

ARGO USER'S MANUAL

Version 3.43.0

June 23rd 2025

ARGO

part of the integrated global observation strategy





Argo data management

Argo User's manual

Authors: Argo data management team

How to cite this document

Argo User's manual, June 2025, <http://dx.doi.org/10.13155/29825>

Table of contents

Table des matières

| | |
|--|----|
| 1 Introduction | 14 |
| 1.1 Notice on file format change transition | 14 |
| 1.2 User Obligations | 14 |
| 1.3 Disclaimer | 14 |
| 1.4 Further information sources and contact information | 15 |
| 1.5 Argo program, data management context | 15 |
| 1.6 Argo float cycles | 15 |
| 1.7 Real-time, real-time adjusted data and delayed mode data | 16 |
| 2 Formats description | 17 |
| 2.1 Overview of the formats | 17 |
| 2.2 Core-Argo profile format version 3.1 | 17 |
| 2.2.1 Global attributes | 18 |
| 2.2.2 Dimensions | 18 |
| 2.2.3 General information on the profile file | 19 |
| 2.2.4 General information for each profile | 19 |
| 2.2.5 Measurements for each profile | 23 |
| 2.2.6 Calibration information for each profile | 25 |
| 2.2.7 History information for each profile | 26 |
| 2.3 Argo trajectory format version 3.2 | 28 |
| 2.3.1 Global attributes | 29 |
| 2.3.2 Dimensions and definitions | 30 |
| 2.3.3 General information on the trajectory file | 30 |
| 2.3.4 General information on the float | 31 |
| 2.3.5 N_MEASUREMENT dimension variable group | 32 |
| 2.3.6 N_CYCLE dimension variable group | 38 |
| 2.3.7 Scientific calibration section | 45 |
| 2.3.8 JULD calibration section | 46 |
| 2.3.9 History information | 47 |
| 2.4 Metadata format version 3.1 | 48 |
| 2.4.1 Global attributes | 48 |
| 2.4.2 Dimensions and definitions | 49 |
| 2.4.3 General information on the metadata file | 50 |
| 2.4.4 Float characteristics | 50 |
| 2.4.5 Float deployment and mission information | 53 |
| 2.4.6 Configuration parameters | 55 |
| 2.4.7 Float sensor and parameter information | 59 |
| 2.4.8 Float calibration information | 61 |
| 2.4.9 Mandatory metadata parameters | 61 |

| | |
|--|----|
| 2.5 Technical information format version 3.1 | 64 |
| 2.5.1 Global attributes | 64 |
| 2.5.2 Dimensions and definitions | 64 |
| 2.5.3 General information on the technical data file | 65 |
| 2.5.4 Technical data | 65 |
| 2.6 B-Argo profile format additional features | 67 |
| 2.6.1 Pressure axis management in core-Argo and b-Argo profile files | 67 |
| 2.6.3 Management of multi-dimensional parameters | 68 |
| 2.6.4 Parameter variable type | 69 |
| 2.6.5 PARAMETER_DATA_MODE | 69 |
| 2.6.6 N_PARAM management in b-Argo profile files | 70 |
| 2.6.7 QC and ADJUSTED variables in b-Argo profile files | 70 |
| 2.6.8 PARAMETER and PARAMETER_SENSOR need expansion | 71 |
| 2.6.9 DATA_TYPE dimension extended from 16 to 32 characters | 71 |
| 2.6.10 Predeployment calibration variable dimension | 71 |
| 2.7 GDAC FTP directory file format | 71 |
| 2.7.1 Profile directory file format 2.0 | 71 |
| 2.7.2 Profile directory file format version 2.1 | 72 |
| 2.7.3 Trajectory index file format 2.0 | 74 |
| 2.7.4 Metadata directory file format 2.0 | 75 |
| 3 Reference tables | 76 |
| 3.1 Reference table 1: data type | 76 |
| 3.2 Reference table 2: Argo quality control flag scale | 76 |
| 3.2.1 Reference table 2: quality control flag scale | 76 |
| 3.2.2 Reference table 2a: overall profile quality flag | 77 |
| 3.3 Reference table 3: parameter code table | 77 |
| 3.3.1 How to indicate multiple sensors for the same parameter | 78 |
| 3.3.2 Oxygen related parameters | 78 |
| 3.4 Reference table 4: data centres and institutions codes | 79 |
| 3.5 Reference table 5: location classes | 79 |
| 3.6 Reference table 6: data state indicators | 79 |
| 3.7 Reference table 7: history action codes | 80 |
| 3.8 Reference table 8: instrument types | 80 |
| 3.9 Reference table 9: positioning system | 80 |
| 3.10 Reference table 10: transmission system | 80 |
| 3.11 Reference table 11: QC test binary IDs | 80 |
| 3.12 Reference table 12: history steps codes | 81 |
| 3.13 Reference table 13: ocean codes | 81 |
| 3.14 Reference table 14: technical parameter names | 81 |
| 3.15 Reference Table 15: codes of trajectory measurements performed within a cycle | 82 |
| 3.16 Reference table 16: vertical sampling schemes | 83 |
| 3.17 Reference table 17: obsolete | 85 |

| | |
|--|----|
| 3.18 Reference table 18: metadata configuration parameter names..... | 85 |
| 3.19 Reference table 19: STATUS flags..... | 85 |
| 3.20 Reference table 20: GROUNDED flags..... | 86 |
| 3.21 Reference table 21: REPRESENTATIVE_PARK_PRESSURE_STATUS..... | 86 |
| 3.22 Reference table 22: PLATFORM_FAMILY..... | 86 |
| 3.23 Reference table 23: PLATFORM_TYPE..... | 86 |
| 3.24 Reference table 24: PLATFORM_MAKER..... | 86 |
| 3.26 Reference table 26: SENSOR_MAKER..... | 86 |
| 3.27 Reference table 27: SENSOR_MODEL..... | 86 |
| 3.28 Reference table 28: CONTROLLER_BOARD_TYPE_PRIMARY..... | 87 |
| 3.29 Reference table 29: BATTERY_TYPE..... | 87 |
| 3.30 Reference table 30: BATTERY_PACKS (optional field)..... | 88 |
| 3.31 Reference table 31: ENDING_CAUSE and ENDING_CAUSE_CATEGORY..... | 88 |
| 3.32 Reference table 32: SPECIAL_FEATURE..... | 89 |
| 3.33 Reference table 40 : PI_NAME..... | 89 |
| 4 Data access..... | 90 |
| 4.1 File naming convention on GDACs..... | 90 |
| 4.1.1 Core-Argo individual profile files..... | 90 |
| 4.1.2 B-Argo data file..... | 91 |
| 4.1.3 Argo trajectory data file..... | 91 |
| 4.1.5 Metadata file..... | 92 |
| 4.1.6 Technical Data file..... | 92 |
| 4.1.7 Auxiliary data files..... | 92 |
| 4.2 Other data sources..... | 92 |
| 5 Using the History section of the Argo netCDF Structure..... | 93 |
| 5.1 Recording information about the Delayed Mode QC process..... | 93 |
| 5.2 Recording processing stages..... | 93 |
| 5.3 Recording QC Tests Performed and Failed..... | 94 |
| 5.4 Recording changes in values..... | 95 |
| 6 DAC-GDAC data-management..... | 97 |
| 6.1 File submission from DAC to GDACs..... | 97 |
| 6.3 Archived DOI datasets..... | 97 |
| 6.4 Compressed files data submission..... | 98 |
| 7 Glossary, definitions..... | 99 |
| 7.1 Float..... | 99 |
| 7.2 Sensor..... | 99 |
| 7.3 Parameter measured by the sensor..... | 99 |
| 7.4 Calibration of the parameter measured by the sensor..... | 99 |
| 7.5 Principal Investigator (PI)..... | 99 |
| 7.6 Global Data Assembly Centre (GDAC)..... | 99 |
| 7.7 Data Assembly Centre (DAC)..... | 99 |
| 7.8 GTS..... | 99 |

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 | metadata 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 accommodate 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 metadata 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 : example of sensor type in metadata |
| 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 metadata 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 description of table 11. 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 Scanderbeg : 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 Van Wijk : metadata format version 2.4 |
| 2.4 | 10/02/2012 | §2.2.3 Thierry Carval : global attributes and parameter attributes for CF compatibility |
| 2.4 | 13/02/2012 | §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 | §2.4.6: CONFIGURATION_MISSION_COMMENT: FillValue is equal to " "; |
| 2.41 | 07/11/2012 | §2.4 : metadata format, additions from Esmeé |
| 2.41 | 10/11/2012 | §6.3 : compressed files distribution |
| 2.41 | 10/11/2012 | §6.5 : compressed files data submission |
| 2.41 | 07/01/2013 | §2.2 §2.3 §2.4 §2.4 all format versions renamed as 3.0 |
| 3.0 | 03/05/2013 | §2.3 : trajectory format entirely revisited by Megan Scanderbeg, Jean-Philippe Rannou and John Gilson |
| 3.01 | 06/05/2013 | §3.15 : move reference table 15 from §2.3 to §3.15 and replace the previous version. |
| 3.01 | 06/05/2013 | §3.20 : reference table 20 "grounded flags" updated |
| 3.01 | 06/05/2013 | §3.21 : reference table 21 added, "representative park pressure status" |
| 3.01 | 06/05/2013 | §4.1 : file naming convention, update on trajectory file names |
| 3.01 | 06/05/2013 | §2.3.6 : table n_cycle revisited, 2 variables added |
| 3.01 | 19/06/2013 | §2.2 : more information to describe the relations between profiles – vertical sampling scheme – cycle |
| 3.01 | 19/06/2013 | §2.2.4 : Replace INST_REFERENCE with FLOAT_SERIAL_NO |
| 3.01 | 19/06/2013 | §2.2.4 : FIRMWARE_VERSION : dimension set to STRING16, remove attribute "conventions" |
| 3.01 | 19/06/2013 | §2.2.4 : add a CONFIG_MISSION_NUMBER variable in the profile file |
| 3.01 | 19/06/2013 | §2.4.4 : remove IMEI number |
| 3.01 | 19/06/2013 | §2.4.4 : TRANS_FREQUENCY dimension set to (N_TRANS_SYSTEM, STRING16) |
| 3.01 | 19/06/2013 | §2.4.4: PLATFORM_TYPE, STANDARD_FORMAT_ID, DAC_FORMAT_ID : add a link to the draft reference table (an Internet spreadsheet) |
| 3.01 | 19/06/2013 | §2.4.4: remove variable SAMPLING_MODE |
| 3.01 | 19/06/2013 | §2.4.4: remove variable ARGO_GROUP |
| 3.01 | 19/06/2013 | §2.4.4: BATTERY_PACKS moved from mandatory metadata to highly desirable metadata |
| 3.01 | 19/06/2013 | §3.3: reference table 3: parameter code table, oxygen related parameters . Remove pres_doxy . Update long names . Add TPHASE_DOXY, C1PHASE_DOXY, C2PHASE_DOXY, MLPL_DOXY Resolution is mandatory but its content is sensor dependent. |
| 3.01 | 19/06/2013 | §3.8 : reference table 8: instrument types Add 10 new instruments types |
| 3.01 | 19/06/2013 | §3.17 : reference table 17: obsolete Remove reference table 17. |
| 3.01 | 19/06/2013 | §4.1 : trajectory : new file naming convention |
| 3.01 | 20/06/2013 | §2.2.4 : Add the PLATFORM_TYPE variable Add an N_PROF dimension in FLOAT_SERIAL_NO. |
| 3.01 | 20/06/2013 | §2.3.4 : add the PLATFORM_TYPE variable |
| 3.01 | 26/06/2013 | §2.2.4 : CONFIG_MISSION_NUMBER long name updated |
| 3.01 | 26/06/2013 | Remarks from Megan Scanderbeg §2.3.1 : add a global attribute in the profile file : :comment_on_resolution = "PRES variable resolution depends on measurement code"; §2.3.4 : remove the STRING2 dimension from SATELLITE_NAME char SATELLITE_NAME(N_MEASUREMENT, STRING2); §3.15 : remove the old reference table 15 (but keep the new one) |
| 3.01 | 27/06/2013 | configuration_mission_number graphic updated |
| 3.01 | 28/06/2013 | §2.3.2 : add a STRING32 dimension |
| 3.02 | 18/07/2013 | §3.3 : remove the sensor names from oxygen related parameter's long name attributes |
| 3.03 | 28/08/2013 | §2.4.2: add a STRING64 dimension §2.4.4: CONTROLLER_BOARD_SERIAL_NO_PRIMARY long_name : remove "The " §2.4.1: //global attributes:user_manual_version="3.03" |
| 3.03 | 30/08/2013 | §2.4.2 : add a STRING32 dimension in definition column |
| 3.1 | 30/11/2013 | §1.2 : use DOI for data and document citation §1.6 : update cycle definition §2.2.4 §2.3.4 §2.4.4 :FLOAT_SERIAL_NO dimension extended to 32 instead of 16 §2.2.4, §2.3.4, §2.4.4 : add a "conventions" attribute to PLATFORM_TYPE §2.3.5 : add a "resolution attribute" to JULD and JULD_ADJUSTED §2.3.5.1 How to report unusual Pressure resolutions in the N_MEASUREMENT variable group of the TRAJ file §2.4 : general revision of metadata format; §2.4.6 adding launch configuration parameters; §2.4.7 discriminate float sensor and float parameter information §2.4.4 : add a "conventions" attribute to PLATFORM_FAMILY and PLATFORM_MAKER §2.4.5 : add STARTUP_DATE and STARTUP_DATE_QC §2.4.7.1 : add a "conventions" attribute to SENSOR_MAKER and SENSOR_MODEL |

