Argo data management

ar-um-02-01

ARGO USER’S MANUAL

Version 3.0

December 4th 2012

DRAFT





Argo data management

User’s manual

Ref: ar-um-02-01  
Ref ifremer : cor-do/dti-mut/02-084

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, Anh Tran / MEDS

**Table of contents**

[1 Introduction 10](#_Toc320976504)

[1.1 Notice on file format change transition 10](#_Toc320976505)

[1.2 User Obligations 10](#_Toc320976506)

[1.3 Disclaimer 10](#_Toc320976507)

[1.4 Further information sources and contact information 10](#_Toc320976508)

[1.5 Argo program, data management context 11](#_Toc320976509)

[1.6 Argo float cycles 11](#_Toc320976510)

[1.7 Real-time and Delayed mode data 12](#_Toc320976511)

[2 Formats description 13](#_Toc320976512)

[2.1 Overview of the formats 13](#_Toc320976513)

[2.2 Profile format version 2.3 14](#_Toc320976514)

[2.2.1 Global attributes, dimensions and definitions 14](#_Toc320976515)

[2.2.1.1 Global attributes 14](#_Toc320976516)

[2.2.1.2 Dimensions 15](#_Toc320976517)

[2.2.2 General information on the profile file 16](#_Toc320976518)

[2.2.3 General information for each profile 16](#_Toc320976519)

[2.2.4 Measurements for each profile 20](#_Toc320976520)

[2.2.5 Calibration information for each profile 23](#_Toc320976521)

[2.2.6 History information for each profile 24](#_Toc320976522)

[2.3 Trajectory format version 2.3 26](#_Toc320976523)

[2.3.1 Global attributes, dimensions and definitions 26](#_Toc320976524)

[2.3.1.1 Global attributes 26](#_Toc320976525)

[2.3.1.2 Dimensions 26](#_Toc320976526)

[2.3.2 General information on the trajectory file 27](#_Toc320976527)

[2.3.3 General information on the float 27](#_Toc320976528)

[2.3.4 Locations and measurements from the float 29](#_Toc320976529)

[2.3.5 Cycle information from the float 31](#_Toc320976530)

[2.3.6 History information 34](#_Toc320976531)

[2.4 Metadata format version 2.4 36](#_Toc320976532)

[2.4.1 Global attributes, dimensions and definitions 36](#_Toc320976533)

[2.4.1.1 Global attributes 36](#_Toc320976534)

[2.4.1.2 Dimensions and definitions 36](#_Toc320976535)

[2.4.2 General information on the meta-data file 38](#_Toc320976536)

[2.4.3 Float characteristics 39](#_Toc320976537)

[2.4.4 Float deployment and mission information 41](#_Toc320976538)

[2.4.5 Configuration parameters 42](#_Toc320976539)

[2.4.5.1 Note on floats with multiple configurations 44](#_Toc320976540)

[2.4.5.2 Determining which mission applies to a particular float cycle 44](#_Toc320976541)

[2.4.6 Float sensor information 46](#_Toc320976542)

[2.4.7 Float calibration information 46](#_Toc320976543)

[2.4.8 Mandatory meta-data parameters 47](#_Toc320976544)

[2.5 Technical information format version 2.4 49](#_Toc320976545)

[2.5.1 Global attributes, dimensions and definitions 49](#_Toc320976546)

[2.5.1.1 Global attributes 49](#_Toc320976547)

[2.5.1.2 Dimensions and definitions 49](#_Toc320976548)

[2.5.2 General information on the technical data file 50](#_Toc320976549)

[2.5.3 Technical data 50](#_Toc320976550)

[2.6 GDAC FTP directory file format 52](#_Toc320976551)

[2.6.1 Profile directory file format 52](#_Toc320976552)

[2.6.2 Profile directory file format version 2.1 53](#_Toc320976553)

[2.6.3 Trajectory directory format 54](#_Toc320976554)

[2.6.4 Meta-data directory format 55](#_Toc320976555)

[3 Reference tables 57](#_Toc320976556)

[3.1 Reference table 1: data type 57](#_Toc320976557)

[3.2 Reference table 2: Argo quality control flag scale 58](#_Toc320976558)

[3.2.1 Reference table 2: measurement flag scale 58](#_Toc320976559)

[3.2.2 Reference table 2a: profile quality flag 59](#_Toc320976560)

[3.3 Reference table 3: parameter code table 60](#_Toc320976561)

[3.3.1 Parameters from duplicate sensors 61](#_Toc320976562)

[3.3.2 Oxygen related parameters 61](#_Toc320976563)

[3.4 Reference table 4: data centres and institutions codes 62](#_Toc320976564)

[3.5 Reference table 5: location classes 63](#_Toc320976565)

[3.6 Reference table 6: data state indicators 64](#_Toc320976566)

[3.7 Reference table 7: history action codes 66](#_Toc320976567)

[3.8 Reference table 8: instrument types 66](#_Toc320976568)

[3.9 Reference table 9: positioning system 66](#_Toc320976569)

[3.10 Reference table 10: transmission system 67](#_Toc320976570)

[3.11 Reference table 11: QC test binary IDs 67](#_Toc320976571)

[3.12 Reference table 12: history steps codes 68](#_Toc320976572)

[3.13 Reference table 13: ocean codes 69](#_Toc320976573)

[3.14 Reference table 14: technical parameter names 70](#_Toc320976574)

[3.15 Reference table 15: codes of trajectory measurements performed within a cycle 71](#_Toc320976575)

[3.16 Reference table 16: vertical sampling schemes 74](#_Toc320976576)

[3.17 Reference table 17: Argo group 76](#_Toc320976577)

[3.18 Reference table 18: metadata configuration parameter names 77](#_Toc320976578)

[4 Data access 78](#_Toc320976579)

[4.1 File naming convention on GDACs 78](#_Toc320976580)

[4.2 Other data sources 79](#_Toc320976581)

[5 Using the History section of the Argo netCDF Structure 80](#_Toc320976582)

[5.1 Recording information about the Delayed Mode QC process 80](#_Toc320976583)

[5.2 Recording processing stages 81](#_Toc320976584)

[5.3 Recording QC Tests Performed and Failed 82](#_Toc320976585)

[5.4 Recording changes in values 83](#_Toc320976586)

[6 DAC-GDAC data-management 85](#_Toc320976587)

[6.1 Greylist files operations 85](#_Toc320976588)

[6.1.1 Greylist definition and management 85](#_Toc320976589)

[6.1.2 Greylist files collection 86](#_Toc320976590)

[6.2 GDAC files removal 87](#_Toc320976591)

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 | §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\_STAGE" 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 §2.6 : technical data format 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 add a RAFOS positioning system  add a note on qc flag and qc manual  add a description of greylist use for users  trajectory format : move date\_creation and date\_update in the file information section |
| 2.2 | 27/11/2009 | §1.1: “Notice on file format change” chapter added  §1.2: “User Obligations” chapter added  §1.3: “Disclaimer” chapter added  §1.4: “Further information sources and contact information” chapter added  §2.3.1 and §2.3.6: remove N\_HISTORY2 dimension from trajectory format  §2.3.2: move DATE\_CREATION and DATE\_UPDATE to “General information on the trajectory file” chapter  §2.3.4: revisit PARAM and PARAM\_QC policy in real-time/delayed mode  §2.5.4: CONFIGURATION\_PHASE\_REPETITION is removed from the configuration parameter chapter.  §2.5.4: new example with a graphic  §2.8.2: Profile directory file format statement transition added.  §3.2.1: add a reference to quality control manual.  §3.11: add a descripion of table11. Add a new column in the table to explain the link between QC test binary ID and test number.  §3.14: table 14 “technical parameter names” revision, links to naming convention and list of technical parameters added.  §6.1.1: “Greylist definition” chapter added  §6.1.1: Who/when/how to add a float in the greylist  §6.1.1: Who/when/how to remove floats from the greylist  §6.1.1: How users should use the greylist |
| 2.2 | 31/12/2009 | §1.3: Disclaimer; argo data are continuously managed and updated  §2.3.4: Trajectory locations and measurements  Remove DC\_REFERENCE  Do not report DATA\_MODE in this section  report CYCLE\_NUMBER in this section  §2.3.5: Trajectory cycle information from the float  Missing cycle management  Report DATA\_MODE in this section  §3.2.1: Reference table 2: measurement flag scale  For flag 2 comment is “Treat as good data” instead of “Probably good data”  §3.3.2: Oxygen data management  §3.14 Reference table 14: technical parameter names  How to require new technical parameters |
| 2.2 | 08/01/2010 | Address the following messages listed and commented in argo-user-manual-comment-toulouse.doc :  04/01/2010 22:32 Annie Wong  31/12/2009 22:49 Claudia Schmid  31/12/2009 20:35 Claudia Schmid  31/12/2009 19:12 Annie Wong |
| 2.31 | 08/09/2010 | T. Carval : CONCENT\_DOXY is renamed MOLAR\_DOXY to be compliant with the document "Processing Argo OXYGEN data at the DAC level", version 1.0 |
| 2.31 | 14/06/2011 | T. Carval : Add a NMDIS Chinese DAC |
| 2.4 | 19/11/2011 | Thierry Carval : general revision of the document presentation |
| 2.4 | 19/11/2011 | §2.3 Megan Scanderberg : update of trajectory format following Seoul trajectory & ADMT12 meeting |
| 2.4 | 19/11/2011 | §3.3 Thierry Carval : CNDC (conductivity) valid min is set to 8.5 instead of 60.0 |
| 2.4 | 10/02/2012 | §2.2.3 Thierry Carval : vertical sampling scheme to manage profiles performed on different vertical axes |
| 2.4 | 10/02/2012 | §2.4 Esmee Vanwijk : meta-data format version 2.4 |
| 2.4 | 10/02/2012 | §2.2.3 Thierry Carval : global attributes and parameter attributes for CF compatibility |
| 2.4 | 13/02/0212 | §2.5 Thierry Carval : remove chapter “technical information format version 2.2”; keep “technical information format version 2.3” |
| 2.4 | 20/02/2012 | Feedbacks from the draft "User's manual" sent on 13/02/2012.  The changes are highlighted in green.  The comments are available in argo-dm-user-manual-seoul-update-comment.docx |
| 2.4 | 14/03/2012 | Feedbacks from the draft "User's manual" sent on 14/03/2012.  The changes are highlighted in grey.  The comments are available in argo-dm-user-manual-seoul-update-comment.docx |
| 2.4 | 30/03/2012 | The version 2.4 of Argo user's manual is officially released. |
| 2.41 | 19/06/2012 | §2.4.5 : CONFIGURATION\_MISSION\_COMMENT: FillValue is equal to " "; |
| 2.41 | 07/11/2012 | §2.4 : metadata format, additions from Esmee |
| 2.41 | 10/11/2012 | §6.4 : compressed files submission |
| 2.41 | 07/01/2013 | All format versions renamed as 3.0 |

# Introduction

This document is the Argo data user’s manual.

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

## Notice on file format change transition

This version of the "User's manual" is adjusting the file formats to the growing variety of floats and user needs. It introduces a complete revision of metadata and technical files. To cope with this radical change, during a transition period the version 2.2 and 3.0 of the technical and metadata file will be valid among Argo data system.

## User Obligations

A user of Argo data is expected to read and understand this manual and the documentation about the data contained in the “attributes” of the NetCDF data files, as these contain essential information about data quality and accuracy.

A user should acknowledge use of Argo data in all publications and products where such data are used, preferably with the following standard sentence:

**“These data were collected and made freely available by the international Argo project and the national programs that contribute to it.”**

## Disclaimer

Argo data are published without any warranty, express or implied.

The user assumes all risk arising from his/her use of Argo data.

Argo data are intended to be research-quality and include estimates of data quality and accuracy, but it is possible that these estimates or the data themselves may contain errors.

It is the sole responsibility of the user to assess if the data are appropriate for his/her use, and to interpret the data, data quality, and data accuracy accordingly.

Argo welcomes users to ask questions and report problems to the contact addresses listed on the Argo internet page.

Argo data are continuously managed; the user should be aware that after he downloaded data, those data may have been updated on Argo data server.

