conductvity in mhos/m
SENSOR_MAKER	char SENSOR_MAKER(N_PARAM,STRING256); SENSOR_MAKER:long_name = "The name of the manufacturer ";	Name of the manufacturer of the sensor. Example : SEABIRD
SENSOR_MODEL	char SENSOR_MODEL (N_PARAM,STRING256); SENSOR_MODEL:long_name = "Type of sensor ";	Model of sensor. Example : Salinity sensor
SENSOR_SERIAL_NO	char SENSOR_SERIAL_NO(N_PARAM,STRING16); SENSOR_SERIAL_NO:long_name = "The serial number of the sensor ";	Serial number of the sensor. Example : SBE211
SENSOR_UNITS	char SENSOR_UNITS(N_PARAM, STRING16); SENSOR_UNITS:long_name = "The units of accuracy and resolution of the sensor";	Units of accuracy of the sensor. Example : psu
SENSOR_ACCURACY	float SENSOR_ACCURACY(N_PARAM); SENSOR_ ACCURACY:long_name = "The accuracy of the sensor";	Accuracy of the sensor. Example : 0.005
SENSOR_RESOLUTION	float SENSOR_RESOLUTION(N_PARAM); SENSOR_RESOLUTION:long_name = "The resolution of the sensor";	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 scientific calibration, based on a data analysis is described in the profile format.

Name	Definition	Comment
PARAMETER	char PARAMETER(N_PARAM,STRING4); PARAMETER:long_name = "List of parameters with calibration information"; PARAMETER:conventions = "GF3 codes";	Parameters measured on this float. The parameter names are in GF3 code list (see reference table 4.3). Examples : TEMP, PSAL, CNDC TEMP : temperature in celsius PSAL : practical salinity in psu

		CNDC : conductvity in mhos/m
PREDEPLOYMENT_CALIB_EQUATION	char PREDEPLOYMENT_CALIB_EQUATION(N_PARAM,STRING256); PREDEPLOYMENT_CALIB_EQUATION:long_name = "Calibration equation for this parameter";	Calibration equation for this parameter. Example : Tc = a1 * T + a0
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 : a1=0.99997 , a0=0.0021
PREDEPLOYMENT_CALIB_COMMENT	char PREDEPLOYMENT_CALIB_COMMENT(N_PARAM,STRING256); PREDEPLOYMENT_CALIB_ COMMENT:long_name = "Comment applying to this parameter calibration";	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 5th 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 days"; 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 day Example : 10.0 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 days"; 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 : 9.25 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.

2.5. Technical information format 2.0

The current version of format is 2.0 .

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 an other. 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.

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 and time values are always in universal time coordinates (UTC). Date_time convention is : YYYYMMDDHHMISS • YYYY : year • MM : month • DD : day • HH : hour of the day • MI : minutes • SS : seconds Examples : 20010105172834 : January 5th 2001 17:28:34 19971217000000 : December 17th 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_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.

2.5.2. 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 : QA9IIIII";	WMO float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : Q6900045
DATA_TYPE	char DATA_TYPE(STRING32); DATA_TYPE:comment = "Data type";	This field contains the type of data contained in the file. The list of acceptable data types is in the reference table 4.3. Example : Technical data
FORMAT_VERSION	char FORMAT_VERSION(STRING4); FORMAT_VERSION:comment = "File format version ";	File format version Example : «2.0»
HANDBOOK_VERSION	float HANDBOOK_VERSION; HANDBOOK_VERSION:comment = "Data handbook version";	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 = "GTSPP table";	Code of the data centre in charge of the float data management. The data centre codes are described in the reference table 4.4. Example : ME for MEDS
DATE_CREATION	char DATE_CREATION(DATE_TIME); DATE_ CREATION:comment = "Date of file creation "; DATE_ CREATION:conventions = "YYYYMMDDHHMISS";	Date and time (UTC) of creation of this file. Format : YYYYMMDDHHMISS Example : 20011229161700 : December 29th 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 and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30th 2001 09 :05 :00

2.5.3. 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.

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";	Name of the technical parameter. Example : "Battery voltage"
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";	Value of the technical parameter. Example : "11.5"

3. Reference tables

3.1. Argo data type

The following table contains the list of acceptable contents for DATA_TYPE field.