| | | |
|-----|------------|--|
| | | <p>§2.6 : B-Argo profile additional format features</p> <p>§3.22, §3.23, §3.24, §3.25, §3.26, §0 : new reference tables</p> <p>§3.3 : separation of core-Argo and B-Argo parameters in reference table 3</p> <p>§0 : separation of core-Argo data files and B-Argo data files</p> <p>§7 : add a glossary</p> |
| 3.1 | 19/12/2013 | §3.1 : add the B-Argo profile and B-Argo trajectory data types in reference table 1. |
| 3.1 | 21/02/2014 | §2.6.1 : manage pressure axis between core-argo and b-argo files |
| 3.1 | 18/04/2014 | <p>§3.3 : revision of valid_min and valid_max on TEMP, PSAL, DOXY to be compliant with Argo quality control manual.</p> <p>§2.3.6: add a "conventions" attribute to each cycle timing</p> <p>§2.2.3, §2.3.3: add a FILE_TYPE variable to discriminate C, B or M files.</p> <p>§2.2.6: list parameter names in the PARAMETER variables even if no calibration is performed yet.</p> <p>Page 1 : add a DOI to the manual</p> <p>Page 2 : add a "How to cite this document" chapter</p> <p>§6.4: add a chapter on Argo DOIs</p> |
| 3.1 | 22/04/2014 | §4 : change the usgodae ftp server |
| 3.1 | 06/05/2014 | §2.2.4 : add "JULD:resolution = X;" and "JULD_LOCATION:resolution = X;" |
| 3.1 | 21/05/2014 | <p>§1.6 : improve end cycle definition (John)</p> <p>§2.2.5.1 : do not report resolution attribute if not known</p> <p>§2.2, 2.3, 2.4, 2.5 : a proper definition replaces "eponymous" : "The version number of the user manual"</p> <p>§3.22....3.27 : add a link to Matthieu's tiny url reference tables</p> <p>§2.2, §2.3 : no need for a file_type variable</p> <p>precisions for use of array N_VALUES##</p> |
| 3.1 | 31/05/2014 | <p>§2.6.1 : core-Argo b-Argo parameters separation, improvement of the chapter, add an example</p> <p>§2.6.3 : management of spectral parameters, revision of the chapter name</p> <p>§3.11 : reference table 11, addition of the new QC tests 21 and 22</p> <p>§2.3.6 : JULD_DEEP_PARK_START typo correction in long name</p> |
| 3.1 | 03/06/2014 | <p>§2.3 : update REPRESENTATIVE_PARK_PRESSURE:long_name</p> <p>§2.3 : add a long_name attribute to all *_STATUS variable</p> <p>§2.3 : JULD_ASCENT_END:long_name...remove the word "float's"</p> <p>§2.3 : replace any "0..N" with "0...N"</p> |
| 3.1 | 23/06/2014 | <p>§2.4 : update config_parameter_value and launch_config_parameter_value fill values</p> <p>§2.3 : update/create long name attributes in trajectory section</p> <p>General : replace all "as part of day" with "as parts of day"</p> <p>§2.6.9 : extend data_type dimension from 16 to 32 in b-argo and merged files</p> |
| 3.1 | 09/07/2014 | <p>§2.4.7.2 : PARAMETER_UNITS dimension set to 32 instead of 16</p> <p>§2.2.6 : SCIENTIFIC_CALIB_DATE:conventions = "YYYYMMDDHHMISS";</p> <p>§2.3.6 JULD_DEEP_DESCENT_END:long_name = "Deep descent end date of the cycle";</p> <p>§2.6.1.2 : Pressure axis management in core-Argo and b-Argo trajectory files</p> <p>§2.6.2 Cycle timings management in core-Argo and b-Argo trajectory files</p> |
| 3.1 | 11/07/2014 | <p>§2.2.4 : CONFIG_MISSION_NUMBER:long_name updated to be consistent with trajectory and metadata formats.</p> <p>§2.3.6 : typo error corrected in JULD_FIRST_LOCATION:long_name and JULD_FIRST_LOCATION_STATUS:long_name</p> <p>§2.2.5 : *_ADJUSTED_ERROR:long_name updated to be consistent with trajectory format.</p> <p>§3.25 §3.26 §0 : update reference tables 25, 26 and 27 (new sensors and platforms)</p> |
| 3.1 | 15/07/2014 | <p>§2.4.6.2 : config_mission_number dimension in prof and traj files</p> <p>§2.3.6 : update long name JULD_LAST_LOCATION_STATUS:long_name = "Status of date of latest location";</p> <p>§3.3 : update of B-Argo parameters and I-Argo parameters</p> |
| 3.1 | 18/07/2014 | §2.2.4 §2.3.4 §2.4.4 : extend FIRMWARE_VERSION from STRING16 to STRING32 |
| 3.1 | 15/09/2014 | <p>§2.3.5.1 : clarification on how to report unusual pressure resolutions in the N_MEASUREMENT variable group of the TRAJ file</p> <p>§2.2.5.1 : how to report unusual parameter resolutions in a profile</p> |
| 3.1 | 07/11/2014 | §2.6 : the variables PROFILE_PRES_QC, PRES_QC, PRES_ADJUSTED, PRES_ADJUSTED_ERROR are not duplicated in the b-Argo profile files. |
| 3.1 | 18/03/2015 | §2.2.5: <PARAM>_ADJUSTED_ERROR:long_name updated to be consistent between profile and trajectory. |
| 3.2 | 10/11/2015 | <p>§2.2.5 clarification on DATA_MODE</p> <p>§2.6.8 clarification on PARAMETER_DATA_MODE</p> <p>§2.6.7 clarification on QC and ADJUSTED variables in b-Argo profile files</p> <p>§6.2.2 the GDAC greylist operation is transferred in the GDAC cookbook</p> <p>§6.3 the "GDAC file removal" chapter is transferred to the GDAC cookbook</p> |
| 3.2 | 29/12/2015 | <p>§3.4 add MBARI in reference table 4 from Annie Wong</p> <p>All chapters: various corrections from Jean-Philippe Rannou (typo, unit names in examples)</p> <p>§2.6.6 N_PARAM management in b-Argo profile files from Annie Wong</p> <p>§2.3 trajectory format revision from Megan Scanderbeg</p> |
| 3.2 | 02/10/2016 | Page 2: replace the obsolete "Authors" list with "Argo data management team". The DOI landing page lists the contributors. |
| 3.2 | 02/10/2016 | §4.1 note on cycle numbers greater than 999 |
| 3.2 | 02/10/2016 | §3.2.1 QC flag 8, replace "interpolated" with a more general "estimated". |

| | | |
|------|------------|---|
| 3.2 | 08/10/2016 | §2.4.6.3 add a chapter "highly desirable configuration parameters" |
| 3.2 | 10/10/2016 | §2.6.8 Parameter names on 64 characters |
| 3.2 | 10/10/2016 | §2.6.10 Predeployment calibration variable dimension When necessary (such as for Nitrate sensor), the string length is set to 4096. |
| 3.2 | 14/10/2016 | §2.6.3 clarification PARAM_QC do not have a N_VALUES dimension |
| 3.2 | 17/10/2016 | §2.6.7 clarification on *_ADJUSTED, *_ADJUSTED_QC, and *_ADJUSTED_ERROR. They are optional as a group they must either all be present or all be absent |
| 3.2 | 17/10/2016 | §3.5 Reference table 5 "Location classes" : add "A", "B" and "Z" Argos location classes |
| 3.2 | 06/08/2016 | §4.1.1 Note on <R/D> in profile file name |
| 3.2 | 08/08/2017 | §2.7.1 Note on date or position with a bad quality control flag in index files |
| 3.2 | 11/09/2017 | §3.25 §3.26 §3.27 update SENSOR, SENSOR_MAKER and SENSOR_MODEL according to their online content |
| 3.3 | 23/10/2019 | §1.7 Real-time, real-time adjusted data and delayed mode data: additional precisions for BGC data §2.2.4 allow CONFIG_MISSION_NUMBER = 0 ADMT19 Action #19 §2.2.4 Note on POSITION_QC is 8 §2.2.4 §2.3.4 §2.4.4 FIRMWARE_VERSION dimension changed to length 64, but still accept 32, ADMT19 action 37 §3.2 remove "not used in real time" for QC flag 2 §3.4 add Laboratoire de Villefranche in reference table 4 §3.5 add accuracy for Iridium, Beidou, GLONASS, GALILEO, GNSS, RAFOS accuracy and estimated position §3.9 add positioning system for BEIDOU, GLONASS, GALILEO, GNSS, NONE §3.15 add measurement codes for in water/in air samples §3.15 add RAFOS MC (measurement codes) §3.15 remove 1076-1099 and 1100 MC (measurement codes) §3.21 typo correction (PET instead of DDST) §6.3 "compressed file distribution" chapter remove - ADMT19 action #17 "Remove zipped GDAC files from Coriolis. Only serve them on DOI page." §6.3 Archived DOI dataset description §2.2.4 add POSITION_ERROR and POSITION_COMMENT §3.28 add reference table 28:CONTROLLER_BOARD_TYPE_PRIMARY §3.29 add reference table 29: BATTERY_TYPE §3.30 add reference table 30: BATTERY_PACKS (optional field) |
| 3.4 | 28/11/2019 | Initiate version v3.4 |
| 3.4 | 28/02/2020 | §4.1.7 Auxiliary data files chapter |
| 3.4 | 28/12/2020 | §2.1 NetCDF references, UDUNIT references §2.2.2 CF convention reference §2.2.2 user_manual_version and Convention version 3.4 §2.2.4 POSITION_ERROR_REPORTED and POSITION_ERROR_ESTIMATED revision §2.3 Argo trajectory format 3.2 §2.3.2 Trajectory additional dimensions §2.3.4 trajectory_parameters dimension increased §2.3.5 addition of variables TRAJECTORY_PARAMETER_DATA_MODE, JULD_DATA_MODE, <PARAM>_MED, <PARAM>_STD §2.3.7 add a trajectory Scientific calibration section §2.3.8 HISTORY_PARAMETER dimension added §2.4.1 user_manual_version and Convention version 3.4 §2.4.4 add PLATFORM_WIGOS_ID variable §2.4.9 remove PARAMETER duplicate §2.5.4 add "Timeseries of technical data" chapter §3 mention Argo vocabulary server §3.2.1 QC flags comments revisited §3.3 add intermediate parameters for Core-Argo §3.3 add parameter NB_SAMPLE_<short_sensor_name> §3.4 add IOPAN and AWI §3.5 revisit location classes label §3.8 add new instruments §3.15 revisit "Relative Measurement code" §3.20 revisit grounded flags §4.1.2.2 BGC Argo synthetic profile §4.1.7 add Auxiliary data files |
| 3.4 | 20/01/2021 | Additional feedbacks from Annie Wong, Claudia Schmid, Jean-Philippe Rannou, Gabi Semolini §2.4.2 add string17 dimension for PLATFORM_WIGOS_ID §3.20 warning on grounded flags "S" and "U", could become obsolete §4.1.2.2 link to BGC-Argo synthetic profiles documentation |
| 3.41 | 20/07/2021 | §2.1 "Overview of the formats" : Argo data uses the UDUNITS convention when applicable §2.3.3 "Trajectory general information" : use 3.2 in FORMAT_VERSION example §2.3.4 "Trajectory general information on the float" : TRAJECTORY_PARAMETERS: "List of available parameters" instead of "List of available parameters for the station" §2.3.5 "Trajectory measurements group" : <param>_adjusted, _adjusted_qc, _adjusted_error are not mandatory for intermediate parameters |

| | | |
|--------|------------|--|
| | | <p>§2.3.7 "Trajectory scientific calibration" : Replace SCIENTIFIC_CALIB_PARAMETER STRING256 with STRING64</p> <p>§2.3.8 "Trajectory history information" : HISTORY_PARAMETER: "Parameter action is performed on", instead of "Station parameter action is performed on"</p> <p>§2.5.2 "Technical information dimensions" : add N_TECH_MESAUREMENT optional dimension</p> <p>§2.5.4 "Technical data" : <TECH_PARAM> long name and units attributes</p> <p>§3.3 "Reference tables": in Table 3, NB_SAMPLE_<short_sensor_name> is now NB_SAMPLE_<parameter_sensor_name></p> |
| 3.41.1 | 05/07/2022 | <p>§2.3.2 N_CALIB_JULD dimension added</p> <p>§2.3.2 N_CALIB replaced with N_CALIB_PARAM</p> <p>§2.3.5 AXES_ERROR_ELLIPSE_MAJOR and AXES_ERROR_ELLIPSE_MINOR long names updated (semi-major and semi-minor ellipses) see https://github.com/euroargodev/ArgoNetCDF/issues/3</p> <p>§2.3.5 <PARAM>_MED and <PARAM>_STD removed from the manual, see https://github.com/euroargodev/ArgoNetCDF/issues/4</p> <p>§2.3.7 N_CALIB replaced with N_CALIB_PARAM</p> <p>§2.3.8 JULD calibration chapter added</p> <p>§3.2 Reference Table 2a: clean up of comments</p> <p>§3.2 Notes on QC removed from the manual, these notes are maintained in the QC manuals.</p> <p>§3.31 reference table 31 added: ENDING_CAUSE and ENDING_CAUSE_CATEGORY</p> <p>§all document: remove tinyurl links (use the original link)</p> |
| 3.42.0 | 26/05/2025 | <p>§2.2.1 optional global attribute "id" contains "Argo GDAC DOI"</p> <p>Argo GDAC data DOI global attribute is optional and recommended https://github.com/nvs-vocabs/ArgoVocabs/issues/43 and https://github.com/OneArgo/ADMT/issues/72</p> <p>§3.22 ... §3.28: remove the remaining tyinurl</p> <p>§3.1 add "Argo auxiliary meta-data" data type</p> <p>Throughout the document: added links to individual Reference tables in the NVS</p> <p>§2.2.4 §2.3.4 §2.4.4 PI_NAME has a controlled vocabulary https://github.com/OneArgo/ArgoVocabs/issues/6</p> <p>§2.4.7.1 SENSOR_FIRMWARE_VERSION https://github.com/OneArgo/ArgoVocabs/issues/55</p> <p>§2.4.5 deployment_platform dimension extended to string128 https://github.com/OneArgo/ArgoVocabs/issues/2#issuecomment-243742119</p> <p>§3.2.1: add a comment to provide examples when *_qc= `` can occur as per https://github.com/OneArgo/ArgoVocabs/issues/21</p> <p>§3.33 add PI_NAME reference table</p> <p>§3.11 change from grey list to supplemental sensor exclusion list in reference table 11 (example) https://github.com/OneArgo/ADMT/discussions/70</p> <p>§6.2 remove greylis paragraph https://github.com/OneArgo/ADMT/discussions/70</p> <p>§6.2 "Grey list" removed: the greylis management is internal, documented in the Quality control manual.</p> |
| 3.43.0 | 23/06/2025 | <p>§2.2.4 §2.3.4 §2.4.4 PROJECT_NAME comment revision</p> <p>§2.4.4 add PROGRAM_NAME variable https://github.com/OneArgo/ArgoVocabs/issues/5</p> <p>§2.4.4 PROJECT_NAME long_name changed</p> <p>§2.4.7.1 SENSOR_MODEL comment clarification: the new definition of SENSOR_MODEL is "as defined by the sensor manufacturer" (reference table R27) https://github.com/OneArgo/ArgoVocabs/issues/55</p> <p>§ 3.3. remove the sentence about FillValue = 99999.f. The FillValue of various parameters are now specified in R03.</p> <p>§3.3.1 How to indicate multiple sensors for the same parameter https://github.com/OneArgo/ArgoVocabs/issues/81</p> |

| | | |
|--|--|---|
| | | § 4.1.2.2. Under File Naming Convention, change the filename prefix of the synthetic files from "M" to "S". |
|--|--|---|

1 Introduction

This document is the Argo data user's manual. It contains the description of the formats and files produced by the Argo DACs.