## Further information sources and contact information

* Argo website: http://www.argo.net/
* If you detect any problem in the Argo data set, please give us your feedback via [support@argo.net](mailto:support@argo.net)

## 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 :

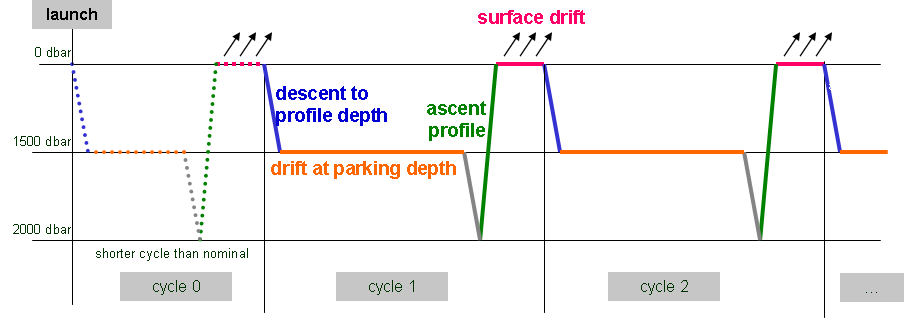
* It is a widely accepted data format by the user community,
* It is a self-describing format for which tools are widely available,
* It is a reliable and efficient format for data exchange.

## 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 stages:

* 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.

## 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.

# Formats description

## Overview of the formats

Argo data formats are based on NetCDF from UNIDATA.

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](http://www.unidata.ucar.edu/packages/netcdf/copyright.html) source can be obtained as [a compressed tar file](ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf.tar.Z) or [a zip file](ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf.ZIP) from Unidata or from other [mirror sites](http://www.unidata.ucar.edu/packages/netcdf/mirrors.html).

* 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 time: all date and time have to be given in Universal Time coordinates.

## Profile format version 3.0

An Argo single-cycle profile file contains a set of profiles from a single cycle. The minimum number is one profile per cycle. There is no defined maximum number of profiles per cycle.

A profile contains all parameters that are measured with the same vertical sampling scheme. For example, all Argo floats collect at least one profile per cycle that contains the CTD measurements.

Some speciality floats collect additional profiles per cycle. These speciality profiles contain parameters measured at pressure levels that are different from the CTD levels. Some examples of speciality profiles with different vertical sampling schemes are:

* Bouncing profiles: a series of shallow profiles performed during one cycle.
* High resolution near-surface observations: higher resolution vertical sampling near the surface from unpumped CTD.
* Oxygen profiles: dissolved oxygen measured on vertical levels that are not the CTD levels.
* Optical profiles: a series of optical profiles performed during one cycle.

For single-cycle profile file naming conventions, see §4.1.

### Global attributes, dimensions and definitions

#### Global attributes

The global attributes section is used for data discovery. The following 9 global attributes should appear in the global section. The NetCDF Climate and Forecast (CF) Metadata Conventions (version 1.6, 5 December, 2011) are available from:

* http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.6/cf-conventions.pdf

// global attributes:

:title = "Argo float vertical profile";

:institution = "CSIRO";

:source = "Argo float";

:history = "2011-04-22T06:00:00Z creation";

:references = "http://www.argodatamgt.org/Documentation";

:comment = "free text";

:user\_manual\_version = "3.0" ;

:Conventions = "Argo-3.0 CF-1.6" ;

:featureType = "trajectoryProfile";

|  |  |
| --- | --- |
| Global attribute name | Definition |
| Title | A succinct description of what is in the dataset |
| Institution | Specifies where the original data was produced. |
| Source | The method of production of the original data. If it was model-generated, source should name the model and its  version, as specifically as could be useful. If it is observational, source should characterize it (e.g., "surface  observation" or "radiosonde"). |
| History | Provides an audit trail for modifications to the original data. Well-behaved generic netCDF filters will  automatically append their name and the parameters with which they were invoked to the global history attribute  of an input netCDF file. We recommend that each line begin with a timestamp indicating the date and time of  day that the program was executed. |
| references | Published or web-based references that describe the data or methods used to produce it. |
| Comment | Miscellaneous information about the data or methods used to produce it. |

#### Dimensions

|  |  |  |
| --- | --- | --- |
| 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. |
| N\_SPECTRUM\_DIMENSION | N\_SPECTRUM\_DIMENSION = 47 | Additional dimension for spectrum variables such a SUNA nitrate sensor. |

### 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:long\_name = "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:long\_name = "File format version";  FORMAT\_VERSION:\_FillValue = " "; | File format version  Example : «3.0» |
| HANDBOOK\_VERSION | char HANDBOOK\_VERSION(STRING4);  HANDBOOK\_VERSION:long\_name = "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: long\_name = "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 1st 1950 00:00:00 |
| DATE\_CREATION | char DATE\_CREATION(DATE\_TIME);  DATE\_CREATION: long\_name = "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 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\_UPDATE:\_FillValue = " "; | Date and time (UTC) of update of this file.  Format : YYYYMMDDHHMISS  Example :  20011230090500 : December 30th 2001 09 :05 :00 |

### 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 : A9IIIII"; 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: long\_name = "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: long\_name = "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);  STATION\_PARAMETERS:long\_name = "List of available parameters for the station";  STATION\_PARAMETERS:conventions = "Argo reference table 3";  STATION\_PARAMETERS:\_FillValue = " "; | List of parameters contained in this profile.  The parameter names are listed in reference table 3.  Examples : TEMP, PSAL, CNDC  TEMP : temperature  PSAL : practical salinity  CNDC : conductvity |
| 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:standard\_name = "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.;  JULD:axis = "T" ; | Julian day of the profile.  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\_LOCATION:long\_name = "Julian day (UTC) of the location relative to REFERENCE\_DATE\_TIME ";  JULD\_LOCATION:units = "days since 1950-01-01 00:00:00 UTC";  JULD\_LOCATION:conventions = "Relative julian days with decimal part (as parts of day)";  JULD\_LOCATION:\_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:standard\_name = "latitude" ;  LATITUDE:units = "degree\_north";  LATITUDE:\_FillValue = 99999.;  LATITUDE:valid\_min = -90.;  LATITUDE:valid\_max = 90.;  LATITUDE:axis = "Y" ; | 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:standard\_name = "longitude" ;  LONGITUDE:units = "degree\_east";  LONGITUDE:\_FillValue = 99999.;  LONGITUDE:valid\_min = -180.;  LONGITUDE:valid\_max = 180.;  LONGITUDE:axis = "X" ; | 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 |
| VERTICAL\_SAMPLING\_SCHEME | char VERTICAL\_SAMPLING\_SCHEME (N\_PROF, STRING256);  VERTICAL\_SAMPLING\_SCHEME:long\_name = "Vertical sampling scheme" ;  VERTICAL\_SAMPLING\_SCHEME:conventions = "Argo reference table 16" ;  VERTICAL\_SAMPLING\_SCHEME:\_FillValue = " " ; | This variable is mandatory.  Use vertical sampling scheme to differentiate and identify profiles from a single-cycle with different vertical sampling schemes. |

### 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.

<PARAM> contains the raw values telemetered from the floats.

The values in <PARAM> should never be altered. <PARAM\_QC> contains qc flags that pertain to the values in <PARAM>. Values in <PARAM\_QC> are set initially in 'R' and 'A' modes by the automatic real-time tests.

They are later modified in 'D' mode at levels where the qc flags are set incorrectly by the real-time procedures, and where erroneous data are not detected by the real-time procedures.

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 adjustment 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>:standard\_name = "<X>";  <PARAM>:\_FillValue = <X>;  <PARAM>:units = "<X>"; <PARAM>:valid\_min = <X>;  <PARAM>:valid\_max = <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>:standard\_name = "<X>";  <PARAM>\_ADJUSTED:\_FillValue = <X>;  <PARAM>\_ADJUSTED:units = "<X>"; <PARAM>\_ADJUSTED:valid\_min = <X>;  <PARAM>\_ADJUSTED:valid\_max = <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: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. |

Note on vertical axis associated to PRES

The variable PRES (pressure) is the vertical axis. The PRES declaration contains the variable attribute

PRES:axis = "Z" ;

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

|  |
| --- |
| Parameter definition : PRES, TEMP, TEMP\_ADJUSTED |
| float TEMP(N\_PROF, N\_LEVELS);  TEMP:long\_name = "SEA TEMPERATURE IN SITU ITS-90 SCALE";  TEMP:standard\_name = "sea\_water\_temperature" ;  TEMP:\_FillValue = 99999.f;  TEMP:units = "degree\_Celsius"; TEMP:valid\_min = -2.f;  TEMP:valid\_max = 40.f;  TEMP:C\_format = "%9.3f";  TEMP:FORTRAN\_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 IN SITU ITS-90 SCALE";  TEMP:standard\_name = "sea\_water\_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:C\_format = "%9.3f";  TEMP\_ADJUSTED:FORTRAN\_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 IN SITU ITS-90 SCALE";  TEMP\_ADJUSTED\_ERROR:\_FillValue = 99999.f;  TEMP\_ADJUSTED\_ERROR:units = "degree\_Celsius"; TEMP\_ADJUSTED\_ERROR :C\_format = "%9.3f";  TEMP\_ADJUSTED\_ERROR :FORTRAN\_format= "F9.3";  TEMP\_ADJUSTED\_ERROR:resolution= 0.001f; |

#### Management of spectrum observation

To manage spectrum observations from sensors such as the SUNA optical sensor of Nitrate, an additional dimension is added to report the N spectrum observation performed on each level.

float <PARAM>(N\_PROF, N\_LEVELS, N\_SPECTRUM\_DIMENSION);

Example of 45 measurements of “Spectrum\_” parameter performed at each 60 levels of an individual profile.

* NPROF = 1
* N-LEVELS = 60
* N\_SPECTRUM\_DIMENSION = 45

### 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 :  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";  SCIENTIFIC\_CALIB\_COEFFICIENT:\_FillValue = " "; | 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";  SCIENTIFIC\_CALIB\_COMMENT:\_FillValue = " ";f | Comment about this calibration  Example :  The sensor is not stable |
| SCIENTIFIC\_CALIB\_DATE | Char SCIENTIFIC\_CALIB\_DATE (N\_PROF N\_CALIB, N\_PARAM, DATE\_TIME) SCIENTIFIC\_CALIB\_DATE:\_FillValue = " ";  SCIENTIFIC\_CALIB\_DATE:long\_name = "Date of calibration"; | Date of the calibration.  Example : 20011217161700 |
| PARAMETER\_DATA\_MODE | char PARAMETER\_DATA\_MODE(N\_ PROF,N\_PARAM);  PARAMETER\_DATA\_MODE:long\_name = "Delayed mode or real time data";  PARAMETER\_DATA\_MODE:conventions = "R : real time; D : delayed mode; A : real time with adjustment";  PARAMETER\_DATA\_MODE:\_FillValue = " "; | Describe the data mode of the individual parameter :  R : real time data  D : delayed mode data  A : real time data with adjusted values |

### 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";  HISTORY\_DATE:\_FillValue = " "; | 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 = "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, N\_PROF, 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\_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 action applied on";  HISTORY\_STOP\_PRES:\_FillValue = 99999.f;  HISTORY\_STOP\_PRES:units = "decibar"; | Stop pressure the action is applied to. This should be greater than START\_PRES.  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$; tests failed when ACTION=QCF$";  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 the History section is described in §5 "Using the History section of the Argo netCDF Structure".

## Trajectory format version 3.0

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.

For file naming conventions, see §4.1.