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

3.2. Quality flags

Value	Description
0	no quality control (QC) was performed
1	QC was performed; good data
2	Probably good data but value may be inconsistent with statistics (differ from climatology)
3	Probably bad data (spike, gradient, ... if other tests passed)
4	Bad data, impossible value (out of scale, vertical instability, constant profile, ...)
5	Value modified during quality control
6	Not used
7	Not used
8	Interpolated value
9	Missing value

3.3. Physical parameter codes (GF3)

The following table describes the GF3 physical oceanographic codes.

The global list of GF3 physical parameter codes is available at :

http://www.meds-sdmm.dfo-mpo.gc.ca/jgofs/jgofs_cd/refrnce/format/PARMCODE.TXT

Code	Parameter name	Units
CAST	HYDROGRAPHIC CAST TYPE	code
CHLR	CHLORINITY (PARTS/THOUSAND)	g/kg
CHLS	CHLOROSITY	kg/m**3
CNDC	ELECTRICAL CONDUCTIVITY	mhos/m
DENS	SEA DENSITY	kg/m**3
DOXY	DISSOLVED OXYGEN	mmol/m**3

DRDP	DEPTH OF DROGUE	metres
EWCM	EAST (MAGNETIC) COMPONENT OF CURRENT	m/s
EWCT	EAST (TRUE) COMPONENT OF CURRENT	m/s
HCDM	DIRECTION TO WHICH HORIZONTAL CURRENT IS FLOWING (relative to Magnetic North)	degrees
HCDT	DIRECTION TO WHICH HORIZONTAL CURRENT IS FLOWING (relative to True North)	degrees
HCSP	HORIZONTAL CURRENT SPEED	m/s
ICEF	FLAG FOR ICE IN VICINITY OF HYDROGRAPHIC OBSERVATIONS	code
LVLS	SELECTION OF DEPTH LEVELS	code
NSCM	NORTH (MAGNETIC) COMPONENT OF CURRENT	m/s
NSCT	NORTH (TRUE) COMPONENT OF CURRENT	m/s
POTM	POTENTIAL TEMPERATURE	degrees C
PRES	SEA PRESSURE (sea surface = 0)	decibars
PRHB	PROBE HIT BOTTOM INDICATOR	code
PSAL	PRACTICAL SALINITY	----
RELP	RELATIVE TOTAL PRESSURE	decibars
SALD	SALINITY UNITS FLAG	code
SCDT	DIRECTION TO WHICH SEA SURFACE CURRENT IS FLOWING (relative to True North)	degrees
SCSP	SEA SURFACE CURRENT SPEED	m/s
SECC	SECCHI DISC DEPTH	metres
SLEV	OBSERVED SEA LEVEL	metres
SSAL	SALINITY (PRE-1978 DEFINITION) (PARTS/THOUSAND)	g/kg
SSPS	SEA SURFACE PRACTICAL SALINITY	----
SSSL	SEA SURFACE SALINITY (PRE-1978 DEFINITION) (PARTS/THOUSAND)	g/kg
SSTP	SEA SURFACE TEMPERATURE	degrees C
SVEL	SOUND VELOCITY	m/s
TDFL	TRACE DIRECTION FLAG	code
TEMP	SEA TEMPERATURE	degrees C
TGRD	SEA TEMPERATURE GRADIENT	degC/m
TOTP	TOTAL PRESSURE (atmosphere+sea pressure)	decibars
USAL	UNDEFINED SALINITY (Prac. Salin or parts/thousand) - see also SALD7AAN	----
WCLR	WATER COLOUR (FOREL-ULE SCALE)	code

3.4. Data centres and institutions codes (MEDS)

Data centres and institutions	
ME	MEDS, Canada
AO	AOML, USA
SI	SIO, Scripps, USA
CS	CSIRO, Australia
GE	BSH, Germany
JA	JMA, Japan
JM	Jamstec, Japan
IF	Ifremer, France

RU	Russia
VL	Far Eastern Regional Hydrometeorological Research Institute of Vladivostock, Russia
SP	Spain
CI	Institute of Ocean Sciences, Canada

3.5. Location classes (Argos)

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

3.6. Data state indicators (OOPC)

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 thermodynamic 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.	**-** Some reductions or subsampling has been performed, but the original record is available. **+** Geospatial and temporal properties are checked. Geophysical values are validated. If not validated, this is clearly indicated.
B	Data have been scrutinized and evaluated against a defined and documented set of measures. The process is often automated	**-** Measures are completely automated, or documentation is not widely available.