The changes from the previous version are highlighted in yellow.

1.1 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 3.1 of the technical and metadata file will be valid in the Argo data system.

1.2 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 and DOI:

"These data were collected and made freely available by the international Argo project and the national programs that contribute to it. <http://doi.org/10.17882/42182>".

The Argo dataset can also be added to the references section using the following citation:

Argo (2019). Argo float data and metadata from Global Data Assembly Centre (Argo GDAC). SEANOE. <http://doi.org/10.17882/42182>

More information about the available DOIs for both Argo documents and the dataset versions can be found here:

- <http://www.argodatamgt.org/Access-to-data/Argo-DOI-Digital-Object-Identifier>

1.3 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 the Argo data server.

1.4 Further information sources and contact information

- Argo website: <http://www.argo.net/>
- If you detect any problem in the Argo dataset, please give us your feedback via support@argo.net

1.5 Argo program, data management context

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

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

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

The Argo data formats are based on NetCDF because:

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

1.6 Argo float cycles

A typical Argo float drifts for four years or more in the ocean. It continuously performs measurement cycles.

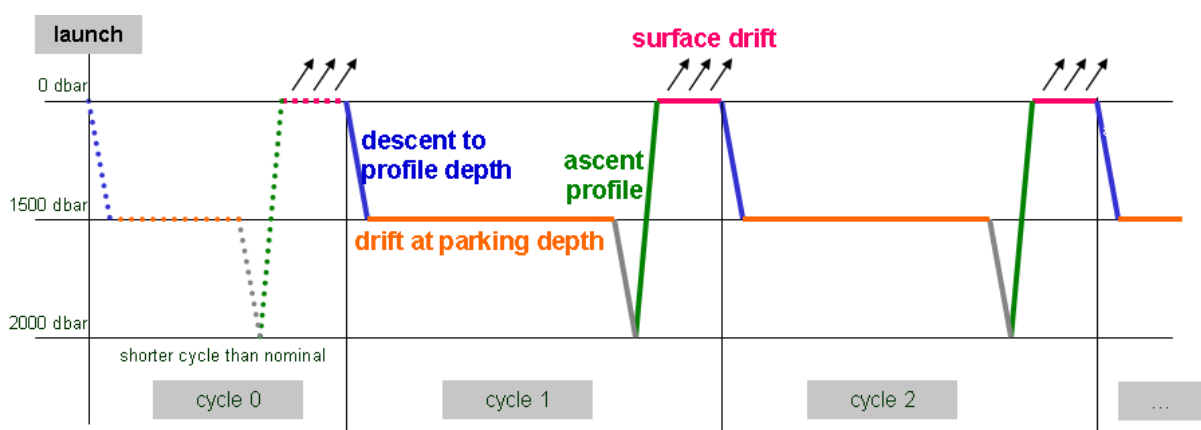
A cycle is defined as a series of actions made by a float and includes either a descending profile or an ascending profile (or, rarely, both); it may also include immersion drift or surface drift.

An Argo cycle starts with a descent toward deep water, usually from the surface.

It ends after the next programmed ascent to the surface, and if begun, after the full surface interval has been completed. During the surface interval, data transmission typically occurs but it is not a requirement for a cycle to have occurred

Each cycle of a float has a unique number, increased by one after each ascent to shallow water. For most floats, this will be the cycle number transmitted by the float. In some cases, this number will need to be calculated by the operator.

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



A typical Argo float cycle during 3 years or more in the ocean.

A more detailed cycle description is available in reference table 15 (<https://vocab.nerc.ac.uk/collection/R15/>), 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 descend 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 after its deployment.

1.7 Real-time, real-time adjusted data 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 Assembly Centres (GDACs) within 12 hours of surfacing, or as quickly thereafter as possible. These are called real-time data or raw data. They are denoted by `DATA_MODE = 'R'`.

The data are also sent to the Principal Investigators, the scientists who apply additional procedures to check data quality and apply adjustments where necessary. These constitute delayed-mode data and are denoted by `DATA_MODE = 'D'`. For the core CTD data, the target is for the delayed-mode data to be available at the GDACs about 12 months after the raw data are transmitted. For biogeochemical (BGC) data, their delayed-mode data can be available as early as 5 to 6 cycles after data transmission.

The difference in timing of availability between delayed-mode CTD and BGC data are due to the difference in maturity of the sensors. CTD sensors are relatively mature sensors and their raw data are usually of good quality. Their delayed-mode assessment involves evaluation of the long-term sensor stability, which typically requires a float data record of about 12 months. On the other hand, BGC sensors are relatively new and their raw data are typically unfit for scientific usage. Therefore early adjustments (after the first 5 to 6 cycles) can increase the accuracy of the raw BGC data significantly. Re-assessment of delayed-mode BGC data is carried out when longer float data records become available.

Some adjustments can be applied to the raw data automatically in real-time (such as adjustments based on previous delayed-mode corrections). These real-time adjusted data are denoted by `DATA_MODE = 'A'`. The real-time adjustment can improve the data quality for users who need to access data in real-time, but they are only an interim adjustment and are revised in delayed-mode.

2 Formats description

2.1 Overview of the formats

Argo data format is an implementation of NetCDF format developed by UCAR/Unidata (<http://doi.org/10.5065/D6H70CW6>).

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.

Argo formats are divided in 4 sections:

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

Argo data comply with NetCDF Climate and Forecast (CF) Metadata Conventions (version 1.6).

- <http://cfconventions.org/>

Argo data uses the UDUNITS convention when applicable to populate the variable attribute "unit" in the Data section. The UDUNITS standard, database and supporting information is available from

- <https://doi.org/10.5065/D6KD1WN0>.

Argo date and time: all date and time have to be given either in Universal Time (UTC) or in float's time.

2.2 Core-Argo profile format version 3.1

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.

Core-Argo profile contains the CTD sensor parameters (pressure, temperature, salinity, conductivity) that are measured with the same vertical sampling scheme and at the same location and time. Additional parameters from other sensors are stored in a B-Argo profile file. The B-profile file is very similar to core-Argo profile file; its additional features are listed in §2.6

Some specialty floats collect additional profiles per cycle. These specialty profiles contain parameters measured at pressure levels that are different from the CTD levels, and can be at locations and time that are different from the primary profile. When multiple profiles exist in a single cycle, users are urged to check the information associated with each profile in order to determine their spatial and temporal relations. 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.

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

2.2.1 Global attributes

The global attributes section is used for data discovery. The following global attributes should appear in the global section:

```
// 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";
:id = "https://doi.org/10.17882/42182";
:comment = "free text";
:user_manual_version = "3.4";
:Conventions = "Argo-3.1 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. |
| id | The Argo GDAC data DOI: https://doi.org/10.17882/42182 This attribute is optional and recommended |
| comment | Miscellaneous information about the data or methods used to produce it. |
| user_manual_version | The version number of the user manual |
| Conventions | The conventions supported by this file, blank separated |
| featureType | The NetCDF CF feature type. |
| comment_on_resolution | Optional comment on parameter resolution |

2.2.2 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 5 th 2001 17:28:34 19971217000000 : December 17 th 1997 00:00:00 |
| STRING256 STRING64 STRING32 STRING16 STRING8 STRING4 STRING2 | STRING256 = 256; STRING64 = 64; STRING32 = 32; STRING16 = 16; STRING8 = 8; STRING4 = 4; STRING2 = 2; | String dimensions from 2 to 256. |
| N_PROF | N_PROF = <int value>; | Number of profiles contained in the file. This dimension depends on the data set. A file contains at least one profile. There is no defined limit on the maximum number of profiles in a file. |

| | | |
|-----------|-------------------------|--|
| | | Example : N_PROF = 100 |
| N_PARAM | N_PARAM = <int value>; | Maximum number of parameters measured or calculated for a pressure sample. This dimension depends on the data set. Examples : (pressure, temperature) : N_PARAM = 2 (pressure, temperature, salinity) : N_PARAM = 3 (pressure, temperature, conductivity, salinity) : N_PARAM = 4 |
| N_LEVELS | N_LEVELS = <int value>; | Maximum number of pressure levels contained in a profile. This dimension depends on the data set. Example : N_LEVELS = 100 |
| N_CALIB | N_CALIB = <int value>; | Maximum number of calibrations performed on a profile. This dimension depends on the data set. Example : N_CALIB = 10 |
| N_HISTORY | N_HISTORY = UNLIMITED; | Number of history records. |

2.2.3 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:conventions = "Argo reference table 1"; 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 (https://vocab.nerc.ac.uk/collection/R01/). Example : Argo profile |
| FORMAT_VERSION | char FORMAT_VERSION(String4); FORMAT_VERSION:long_name = "File format version"; FORMAT_VERSION:_FillValue = " "; | File format version Example : "3.1" |
| 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 1 st 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 29 th 2001 16 :17 :00 |
| DATE_UPDATE | char DATE_UPDATE(Date_Time); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " "; | Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 th 2001 09 :05 :00 |

2.2.4 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 that operates the float. Multiple projects can be separated by commas. Example: "GYROSCOPE, GMMC" |
| 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 responsible for the profiling float. Example: "Julia UITZ". Valid PI names are listed in reference table R40 (https://vocab.nerc.ac.uk/collection/R40/). Multiple names can be concatenated, separated by commas. |
| 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 (https://vocab.nerc.ac.uk/collection/R03/). Examples : TEMP, PSAL, CNDC TEMP : temperature PSAL : practical salinity CNDC : conductivity |
| CYCLE_NUMBER | int CYCLE_NUMBER(N_PROF); CYCLE_NUMBER:long_name = "Float cycle number"; CYCLE_NUMBER:conventions = "0...N, 0 : launch cycle (if exists), 1 : first complete cycle"; CYCLE_NUMBER:_FillValue = 99999; | Float cycle number. See §1.6: float cycle definition. |
| 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 (https://vocab.nerc.ac.uk/collection/R04/). 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 (https://vocab.nerc.ac.uk/collection/R06/). |
| 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 |
| PLATFORM_TYPE | char PLATFORM_TYPE(N_PROF, STRING32); PLATFORM_TYPE:long_name = "Type of float"; PLATFORM_TYPE:conventions = "Argo reference table 23"; PLATFORM_TYPE:_FillValue = " "; | Type of float listed in reference table 23 (https://vocab.nerc.ac.uk/collection/R23/). Example: SOLO, APEX, PROVOR, ARVOR, NINJA |
| FLOAT_SERIAL_NO | char FLOAT_SERIAL_NO(N_PROF, STRING32); FLOAT_SERIAL_NO:long_name = "Serial | Serial number of the float. Example 1679 |

| | | |
|--------------------|--|---|
| | number of the float"; FLOAT_SERIAL_NO: FillValue = " "; | |
| FIRMWARE_VERSION | char FIRMWARE_VERSION(N_PROF, STRING64); FIRMWARE_VERSION:long_name = "Instrument firmware version"; FIRMWARE_VERSION: FillValue = " "; | Firmware version of the float. Example : "013108" The dimension STRING32 instead of STRING64 remains accepted. |
| 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 (https://vocab.nerc.ac.uk/collection/R08/). 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:resolution = X; 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. 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 tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed mode). 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:resolution = X; JULD_LOCATION: FillValue = 999999.; | Julian day of the location 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. 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 tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed mode). Example: 1: position seems correct. |
| POSITIONING_SYSTEM | char POSITIONING_SYSTEM(N_PROF, STRING8); POSITIONING_SYSTEM:long_name = "Positioning system"; | Name of the system in charge of positioning the float locations from reference table 9 (https://vocab.nerc.ac.uk/collection/R09/). Examples : ARGOS |

| | | |
|----------------------------------|---|---|
| | POSITIONING_SYSTEM:_FillValue = " "; | |
| POSITION_ERROR_REPORTED | float POSITION_ERROR_REPORTED(N_PROF); POSITION_ERROR_REPORTED:long_name = "Position error reported by the positioning system"; POSITION_ERROR_REPORTED:units = "meters"; POSITION_ERROR_REPORTED:_FillValue = 99999.; | This is an optional variable. Position error reported by the positioning system. |
| POSITION_ERROR_ESTIMATED | float POSITION_ERROR_ESTIMATED(N_PROF); POSITION_ERROR_ESTIMATED:long_name = "Position error estimated by the real-time or delayed-mode process"; POSITION_ERROR_ESTIMATED:units = "meters"; POSITION_ERROR_ESTIMATED:_FillValue = 99999.; | This is an optional variable. Position error estimated by the real-time or delayed-mode process. |
| POSITION_ERROR_ESTIMATED_COMMENT | char POSITION_ERROR_ESTIMATED_COMMENT(N_PROF, STRING1024); POSITION_ERROR_ESTIMATED_COMMENT:long_name = "Comment on the method used to determine POSITION_ERROR_ESTIMATED"; POSITION_ERROR_ESTIMATED_COMMENT:_FillValue = " "; | This is an optional variable. Comment on the method used to determine POSITION_ERROR_ESTIMATED. |
| 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 (https://vocab.nerc.ac.uk/collection/RP2/). 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 = " "; | Use the vertical sampling scheme to differentiate and identify profiles from a single-cycle with different vertical sampling schemes. See reference table 16 (https://vocab.nerc.ac.uk/collection/R16/). |
| CONFIG_MISSION_NUMBER | int CONFIG_MISSION_NUMBER(N_PROF); CONFIG_MISSION_NUMBER:long_name = "Unique number denoting the missions performed by the float"; CONFIG_MISSION_NUMBER:conventions = "1...N, 1 : first complete mission"; CONFIG_MISSION_NUMBER:_FillValue = 99999.; | Unique number of the mission to which this profile belongs. See note on floats with multiple configurations §2.4.6.1. The number 0 (zero) can be used to denote the float's mission prelude, if it exists. |

Note: how to sort STATION_PARAMETERS variable

The parameters listed in STATION_PARAMETERS should be sorted in the same order within a given float's data file.

Examples with POSITION_QC is 8 (estimated value)

Case 1: only one system has been used for the interpolation. That system is shown in POSITIONING_SYSTEM.

Case 2: When no position was transmitted and an estimate of the position in delayed mode relies on information not from a positioning system (a numerical method, model data. etc.) the **POSITIONG_SYSTEM** should be “NONE”

2.2.5 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 transmitted by the float.

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.

When a profile has DATA_MODE = 'R', no adjusted data are available. Hence the adjusted section (<PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC and <PARAM>_ADJUSTED_ERROR) should be filled with FillValue.

When N_PROF > 1, DATA_MODE for each profile can be assigned differently. This is because when there are multiple profiles, delayed-mode or near real-time adjustments can become available at different times.

The adjusted section for each N_PROF should then be filled independently according to its DATA_MODE.

For example, in a profile file with 2 profiles, it is possible that

- DATA_MODE = 'D' in N_PROF = 1, and
- DATA_MODE = 'R' in N_PROF = 2.

In this case:

- the adjusted section in N_PROF=1 with DATA_MODE='D' should be filled with their adjusted values;
- the adjusted section in N_PROF=2 with DATA_MODE='R' should be filled with FillValue.

The delayed mode QC for core Argo profiles is described in "Argo quality control manual" by Annie Wong et al. (see <http://www.argodatamgt.org/Documentation>).

| 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> contains the original values of a parameter listed in reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). <X> : this field is specified in the reference table 3 |

| | | |
|------------------------|---|--|
| | <pre> <PARAM>:C_format = "<X>"; <PARAM>:FORTRAN_format = "<X>"; <PARAM>:resolution = <X>; </pre> | (https://vocab.nerc.ac.uk/collection/R03/) . |
| <PARAM>_QC | <pre> char <PARAM>_QC(N_PROF, N_LEVELS); <PARAM>_QC:long_name = "quality flag"; <PARAM>_QC:conventions = "Argo reference table 2"; <PARAM>_QC:_FillValue = " "; </pre> | <p>Quality flag applied on each <PARAM> value.</p> <p>The flag scale is specified in reference tables 2</p> <p>(https://vocab.nerc.ac.uk/collection/RR2/) for real time and</p> <p>https://vocab.nerc.ac.uk/collection/RD2/ for delayed mode).</p> |
| <PARAM>_ADJUSTED | <pre> float <PARAM>_ADJUSTED(N_PROF, N_LEVELS); <PARAM>_ADJUSTED:long_name = "<X>"; <PARAM>_ADJUSTED: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>; </pre> | <p><PARAM>_ADJUSTED contains the adjusted values derived from the original values of the parameter.</p> <p><X> : this field is specified in the reference table 3</p> <p>(https://vocab.nerc.ac.uk/collection/R03/).</p> <p><PARAM>_ADJUSTED is mandatory.</p> <p>When no adjustment is performed, the FillValue is inserted.</p> |
| <PARAM>_ADJUSTED_QC | <pre> 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 = " "; </pre> | <p>Quality flag applied on each <PARAM>_ADJUSTED value.</p> <p>The flag scale is specified in reference tables 2</p> <p>(https://vocab.nerc.ac.uk/collection/RR2/) for real time and</p> <p>https://vocab.nerc.ac.uk/collection/RD2/ for delayed mode).</p> <p><PARAM>_ADJUSTED_QC is mandatory.</p> <p>When no adjustment is performed, the FillValue is inserted.</p> |
| <PARAM>_ADJUSTED_ERROR | <pre> float <PARAM>_ADJUSTED_ERROR(N_PROF, N_LEVELS); <PARAM>_ADJUSTED_ERROR:long_name = "Contains the error on the adjusted values as determined by the delayed mode QC process"; <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>; </pre> | <p><PARAM>_ADJUSTED_ERROR</p> <p>Contains the error on the adjusted values as determined by the delayed mode QC process.</p> <p><X> : this field is specified in the reference table 3</p> <p>(https://vocab.nerc.ac.uk/collection/R03/).</p> <p><PARAM>_ADJUSTED_ERROR is mandatory.</p> <p>When no adjustment is performed, the FillValue is inserted.</p> |

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: 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 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;