### Global attributes, dimensions and definitions

#### Global attributes

The global attributes section is used for data discovery. The following 8 global attributes should appear in the global section. The NetCDF Climate and Forecast (CF) Metadata Conventions (version 1.6, 5 December, 2011) are available from:

* http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.6/cf-conventions.pdf

// global attributes:

:title = "Argo float trajectory file";

:institution = "CSIRO";

:source = "Argo float";

:history = "2011-04-22T06:00:00Z creation";

:references = "http://www.argodatamgt.org/Documentation";

:comment = "free text";

:user\_manual\_version = "3.0" ;

:Conventions = “Argo-3.0 CF-1.6" ;

:featureType = "trajectory";

#### Dimensions

|  |  |  |
| --- | --- | --- |
| 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> ; | Number of cycles performed by the float.  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 |

### 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:long\_name = "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:long\_name "File format version ";  FORMAT\_VERSION:\_FillValue = " "; | File format version  Example : «3.0» |
| HANDBOOK\_VERSION | char HANDBOOK\_VERSION(STRING4);  HANDBOOK\_VERSION: long\_name "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: long\_name "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 1st 1950 00:00:00 |
| DATE\_CREATION | char DATE\_CREATION(DATE\_TIME);  DATE\_CREATION: long\_name "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 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\_UPDATE:\_FillValue = " "; | Date and time (UTC) of update of this file.  Format : YYYYMMDDHHMISS  Example :  20011230090500 : December 30th 2001 09 :05 :00 |

### 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 : A9IIIII";  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: long\_name = "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: long\_name = "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 : conductvity |
| 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 |
| 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 |

### Locations and measurements from the float

This section contains locations for an individual 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.

<PARAM> contains the raw values telemetered from the floats.

The values in <PARAM> should never be altered.

<PARAM\_QC> contains qc flags that pertain to the values in <PARAM>. Values in <PARAM\_QC> are set initially in 'R'[[1]](#footnote-1) and 'A'[[2]](#footnote-2) modes by the automatic real-time tests.

They are later modified in 'D'[[3]](#footnote-3) mode at levels where the qc flags are set incorrectly by the real-time procedures, and where erroneous data are not detected by the real-time procedures.

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 |
| JULD | double JULD(N\_MEASUREMENT);  JULD:long\_name = "Julian day (UTC) of each measurement relative to REFERENCE\_DATE\_TIME";  JULD:standard\_name = "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.;  JULD:axis = "T" ; | 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:standard\_name = "latitude" ;  LATITUDE:units = "degree\_north";  LATITUDE:\_FillValue = 99999.;  LATITUDE:valid\_min = -90.;  LATITUDE:valid\_max = 90.;  LATITUDE:axis = "Y" ; | 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:standard\_name = "longitude" ;  LONGITUDE:units = "degree\_east";  LONGITUDE:\_FillValue = 99999.;  LONGITUDE:valid\_min = -180.;  LONGITUDE:valid\_max = 180.;  LONGITUDE:axis = "X" ; | 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 17th cycle of the float. |
| MEASUREMENT\_CODE | int MEASUREMENT\_CODE (N\_MEASUREMENT);  MEASUREMENT\_CODE:long\_name = "Code referring to a measurement event in the cycle";  MEASUREMENT\_CODE:conventions = "Argo reference table 15";  MEASUREMENT\_CODE:\_FillValue = 99999; | Code for each event in the cycle which corresponds to Argo reference table 15.  Example:  1: All measurements made at start of descent to drift pressure . Could be time, location, surface pressure, etc. |
| <PARAM> | float <PARAM>(N\_MEASUREMENT);  <PARAM>:long\_name = "<X>";  <PARAM>:standard\_name = "<X>";  <PARAM>:\_FillValue = <X>;  <PARAM>:units = "<X>"; <PARAM>:valid\_min = <X>;  <PARAM>:valid\_max = <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>:standard\_name = "<X>";  <PARAM>\_ADJUSTED:\_FillValue = <X>;  <PARAM>\_ADJUSTED:units = "<X>"; <PARAM>\_ADJUSTED:valid\_min = <X>;  <PARAM>\_ADJUSTED:valid\_max = <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 = "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\_MEASUREMENT);  <PARAM>\_ADJUSTED\_ERROR:long\_name = "<X>";  <PARAM>\_ADJUSTED\_ERROR:\_FillValue = <X>;  <PARAM>\_ADJUSTED\_ERROR:units = "<X>"; <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. |

### 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.

When a cycle is missing (e.g. no data received), all cycle information is reported as fill values.

|  |  |  |
| --- | --- | --- |
| 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  3: date is determined by positioning system  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  3: date is determined by positioning system  9 : date is unknown |
| JULD\_DESCENT\_START | 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.; | Julian day (UTC) of the beginning of the descending profile.  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  3: date is determined by positioning system  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  3: date is determined by positioning system  9 : date is unknown |
| JULD\_TRANSMISSION\_START | double JULD\_TRANSMISSION\_START(N\_CYCLE);  JULD\_TRANSMISSION\_START:long\_name = "Start date of transmssion";  JULD\_TRANSMISSION\_START:units = "days since 1950-01-01 00:00:00 UTC";  JULD\_TRANSMISSION\_START:conventions = "Relative julian days with decimal part (as part of day)"  JULD\_TRANSMISSION\_START:\_FillValue=999999.; | Julian day (UTC) of the beginning of data transmission.  Example :  18833.8013889885 : July 25 2001 19:14:00 |
| JULD\_TRANSMISSION\_START\_STATUS | char JULD\_TRANSMISSION\_START\_STATUS(N\_CYCLE);  JULD\_TRANSMISSION\_START\_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";  JULD\_TRANSMISSION\_START\_STATUS:\_FillValue = " "; | 0 : date comes from the float meta data  1 : date is estimated  2 : date is transmitted by the float  3: date is determined by positioning system  9 : date is unknown |
| JULD\_FIRST\_STABILIZATION | Double JULD\_FIRST\_STABILIZATION(N\_CYCLE);  JULD\_FIRST\_STABILIZATION:long\_name = “Time of float’s first stabilization after leaving the surface”;  JULD\_FIRST\_STABILIZATION:units = days since 1950-01-01 00:00:00 UTC”;  JULD\_FIRST STABILIZATION: Relative julian days with decimal part (as part of day) ";  JULD\_FIRST\_STABILIZATION:\_FillValue=999999 | Julian day (UTC) of the first stabilization after the start of descent to  the drift pressure.  Example :  18833.8013889885 : July 25 2001 19:14:00 |
| JULD\_FIRST\_STABILIZATION\_STATUS | char JULD\_FIRST\_STABILIZATION\_STATUS(N\_CYCLE);  JULD\_FIRST\_STABILIZATION\_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";  JULD\_FIRST\_STABILIZATION\_STATUS:\_FillValue = " "; | 0 : date comes from the float meta data  1 : date is estimated  2 : date is transmited by the float  3: date is determined by positioning system  9 : date is unknown |
| JULD\_DEEP\_DESCENT\_START | double JULD\_DEEP\_DESCENT\_START(N\_CYCLE);  JULD\_DEEP\_DESCENT\_START:long\_name = "Deep Descent start date of the cycle";  JULD\_DEEP\_DESCENT\_START:units = "days since 1950-01-01 00:00:00 UTC";  JULD\_DEEP\_DESCENT\_START:conventions = "Relative julian days with decimal part (as part of day) ";  JULD\_DEEP\_DESCENT\_START:\_FillValue=999999.; | Julian day (UTC) of the start of the deep descent to profile pressure at the end  of the drift phase.  Example :  18833.8013889885 : July 25 2001 19:14:00 |
| JULD\_DEEP\_DESCENT\_START\_STATUS | char JULD\_DEEP\_DESCENT\_START\_STATUS(N\_CYCLE);  JULD\_DEEP\_DESCENT\_START\_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";  JULD\_DEEP\_DESCENT\_START\_STATUS:\_FillValue = " "; | 0 : date comes from the float meta data  1 : date is estimated  2 : date is transmited by the float  3: date is determined by positioning system  9 : date is unknown |
| JULD\_DEEP\_DESCENT\_END | double JULD\_DEEP\_DESCENT\_END(N\_CYCLE);  JULD\_DEEP\_DESCENT\_END:long\_name = "Deep Descent end date of the cycle";  JULD\_DEEP\_DESCENT\_END:units = "days since 1950-01-01 00:00:00 UTC";  JULD\_DEEP\_DESCENT\_END:conventions = "Relative julian days with decimal part (as part of day) ";  JULD\_DEEP\_DESCENT\_END:\_FillValue=999999.; | Julian day (UTC) of the end of the deep descent to profile pressure.  Example :  18833.8013889885 : July 25 2001 19:14:00 |
| JULD\_DEEP\_DESCENT\_END\_STATUS | char JULD\_DEEP\_DESCENT\_END\_STATUS(N\_CYCLE);  JULD\_DEEP\_DESCENT\_END\_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";  JULD\_DEEP\_DESCENT\_END\_STATUS:\_FillValue = " "; | 0 : date comes from the float meta data  1 : date is estimated  2 : date is transmited by the float  3: date is determined by positioning system  9 : date is unknown |
| JULD \_TRANSMISSION\_END | double JULD \_TRANSMISSION\_END (N\_CYCLE);  JULD \_TRANSMISSION\_END:long\_name = "Transmssion end date";  JULD \_TRANSMISSION\_END:units = "days since 1950-01-01 00:00:00 UTC";  JULD \_TRANSMISSION\_END:conventions = "Relative julian days with decimal part (as part of day)";  JULD \_TRANSMISSION\_END:\_FillValue=999999.; | Julian day (UTC) of the end of transmission.  Example :  18833.8013889885 : July 25 2001 19:14:00 |
| JULD \_TRANSMISSION\_END \_STATUS | char JULD \_TRANSMISSION\_END \_STATUS (N\_CYCLE);  JULD \_TRANSMISSION\_END \_STATUS:conventions = “0 : Nominal, 1 : Estimated, 2 :Transmitted";  JULD \_TRANSMISSION\_END \_STATUS:\_FillValue = " "; | 0 : date comes from the float meta data  1 : date is estimated  2 : date is transmitted by the float  3: date is determined by positioning system  9 : date is unknown |
| GROUNDED | char GROUNDED(N\_CYCLE);  GROUNDED:long\_name = "Did the profiler touch the ground for that part of the cycle”;  GROUNDED:conventions = "Y,P,N,U";  GROUNDED:\_FillValue = " "; | GROUNDED indicates if the float touched the ground for that part of the cycle.  Format : Y, N, U  Examples :  Y : yes the float touched the ground during drift  P : yes the float touched the ground during descent to profile  N : no  U : unknown |
| CONFIG\_MISSION\_NUMBER | int CONFIG\_MISSION\_NUMBER (N\_CYCLE);  CONFIG\_MISSION\_NUMBER:long\_name = " mission number of unique cycles performed by the float”;  CONFIG\_MISSION\_NUMBER:\_FillValue = " "; | Mission number of the configuration parameter.  Example : 1  See §2.4.5 "Configuration parameters". |
| CYCLE\_NUMBER\_ACTUAL | int CYCLE\_NUMBER\_ACTUAL(N\_CYCLE);  CYCLE\_NUMBERc: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 one cycle number, there is a collection of useful information recorded (e.g. grounded or not).  Example : 17 for measurements performed during the 17th cycle of the float. |
| DATA\_MODE | char DATA\_MODE(N\_ CYCLE);  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 |

### 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);  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 HISORY\_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; | Stopt 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, 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".

## Metadata format version 3.0

The format version 3.0 of Argo metadata will replace version 2.2 gradually. During the transition period, all formats will be valid. However, when a Data Assembly Center (DAC) produces metadata files with the new 3.0 format, all its metadata files must be provided in version 3.0.

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

For file naming conventions, see §4.1.