	(i.e. has no human intervention) and the measures are published and widely available.	**+** The measures have been tested on independent data sets for completeness and robustness and are widely accepted.
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.	**-** Procedures are not published or widely available. Procedures have not undergone full scrutiny and testing. **+** Data are fully quality controlled, peer reviewed and are widely accepted as valid. Documentation is complete and widely available.

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 iintercomparisons 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. History codes (GTSPP)

Code	Meaning
CF	Change a quality flag
CR	Create record
CT	Constant temperature, hit bottom. The temperature profile exhibits constant temperature which is considered to be erroneous when compared to neighbouring profiles and/or climatology and/or known characteristics of the region. The constant temperatures could occur over all of the profile or part of it. (The reasons could be sensor failure, depth calculation error, nose falling off the XBT, XBT sitting on the bottom, etc.)
CV	Change value
DC	Station was checked by duplicate checking software
ED	Edit a parameter value
FS	Fine structure error: leakage, PET fault, cusping, sticking bit. The temperature profile exhibits erroneous fine structure when compared to neighbouring profiles and/or climatology and/or known characteristics of the region. (The reasons could be signal leakage, XBT recording system failures (sticking bit, cusping, PET Fault, etc), complete instrument failure, etc)
IP	This history group operates on the complete input record
ML	Mixed layer error: bowing. The temperature profile exhibits erroneous features (such as false inversions) in the mixed layer when compared to neighbouring profiles and/or climatology and/or known characteristics of the region. (The reasons could be XBT bowing problem, instrument drift, etc).
NG	No good trace
PE	Position error. Profile position has been erroneously encoded. Corrected if possible.
QC	Quality Control
SF	Surface feature error: surface anomaly, chopped surface value. The temperature exhibits erroneous features at the surface when compared to neighbouring profiles and known characteristics of the region. (The reasons could be XBT start-up transients, general instrument equilibration problems, sensors recording prematurely before entering water, etc)
SV	Set a value
TE	Time error. Profile date/time has been erroneously encoded. Corrected if possible.
TG	Temperature gradient error: insulation penetration, spikes, high frequency noise, wire break, modulo 10 spikes. The temperature profile exhibits erroneous temperature gradients when compared to neighbouring profiles and/or climatology and/or known characteristics of the region. (The reasons could be spiking, interference, XBT wire break, modulo 10 problem, XBT wire insulation penetration, etc)
TI	Temperature inversion error: wire stretch. The temperature profile exhibits erroneous temperature inversions when compared to neighbouring profiles and/or climatology and/or known characteristics of the region. (The reasons could be XBT wire stretch, sensor drift, encoding error, etc).

TO	Temperature/depth offset. The temperature profile exhibits erroneous temperature/depth offsets compared to neighbouring profiles and/or climatology and/or known characteristics of the region. The offsets can occur at depth, or over sections of the profile, or over the complete profile (the reason could be instrument drift/sensor failure, encoding error, XBT fall rate error, XBT start-of-descent timing error, etc).
UP	Station passed through the update program
UR	Under resolved. Temperature data is encoded at standard depths/levels and can't be used to reconstitute a profile accurately.

3.8. WMO instrument types – table 1770

The WMO instrument types are maintained on the following web site :

http://www.meds-sdmm.dfo-mpo.gc.ca/meds/Prog_Int/J-COMM/CODES/wmotable_e.htm#ct1770

Code Figure	Instrument
831	P-Alace float
840	Provor T, no conductivity
841	Provor CT, Seabird c. sensor
842	Provor CT, FSI c. sensor
845	Webb Research, no conductivity
846	Webb Research, Seabird sensor
847	Webb Research, FSI sensor
850	Solo, no conductivity
851	Solo, Seabird c. sensor
852	Solo, FSI c. sensor

3.9. Postioning system

Code	Description
ARGOS	Argos positioning system
GPS	GPS positioning system

3.10. Transmission system

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

4. Data access

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

The internet address are :

http://www.ifremer.fr/coriolis/cdc/argo

http://www.usgodae.fnmoc.navy.mil

The 2 GDACs offer the same data set (the Argo data are mirrored in real time).