```

2.2.5.1 How to report unusual parameter resolutions in a profile

The resolution of a parameter is reported in “resolution” attribute.

For specific floats, the resolution of a parameter may depend on the profile level.

How to keep the information in the file?

- add a "comment_on_resolution" attribute on the variable to inform the user,

2.2.6 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, PARAMETER is filled with the list of parameter names, and all values of the 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 (https://vocab.nerc.ac.uk/collection/R03/). 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 | Calibration equation applied to the parameter. Example : $T_c = a_1 * T + a_0$ |

| | | |
|------------------------------|--|--|
| | parameter"; SCIENTIFIC_CALIB_EQUATION:_FillValue = " "; | |
| 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 = " "; | 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:long_name = "Date of calibration"; SCIENTIFIC_CALIB_DATE:conventions = "YYYYMMDDHHMISS"; SCIENTIFIC_CALIB_DATE:_FillValue = " "; | Date of the calibration. Example : 20011217161700 |

2.2.7 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 (<https://vocab.nerc.ac.uk/collection/R07/>).

On the GDAC, the history section of the multi-cycle files is empty to reduce the size of the file. The history section is available in the single-cycle files

| 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 (https://vocab.nerc.ac.uk/collection/R04/). 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 (https://vocab.nerc.ac.uk/collection/R12/). 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 (https://vocab.nerc.ac.uk/collection/R07/). 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. The parameters are described in reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). 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 or equal to 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)"; | 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 described in reference table 11 (https://vocab.nerc.ac.uk/collection/R11/). |

| | | |
|--|---|------------------------------------|
| | HISTORY_QCTEST:conventions = "Write tests performed when ACTION=QCP\$; tests failed when ACTION=QCF\$"; HISTORY_QCTEST: _FillValue = " "; | Example : 0A (in hexadecimal form) |
|--|---|------------------------------------|

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

2.3 Argo trajectory format version 3.2

Argo trajectory files contain all received Argos and GPS locations of Argo floats. The trajectory file also contains cycle timing information important for making velocity calculations. These times may come directly from the float in real time, from calculations based on float information in real time, from the satellite system in real time, or from estimations done in delayed mode.

In addition to locations and cycle timing information, a trajectory file often contains measurements such as pressure, temperature, salinity or conductivity performed at various intermediate times during the cycle. The full pressure, temperature and salinity profile collected upon ascent is not included in the trajectory file. This is stored in the profile file.

An Argo trajectory file contains the CTD and BGC sensor parameters that are measured outside the vertical profiles.

There may be two Argo trajectory files at one time for a float - a real time trajectory file ("R") and a delayed mode trajectory file ("D"). For naming conventions, see §4.1.3. The real time trajectory file will contain all the data obtained in real time for all the cycles the float has performed. The "R" file will exist until the float dies and DMQC is finalized. A delayed mode trajectory file exists for the entire float lifetime.

The delayed mode trajectory file will contain both real time and delayed mode data. The delayed mode data will be the highest quality data available for each cycle that has been delayed mode quality controlled. However, delayed mode quality control may not be performed on all the float's cycles. In this case, the "D" file will contain both the real time and delayed mode data only for the cycles for which delayed mode quality control has been performed. Therefore, if both an "R" and "D" trajectory file exist, to obtain the best quality data for the entire float record, it might be necessary to look at the "D" file for the cycles that have been delayed mode quality controlled and then in the "R" file for the rest of the cycles which have not yet been delayed mode quality controlled. Once a float dies and the entire float record has been quality controlled, the "D" file will be the only file available on the GDAC and will contain both adjusted and not adjusted data.

The trajectory file contains two groups of data variables. In this document the groups are differentiated by their dimension.

The variable group described in §2.3.5 which includes the locations, cycle timing information, and measurements from the float is N_MEASUREMENT long. It includes all the data from the float. If filled, the best timing information is kept in the JULD_ADJUSTED variable. If this is filled in real time, that means either clock drift has been determined and adjustment has been applied (inclusive of adjustment of zero) or another timing estimate has been done based on typical float behavior. Simultaneously, the DATA_MODE should be marked as "A" indicating an adjusted float, and the CLOCK_OFFSET variable should be appropriately filled.

The variable group described in §2.3.6 which includes the cycle timing information and other cycle descriptive variables is N_CYCLE long. The cycle timing information is a subset of the information found in the N_MEASUREMENT array. This array includes the best timing information which matches the JULD_ADJUSTED times if filled, else JULD times from the N_MEASUREMENT array. The times can be corrected for float clock drift or estimated. The JULD_*_STATUS variables provide information

on the state of the timing information. The N_CYCLE array also includes several variables that pertain only to the entire cycle such as GROUNDED, CONFIG_MISSION_NUMBER, etc.

In the N_MEASUREMENT group, the MEASUREMENT_CODE variable must be correctly understood. This variable is designed to indicate where in the cycle the location, times and important float timing events. The Measurement Code Table (Reference Table 15 - <https://vocab.nerc.ac.uk/collection/R15/>) contains all the flags and their meanings for the MEASUREMENT_CODE variable. This table is comprised of two parts - a) Absolute codes: measurement code (MC) values can be primary (mandatory) or secondary (highly desirable), and b) Relative codes: measurement code values are relative to an absolute code and are further divided into two parts: generic codes that can be used by a wide variety of floats and specific codes that are directly important to a specific float/measurement.

All Primary and Secondary MC events that are experienced by the float are required to be present in the N_MEASUREMENT array and redundantly in the N_CYCLE variables. Secondary codes are codes that not as crucial as the primary codes, but it is still recommended they be filled. All other codes are voluntary. Please note the term 'experienced by the float'. It is not necessary, nor best practice, to include measurement codes including primary or secondary codes, if the float is not programmed to activate an action described by the measurement code. For example a float alternates cycle missions. In even cycles n, the float does not enter a drift phase, but instead rises directly back to the surface after falling to depth. In odd cycles n+1, the float enters a drift phase. In the N_MEASUREMENT array the even cycles n would not include measurement codes indicative of drift, such as MC250 or MC300. But these codes would be included in odd cycles n+1.

If the float experiences an event but the time is not able to be determined, then most JULD variables are set to fill value and a *_STATUS = '9' is used in both the N_MEASUREMENT and N_CYCLE arrays. This indicates that it might be possible to estimate the timing of the event in the future and acts as a placeholder.

If a float does not experience an event, then the fill values are used for all N_CYCLE variables. These non-events do not get a placeholder in the N_MEASUREMENT arrays as described above.

For file naming conventions, see §4.1.3.

2.3.1 Global attributes

The global attributes section is used for data discovery. It complies with NetCDF Climate and Forecast (CF) Metadata Conventions (version 1.6).