### Global attributes, dimensions and definitions

#### Global attributes

The global attributes section is used for data discovery. The following 8 global attributes should appear in the global section. The NetCDF Climate and Forecast (CF) Metadata Conventions (version 1.6, 5 December, 2011) are available from:

* http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.6/cf-conventions.pdf

// global attributes:

:title = "Argo float metadata file";

:institution = "CSIRO";

:source = "Argo float";

:history = "2011-04-22T06:00:00Z creation";

:references = "http://www.argodatamgt.org/Documentation";

:comment = "free text";

:user\_manual\_version = "3.0" ;

:Conventions = “Argo-3.0 CF-1.6" ;

#### 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 |
| STRING1024  STRING256 | STRING1024 = 1024;  STRING256 = 256; | String dimensions from 2 to 1024. |
| STRING64 | STRING64 = 64; |  |
| STRING32 | STRING32 = 32; |  |
| STRING16 | STRING16 = 16; |  |
| STRING8 | STRING8 = 8; |  |
| STRING4 | STRING4 = 4; |  |
| STRING2 | STRING2 = 2; |  |
| 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 |
| N\_CONF\_PARAM | N\_CONF\_PARAM=<int value>; | Number of configuration parameters. |
| N\_MISSIONS | N\_MISSIONS=<unlimited>; | Number of missions. |
| N\_POSITIONING\_SYSTEM | N\_POSITIONING\_SYSTEM=<int value>; | Number of positioning systems. |
| N\_TRANS\_SYSTEM | N\_TRANS\_SYSTEM=<int value>; | Number of transmission systems. |
|  |  |  |

### 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: long\_name = "Data type"; DATA\_TYPE:\_FillValue = " "; | MANDATORY  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: long\_name = "File format version ";  FORMAT\_VERSION:\_FillValue = " "; | File format version Example : «3.0» |
| HANDBOOK\_VERSION | char HANDBOOK\_VERSION(STRING4); HANDBOOK\_VERSION: long\_name = "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: long\_name = "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 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\_UPDATE:\_FillValue = " "; | Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30th 2001 09:05:00 |

### 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. |
| IMEI | char IMEI (STRING256); IMEI:long\_name = "Transmission identifier (ARGOS, ORBCOMM, etc.)"; IMEI:\_FillValue = " "; | Optional: The IMEI number for the float. For security reasons recommend that only the last six digits (after removing the check sum) are reported. Example : 423978. |
| TRANS\_SYSTEM | char TRANS\_SYSTEM(N\_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(N\_TRANS\_SYSTEM, 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 Orbcomm) 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 |
| POSITIONING\_SYSTEM | char POSITIONING\_SYSTEM(N\_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 |
| PLATFORM\_FAMILY | char PLATFORM\_FAMILY (STRING256); PLATFORM\_FAMILY:long\_name = "Category of instrument ";  PLATFORM\_FAMILY:\_FillValue = " "; | Category of instrument. Example: Float, POPS, ITP |
| PLATFORM\_TYPE | char PLATFORM\_TYPE (STRING32); PLATFORM\_TYPE:long\_name = "Type of float "; PLATFORM\_TYPE:\_FillValue = " "; | Type of float. Example: SOLO, APEX, PROVOR, ARVOR, NINJA |
| 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 Corporation |
| FIRMWARE\_VERSION | char FIRMWARE\_ VERSION (STRING16);  FIRMWARE\_ VERSION:long\_name = “The firmware version for the float.”;  FIRMWARE\_ VERSION:\_FillValue = " "; | The firmware version. This is specified as per the format on the manufacturers manual. Example: 072804 |
| MANUAL\_VERSION | char MANUAL\_ VERSION (STRING16);  MANUAL\_ VERSION:long\_name = “The manual version for the float.”;  MANUAL\_ VERSION:\_FillValue = " "; | The version date or number for the manual for each float. Example 110610 or 004 |
| FLOAT\_SERIAL\_NO | char FLOAT\_SERIAL\_NO(STRING16);  long\_name = "The serial number of the float"; FLOAT\_SERIAL\_NO:\_FillValue = " "; | This field should contain only the serial number of the float. Example 1679 |
| STANDARD\_FORMAT\_ID | char STANDARD\_FORMAT\_ID(STRING16);  FORMAT\_NUMBER:long\_name = "A standard format number to describe the data format type for each float."; FORMAT\_NUMBER:\_FillValue = " "; | Standardised format number as described in the reference table online (host site yet to be determined, this table cross references to individual DAC format numbers. Example: 1 |
| DAC\_FORMAT\_ID | char DAC\_FORMAT\_ID(STRING16);  FORMAT\_NUMBER:long\_name = "The format number used by the DAC to describe the data format type for each float."; FORMAT\_NUMBER:\_FillValue = " "; | Format numbers used by individual DACs to describe each float type. This is cross-referenced to a standard format id by a reference table online, host site yet to be determined. |
| 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 : 846 : Webb Research float, Seabird sensor |
| 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:long\_name = "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.” |
| BATTERY\_TYPE | char BATTERY\_TYPE(STRING64);  BATTERY\_TYPE: long\_name = “The type of battery packs in the float.”;  BATTERY\_TYPE:\_FillValue = " "; | Describes the type of battery packs in the float.  Example: Alkaline, Lithium or Alkaline and Lithium |
| BATTERY\_PACKS | char BATTERY\_PACKS(STRING64);  BATTERY\_PACKS: long\_name = “The configuration of battery packs in the float.”;  BATTERY\_PACKS:\_FillValue = " "; | Describes the configuration of battery packs in the float, number and type. Example: 4DD Li + 1C Alk |
| CONTROLLER\_BOARD\_TYPE\_PRIMARY | char CONTROLLER\_BOARD\_TYPE\_PRIMARY(STRING32);  CONTROLLER\_BOARD\_TYPE\_PRIMARY: long\_name = “The type of controller board.”;  CONTROLLER\_BOARD\_TYPE\_PRIMARY: FillValue = " "; | Describes the type of controller board. Example: APF8, APF9i |
| CONTROLLER\_BOARD\_TYPE\_SECONDARY | char CONTROLLER\_BOARD\_TYPE\_SECONDARY(STRING32);  CONTROLLER\_BOARD\_TYPE\_SECONDARY: long\_name = “The secondary type of controller board.”;  CONTROLLER\_BOARD\_TYPE\_SECONDARY: FillValue = " "; | Only applicable if there is more than one controller board in the float. Describes the second type of controller board.  Example: APF8, APF9i |
| CONTROLLER\_BOARD\_SERIAL\_NO\_PRIMARY | char CONTROLLER\_BOARD\_SERIAL\_NO\_PRIMARY(STRING32);  CONTROLLER\_BOARD\_SERIAL\_NO\_PRIMARY: long name = “The serial number of the primary controller board”;  CONTROLLER\_BOARD\_SERIAL\_NO\_PRIMARY: FillValue = " "; | The serial number for the primary controller board.  Example: 4567 |
| CONTROLLER\_BOARD\_SERIAL\_NO\_SECONDARY | char CONTROLLER\_BOARD\_SERIAL\_NO\_SECONDARY(STRING32);  CONTROLLER\_BOARD\_SERIAL\_NO\_SECONDARY: long name = “The serial number of the secondary controller board”;  CONTROLLER\_BOARD\_SERIAL\_NO\_SECONDARY: FillValue = " "; | Only applicable if there is more than one controller board in the float. The serial number for the secondary controller board.  Example: 4567 |
| SPECIAL\_FEATURES | char SPECIAL\_FEATURES (STRING1024); SPECIAL\_FEATURES:long\_name = "Extra features of the float (algorithms, compressee etc.)"; SPECIAL\_FEATURES:\_FillValue = " "; | Additional float features can be specified here such as algorithms used by the float (Ice Sensing Algorithm, Interim Storage Algorithm, grounding avoidance) or additional hardware such as a compressee (buoyancy compensator).  Example : … |
| SAMPLING\_MODE | char SAMPLING\_MODE (STRING32); SAMPLING\_MODE:long\_name = "Float sampling mode"; SAMPLING\_MODE:\_FillValue = " "; | The float sampling mode.  Example: Spot, Continuous or Mixed. |
| REDEPLOYED | char REDEPLOYED (STRING32); REDEPLOYED:long\_name = "Indicates if the float has been previously deployed.”; REDEPLOYED:\_FillValue = " "; | Indicates if the float has been previously deployed, i.e. recovered after first mission and deployed with a new wmo id. In this case you would fill this field with the previous wmo id.  Example: … |
| FLOAT\_OWNER | char FLOAT\_OWNER (STRING64); FLOAT\_OWNER:long\_name = "The float owner"; FLOAT\_OWNER:\_FillValue = " "; | The owner of the float (may be different from the data centre and operating institution).  Example: Tom Trull |
| OPERATING\_INSTITUTION | char OPERATING\_INSTITUTION (STRING64); OPERATING\_INSTITUTION:long\_name = "The operating institution of the float"; OPERATING\_INSTITUTION:\_FillValue = " "; | The operating institution of the float (may be different from the float owner and data centre).  Example: ACE CRC |
| CUSTOMISATION | char CUSTOMISATION (STRING1024); CUSTOMISATION:long\_name = "Float customisation, i.e. (institution and modifications)"; CUSTOMISATION:\_FillValue = " "; | Free form field to record changes made to the float after manufacture and before deployment, i.e. this could be the customisation institution plus a list of modifications.  Example:… |
| ARGO\_GROUP | char ARGO\_GROUP (STRING64); ARGO\_GROUP:long\_name = "The Argo group to which the float belongs"; ARGO\_GROUP:\_FillValue = " ";  ARGO\_GROUP:conventions = "Argo reference table 17"; | The Argo group to which the float belongs.  The valid Argo groups are listed in reference table 17.  Example: Core Argo, Bio Argo, Argo Equivalent |

### 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 30th 2001  03:05:00 |
| LAUNCH\_LATITUDE | double LAUNCH\_LATITUDE; | Latitude of the launch. |
| LAUNCH\_LATITUDE:long\_name = "Latitude of the float when deployed";  LAUNCH\_LATITUDE:units = "degrees\_north"; LAUNCH\_LATITUDE:\_FillValue = 99999.; | Unit : degree north Example : 44.4991 : 44° 29’ 56.76’’ N |
| LAUNCH\_LATITUDE:valid\_min = -90.; LAUNCH\_LATITUDE:valid\_max = 90.; |  |
|  |  |
| LAUNCH\_LONGITUDE | double LAUNCH\_LONGITUDE; LAUNCH\_LONGITUDE:long\_name = "Longitude of the float when deployed"; LAUNCH\_LONGITUDE:units = "degrees\_east"; LAUNCH\_LONGITUDE:\_FillValue = 99999.; | Longitude of the launch. Unit : degree east Example : 16.7222 : 16° 43’ 19.92’’ E |
|  | LAUNCH\_LONGITUDE:valid\_min = -180.; |  |
|  | LAUNCH\_LONGITUDE:valid\_max = 180.; |  |
| LAUNCH\_QC | char LAUNCH\_QC; LAUNCH\_QC:long\_name = "Quality on launch date, time | Quality flag on launch date, time and location. |
|  | and location"; LAUNCH\_QC:conventions = "Argo reference table 2"; | The flag scale is described in the reference table 2. |
|  | LAUNCH\_QC:\_FillValue = " "; | Example : 1 : launch location seems correct. |
| START\_DATE | char START\_DATE(DATE\_TIME); | Date and time (UTC) of the first descent |
|  | START\_DATE:long\_name = "Date (UTC) of the first | of the float. |
|  | descent of the float."; | Format : YYYYMMDDHHMISS |
|  | START\_DATE:conventions = "YYYYMMDDHHMISS"; START\_DATE:\_FillValue = " "; | Example : 20011230090500 : December 30th 2001 |
|  |  | 06 :05 :00 |
| START\_DATE\_QC | char START\_DATE\_QC; START\_DATE\_QC:long\_name = "Quality on start date"; | Quality flag on start date. The flag scale is described in the |
|  | START\_DATE\_QC:conventions = "Argo reference table 2"; | reference table 2. Example : |
|  | START\_DATE\_QC:\_FillValue = " "; | 1 : start date seems correct. |
| DEPLOYMENT\_PLATFORM | char DEPLOY\_PLATFORM(STRING32); DEPLOY\_PLATFORM:long\_name = "Identifier of the deployment platform"; | Identifier of the deployment platform, i.e. vessel or ship name. Example : L’ATALANTE |
|  | DEPLOY\_PLATFORM:\_FillValue = " "; |  |
| ~~DEPLOY\_MISSION~~  DEPLOYMENT\_CRUISE\_ID | char DEPLOYMENT\_CRUISE\_ID(STRING32); DEPLOYMENT\_CRUISE\_ID:long\_name = "Identification number or reference number of the cruise used to deploy the float”; | Identification number or reference number of the cruise used to deploy the platform. |
|  | DEPLOY\_MISSION:\_FillValue = " "; | Example : POMME2 |
| ~~DEPLOY\_AVAILABLE\_PRO FILE\_ID~~  DEPLOYMENT\_REFERENCE\_STATION\_ID | char DEPLOYMENT\_REFERENCE\_STATION\_ID(STRING256); DEPLOYMENT\_REFERENCE\_STATION\_ID:long\_name = | Identifier or reference number of co-located CTD or XBT stations used to verify the first profile. |
|  | "Identifier or reference number of co-located stations used to verify the first profile"; DEPLOY\_AVAILABLE\_PROFILE\_ID:\_FillValue = " "; | Example : 58776, 58777 |
| END\_MISSION\_DATE | char END\_MISSION\_DATE (DATE\_TIME); | Date (UTC) of the end of mission of the |
|  | END\_MISSION\_DATE:long\_name = "Date (UTC) of the | float. |
|  | end of mission of the float"; | Format : YYYYMMDDHHMISS |
|  | END\_MISSION\_DATE:conventions = "YYYYMMDDHHMISS"; | Example : 20011230090500 : December 30th 2001 |
|  | END\_MISSION\_DATE:\_FillValue = " "; | 03:05:00 |
| END\_MISSION\_STATUS | char END\_MISSION\_STATUS; | Status of the end of mission of the float. |
|  | END\_MISSION\_STATUS:long\_name = "Status of the end of mission of the float"; | T:No more |
|  | END\_MISSION\_STATUS:conventions = "T:No more | transmission received, |
|  | transmission received, | R:Retrieved |
|  | R:Retrieved"; |  |
|  | END\_MISSION\_STATUS:\_FillValue = " "; |  |

### Configuration parameters

This section describes the configuration parameters for a float. It is important to note that configuration parameters are float settings, not measurements reported by the float.