- <http://cfconventions.org/>

The following global attributes should appear in the global section:

```
// global attributes:
:title = "Argo float trajectory file";
:institution = "CORIOLIS";
:source = "Argo float";
:history = "2011-04-22T06:00:00Z creation";
:references = "http://www.argodatamgt.org/Documentation";
:comment = "free text";
:user_manual_version = "3.4";
:Conventions = "Argo-3.2 CF-1.6";
:featureType = "trajectory";
:comment_on_resolution = "PRES variable resolution depends on measurement code";
```

| 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. |
| user_manual_version | The version number of the user manual |
| Conventions | The conventions supported by this file, blank separated |
| featureType | The NetCDF CF feature type. |
| comment_on_resolution | Optional comment on parameter resolution |

2.3.2 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 either in Universal Time (UTC) or float's time. Examples : 20010105172834 : January 5 th 2001 17:28:34 19971217000000 : December 17 th 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, cycle timings and measurements of the file. |
| N_CYCLE | N_CYCLE = <int value>; | Number of collected float cycles. If all the cycles have been collected (i.e. if there are no missing cycles), it is the number of cycles performed by the float. In this particular case, as some floats begin cycle numbering at 0, others at 1, in the former, N_CYCLE = max(CYCLE_NUMBER) + 1. In the latter, N_CYCLE = max(CYCLE_NUMBER) 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 Example : N_HISTORY = 10 |
| N_VALUESxx | N_VALUESxx = <int value>; | Maximum number of parameter measurements sampled at a given pressure level. This dimension depends on the data set. Example: N_VALUES41 = 41. |
| N_CALIB_PARAM | N_CALIB_PARAM = <int value>; | Maximum number of calibrations performed on a float to fill the parameter adjusted fields. This dimension depends on the data set. Example: N_CALIB_PARAM = 3. |
| N_CALIB_JULD | N_CALIB_JULD = <int value>; | Maximum number of calibrations performed on a float to fill the JULD adjusted fields. This dimension depends on the data set. Example: N_CALIB_JULD = 3. |

2.3.3 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:conventions = "Argo reference table 1"; 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 (https://vocab.nerc.ac.uk/collection/R01/). Example : Argo trajectory |
| FORMAT_VERSION | char FORMAT_VERSION(String4); FORMAT_VERSION:long_name = "File format version"; FORMAT_VERSION:_FillValue = " "; | File format version Example : "3.2" |
| 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 : "1.2" |
| 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 1 st 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 29 th 2001 16 :17 :00 |
| DATE_UPDATE | char DATE_UPDATE(Date_Time); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " "; | Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 th 2001 09 :05 :00 |

2.3.4 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 that operates the float. Multiple projects can be separated by commas. Example: "GYROSCOPE, GMMC" |
| PI_NAME | char PI_NAME (String64); PI_NAME:long_name = "Name of the principal investigator"; PI_NAME:_FillValue = " "; | Name of the principal investigator responsible for the profiling float. Example: "Julia UITZ". Valid PI names are listed in reference table R40: https://vocab.nerc.ac.uk/collection/R40/ Multiple names can be concatenated, separated by commas. |
| TRAJECTORY_PARAMETERS | char TRAJECTORY_PARAMETERS(N_PARAM, String64); TRAJECTORY_PARAMETERS:long_name = "List of available parameters"; 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 (https://vocab.nerc.ac.uk/collection/R03/). Examples : "PRES", "TEMP", "PSAL", "CNDC", "DOXY", etc "PRES": pressure "TEMP" : temperature "PSAL" : practical salinity "CNDC" :electrical conductivity "DOXY" : dissolved oxygen |
| DATA_CENTRE | char DATA_CENTRE(String2); DATA_CENTRE:long_name = "Data centre in charge of float data processing"; DATA_CENTRE:conventions = "Argo | Code for the data centre in charge of the float data management. The data centre codes are described in the reference table 4 (https://vocab.nerc.ac.uk/collection/R04/). |

| | | |
|----------------------|---|---|
| | reference table 4"; DATA_CENTRE:_FillValue = " "; | 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 (https://vocab.nerc.ac.uk/collection/R06/). |
| PLATFORM_TYPE | char PLATFORM_TYPE(String32); PLATFORM_TYPE:long_name = "Type of float"; PLATFORM_TYPE:conventions = "Argo reference table 23"; PLATFORM_TYPE:_FillValue = " "; | Type of float listed in reference table 23 (https://vocab.nerc.ac.uk/collection/R23/). Example: SOLO, APEX, PROVOR, ARVOR, NINJA. |
| FLOAT_SERIAL_NO | char FLOAT_SERIAL_NO(String32); FLOAT_SERIAL_NO:long_name = "Serial number of the float"; FLOAT_SERIAL_NO:_FillValue = " "; | This field should contain only the serial number of the float. Example : 1679 |
| FIRMWARE_VERSION | char FIRMWARE_VERSION(String64); FIRMWARE_VERSION:long_name = "Instrument firmware version"; FIRMWARE_VERSION:_FillValue = " "; | Firmware version of the float. Example : "013108" The dimension STRING32 instead of STRING64 remains accepted. |
| 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 (https://vocab.nerc.ac.uk/collection/R08/). 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 (https://vocab.nerc.ac.uk/collection/R09/). Example : ARGOS |

2.3.5 N_MEASUREMENT dimension variable group

This section describes the variables found in the N_MEASUREMENT dimension variable group. In this variable group you find the unadjusted data as reported by the float, adjusted timing, the reported locations, as well as measurements performed along the surface and subsurface trajectory.

N_MEASUREMENT is the number of locations, cycle timings, and measurements received or estimated from information sent by the float. If a cycle is missed, nothing is entered into the N_MEASUREMENT array - e.g. no fill values are allowed to indicate a missing cycle.

The N_MEASUREMENT array should be arranged first by CYCLE_NUMBER and then by the order the events for that cycle occurred. Some data within the netCDF may not have a time attached to it, but it should still be placed as close as possible to its origination time. Because the order of the N_MEASUREMENT array is based on time, MEASUREMENT_CODE will not be ascending for every cycle, but JULD is usually ascending (unless a clock offset has been applied and then the JULD variable may have an inversion) and JULD_ADJUSTED is always ascending. To construct the trajectory netCDF a full understanding of when the float data was gathered within the cycle is necessary.

JULD contains the raw timing values either from the satellite system or from the float. The values in JULD cannot be estimated, nor altered such as for clock drift.

JULD_ADJUSTED contains the best estimate of float timing available for this float. If necessary, it contains adjusted timing variables due to clock drift. The times can be adjusted either in real time or in delayed mode due to clock drift or estimation of times based on float behavior by a float expert. The JULD_ADJUSTED_STATUS variable indicates how the JULD_ADJUSTED value is filled and indicates whether the time is estimated or measured. The JULD_ADJUSTED_QC contains the QC flags

for the adjusted times. This may lead to times where JULD_ADJUSTED is filled, but JULD contains 'FillValue'. That is because the time is estimated rather than measured.

In R-mode, no times are adjusted, but times may be estimated and placed within JULD_ADJUSTED with the JULD_ADJUSTED_STATUS flag set to '1' indicating an estimated value. Non-adjusted times do not need to be carried down to the JULD_ADJUSTED array, so this array may be sparse or even empty. No estimated times are allowed in the JULD variable.

In A-mode, adjustments are made, typically to PSAL and PRES, but others may be adjusted as well such as JULD if real-time correction of clock drift is applied. If an adjustment is made, all values must be carried down to the ADJUSTED variables with the appropriate adjustment applied. Real time estimates will also be present in 'A' files. Non-zero adjustments will need to be applied to any present estimates.

In D-mode, all adjustments and estimations are complete. Similar to the 'A' file, there must be correspondence between the ADJUSTED and non-ADJUSTED fields. This means that all values must be carried down to the ADJUSTED field. ADJUSTED variables may have a value while the corresponding non-ADJUSTED variable is 'FillValue' due to the presence of an estimated value. The opposite is not allowed. There cannot be a non-ADJUSTED value and 'FillValue' in the corresponding ADJUSTED variable.

CYCLE_NUMBER contains the cycle number of the cycle that is assigned in real time. This cycle number must match the profile cycle number, which is the number recorded in the CYCLE_NUMBER(N_PROF) variable in profile files.

CYCLE_NUMBER_INDEX indicates which cycle number information is contained in that index of the N_CYCLE array. For example, CYCLE_NUMBER_INDEX(4)=3 means the 4th element of all 34 N_CYCLE variables is associated with the WMO_003.nc profile file. This might happen if the float's first cycle has a cycle number of zero rather than one. Additionally, all the elements of the N_MEASUREMENT variables for which CYCLE_NUMBER = 3 are likewise associated with the 4th N_CYCLE elements and with the WMO_003.nc profile file. This clearly links the index in the N_CYCLE array to the cycle number in the N_MEASUREMENT array.

Additionally, CYCLE_NUMBER = -1 indicates the float's launch and the JULD and LATITUDE and LONGITUDE variables should contain the float's launch time and location.

CYCLE_NUMBER_ADJUSTED contains a cycle numbering which has been assessed and adjusted to be correct, especially for the purposes of trajectory calculations. If a cycle is recovered during delayed mode and no profile file is created, the cycle must be added into the CYCLE_NUMBER_ADJUSTED and CYCLE_NUMBER_ADJUSTED_INDEX variables. Two examples of recovered cycles are below.

The first example is where cycle number 5 is recovered in delayed mode. The cycle number variables must be rewritten as follows:

| | |
|-----------------------------|---|
| CYCLE_NUMBER | 1, 2, 3, 4, _, 6, 7, 8, 9, 10, 11, ..., |
| CYCLE_NUMBER_INDEX | 1, 2, 3, 4, _, 6, 7, 8, 9, 10, 11, ..., |
| CYCLE_NUMBER_ADJUSTED | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, _ |
| CYCLE_NUMBER_ADJUSTED_INDEX | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, _ |

Here, FillValue is added to CYCLE_NUMBER and CYCLE_NUMBER_INDEX to indicate that no profile file exists with cycle number 5.

A second example of an error that might be discovered in cycle number in delayed mode involves floats that do not send cycle number and for which cycle number must be calculated. Here, cycle number 5 was incorrectly skipped in real time and is introduced in delayed mode:

| | |
|-----------------------|---|
| CYCLE_NUMBER | 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, ... |
| CYCLE_NUMBER_INDEX | 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, ... |
| CYCLE_NUMBER_ADJUSTED | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, _, _ |

CYCLE_NUMBER_ADJUSTED_INDEX 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, _, _

Here, CYCLE_NUMBER = 6 which was assigned in real time, should actually be assigned cycle number = 5 as reflected by the CYCLE_NUMBER_ADJUSTED variable.

CYCLE_NUMBER always corresponds to the profile cycle number. To look for the cycle that matches the profile cycle number, users must look in the CYCLE_NUMBER variable for the cycle number they are interested in. If the CYCLE_NUMBER_ADJUSTED variable is 'FillValue', then this cycle is in real time mode and no corrected cycle number exists. If the CYCLE_NUMBER_ADJUSTED variable is filled, this is the correct cycle number as determined during delayed mode. For the previous example, profile cycle number 6 corresponds to CYCLE_NUMBER = 6 and CYCLE_NUMBER_ADJUSTED = 5.

<PARAM> contains the uncorrected real-time data transmitted by 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 real time 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 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 A-mode processing contains adjusted sections with fill values (<PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC and <PARAM>_ADJUSTED_ERROR). This is the same for all other ADJUSTED variables (e.g. JULD_ADJUSTED).

When no parameter is measured along the trajectory, N_PARAM (number of parameters) and any fields relative to parameter are not in the file : <PARAM>, <PARAM>_QC, <PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC, <PARAM>_ADJUSTED_ERROR and TRAJECTORY_PARAMETERS.

Iridium and estimated positions:

Iridium-based or estimated positions are included if no GPS fix could be obtained that is considered good in real-time. POSITIONING_SYSTEM is used to indicate that multiple positioning systems are present in the trajectory file (see reference table 9 - <https://vocab.nerc.ac.uk/collection/R09/>). Used are "I" (for Iridium) or "U" (for estimated) in POSITION_ACCURACY (Table 5 - <https://vocab.nerc.ac.uk/collection/R05/>). If both AXES_ERROR_ELLIPSE_MAJOR and AXES_ERROR_ELLIPSE_MINOR values are the same AXES_ERROR_ELLIPSE_ANGLE is not needed (set to FillValue).

RAFOS positions:

A similar approach is used storing the error and indicating that the position comes from RAFOS estimates. POSITIONING_SYSTEM remains filled with the primary system, i.e. GPS or Argos rather than RAFOS. POSITION_ACCURACY becomes 'R' and the MEASUREMENT_CODE for positions and associated times during drift is 275.

Details on how the RAFOS positions were derived can not be stored in the traj.nc file. These will be in a traj_aux file.

| Name | definition | comment |
|------|-----------------------------|--|
| JULD | double JULD(N_MEASUREMENT); | Julian day of the location (or 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 day)"; JULD:resolution = X; JULD:_FillValue = 999999.; JULD:axis = "T"; | 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_STATUS | char JULD_STATUS(N_MEASUREMENT); JULD_STATUS:long_name="Status of the date and time" JULD_STATUS:conventions = "Argo reference table 19"; JULD_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: 2 : Value is transmitted by the float |
| 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 tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real-time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed-mode). Example : 1: The date and time seem correct. |
| JULD_ADJUSTED | double JULD_ADJUSTED(N_MEASUREMENT); JULD_ADJUSTED:long_name = "Adjusted julian day (UTC) of each measurement relative to REFERENCE_DATE_TIME"; JULD_ADJUSTED:standard_name = "time"; JULD_ADJUSTED:units = "days since 1950-01-01 00:00:00 UTC"; JULD_ADJUSTED:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_ADJUSTED:resolution = X; JULD_ADJUSTED:_FillValue = 999999.; JULD_ADJUSTED:axis = "T"; | Adjusted 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. The date may be adjusted due to float clock drift or expert review. Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_ADJUSTED_STATUS | char JULD_ADJUSTED_STATUS(N_MEASUREMENT); JULD_ADJUSTED_STATUS:long_name="Status of the JULD_ADJUSTED date" JULD_ADJUSTED_STATUS:conventions = "Argo reference table 19"; JULD_ADJUSTED_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: 2 : Value is transmitted by the float |
| JULD_ADJUSTED_QC | char JULD_ADJUSTED_QC(N_MEASUREMENT); JULD_ADJUSTED_QC:long_name = "Quality on adjusted date and time"; JULD_ADJUSTED_QC:conventions = "Argo reference table 2"; JULD_ADJUSTED_QC:_FillValue = " "; | Quality flag on JULD_ADJUSTED date and time. The flag scale is described in the reference tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real-time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed-mode). Example : 1 : The date and time seem 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 received from the positioning system. The location classes from ARGOS are described in the reference table 5 (https://vocab.nerc.ac.uk/collection/R05/). A "G" indicates the GPS positioning system. |

| | | |
|------------------------|--|--|
| | POSITION_ACCURACY:conventions = "Argo reference table 5"; POSITION_ACCURACY:_FillValue = " "; | Examples : 3 for a latitude and longitude accuracy < 250 m. G for GPS accuracy |
| 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 tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real-time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed-mode). 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 series of measurements, locations and timings. Some floats begin with a cycle 0 and some begin at cycle number 1. Cycle number is -1 for the float's launch and includes the time and location. For one cycle number, there are usually several locations/measurements received. This cycle number must match the profile cycle number. Example : 17 for measurements performed during the 17 th cycle of the float. |
| CYCLE_NUMBER_ADJUSTED | int CYCLE_NUMBER_ADJUSTED(N_MEASUREMENT); CYCLE_NUMBER_ADJUSTED:long_name = "Adjusted float cycle number of the measurement"; CYCLE_NUMBER_ADJUSTED:conventions = "0...N, 0 : launch cycle, 1 : first complete cycle"; CYCLE_NUMBER_ADJUSTED:_FillValue = 99999; | Adjusted cycle number of the float for this series of measurements, locations and timings. Some floats begin with a cycle 0 and some begin at cycle number 1. For one cycle number, there are usually several locations/measurements received. Sometimes cycle numbers are assigned erroneously and need to be corrected. This variable contains the corrected cycle numbers. Example : 17 for measurements performed during the 17 th cycle of the float. |
| MEASUREMENT_CODE | int MEASUREMENT_CODE(N_MEASUREMENT); MEASUREMENT_CODE:long_name = "Flag referring to a measurement event in the cycle"; MEASUREMENT_CODE:conventions = "Argo reference table 15"; MEASUREMENT_CODE:_FillValue = 99999; | Flag for each event in the cycle which corresponds to Argo reference table 15 (https://vocab.nerc.ac.uk/collection/R15/). Example: 100 : 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> contains the original values of a parameter listed in the "code" column of reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). <X> : these fields are specified in the columns of the reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). |
| <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> value. The flag scale is specified in tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real-time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed-mode). |
| <PARAM>_ADJUSTED | float <PARAM>_ADJUSTED(N_MEASUREMENT); <PARAM>_ADJUSTED:long_name = "<X>"; <PARAM>_ADJUSTED:standard_name = "<X>"; <PARAM>_ADJUSTED:_FillValue = <X>; <PARAM>_ADJUSTED:units = "<X>"; <PARAM>_ADJUSTED:valid_min = <X>; <PARAM>_ADJUSTED:valid_max = <X>; <PARAM>_ADJUSTED:comment = "<X>"; <PARAM>_ADJUSTED:C_format = "<X>"; <PARAM>_ADJUSTED:FORTRAN_format = "<X>"; <PARAM>_ADJUSTED:resolution= <X>; | <PARAM>_ADJUSTED contains the adjusted values derived from the original values of the parameter. <X> : these fields are specified in the columns of the reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). When no adjustment is performed, the FillValue is inserted. <PARAM>_ADJUSTED is mandatory for core and bgc parameters, but not for intermediate parameters. |
| <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 tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real-time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed-mode). When no adjustment is performed, the FillValue is inserted. <PARAM>_ADJUSTED_QC is mandatory for core and bgc parameters, but not for intermediate parameters. |
| <PARAM>_ADJUSTED_ERROR | float | <PARAM>_ADJUSTED_ERROR contains the error on the |

| | | |
|--------------------------------|--|---|
| D_ERROR | <pre> <PARAM>_ADJUSTED_ERROR(N_MEASUREMENT); <PARAM>_ADJUSTED_ERROR:long_name = "Contains the error on the adjusted values as determined by the delayed mode QC process."; <PARAM>_ADJUSTED_ERROR:_FillValue = <X>; <PARAM>_ADJUSTED_ERROR:units = "<X>"; <PARAM>_ADJUSTED_ERROR:C_format = "<X>"; <PARAM>_ADJUSTED_ERROR:FORTTRAN_for mat = "<X>"; <PARAM>_ADJUSTED_ERROR:resolution= <X>; </pre> | <p>adjusted values of the parameter. <X> : these fields are specified in the columns of the reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). When no adjustment is performed, the FillValue is inserted. <PARAM>_ADJUSTED_ERROR is mandatory for core and bgc parameters, but not for intermediate parameters.</p> |
| AXES_ERROR_ELLIPSE_MAJOR | <pre> float AXES_ERROR_ELLIPSE_MAJOR(N_MEASUREMENT); AXES_ERROR_ELLIPSE_MAJOR.long_name = "Semi-major axis of error ellipse from positioning system"; AXES_ERROR_ELLIPSE_MAJOR:units = "meters"; AXES_ERROR_ELLIPSE_MAJOR:_FillValue = 99999.; </pre> | Semi-major axis of error ellipse reported by the positioning system. |
| AXES_ERROR_ELLIPSE_MINOR | <pre> float AXES_ERROR_ELLIPSE_MINOR(N_MEASUREMENT); AXES_ERROR_ELLIPSE_MINOR.long_name = "Semi-minor axis of error ellipse from positioning system"; AXES_ERROR_ELLIPSE_MINOR:units = "meters"; AXES_ERROR_ELLIPSE_MINOR:_FillValue = 99999.; </pre> | Semi-minor axis of error ellipse reported by the positioning system. |
| AXES_ERROR_ELLIPSE_ANGLE | <pre> float AXES_ERROR_ELLIPSE_ANGLE(N_MEASUREMENT); AXES_ERROR_ELLIPSE_ANGLE.long_name = "Angle of error ellipse from positioning system"; AXES_ERROR_ELLIPSE_ANGLE:units = "Degrees (from North when heading East)"; AXES_ERROR_ELLIPSE_ANGLE:_FillValue = 99999.; </pre> | Angle of error ellipse reported by the positioning system. |
| SATELLITE_NAME | <pre> char SATELLITE_NAME(N_MEASUREMENT); SATELLITE_NAME.long_name = "Satellite name from positioning system"; SATELLITE_NAME.FillValue = " "; </pre> | Satellite name from positioning system. SATELLITE_NAME is only used for Argos position |
| TRAJECTORY_PARAMETER_DATA_MODE | <pre> char TRAJECTORY_PARAMETER_DATA_MODE(N_MEASUREMENT, N_PARAMETER); TRAJECTORY_PARAMETER_DATA_MODE:long_name = "Delayed mode or real time data"; TRAJECTORY_PARAMETER_DATA_MODE:conventions = "R : real time; D : delayed mode; A : real time with adjustment"; TRAJECTORY_PARAMETER_DATA_MODE:_FillValue = " "; </pre> | Indicates the availability of adjustment of each TRAJECTORY_PARAMETERS. 'R' = no adjusted value is available. 'A' = real-time adjusted value is available. 'D' = delayed-mode adjusted value is available. |
| JULD_DATA_MODE | <pre> char JULD_DATA_MODE(N_MEASUREMENT); JULD_DATA_MODE:long_name = "Delayed mode or real time data"; JULD_DATA_MODE:conventions "R : real time; D : delayed mode; A : real time with adjustment"; JULD_DATA_MODE:_FillValue = " "; </pre> | Indicates the availability of adjustment/estimation of the Julian day of the measurement, which is stored in JULD_ADJUSTED. The raw parameter JULD stores the original telemetered value, or FillValue (if not available). If JULD_DATA_MODE = 'R', JULD_ADJUSTED = FillValue. If JULD_DATA_MODE = 'A', JULD_ADJUSTED = real-time adjusted/estimated value. If JULD_DATA_MODE = 'D', JULD_ADJUSTED = delayed-mode adjusted/estimated value. |

2.3.5.1 How to report unusual Pressure resolutions in the N_MEASUREMENT variable group of the TRAJ file

In the N_MEASUREMENT array of the TRAJ file, the pressure resolution may differ according to the MEASUREMENT_CODE.

How to keep the information in the file?

- add a "comment_on_resolution" attribute to the variable to inform the user,
- add a "comment_on_resolution" global attribute to the file

For example, for APEX and PROVOR floats some pressures are provided in bars whereas most of them are in dbar. Thus, in this case:

- PRES:resolution = 0.1f;
- PRES:comment_on_resolution = "PRES resolution is 0.1 dbar, except for measurement codes [150 189 198 289 297 298 389 398 489 497 498 589 901] for which PRES resolution is 1 bar";

You may add a comment_on_resolution global attribute

- :comment_on_resolution = "PRES variable resolution may be lower than nominal depending on measurement codes"

2.3.6 N_CYCLE dimension variable group

This section contains information on the variables with dimension N_CYCLE. They include variables that contain the best estimate of float timing.

Each field in this section has a N_CYCLE dimension.

N_CYCLE is the number of collected cycles performed by the float. It is a dimension, thus it may not equal the maximum cycle number within the file.

The N_CYCLE array should be ordered by CYCLE_NUMBER_INDEX.

The cycle definition is available at §1.6. Cycle is defined as a series of actions, including collection of data, made by a float that ends with transmission of data. If the float fails to collect nor transmit data, a cycle has not occurred and CYCLE_NUMBER_INDEX should not be incremented.

Floats begin with different cycle numbers depending on float type. To understand how the N_CYCLE dimension variable group relates to the N_MEASUREMENT variable group, the user must consult the CYCLE_NUMBER_INDEX variable. This variable indicates the cycle number of the float information that is contained in that particular N_CYCLE index. For example, to find the N_CYCLE information that corresponds to CYCLE_NUMBER = 1, look for CYCLE_NUMBER_INDEX = 1.

Additionally, CYCLE_NUMBER_INDEX is the number of the profile cycle associated with the trajectory cycle in that index of the N_CYCLE array.

If any errors are discovered in how the cycle numbers were assigned in real-time, or if additional cycles are recovered in delayed mode, the CYCLE_NUMBER_INDEX_ADJUSTED variable is adjusted accordingly.

CYCLE_NUMBER_INDEX always corresponds to the profile cycle number. To look for the cycle that matches the profile cycle number, users must look in the CYCLE_NUMBER_INDEX variable for the cycle number they are interested in. If the CYCLE_NUMBER_INDEX_ADJUSTED variable contains 'FillValue', then this cycle is in real time mode and no corrected cycle number exists. If the

CYCLE_NUMBER_INDEX_ADJUSTED variable is filled, this is the correct cycle number as determined during delayed mode.

When a cycle is missing (e.g. no data received), no fill values are used to indicate a missing cycle.

| Name | Definition | Comment |
|---------------------------------|--|---|
| JULD_DESCENT_START | double JULD_DESCENT_START(N_CYCLE); JULD_DESCENT_START:long_name = "Descent start date of the cycle"; JULD_DESCENT_START:standard_name = "time"; JULD_DESCENT_START:units = "days since 1950-01-01 00:00:00 UTC"; JULD_DESCENT_START:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_DESCENT_START:resolution = X; JULD_DESCENT_START:_FillValue = 999999.; | Julian day (UTC) when float leaves the surface and begins descent Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_DESCENT_START_STATUS | char JULD_DESCENT_START_STATUS(N_CYCLE); JULD_DESCENT_START_STATUS:long_name = "Status of descent start date of the cycle"; JULD_DESCENT_START_STATUS:conventions = "Argo reference table 19"; JULD_DESCENT_START_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: '2' : Value is transmitted by the float |
| JULD_FIRST_STABILIZATION | double JULD_FIRST_STABILIZATION(N_CYCLE); JULD_FIRST_STABILIZATION:long_name = "Time when a float first becomes water-neutral"; JULD_FIRST_STABILIZATION:standard_name = "time"; JULD_FIRST_STABILIZATION:units = "days since 1950-01-01 00:00:00 UTC"; JULD_FIRST_STABILIZATION:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_FIRST_STABILIZATION:resolution = X; JULD_FIRST_STABILIZATION:_FillValue = 999999. | Julian day (UTC) of time when a float first becomes water-neutral. Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_FIRST_STABILIZATION_STATUS | char JULD_FIRST_STABILIZATION_STATUS(N_CYCLE); JULD_FIRST_STABILIZATION_STATUS:long_name = "Status of time when a float first becomes water-neutral"; JULD_FIRST_STABILIZATION_STATUS:conventions = "Argo reference table 19"; JULD_FIRST_STABILIZATION_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: '2' : Value is transmitted by the float |
| JULD_DESCENT_END | double JULD_DESCENT_END(N_CYCLE); JULD_DESCENT_END:long_name = "Descent end date of the cycle"; JULD_DESCENT_END:standard_name = "time"; JULD_DESCENT_END:units = "days since 1950-01-01 00:00:00 UTC"; JULD_DESCENT_END:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_DESCENT_END:resolution = X; | Julian day (UTC) when float first approaches within 3% of the eventual drift pressure. Float may be transitioning from the surface or from a deep profile. Example : 18833.8013889885 : July 25 2001 19:14:00 |

| | | |
|------------------------------|---|---|
| | JULD_DESCENT_END:_FillValue = 999999.; | |
| JULD_DESCENT_END_STATUS | char JULD_DESCENT_END_STATUS(N_CYCLE); JULD_DESCENT_END_STATUS:long_name = "Status of descent end date of the cycle"; JULD_DESCENT_END_STATUS:conventions = "Argo reference table 19"; JULD_DESCENT_END_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: '2' : Value is transmitted by the float |
| JULD_PARK_START | double JULD_PARK_START(N_CYCLE); JULD_PARK_START:long_name = "Drift start date of the cycle"; JULD_PARK_START:standard_name = "time"; JULD_PARK_START:units = "days since 1950-01-01 00:00:00 UTC"; JULD_PARK_START:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_PARK_START:resolution = X; JULD_PARK_START:_FillValue = 999999.; | Julian day (UTC) when the float transitions to its Park or Drift mission. This variable is based on float logic based on a descent timer (i.e. SOLO), or be based on measurements of pressure (i.e. Provor).Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_PARK_START_STATUS | char JULD_PARK_START_STATUS(N_CYCLE); JULD_PARK_START_STATUS:long_name = "Status of drift start date of the cycle"; JULD_PARK_START_STATUS:conventions = "Argo reference table 19"; JULD_PARK_START_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: '2' : Value is transmitted by the float |
| JULD_PARK_END | double JULD_PARK_END(N_CYCLE); JULD_PARK_END:long_name = "Drift end date of the cycle"; JULD_PARK_END:standard_name = "time"; JULD_PARK_END:units = "days since 1950-01-01 00:00:00 UTC"; JULD_PARK_END:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_PARK_END:resolution = X; JULD_PARK_END:_FillValue = 999999.; | Julian day (UTC) when float exits from its Park or Drift mission. It may next rise to the surface (AST) or sink to profile depth (DDET)Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_PARK_END_STATUS | char JULD_PARK_END_STATUS(N_CYCLE); JULD_PARK_END_STATUS:long_name = "Status of drift end date of the cycle"; JULD_PARK_END_STATUS:conventions = "Argo reference table 19"; JULD_PARK_END_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: '2' : Value is transmitted by the float |
| 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:standard_name = "time"; 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 parts of day)"; JULD_DEEP_DESCENT_END:resolution = X; JULD_DEEP_DESCENT_END:_FillValue = 999999.; | Julian day (UTC) when float first approaches within 3% of the eventual deep drift/profile pressure. Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_DEEP_DESCENT_END_STATUS | char JULD_DEEP_DESCENT_END_STATUS(N_CYCLE); | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). |