Configuration parameters may or may not be reported by a float.

Configuration parameters are identified by the “CONFIG” prefix.

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

|  |  |  |
| --- | --- | --- |
| Name | Definition | Comment |
| CONFIG\_PARAMETER\_NAME | char CONFIG\_PARAMETER\_NAME (N\_CONF\_PARAM, STRING128)  CONFIG\_PARAMETER\_NAME:long\_name=”Name of configuration parameter”;  CONFIG\_PARAMETER\_NAME:\_FillValue = " "; | Name of the configuration parameter. Example :  “CONFIG\_ParkPressure\_dBAR” See reference table 14b for standard configuraton parameter names. |
| CONFIG\_PARAMETER\_VALUE | int CONFIG\_ PARAMETER\_VALUE (N\_MISSIONS, N\_CONF\_PARAM)  CONFIG\_ PARAMETER\_VALUE:long\_name=”Value of configuration parameter”;  CONFIG\_ PARAMETER\_VALUE:\_FillValue = " "; | Value of the configuration parameter. Example : "1500" |
|  |  |
|  |  |
| CONFIG\_MISSION\_NUMBER | int CONFIG\_MISSION\_NUMBE R (N\_MISSIONS); CONFIG\_MISSION\_NUMBE R:long\_name = "Unique number denoting the missions performed by the floatMission";  CONFIG\_MISSION\_NUMBER:conventions = "0..N, 0 : launch  mission (if exists), 1 : first complete mission"; CONFIG\_MISSION\_NUMBER:\_FillValue = 99999; | Unique number of the mission to which this parameter belongs. Example : 0 See note on floats with multiple configurations. |
|  |
| CONFIG\_MISSION\_COMMENT | char CONFIGURATION\_MISSION\_COMM ENT (N\_MISSIONS, STRING256) CONFIGURATION\_MISSION\_COMMENT:long\_name  =”Comment on configuration”;  CONFIGURATION\_MISSION\_COMMENT: FillValue= " "; | Comment on this configuration mission. Example : “This mission follows a 1000 dbar meddie during parking” |
|  |
|  |

The mission settings or parameter values are recorded as numbers. In this scheme, strings will need to be converted to numbers and will require measurement codes for the relevant parameters. The numeric codes for the affected parameters are defined in the “Explanation” section of the Configuration parameter names table (please see reference table 18). Only a few existing parameters are affected. If new floats with new configuration parameters (as strings) are introduced, then equivalent numeric flags must also be added to the table by the proposer of the new configuration parameter.

All parameter names are standardized and are available in reference table 18.   
The mission is used to record information that changes from cycle to cycle, for instance when a float changes its mission from 3 shallow profiles to 1 deep profile. The shallow and deep profiles will have different mission numbers. The value of the mission number is recorded in CONFIG\_MISSION\_NUMBER.

Mission 0 parameters are pre-deployment or launch instructions. They are configuration parameters that are ‘configured’ but not changeable and are therefore designated mission 0 so that they are clearly differentiated from the other mission variables that may change during the float lifetime.

The parameter CONFIG\_MISSION \_COMMENT can be used to store information about the mission or whether the mission was set pre-deployment or transmitted by the float (free form field).

##### Note on floats with multiple configurations

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

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

When there is only one configuration, CONFIG\_MISSION\_NUMBER is set to 1: all the cycles are programmed to be the same. Note that in this case; floats will still have mission “0” that contains the pre-deployment or launch information. So for a float with one basic mission, it will have missions 0 and 1.

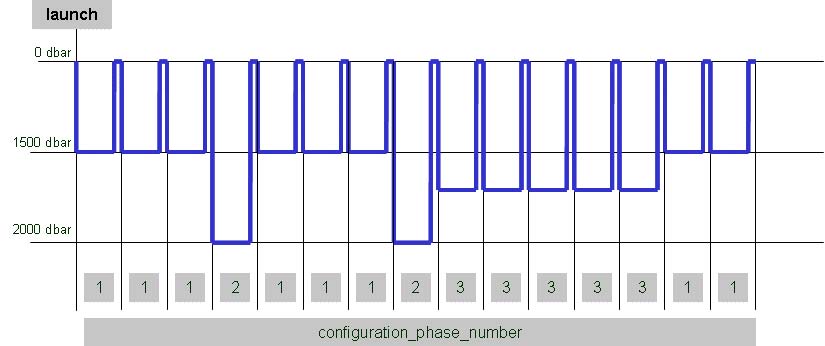
When there are multiple configurations, the configuration from the first cycle has CONFIG\_MISSION\_NUMBER set to 1. Each subsequent configuration change will be recorded as additional entries in CONFIG\_MISSION\_NUMBER, with the value increased sequentially by the integer one. All variables from mission 1 must be repeated in subsequent missions. Floats with multiple configurations still record pre-deployment or launch information in CONFIG\_MISSION\_NUMBER = 0.

Argo best practice and our recommendation to users, is a minimum of configuration missions; i.e. if there is a change to configuration parameters that does not repeat a previous configuration then a new mission number should be used. If the configuration parameters change, but mirror a previous mission then that mission number should be re-used. In extremely complex cases where mission changes are unclear, then a new mission number can be used for each cycle. Users should be aware that the metafile will need to be rewritten each time a new mission number is added.

#### Determining which mission applies to a particular float cycle

Users are able to determine which mission applies to each cycle by looking at the CONFIG

\_MISSION\_NUMBER(N\_CYCLE) variable located in the trajectory file (see section “2.3.5 cycle information from the float” in the “Trajectory format version 3.0” section of this User’s manual).



Mission 0 = pre-deployment or launch information

In the above example, there are 3 different float behaviours to record, (with park depth varying between 1500, 2000 and 1700 db). Each of these new behaviours requires a new mission number. This is in addition to the pre-deployment or launch info contained in mission 0:

CONFIG\_PARAMETER\_NAME = “CONFIG\_ParkPressure\_dBAR”   
CONFIG\_PARAMETER\_VALUE = "1500"   
CONFIG\_MISSION\_NUMBER = 1

CONFIG\_PARAMETER\_NAME = “CONFIG\_ParkPressure\_dBAR”   
CONFIG\_PARAMETER\_VALUE = "2000"   
CONFIG\_MISSION\_NUMBER = 2

CONFIG\_PARAMETER\_NAME = “CONFIG \_ParkPressure\_dBAR”   
CONFIG\_PARAMETER\_VALUE = "1700"   
CONFIG\_MISSION\_NUMBER = 3

A further example for a float with multiple missions is shown below. For this float the only change to the mission behaviour is the depth at which the float parks (with changes in two configuration parameters). However all configuration parameters from mission 1 must still be reported for each subsequent mission, even those that do not change. In this example there is one configuration mission 0 variable, which is set before launch, then there are another 6 variables that may change and control the float behaviour in subsequent missions (missions 1 to n). In this example, even though only CONFIG\_ParkPressure\_dBAR and CONFIG\_ParkPistonPosition\_COUNT are changing, the other mission variables are also repeated for each subsequent mission.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CONFIG\_PARAMETER\_NAME**  **(N\_CONF\_PARAM)** | **Mission Settings**  **(N\_MISSIONS, N\_CONF\_PARAM)** | | | | |
| CONFIG\_Mission\_Number | 0 | 1 | 2 | … | … |
| CONFIG\_PistonPositionPressureActivation\_COUNT | 100 | 100 | 100 |  |  |
| CONFIG\_ParkPressure\_dBAR |  | 1000 | 1500 |  |  |
| CONFIG\_ProfilePressure\_dBAR |  | 2000 | 2000 |  |  |
| CONFIG\_Direction\_LOGICAL |  | 1\* | 1\* |  |  |
| CONFIG\_AscentToSurfaceTimeout\_DecimalHour |  | 3 | 3 |  |  |
| CONFIG\_ParkPistonPosition\_COUNT |  | 113 | 75 |  |  |
| CONFIG\_MeasureBattery\_LOGICAL |  | 0 ^ | 0 ^ |  |  |
| … |  |  |  |  |  |
| … |  |  |  |  |  |

\* 1 = Ascending, 2 = Descending

^ 0 = No, 1 = Yes

### 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 : conductvity 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 | char SENSOR\_ACCURACY(N\_PARAM, STRING32); SENSOR\_ACCURACY:long\_name = "The accuracy of the sensor";  SENSOR\_ACCURACY:\_FillValue = " "; | Accuracy of the sensor.  Example: "8 micromole/l or 5%" |
| SENSOR\_RESOLUTION | char SENSOR\_RESOLUTION(N\_PARAM, STRING32); SENSOR\_RESOLUTION:long\_name = "The resolution of the sensor";  SENSOR\_RESOLUTION:\_FillValue =" "; | Resolution of the sensor.  Example : 0.001 micromole/l |

### 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); | Parameters measured on this float. | |
|  | | PARAMETER:long\_name = "List of parameters with calibration information"; | The parameter names are listed inreference table 3. | |
|  | | PARAMETER:conventions = "Argo reference table 3"; PARAMETER:\_FillValue = " "; | 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,STRING | Calibration equation for this parameter. Example : | |
|  | | 1024); PREDEPLOYMENT\_CALIB\_EQUATION:long\_name = | Tc = a1 \* T + a0 | |
|  | | "Calibration equation for this parameter"; PREDEPLOYMENT\_CALIB\_EQUATION:\_FillValue = " "; |  | |
| PREDEPLOYMENT\_CALIB | | char | Calibration coefficients for this equation. | |
| \_COEFFICIENT | | PREDEPLOYMENT\_CALIB\_COEFFICIENT(N\_PARAM,STRIN | Example : | |
|  | | G1024); | a1=0.99997 , a0=0.0021 | |
|  | | PREDEPLOYMENT\_CALIB\_COEFFICIENT:long\_name =  "Calibration coefficients for this equation";  PREDEPLOYMENT\_CALIB\_COEFFICIENT:\_FillValue = " "; |  | |
| PREDEPLOYMENT\_CALIB | char | | | Comments applying to this parameter | |
| \_COMMENT | PREDEPLOYMENT\_CALIB\_COMMENT(N\_PARAM,STRING  1024); | | | calibration. | |
|  |  | | | Example : | |
|  | PREDEPLOYMENT\_CALIB\_COMMENT:long\_name = | | | The sensor is not stable | |
|  | "Comment applying to this parameter calibration"; | | |  | |
|  | PREDEPLOYMENT\_CALIB\_COMMENT:\_FillValue = " "; | | |  | |

### Mandatory meta-data parameters

Mandatory (formerly known as highly desirable) meta-data parameters should be correctly filled according to the following table.

|  |  |  |
| --- | --- | --- |
| Mandatory meta-data | Mandatory format | Example |
| ARGO\_GROUP | see reference table ? | ARGO\_GROUP = “Core Argo” or “Argo Equivalent” or “Bio Argo”; |
| BATTERY\_TYPE | not empty | BATTERY\_TYPE = “Alkaline” or “Lithium” or “Alkaline and Lithium”; |
| BATTERY\_PACKS | not empty | BATTERY\_PACKS = “4DD Li + 1C Alk”; |
| CONTROLLER\_BOARD\_SERIAL\_NO\_PRIMARY | not empty | CONTROLLER\_BOARD\_SERIAL\_NO\_PRIMARY = 4567 |
| CONTROLLER\_BOARD\_TYPE\_PRIMARY | not empty | CONTROLLER\_BOARD\_TYPE\_PRIMARY = “APF9”; |
| DAC\_FORMAT\_ID | not empty | DAC\_FORMAT\_ID = ‘11’; |
| DATA\_CENTRE | see reference table 4 | DATA\_CENTRE = "AO ; |
| DATA\_TYPE | "Argo meta-data"; | DATA\_TYPE = "Argo meta-data"; |
| DATE\_CREATION | YYYYMMDDHHMISS | DATE\_CREATION = "20040210124422"; |
| DATE\_UPDATE | YYYYMMDDHHMISS | DATE\_UPDATE = "20040210124422"; |
| FIRMWARE\_VERSION | not empty | FIRMWARE\_ VERSION = “042606”; |
| FLOAT\_SERIAL\_NO | not empty | FLOAT\_SERIAL\_NO = “1679” |
| FORMAT\_VERSION | "2.2 "; | FORMAT\_VERSION = "2.2 "; |
| HANDBOOK\_VERSION | "1.2 "; | HANDBOOK\_VERSION = "1.2 "; |
| 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"; |
| MANUAL\_ VERSION | not empty | MANUAL\_ VERSION = “004” or “041708” |
| PARAMETER | see reference table 3 | PARAMETER ="PRES","TEMP","PSAL"; |
| PI\_NAME | not empty | PI\_NAME = “Susan Wijffels”; |
| PLATFORM\_FAMILY | see reference table | PLATFORM\_FAMILY = “subsurface profiling float”, “ITP”, “POPS”; |
| PLATFORM\_MAKER | see reference table | PLATFORM\_MAKER = “Optimare”; |
| PLATFORM\_NUMBER | XXXXX or XXXXXXX | PLATFORM\_NUMBER = "5900077 "; |
| PLATFORM\_TYPE | see reference table | PLATFORM\_TYPE = “SOLO” or “APEX” or “PROVOR”; |
| POSITIONING\_SYSTEM | see reference table 9 | POSITIONING\_SYSTEM = "ARGOS"; |
| PREDEPLOYMENT\_CALIB\_COEFFICIENT | not empty | PREDEPLOYMENT\_CALIB\_COEFFICIENT = “ser# = 3016 temperature coeffs: A0 = -0.0000 A1 = 0.0003 A2 = -0.0000 A3 = 0.0000”; |
| PREDEPLOYMENT\_CALIB\_EQUATION | not empty | PREDEPLOYMENT\_CALIB\_EQUATION = “Temperature ITS-90 = 1/ { a0 + a1[lambda nu (n)] + a2 [lambda nu^2 (n)] + a3 [lambda nu^3 (n)]} - 273.15 (deg C)”; |
| PTT | not empty | PTT = "23978 ";  Default value : "n/a" |
| SENSOR | not empty | SENSOR = “TEMP”, “PRES”,”CNDC”; |
| SENSOR\_MAKER | see reference table | SENSOR\_MAKER = “SEABIRD”; |
| SENSOR\_MODEL | see reference table | SENSOR\_MODEL = “SBE41” |
| SENSOR\_SERIAL\_NO | not empty | SENSOR\_SERIAL\_NO = “6785”; |
| SENSOR\_UNITS | not empty | SENSOR\_UNITS = “deg C”, “decibars”; |
| STANDARD\_FORMAT\_ID | reference table available at ADMT or CORIOLIS website | STANDARD\_FORMAT\_ID = “1”; |
| TRANS\_FREQUENCY | not empty | TRANS\_FREQUENCY = “1/44”; |
| TRANS\_SYSTEM | see reference table 10 | TRANS\_SYSTEM = "ARGOS "; |
| TRANS\_SYSTEM\_ID | not empty | TRANS\_SYSTEM\_ID = "14281"; |
| WMO\_INST\_TYPE | not empty | WMO\_INST\_TYPE = “846”; |

## Technical information format version 3.0

The format version 3.0 of Argo technical data will replace versions 2.3 and 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 3.0 format, all its technical files must be provided in version 3.0.

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.

For file naming conventions, see §4.1.

### Global attributes, dimensions and definitions

#### Global attributes

The global attributes section is used for data discovery. The following 8 global attributes should appear in the global section. The NetCDF Climate and Forecast (CF) Metadata Conventions (version 1.6, 5 December, 2011) are available from:

* http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.6/cf-conventions.pdf

// global attributes:

:title = "Argo float technical data file";

:institution = "CSIRO";

:source = "Argo float";

:history = "2011-04-22T06:00:00Z creation";

:references = "http://www.argodatamgt.org/Documentation";

:comment = "free text";

:user\_manual\_version = "3.0" ;

:Conventions = “Argo-3.0 CF-1.6" ;

#### 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 |
| STRING128,  STRING32 STRING8 STRING4 STRING2 | STRING128 = 128;  STRING32 = 32;  STRING8 = 8;  STRING4 = 4;  STRING2 = 2; | String dimensions from 2 to 128. |
| N\_TECH\_PARAM | N\_TECH\_PARAM = UNLIMITED; | Number of technical parameters. |

### 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 : A9IIIII";  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:long\_name = "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: long\_name = "File format version ";  FORMAT\_VERSION:\_FillValue = " "; | File format version  Example : «3.0» |
| HANDBOOK\_VERSION | char HANDBOOK\_VERSION(STRING4);  HANDBOOK\_VERSION: long\_name = "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:long\_name = "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 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\_UPDATE:\_FillValue = " "; | Date and time (UTC) of update of this file.  Format : YYYYMMDDHHMISS  Example :  20011230090500 : December 30th 2001 09 :05 :00 |

### 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.

All 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 |

## GDAC FTP directory file format

### 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 |

### 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 auto descriptive ASCII with comma separated values.

This directory file format is more detailed than the previous version 2.0, it will eventually replace it.

The directory file contains:

* A header with a list of general information: 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.

The detailed index file is limited to core mission "Argo sampling scheme" : temperature, salinity and oxygen observations.

Compression of the profile directory file

The profile directory file is compressed with gzip.

MD5 signature

For each update of the directory file, an MD5 signature is produced. The MD5 signature file allows user to check that the file he collected through FTP is identical to the original file.

Index file naming convention

* etc/argo\_profile\_detailled\_index.txt.gz
* etc/argo\_profile\_detailled\_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,gdac\_date\_creation,gdac\_date\_update,n\_levels   * 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) * gdac\_date\_creation : création date of the file on GDAC, YYYYMMDDHHMISS * gdac\_date\_update : update date of the file on GDAC, YYYYMMDDHHMISS * 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  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  … |

### 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 fiel 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 |

### Meta-data directory format

The metadata directory file describes all metadata 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.

|  |
| --- |
| 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://usgodae.usgodae.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 |

# Reference tables

## Reference table 1: data type

This table contains the list of acceptable values for DATA\_TYPE field.

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

## Reference table 2: Argo quality control flag scale

### Reference table 2: measurement flag scale

A quality flag indicates the quality of an observation.

The flags are assigned in real-time or delayed mode according to the Argo quality control manual available at:

* <http://www.argodatamgt.org/Documentation>

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | Not used in real-time. | 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 |

A list of real-time QC tests can be found in Table 11.

### Reference table 2a: profile quality flag

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

* QC flag values of 1, 2, 5, or 8 are GOOD data
* QC flag values of 9 (missing) are NOT USED in the computation

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 | ***N*** = 100%; All profile levels contain good data. |
| B | 75% <= ***N*** < 100% |
| C | 50% <= ***N*** < 75% |
| D | 25% <= ***N*** < 50% |
| E | 0% < ***N*** < 25% |
| F | ***N*** = 0%; No profile levels have good data. |

Example: a TEMP profile has 60 levels (3 levels contain missing values).

* 45 levels are flagged as 1
* levels are flagged as 2
* 7 levels are flagged as 4
* 3 levels are flagged as 9 (missing)

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

* PROFILE\_TEMP\_QC = “B”;

## Reference table 3: parameter code table

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Code | long name | standard name | unit | valid\_min | valid\_max | C\_Format FORTRAN\_Format resolution | Fill value |
| CNDC | ELECTRICAL CONDUCTIVITY | sea\_water\_electrical\_conductivity | mhos/m | 0.f | 8.5.f | %10.4f F10.4 0.0001f | 99999.f |
| PRES | SEA PRESSURE | sea\_water\_pressure | decibar | 0.f | 12000.f | %7.1f F7.1 0.1f | 99999.f |
| PSAL | PRACTICAL SALINITY | sea\_water\_salinity | psu | 0.f | 42.f | %9.3f F9.3 0.001f | 99999.f |
| TEMP | SEA TEMPERATURE IN SITU ITS-90 SCALE | sea\_water\_temperature | degree\_Celsius | -2.f | 40.f | %9.3f F9.3 0.001f | 99999.f |
| DOXY | DISSOLVED OXYGEN | moles\_of\_oxygen\_per\_unit\_mass\_in\_sea\_water | micromole/kg | 0.f | 650.f | %9.3f F9.3 0.001f | 99999.f |
| TEMP\_DOXY | SEA TEMPERATURE FROM DOXY SENSOR (ITS-90 SCALE) | temperature\_of\_sensor\_for\_oxygen\_in\_sea\_water | degree\_Celsius | -2.f | 40.f | %9.3f F9.3 0.001f | 99999.f |
| PRES\_DOXY | Sea water pressure at the depth of oxygen sampling | sea\_water\_pressure | decibar | 0.f | 12000.f | %7.1f F7.1 0.1f | 99999.f |
| VOLTAGE\_DOXY | Voltage reported by oxygen sensor | - | volt | 0.f | 100.f | %5.2f  F5.2  0.01f | 99999.f |
| FREQUENCY\_DOXY | Frequency reported by oxygen sensor | - | hertz | 0.f | 25000.f | %7.1f  F7.1  0.1f | 99999.f |
| COUNT\_DOXY | Count reported by oxygen sensor | - |  | 0.f | 100.f | %5.2f  F5.2  0.01f | 99999.f |
| BPHASE\_DOXY | Uncalibrated phase shift reported by oxygen sensor | - | degree | 10.f | 70.f | %8.2f  F8.2  0.01f | 99999.f |
| DPHASE\_DOXY | Calibrated phase shift reported by oxygen sensor | - | degree | 10.f | 70.f | %8.2f  F8.2  0.01f | 99999.f |
| MOLAR\_DOXY | Molar oxygen concentration reported  by the oxygen sensor | mole\_concentration\_of\_dissolved\_molecular\_oxygen\_in\_sea\_water | micromole/litre | 0.f | 650.f | %9.3f F9.3 0.001f | 99999.f |
| PHASE\_DOXY | Phase shift reported by oxygen sensor | - | degree | 10.f | 70.f | %8.2f  F8.2  0.01f | 99999.f |

New bio-geo-chemical parameters

Chlorophyll

CPHL

Nitrate

POC

Ph

CDOM

If new parameters are required, they have to be added to this table before they will be accepted.   
A request for new parameters can be sent to argo-dm-chairman@jcommops.org for approval and inclusion.

### Parameters from 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

### Oxygen related parameters

Some Argo floats perform Oxygen observation from different types of sensors, such as the Aandera Optode or the Seabird SBE 43/IDO.