| | | |
|-----------------------------|---|---|
| | JULD_DEEP_DESCENT_END_STATUS:long_name = "Status of deep descent end date of the cycle"; JULD_DEEP_DESCENT_END_STATUS:conventions = "Argo reference table 19"; JULD_DEEP_DESCENT_END_STATUS:_FillValue = " "; | Example: '2' : Value is transmitted by the float |
| JULD_DEEP_PARK_START | double JULD_DEEP_PARK_START(N_CYCLE); JULD_DEEP_PARK_START:long_name = "Deep park start date of the cycle"; JULD_DEEP_PARK_START:standard_name = "time"; JULD_DEEP_PARK_START:units = "days since 1950-01-01 00:00:00 UTC"; JULD_DEEP_PARK_START:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_DEEP_PARK_START:resolution = X; JULD_DEEP_PARK_START:_FillValue = 999999.; | Julian day (UTC) when the float transitions to its Deep Park or Deep Drift mission. This variable is based on float logic based on a descent timer (i.e. SOLO), or be based on measurements of pressure (i.e. Provor). Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_DEEP_PARK_START_STATUS | char JULD_DEEP_PARK_START_STATUS(N_CYCLE); JULD_DEEP_PARK_START_STATUS:long_name = "Status of deep park start date of the cycle "; JULD_DEEP_PARK_START_STATUS:conventions = "Argo reference table 19"; JULD_DEEP_PARK_START_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: '2' : Value is transmitted by the float |
| JULD_ASCENT_START | double JULD_ASCENT_START(N_CYCLE); JULD_ASCENT_START:long_name = "Start date of the ascent to the surface"; JULD_ASCENT_START:standard_name = "time"; JULD_ASCENT_START:units = "days since 1950-01-01 00:00:00 UTC"; JULD_ASCENT_START:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_ASCENT_START:resolution = X; JULD_ASCENT_START:_FillValue = 999999.; | Julian day (UTC) of the beginning of the float's ascent to the surface Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_ASCENT_START_STATUS | char JULD_ASCENT_START_STATUS(N_CYCLE); JULD_ASCENT_START_STATUS:long_name = "Status of start date of the ascent to the surface"; JULD_ASCENT_START_STATUS:conventions = "Argo reference table 19"; JULD_ASCENT_START_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: '2' : Value is transmitted by the float |
| JULD_DEEP_ASCENT_START | double JULD_DEEP_ASCENT_START(N_CYCLE); JULD_DEEP_ASCENT_START:long_name = "Deep ascent start date of the cycle"; JULD_DEEP_ASCENT_START:standard_name = "time"; JULD_DEEP_ASCENT_START:units = "days since 1950-01-01 00:00:00 UTC"; JULD_DEEP_ASCENT_START:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_DEEP_ASCENT_START:resolution = X; JULD_DEEP_ASCENT_START:_FillValue = 999999.; | Julian day (UTC) when the float begins its rise to drift pressure. Typical for profile-on-descent floats.. Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_DEEP_ASCENT_S | char | Status flag on JULD date and time. |

| | | |
|--------------------------------|---|--|
| TART_STATUS | JULD_DEEP_ASCENT_START_STATUS(N_CYCLE); JULD_DEEP_ASCENT_START_STATUS:long_name = "Status of deep ascent start date of the cycle"; JULD_DEEP_ASCENT_START_STATUS:conventions = "Argo reference table 19"; JULD_DEEP_ASCENT_START_STATUS:_FillValue = " "; | The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/). Example: '2' : Value is transmitted by the float |
| JULD_ASCENT_END | double JULD_ASCENT_END(N_CYCLE); JULD_ASCENT_END:long_name = "End date of ascent to the surface"; JULD_ASCENT_END:standard_name = "time"; JULD_ASCENT_END:units = "days since 1950-01-01 00:00:00 UTC"; JULD_ASCENT_END:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_ASCENT_END:resolution = X; JULD_ASCENT_END:_FillValue = 999999.; | Julian day (UTC) of the end of the ascent to the surface . Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_ASCENT_END_STATUS | char JULD_ASCENT_END_STATUS(N_CYCLE); JULD_ASCENT_END_STATUS:long_name = "Status of end date of ascent to the surface"; JULD_ASCENT_END_STATUS:conventions = "Argo reference table 19"; JULD_ASCENT_END_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/) Example: '2' : Value is transmitted by the float |
| JULD_TRANSMISSION_START | double JULD_TRANSMISSION_START(N_CYCLE); JULD_TRANSMISSION_START:long_name = "Start date of transmission"; JULD_TRANSMISSION_START:standard_name = "time"; JULD_TRANSMISSION_START:units = "days since 1950-01-01 00:00:00 UTC"; JULD_TRANSMISSION_START:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_TRANSMISSION_START:resolution = X; 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:long_name = "Status of start date of transmission"; JULD_TRANSMISSION_START_STATUS:conventions = "Argo reference table 19"; JULD_TRANSMISSION_START_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/) Example: '2' : Value is transmitted by the float |
| JULD_FIRST_MESSAGE | double JULD_FIRST_MESSAGE(N_CYCLE); JULD_FIRST_MESSAGE:long_name = "Date of earliest float message received"; JULD_FIRST_MESSAGE:standard_name = "time"; JULD_FIRST_MESSAGE:units = "days since 1950-01-01 00:00:00 UTC"; JULD_FIRST_MESSAGE:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_FIRST_MESSAGE:resolution = X; JULD_FIRST_MESSAGE:_FillValue = 999999.; | Julian day (UTC) of the earliest float message received. May or may not have a position associated with it. Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_FIRST_MESSAGE | char | Status flag on JULD date and time. |

| | | |
|----------------------------|--|--|
| JULD_FIRST_MESSAGE_STATUS | JULD_FIRST_MESSAGE_STATUS(N_CYCLE); JULD_FIRST_MESSAGE_STATUS:long_name = "Status of date of earliest float message received"; JULD_FIRST_MESSAGE_STATUS:conventions = "Argo reference table 19"; JULD_FIRST_MESSAGE_STATUS:_FillValue = " "; | The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/) Example: '2' : Value is transmitted by the float |
| JULD_FIRST_LOCATION | double JULD_FIRST_LOCATION(N_CYCLE); JULD_FIRST_LOCATION:long_name = "Date of earliest location"; JULD_FIRST_LOCATION:standard_name = "time"; JULD_FIRST_LOCATION:units = "days since 1950-01-01 00:00:00 UTC"; JULD_FIRST_LOCATION:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_FIRST_LOCATION:resolution = X; JULD_FIRST_LOCATION:_FillValue = 999999.; | Julian day (UTC) of the earliest position Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_FIRST_LOCATION_STATUS | char JULD_FIRST_LOCATION_STATUS(N_CYCLE); JULD_FIRST_LOCATION_STATUS:long_name = "Status of date of earliest location"; JULD_FIRST_LOCATION_STATUS:conventions = "Argo reference table 19"; JULD_FIRST_LOCATION_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/) Example: '2' : Value is transmitted by the float |
| JULD_LAST_LOCATION | double JULD_LAST_LOCATION(N_CYCLE); JULD_LAST_LOCATION:long_name = "Date of latest location"; JULD_LAST_LOCATION:standard_name = "time"; JULD_LAST_LOCATION:units = "days since 1950-01-01 00:00:00 UTC"; JULD_LAST_LOCATION:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_LAST_LOCATION:resolution = X; JULD_LAST_LOCATION:_FillValue = 999999.; | Julian day (UTC) of the latest position Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_LAST_LOCATION_STATUS | char JULD_LAST_LOCATION_STATUS(N_CYCLE); JULD_LAST_LOCATION_STATUS:long_name = "Status of date of latest location"; JULD_LAST_LOCATION_STATUS:conventions = "Argo reference table 19"; JULD_LAST_LOCATION_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/) Example: '2' : Value is transmitted by the float |
| JULD_LAST_MESSAGE | double JULD_LAST_MESSAGE(N_CYCLE); JULD_LAST_MESSAGE:long_name = "Date of latest float message received"; JULD_LAST_MESSAGE:standard_name = "time"; JULD_LAST_MESSAGE:units = "days since 1950-01-01 00:00:00 UTC"; JULD_LAST_MESSAGE:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_LAST_MESSAGE:resolution = X; JULD_LAST_MESSAGE:_FillValue = 999999.; | Julian day (UTC) of the latest float message received. May or may not have a position associated with it. Example : 18833.8013889885 : July 25 2001 19:14:00 |
| JULD_LAST_MESSAGE_STATUS | char JULD_LAST_MESSAGE_STATUS(N_CYCLE); JULD_LAST_MESSAGE_STATUS:long_n | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/) Example: |

| | | |
|-------------------------------------|--|---|
| | ame = "Status of date of latest float message received"; JULD_LAST_MESSAGE_STATUS:conventions = "Argo reference table 19"; JULD_LAST_MESSAGE_STATUS:_FillValue = " "; | '2' : Value is transmitted by the float |
| JULD_TRANSMISSION_END | double JULD_TRANSMISSION_END(N_CYCLE); JULD_TRANSMISSION_END:long_name = "Transmission end date"; JULD_TRANSMISSION_END:standard_name = "time"; JULD_TRANSMISSION_END:units = "days since 1950-01-01 00:00:00 UTC"; JULD_TRANSMISSION_END:conventions = "Relative julian days with decimal part (as parts of day)"; JULD_TRANSMISSION_END:resolution = X; 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:long_name = "Status of transmission end date"; JULD_TRANSMISSION_END_STATUS:conventions = "Argo reference table 19"; JULD_TRANSMISSION_END_STATUS:_FillValue = " "; | Status flag on JULD date and time. The flag scale is described in reference table 19 (https://vocab.nerc.ac.uk/collection/R19/) Example: '2' : Value is transmitted by the float |
| CLOCK_OFFSET | double CLOCK_OFFSET(N_CYCLE); CLOCK_OFFSET:long_name = "Time of float clock drift"; CLOCK_OFFSET:units = "days"; CLOCK_OFFSET:conventions = "Days with decimal part (as parts of day)"; CLOCK_OFFSET:_FillValue = 999999.; | Decimal part of day that float clock has drifted. Float clock drift is defined as Float time (provided by the inboard Real Time Clock (RTC) of the float) – UT time. This makes the clock drift less than 0 if float RTC is before UT time. Float clock drift can be corrected in real time or in delayed mode. Real time corrections correspond to a data mode of "A". For "A" mode files, JULD_ADJUSTED = JULD - CLOCK_OFFSET "D" mode files may have corrections for clock drift only or additional time corrections based on expert review. Example : -1.08546: the clock drift is estimated to be equal to – 1 day 2 hours 3 minutes and 4 seconds at the time of the corresponding cycle surfacing |
| GROUNDED | char GROUNDED(N_CYCLE); GROUNDED:long_name = "Did the profiler touch the ground for that cycle?"; GROUNDED:conventions = "Argo reference table 20"; GROUNDED:_FillValue = " "; | GROUNDED indicates the best estimate of whether the float touched the ground for that cycle. The conventions are described in Argo reference table 20 (https://vocab.nerc.ac.uk/collection/R20/) Example: Y : yes, the float touched the ground |
| REPRESENTATIVE_PARK_PRESSURE | float REPRESENTATIVE_PARK_PRESSURE(N_CYCLE); REPRESENTATIVE_PARK_PRESSURE:long_name = "Best pressure value during park phase"; REPRESENTATIVE_PARK_PRESSURE:units = "<X>"; REPRESENTATIVE_PARK_PRESSURE:_FillValue = <X>; | The Representative Park Pressure (RPP) is the best pressure value assigned to the drift phase. See reference table 21 to understand how this pressure was evaluated. It should match the PRES(N_MEASUREMENT) values with MC = 301. <X> : these fields are specified in the columns of the reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). Example: 1025 |
| REPRESENTATIVE_PARK_PRESSURE_STATUS | char REPRESENTATIVE_PARK_PRESSURE_STATUS(N_CYCLE); REPRESENTATIVE_PARK_PRESSURE_STATUS:long_name = "Status of best pressure value during park phase"; REPRESENTATIVE_PARK_PRESSURE_STATUS:conventions = "Argo reference table 21"; REPRESENTATIVE_PARK_PRESSURE_STATUS:_FillValue = " "; | Status flag on the Representative Park Pressure (RPP). The flag scale is described in reference table 21 (https://vocab.nerc.ac.uk/collection/R21/). Example: '2' : mean value, directly provided by the float, of pressure measurements regularly sampled during the drift phase |
| CONFIG_MISSION_NUM | int CONFIG_MISSION_NUMBER | Unique number of the mission to which this profile belongs. |

| | | |
|-----------------------------|--|--|
| BER | (N_CYCLE); CONFIG_MISSION_NUMBER:long_name = "Unique number denoting the missions performed by the float"; CONFIG_MISSION_NUMBER:conventions = "1...N, 1 : first complete mission"; CONFIG_MISSION_NUMBER:_FillValue = 99999; | See note on floats with multiple configurations §2.4.6.1. The number 0 (zero) can be used to denote the float's mission prelude, if it exists. |
| CYCLE_NUMBER_INDEX | int CYCLE_NUMBER_INDEX(N_CYCLE); CYCLE_NUMBER_INDEX:long_name = "Cycle number that corresponds to the current index"; CYCLE_NUMBER_INDEX:conventions = "0...N, 0 : launch cycle, 1 : first complete cycle"; CYCLE_NUMBER_INDEX:_FillValue = 99999; | Cycle number of the float that corresponds to the information contained in the current index. This cycle number must match the profile cycle number, ensuring that the trajectory and profile with the same cycle number contain data from the same cycle. Example: 17 means information for the 17 th cycle of the float is contained in this index. |
| CYCLE_NUMBER_INDEX_ADJUSTED | int CYCLE_NUMBER_INDEX_ADJUSTED(N_CYCLE); CYCLE_NUMBER_INDEX_ADJUSTED:long_name = "Adjusted cycle number that corresponds to the current index"; CYCLE_NUMBER_INDEX_ADJUSTED:conventions = "0...N, 0 : launch cycle, 1 : first complete cycle"; CYCLE_NUMBER_INDEX_ADJUSTED:_FillValue = 99999; | Corrected cycle number of the float that corresponds to the information contained in the current index. Errors may be found in CYCLE_NUMBER_INDEX variable which are corrected and contained in this variable. Example: 17 means information for the 17 th cycle of the float is contained in this index. |
| 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 trajectory cycle contains real time, adjusted or delayed mode data. A delayed mode cycle means the positions, times, cycle number, pressure, temperature, and salinity (if measured) have been quality controlled. Additional parameters like oxygen may not be quality controlled. Floats often have delayed mode data only after they die, but can have both delayed mode and real time data for different parameters while alive. When this occurs, two trajectory files exist - a real time file ("R") with only real time data for all the cycles in the float record and a delayed mode file ("D") with both real time and delayed mode data for all the cycles that have been delayed mode quality controlled. Floats can be adjusted in real time with adjusted time values only in the JULD_ADJUSTED variable and its associated _STATUS and _QC variables. This occurs when floats are corrected in real time for clock drift. Examples : 'R' : real time data 'D' : delayed mode data 'A' : real time data with JULD_ADJUSTED values |