To provide homogeneous observations from heterogeneous sensors, oxygen measurement should be converted and reported as DOXY.

* DOXY is the dissolved oxygen concentration estimated from the telemetered, calibrations coefficients and CTD values: PRES, TEMP (or TEMP\_DOXY) and PSAL.  
  Pressure and salinity compensations (e.g. Optode) are taken into account.
* DOXY unit: micromole/kg
* DOXY\_ADJUSTED is the dissolved oxygen concentration corrected for any sensor drift and offset. DOXY\_ADJUSTED is calculated from the other “ADJUSTED” fields.

Calibration coefficients, equations and references used to convert the telemetered variables in DOXY must be carefully documented in the metadata.

The Argo oxygen data management is described at:

* <http://www.argodatamgt.org/Documentation> , "Oxygen data processing"

## 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 |
| NM | NMDIS, China |
| 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 |

## Reference table 5: location classes

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

## 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. | **-** 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 (i.e. has no human intervention) and the measures are published and widely available. | **-** Measures are completely automated, or documentation is not 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 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.

## 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 |

## Reference table 8: instrument types

The instrument type codes come 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/wmotable_e.htm#ct1770>

|  |  |
| --- | --- |
| **Code number** | **Instrument** |
| 831 | P-Alace float |
| 840 | Provor, no conductivity |
| 841 | Provor, Seabird conductivity sensor |
| 842 | Provor, FSI conductivity sensor |
| *843* | *POPS ice Buoy/Float* |
| 844 | Arvor, Seabird 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 |
| 853 | Solo2, Seabird conductivity sensor |
| 855 | Ninja, no conductivity sensor |
| 856 | Ninja, SBE conductivity sensor |
| 857 | Ninja, FSI conductivity sensor |
| 858 | Ninja, TSK conductivity sensor |
| 859 | Profiling Float, NEMO, no conductivity |
| 860 | Profiling Float, NEMO, SBE conductivity  sensor |
| 861 | Profiling Float, NEMO, FSI conductivity sensor |

## Reference table 9: positioning system

|  |  |
| --- | --- |
| **Code** | **Description** |
| ARGOS | ARGOS positioning system |
| GPS | GPS positioning system |
| RAFOS | RAFOS positioning system |
| IRIDIUM | Iridium positioning system |

## Reference table 10: transmission system

|  |  |
| --- | --- |
| **Code** | **Description** |
| ARGOS | Argos transmission system |
| IRIDIUM | Iridium transmission system |
| ORBCOMM | Orbcomm transmission system |

## Reference table 11: QC test binary IDs

This table is used to record the result of the quality control tests in the history section.

The binary IDs of the QC tests are used to define the history variable HISTORY\_QCTEST, whose value is computed by adding the binary ID together, then translating to a hexadecimal number. An example is given on §5.3.

The test numbers and the test names are listed in the Argo Quality Control Manual:

* §2.1 “Argo Real-Time Quality Control Test Procedures on Vertical Profiles”, and
* §2.2 “Argo Real-Time Quality Control Test Procedures on Trajectories”

See <http://www.argodatamgt.org/Documentation> .

|  |  |  |
| --- | --- | --- |
| **Test number** | **QC test binary ID** | **Test name** |
| 1 | 2 | Platform Identification test |
| 2 | 4 | Impossible Date test |
| 3 | 8 | Impossible Location test |
| 4 | 16 | Position on Land test |
| 5 | 32 | Impossible Speed test |
| 6 | 64 | Global Range test |
| 7 | 128 | Regional Global Parameter test |
| 8 | 256 | Pressure Increasing test |
| 9 | 512 | Spike test |
| *10* | *1024* | *Top and Bottom Spike test (obsolete)* |
| 11 | 2048 | Gradient test |
| 12 | 4096 | Digit Rollover test |
| 13 | 8192 | Stuck Value test |
| 14 | 16384 | Density Inversion test |
| 15 | 32768 | Grey List test |
| 16 | 65536 | Gross Salinity or Temperature Sensor Drift test |
| 17 | 131072 | Visual QC test |
| 18 | 261144 | Frozen profile test |
| 19 | 524288 | Deepest pressure test |
| 20 | 1044576 | Questionable Argos position test |

## Reference table 12: history steps codes

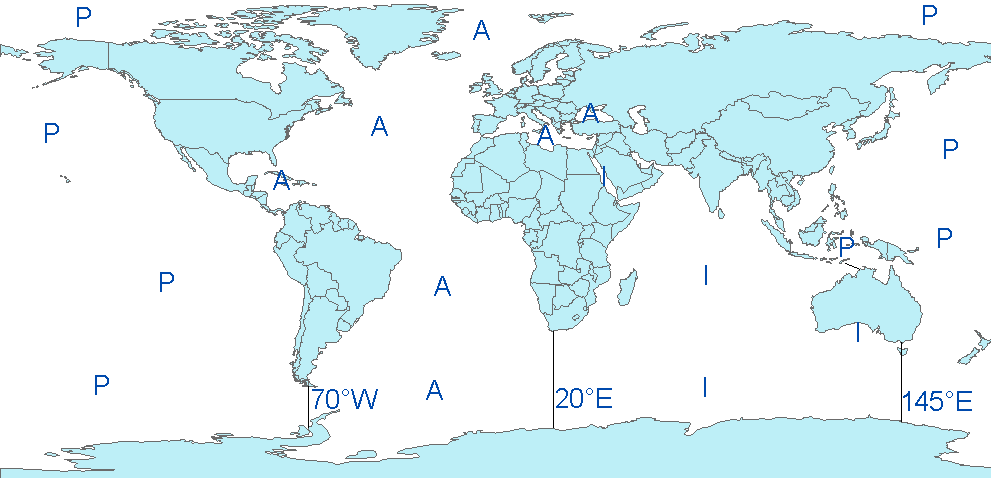
|  |  |
| --- | --- |
| Code | Meaning |
| ARFM | Convert raw data from telecommunications system to a processing format |
| ARGQ | Automatic QC of data reported in real-time has been performed |
| IGO3 | Checking for duplicates has been performed |
| ARSQ | Delayed mode QC has been performed |
| ARCA | 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 |
| COOA | Coriolis objective analysis performed |

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

## 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.

## Reference table 14: technical parameter names

All technical parameter names are standardized.

The list of technical parameter names is available at:

* <http://www.argodatamgt.org/Media/Argo-Data-Management/Argo-Documentation/General-documentation/Data-format/Argo-technical-parameter-names>

The naming convention for technical parameters is available at:

* <http://www.argodatamgt.org/Media/Argo-Data-Management/Argo-Documentation/General-documentation/Data-format/Technical-parameter-naming-convention>

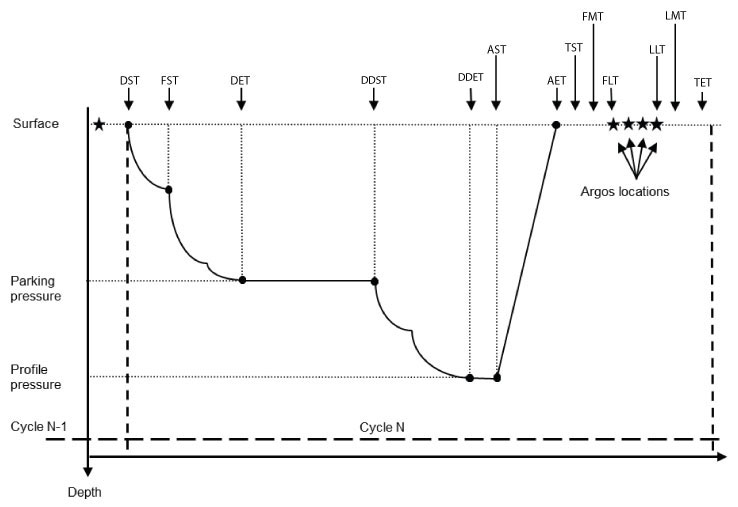
If new names are required as new variables are reported by a float, they must be added to this table before they will be accepted.

Request for new names can be sent to argo-dm-chairman@jcommops.org for approval and inclusion.

Older style files will be accepted for a short time and then all technical files must use approved names for standardized variables

## Reference table 15: codes of trajectory measurements performed within a cycle

In the trajectory file, each measurement is associated with a code (measurement\_code). The code allows matching the measurement with the part of the cycle. One or more measurements might be taken at the time of the action. The code allows matching the measurements with specific times and actions during each cycle.



Typical timings for a float cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DST | Descent Start Time | AST | Ascent Start Time | LLT | Last Location Time |
| FST | First Stabilization Time | AET | Ascent End Time | LMT | Last Message Time |
| DET | Descent End Time | TST | Transmission Start Time | TET | Transmission End Time |
| DDST | Deep Descent Start Time | FMT | First Message Time |  |  |
| DDET | Deep Descent End Time | FLT | First Location Time |  |  |

Measurement codes table

|  |  |  |  |
| --- | --- | --- | --- |
| Measure-ment  code | Meaning | Definition | Transmitted by listed float type |
| 0 | Launch | Launch time and location of the float | All float types |
| 1 | Measurements at the start of descent from the surface to the drift pressure (DST) | All measurements made at the start of the descent of the float to drift pressure.  Time (JULD\_DESCENT\_START)  Location  Surface pressure measurement (APEX)  Surface temperature measurement (PROVOR) | Time: PROVOR, ARVOR, SOLO-II,  WHOI SOLOIR, NEMO, NEMOIR,  APEX APF9, APEXIR APF9 |
| 11 | Stabilisations (more than one time/pres/etc means more than one stabilisation) | Pressure and time of stabilisation of float near surface [is this correct that they occur near the surface? I thought they (also) happen when the float reaches the target depth], soon after start of descent to drift pressure. | PROVOR, ARVOR, SIO SOLO, SOLO-II, POPS |
| 12 | Maximum pressure in descent to drift phase | Maximum pressure during the descent to drift pressure drift [how are 12 and 13 ordered time-wise? Maybe 12 comes after 13? If so: maybe reverse the order and the numbers.] | PROVOR |
| 13 | Measurements made during descent to drift pressure | Any measurements made during descent to drift pressure. Typically, Time and Pressure | PROVOR, SOLO-II |
| 2 | Measurements at the end of the descent from the surface to the drift pressure (DET) | All measurements made at the end of descent to drift pressure.  (1) Time when the float first reaches within 5% of drifting pressure (JULD\_DESCENT\_END).  (2) CTD at start of drift phase. | (1) Time:  PROVOR, ARVOR, SOLO-II, NEMO, NEMOIR, APEX APF9  (2) CTD:  WHOI SOLO  NINJA |
| 21 | Pressure recorded at the end of descent time out | Pressure recorded at the end of the float’s programmed time out during the descent to the drift pressure [somehow, this seems to be a 2, because the time-out seems to be needed to initiate the drift phase] | SIO SOLO |
| 22 | Measurements during drift phase | Pressure/temperature/salinity/etc taken during the drift phase. [If 21 becomes 2, then 22 becomes 21, and so on...] | APEX, SOLO, SOLO-II, PROVOR, ARVOR,  NEMO |
| 23 | Minimum pressure during drift phase | Minimum pressure recorded by the float during the drift phase | APEX, PROVOR |
| 24 | Maximum pressure during drift phase | Maximum pressure recorded by the float during the drift phase | APEX, PROVOR |
| 25 | Mean of measurements done during drift phase | Mean of several measurements done during drift phase |  |
| 26 | Median value of the measurements done during drift phase | Median value of the measurements done during drift phase [ is a difference between 25 and 26 that 25 has the word 'several' or should both have the word 'several'?] | NEMO |
| 3 | Measurements at the start of descent from the drift level to the deep profile level (DDST) | All measurements made at the start of descent from the drift level to deep profile level.  (1) Time (JULD\_DEEP\_DESCENT\_START)  (2) CTD at end of drift phase | (1) Time:  PROVOR (excluding PROVOR MT), ARVOR, SOLO-II, NEMO, NEMOIR, POPS  (2) CTD:  WHOI SOLO |
| 31 | Maximum pressure during the descent to deep profile pressure | Maximum pressure during the descent to deep profile pressure | PROVOR |
| 32 | Maximum pressure during cycle | Maximum pressure during cycle | NINJA |
| 33 | Measurements made during descent to the deep profile pressure | Any measurements made during descent including pressures with corresponding times. [should this one come before 31 and thus shift the others down by 1?] |  |
| 4 | Measurements at end of descent to deep profile pressure (DDET) | All measurements made when the float first reaches within 5% of deep profile pressure  Time (JULD\_DEEP\_DESCENT\_END) | PROVOR CTS3, ARVOR, SOLO-II, POPS |
| 41 | Time at the end of the Down-time | End date of the down-time parameter for APEX floats [is this significantly different from "4"?] | APEX |
| 5 | Measurements at the start of ascent (AST) | All measurements made at the time the ascent starts  (1) Time (JULD\_ASCENT\_START)  (2) CTD right before ascent start | (1) Time:  APEX APF9, PROVOR, ARVOR, SOLO-II, NEMO, NEMOIR, POPS  (2)???? |
| 51 | Measurements taken during ascent (excluding the profile) | All measurements taken during ascent (excluding the profile)  Times and pressures of CTD measurements | PROVOR, NINJA, SOLO-II |
| 52 | CTD taken near surface | CTD measurements taken near the surface [what is this? not the near-surface profile for SST and SSS, I assume.] | APEX |
| 6 | Measurements made at the end of the ascent (AET) | All measurements made at the end of the ascent.  Time (JULD\_ASCENT\_END) , Location, Surface Pressure | PROVOR, ARVOR, SOLO-II, NEMO, NEMOIR, APEX, POPS |
| 7 | Measurements at start of transmission (TST) | Time and location of the start of transmission for the float. | APEX APF9, APEXIR APF9, PROVOR, ARVOR, SOLO-II, NEMO, NEMOIR, POPS |
| 71 | First message received (FMT) | First time message received by telecommunications system – may or may not have a location fix. | All ARGOS floats, [Orbcomm floats? Iridium floats? Some may have it and others don't.] |
| 72 | Surface fixes | Surface times and locations during surface drift. Should be listed in chronological order. | All floats |
| 73 | Last message received (LMT) | Last time message received by the telecommunications system – may or may not have a location fix. | All Argos floats. [Orbcomm floats? Iridium floats? Some may have it and others don't.] |
| 81 | Measurements made on the surface | Any measurements made during surface drift  Surface pressure  Surface temperature (some APEX) | APEX, PROVOR  NEMO |
| 8 | Measurements at end of transmission (TET) | Time and location of the end of transmission for the float. | PROVOR, ARVOR, SOLO-II, APEXIR APF9 |

## Reference table 16: vertical sampling schemes

This variable differentiates the various vertical sampling schemes for multiple profiles from a single cycle. This variable can vary between cycles to accommodate floats with two-way communication capabilities. The profile with N\_PROF=1 is required to be the Primary sampling profile. Other profiles will have N\_PROF > 1 in any order. There can be only one Primary sampling profile, while other vertical sampling schemes can have more than one profile.

|  |  |  |
| --- | --- | --- |
| Code (STRING256)  FORMAT → name: nominal measurement type [full description]  [ ] indicates optional | N\_PROF | Code Description |
| Primary sampling: averaged [description]  *or* Primary sampling: discrete [description]  *or*  Primary sampling: mixed [description] | 1 | Primary CTD measurements and measurements from auxiliary sensors that are taken at the same pressure levels and with the same sampling method as the Primary CTD profile. For auxiliary sensor measurements it is not required that all pressure levels contain data. |
| Secondary sampling: averaged [description]  *or*  Secondary sampling: discrete [description]  *or*  Secondary sampling: mixed [description] | >1 | Excluding “Primary sampling”, this profile includes measurements that are taken at pressure levels different from the Primary CTD profile, or with sampling methods different from the Primary CTD profile. Measurements can be taken by the Primary CTD or by auxiliary sensors. |
| Near-surface sampling: averaged,  pumped/unpumped [description]  *or*  Near-surface sampling: discrete,  pumped/unpumped [description]  *or*  Near-surface sampling: mixed,  pumped/unpumped [description] | >1 | This profile includes near-surface measurements that are focused on the top 5dbar of the sea surface. (For the purpose of cross-calibration, this profile can extend deeper than the top 5dbar so as to overlap with the Primary sampling profile.) These measurements are taken at pressure levels different from the Primary CTD profile, or with sampling methods different from the Primary CTD profile. If the Primary sampling profile measures above 5dbar in the same manner as deeper data, there is no need to place the near-surface data here. |
| Bounce sampling: averaged [description]  *or*  Bounce sampling: discrete [description]  *or*  Bounce sampling: mixed [description] | >1 | This scheme contains profiles that are collected on multiple rises/falls during a single cycle. The profiles are temporally offset from each other and/or the Primary sampling profile. They can be sampled with the Primary CTD or with auxiliary sensors. |
| Use the term 'averaged' if the data in the profile are pressure binned averages using multiple data measurements (pollings) from a sensor. Use the term 'discrete' if the data in the profile are from a single polling from a sensor. If both methods are used in the profile, use the term 'mixed'. | | |

Example for a SOLOII V1.2 float

N\_PROF=1: "Primary sampling: averaged [nominal 2 dbar binned data sampled at 0.5 Hz from a SBE41CP]"

N\_PROF=2: "Near-surface sampling: discrete, pumped [shallowest polling of a SBE41CP]"

Note: In this example, by adding a single data point in N\_PROF=2, the size of the profile file will double.

Example for a Provor bio 5.0 float

This float is equipped with a Seabird CTD and a Wetlab Satrover optical sensor.

CTD sampling scheme:

* The threshold between deep sampling and upper sampling is 200 decibars.
* Upper sampling: 10 decibars slice thickness, 10 seconds sampling rate.
* Deep sampling: 25 decibars slice thickness, 10 seconds sampling rate.

Chlorophyll (optical) sampling scheme:

* The threshold between deep sampling and upper sampling is 300 decibars.
* Upper sampling: 1 decibar slice thickness, 1 seconds sampling rate.
* Deep sampling: 10 decibars slice thickness, 10 seconds sampling rate.
* Deepest sampling: 1000 decibars.

Description of the 2 vertical sampling schemes:

N\_PROF=1: "Primary sampling: averaged [10 seconds sampling, 25 decibars average from bottom to 200 decibars, 10 seconds sampling, 10 decibars average from 200 decibars to surface]"

N\_PROF=2: "Secondary sampling: averaged [10 seconds sampling, 10 decibars average from 1000 decibars to 300 decibars, 1 second sampling, 1 decibar average from 300 decibars to surface]"

Example for an APEX Iridium float with an Optode oxygen sensor and an auxiliary CTD for near-surface measurements

N\_PROF=1: "Primary sampling: averaged [2-dbar bin average]"

N\_PROF=2: "Secondary sampling: discrete [1.1 Hz CTD data, discrete DOXY]"

N\_PROF=3: "Near-surface sampling: discrete, unpumped [auxiliary CTD]"

## Reference table 17: Argo group

The Argo group is a metadata to identify general groups-types of floats.

|  |  |
| --- | --- |
| Argo group | Description |
| core | Argo float core mission (temperature and salinity, 10 days cycles, 2000decibar ascending profile, >=1500 decibar parking drift) , delayed mode data available within a few months |
| equivalent | Equivalent to a core Argo float with possible differences.  Examples : special funding, no delayed mode activity |
| coastal | A profiling float deployed in coastal area |
| bio | A profiling float equipped with bio-geo-chemical sensors. |

## Reference table 18: metadata configuration parameter names

All metadata variable names and configuration parameter names are standardized.

The list of metadata variable names is available at:

* <http://www.argodatamgt.org/Documentation> under “Argo Metadata Files”, “Metadata variable names”

The list of configuration parameter names is available at:

* <http://www.argodatamgt.org/Documentation> under “Argo Metadata Files”, “Configuration parameter names”

If new names are required as new variables are reported by a float, they must be added to this table before they will be accepted.

Please note that in this scheme, configuration parameter values are stored as numerals and therefore any parameters with logical or string input will require an equivalent numeric code to be added to the “Explanation” section of the Configuration parameter names table.

Request for new names can be sent to argo-dm-chairman@jcommops.org for approval and inclusion.

# 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.argodatamgt.org>

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.

More on GDACs organization:

* <http://www.argodatamgt.org/Media/Argo-Data-Management/Argo-Documentation/General-documentation/GDAC-organisation>

## 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:

a) R1900045\_003.nc, R1900045\_003D.nc

b) R1900046\_007\_01.nc, R1900067\_007\_02.nc, R1900067\_007\_03.nc

c) R1900046\_007D\_01.nc, R1900067\_007D\_02.nc, R1900067\_007D\_03.nc

d) 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

## Other data sources

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

Most 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>

# 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.

## 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 | 20030805000000 | 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.

## 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 | 20030805000000 | 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 |

## 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 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) |

## 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.

# DAC-GDAC data-management

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

## File submission from DAC to GDACs

Each DAC submits regularly all its new files to both USGODAE and Coriolis GDACs.

On both GDACs, each DAC has an ftp account with:

* a submit directory to submit files;
* a reject directory that contains the submitted file that were rejected by GDACs files format checker.

Seven types of files are accepted on GDAC:

* A float metadata file
* A float trajectory file
* A float technical data file
* An float’s cycle file
* The DAC’s geylist
* A removal file
* A compressed file containing a series of above files

Each GDAC checks the file format. If agreed, the file is pushed on the GDAC ftp server or processed. Otherwise, the file is moved in the reject directory, an error message is sent to the DAC contact point. Rejected files are kept in the reject directory for one month at least.

## Greylist files operations

### Greylist definition and management

The greylist is used for real-time operations, to detect a sensor malfunction. It is a list of suspicious or malfunctioning float sensors. It is managed by each DAC and available from both GDAC ftp site at:

* ftp://usgodae.org/pub/outgoing/argo/ar\_greylist.txt
* ftp://ftp.ifremer.fr/ifremer/argo/ar\_greylist.txt

The greylist is used in real-time QC test 15 to stop the real-time dissemination on the GTS of measurements from a sensor that is not working correctly.

The grey-list test is described in Argo quality control manual:

* http://www.argodatamgt.org/Media/Argo-Data-Management/Argo-Documentation/General-documentation/Argo-Quality-Control-manual-October-2009

Who/when/how to add a float in the greylist

Under the float’s PI supervision, a DAC inserts a float in the greylist when a sensor is suspicious or malfunctioning.

For each affected parameter, the start/end date of malfunction is recorded and the value of the real-time QC flag to be applied to each observation of this parameter during that period.

The problem is reported in the ANOMALY field of the meta-data file.

Who/when/how to remove floats from the greylist

In collaboration with the PI of the float, a DAC removes a float from the greylist when delayed mode quality control was performed and the suspicious sensor’s observations could be recovered after adjustment.

If the delayed mode quality control decided that the sensor observation cannot be recovered, the float remains in the greylist.

**How users should use the greylist**

The greylist provides an easy way to get information on suspicious floats.

However, the best information on a float’s sensors bad behaviour is recorded in the ANOMALY field of the meta-data file.

### 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 is 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.

## 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 removed files are kept for 3 months in the etc/removed directory and erased after that delay.

## Compressed files data distribution

Once a month, a compressed version of dac and geo directories is distributed on :

<ftp://ftp.ifremer.fr/etc/argozip>

A compléter

## Compressed files data submission

A DAC can push to GDAC a compressed file containing a series of files. The GDAC will process all its content. This is useful to submit an important batch of files (example : delayed mode data).

The compressed file is a tar-ed file or directory compressed with gzip.

Compressed file naming convention

XXX.tar.gz

* XXX : the compressed file name, with no specific requirement
* tar : tar suffix
* gz : gzip suffix

Example : coriolis-201210-DelayedMode.tar.gz

1. R : real-time data with no adjustment [↑](#footnote-ref-1)
2. A : real-time data adjusted automatically in real-time. [↑](#footnote-ref-2)
3. D : delayed-mode data [↑](#footnote-ref-3)