2.3.7 Scientific calibration section

The SCIENTIFIC CALIBRATION section in the V3.2 traj files stores information of parameter calibration during park drift, descend to profile, or surface drift.

| Name | Definition | Comment |
|----------------------------|--|---|
| SCIENTIFIC_CALIB_PARAMETER | char SCIENTIFIC_CALIB_PARAMETER(N_CALIB_PARAM, N_PARAM, STRING64); SCIENTIFIC_CALIB_PARAMETER:long_name = "List of parameters with calibration information"; SCIENTIFIC_CALIB_PARAMETER:conventions = "Argo reference table 3"; SCIENTIFIC_CALIB_PARAMETER:_FillValue = " "; | Name of the calibrated parameter. The list of parameters is in reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). Example: DOXY |
| SCIENTIFIC_CALIB_EQUATION | char SCIENTIFIC_CALIB_EQUATION(N_CALIB_PARAM, N_PARAM, STRING256); SCIENTIFIC_CALIB_EQUATION:long_name = "Calibration equation for this parameter"; SCIENTIFIC_CALIB_EQUATION:_FillValue = " "; | Calibration equation applied to the parameter. Example: $T_c = a_1 * T + a_0$ |

| | | |
|------------------------------|---|---|
| SCIENTIFIC_CALIB_COEFFICIENT | char SCIENTIFIC_CALIB_COEFFICIENT(N_CALIB_PARAM, 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_CALIB_PARAM, N_PARAM, STRING256); SCIENTIFIC_CALIB_COMMENT:long_name = "Comment applying to this parameter calibration"; SCIENTIFIC_CALIB_COMMENT:_FillValue = " "; | Comment about this calibration. Example: "MC<500 and MC>600: DOXY adjustment based on comparison of surface in-air data with NCEP atmospheric reanalysis following Bittig et al. (2018)." |
| SCIENTIFIC_CALIB_DATE | char SCIENTIFIC_CALIB_DATE(N_CALIB_PARAM, N_PARAM, DATE_TIME); SCIENTIFIC_CALIB_DATE:long_name = "Date of calibration"; SCIENTIFIC_CALIB_DATE:conventions = "YYYYMMDDHHMISS"; SCIENTIFIC_CALIB_DATE:_FillValue = " "; | Date of the calibration. Example: 20011217161700 |

2.3.8 JULD calibration section

The JULD CALIBRATION section in the V3.2 traj files stores information of JULD calibration.

| Name | Definition | Comment |
|------------------------|--|--|
| JULD_CALIB_EQUATION | char JULD_CALIB_EQUATION(N_CALIB_JULD, STRING256); JULD_CALIB_EQUATION:long_name = "Calibration equation for JULD"; JULD_CALIB_EQUATION:_FillValue = " "; | Calibration equation applied to JULD. Example: JULD_ADJUSTED = a1*JULD + a0 |
| JULD_CALIB_COEFFICIENT | char JULD_CALIB_COEFFICIENT(N_CALIB_JULD, STRING256); JULD_CALIB_COEFFICIENT:long_name = "Calibration coefficients for JULD equation"; JULD_CALIB_COEFFICIENT:_FillValue = " "; | Calibration coefficients for this equation. Example: a1=0.99997, a0=0.0021 |
| JULD_CALIB_COMMENT | char JULD_CALIB_COMMENT(N_CALIB_JULD, STRING256); JULD_CALIB_COMMENT:long_name = "Comment applying to JULD calibration"; JULD_CALIB_COMMENT:_FillValue = " "; | Comment about this calibration. Example: "Float times have been linearly corrected from float clock offset (reported in CLOCK_OFFSET(N_CYCLE))." |
| JULD_CALIB_DATE | char JULD_CALIB_DATE(N_CALIB_JULD, DATE_TIME); JULD_CALIB_DATE:long_name = "Date of JULD calibration"; JULD_CALIB_DATE:conventions = "YYYYMMDDHHMISS"; JULD_CALIB_DATE:_FillValue = " "; | Date of the calibration. Example: 20011217161700 |

2.3.9 History information

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

Each item of this section has a 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 (https://vocab.nerc.ac.uk/collection/R04/). 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 (https://vocab.nerc.ac.uk/collection/R12/). 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 : "OW" |
| 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, 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 (https://vocab.nerc.ac.uk/collection/R07/). Example : "QCF\$" for QC failed |
| HISTORY_PARAMETER | char HISTORY_PARAMETER (N_HISTORY, STRING64); HISTORY_PARAMETER:long_name = "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); | Parameter or flag of the previous value before action. Example : '2' (probably good) for a flag that was |

| | | |
|-------------------------|--|--|
| | ORY); HISTORY_PREVIOUS_VALUE:long_name = "Parameter/Flag previous value before action"; HISTORY_PREVIOUS_VALUE:_FillValue = 99999; | changed to '1' (good) |
| HISTORY_INDEX_DIMENSION | char HISTORY_INDEX_DIMENSION(N_HISTORY); HISTORY_INDEX_DIMENSION:long_name = "Name of dimension to which HISTORY_START_INDEX and HISTORY_STOP_INDEX correspond"; HISTORY_INDEX_DIMENSION:conventions = "C: N_CYCLE, M: N_MEASUREMENT"; HISTORY_INDEX_DIMENSION:_FillValue = " "; | Name of dimension to which HISTORY_START_INDEX and HISTORY_STOP_INDEX correspond. 'C': N_CYCLE 'M': N_MEASUREMENT |
| HISTORY_START_INDEX | int HISTORY_START_INDEX (N_HISTORY); HISTORY_START_INDEX:long_name = "Start index action applied on"; HISTORY_START_INDEX:_FillValue = 99999; | Start index the action is applied to. This index corresponds to N_MEASUREMENT or N_CYCLE, depending on the corrected parameter Example : 100 |
| HISTORY_STOP_INDEX | int HISTORY_STOP_INDEX (N_HISTORY); HISTORY_STOP_INDEX:long_name = "Stop index action applied on"; HISTORY_STOP_INDEX:_FillValue = 99999; | Stop index the action is applied to. This index corresponds to N_MEASUREMENT or N_CYCLE, depending on the corrected parameter Example : 150 |
| HISTORY_QCTEST | char HISTORY_QCTEST(N_HISTORY, 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 described in reference table 11 (https://vocab.nerc.ac.uk/collection/R11/). Example : "0A" (in hexadecimal form) |

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

2.4 Metadata format version 3.1

The format version 3.1 of Argo metadata will replace versions 2.2 and 2.4 gradually. During the transition period, all formats will be valid. However, when a Data Assembly Center (DAC) produces metadata files with the new 3.1 format, all its metadata files must be provided in version 3.1.

An Argo metadata file contains descriptive information about an Argo float.

For file naming conventions, see §4.1.

2.4.1 Global attributes

The global attributes section is used for data discovery. It complies with NetCDF Climate and Forecast (CF) Metadata Conventions (version 1.6).

- <http://cfconventions.org/>

The following global attributes should appear in the global section:

```
// 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 = "3.4";
:Conventions = "Argo-3.1 CF-1.6";

```

| 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 begins 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. |
| user_manual_version | The version number of the user manual |
| Conventions | The conventions supported by this file, blank separated |

2.4.2 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 STRING128 STRING64 STRING32 STRING17 STRING16 STRING8 STRING4 STRING2 | STRING1024 = 1024; STRING256 = 256; STRING128 = 128; STRING64 = 64; STRING32 = 32; STRING17 = 17; STRING16 = 16; STRING8 = 8; STRING4 = 4; STRING2 = 2; | String dimensions from 2 to 1024. |
| 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_SENSOR | N_SENSOR = <int value>; | Number of sensors mounted on the float and used to measure the parameters. |
| N_CONFIG_PARAM | N_CONFIG_PARAM=<int value>; | Number of configuration parameters. |
| N_LAUNCH_CONFIG_PARAMETER | N_LAUNCH_CONFIG_PARAMETER=<int value>; | Number of pre-deployment or launch 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. |

2.4.3 General information on the metadata 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:conventions = "Argo reference table 1"; DATA_TYPE:_FillValue = " "; | This field contains the type of data contained in the file. The list of acceptable data types is in thereference table 1 (https://vocab.nerc.ac.uk/collection/R01/). Example : Argo meta-data |
| FORMAT_VERSION | char FORMAT_VERSION(String4); FORMAT_VERSION:long_name = "File format version"; FORMAT_VERSION:_FillValue = " "; | File format version. Example : «3.1» |
| 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 29 th 2001 16:17:00 |
| DATE_UPDATE | char DATE_UPDATE(Date_Time); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " "; | Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 th 2001 09:05:00 |

2.4.4 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 |
| PLATFORM_WIGOS_ID | char PLATFORM_WIGOS_ID(String17); PLATFORM_WIGOS_ID:long_name = "Float unique identifier"; PLATFORM_WIGOS_ID:conventions = "WMO WIGOS float identifier: 0-22000-0-A9IIIII"; PLATFORM_WIGOS_ID:_FillValue = " "; | WMO WIGOS float identifier. WMO is the World Meteorological Organization. This platform number is unique. Example : 0-22000-0-6900045 "0-22000-0-" is the Argo floats prefix |
| PTT | char PTT(String256); PTT:long_name = "Transmission identifier (ARGOS, ORBCOMM, etc.)"; PTT:_FillValue = " "; | Transmission identifier of the float. Comma separated list for multi-beacon transmission. Example : "22507": the float is equipped with one ARGOS beacon. "22598, 22768": the float is equipped with 2 ARGOS beacons. |
| TRANS_SYSTEM | char TRANS_SYSTEM(N_TRANS_SYSTEM, String16); TRANS_SYSTEM:long_name = "Telecommunication system used"; TRANS_SYSTEM:_FillValue = " "; | Name of the telecommunication system from reference table 10 (https://vocab.nerc.ac.uk/collection/R10/). Example : ARGOS |

| | | |
|--------------------|---|--|
| TRANS_SYSTEM_ID | char TRANS_SYSTEM_ID(N_TRANS_SYSTEM, STRING32); TRANS_SYSTEM_ID:long_name = "Program identifier used by the transmission system"; TRANS_SYSTEM_ID:_FillValue = " "; | Program identifier of the telecommunication subscription. DACs can use N/A or alternative of their choice when not applicable (e.g. : Iridium or Orbcomm) Example: 38511 is a program number for all the beacons of an ARGOS customer. |
| TRANS_FREQUENCY | char TRANS_FREQUENCY(N_TRANS_SYSTEM, STRING16); TRANS_FREQUENCY:long_name = "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 (https://vocab.nerc.ac.uk/collection/R09/). ARGOS or GPS are 2 positioning systems. Example : ARGOS |
| PLATFORM_FAMILY | char PLATFORM_FAMILY(STRING256); PLATFORM_FAMILY:long_name = "Category of instrument"; PLATFORM_FAMILY:conventions = "Argo reference table 22"; PLATFORM_FAMILY:_FillValue = " "; | Category of instrument. See Argo reference table 22 Example: FLOAT, ICE_TETHERED_PROFILER, FLOAT_DEEP. |
| PLATFORM_TYPE | char PLATFORM_TYPE(STRING32); PLATFORM_TYPE:long_name = "Type of float"; PLATFORM_TYPE:conventions = "Argo reference table 23"; PLATFORM_TYPE:_FillValue = " "; | Type of float. See Argo reference table 23 (https://vocab.nerc.ac.uk/collection/R23/). Example: SOLO, APEX, PROVOR, ARVOR, NINJA |
| PLATFORM_MAKER | char PLATFORM_MAKER(STRING256); PLATFORM_MAKER:long_name = "Name of the manufacturer"; PLATFORM_MAKER:conventions = "Argo reference table 24"; PLATFORM_MAKER:_FillValue = " "; | Name of the manufacturer. Example: MARTEC, MRV, TWR. See Argo reference table 24 (https://vocab.nerc.ac.uk/collection/R24/). |
| FIRMWARE_VERSION | char FIRMWARE_VERSION(STRING64); FIRMWARE_VERSION:long_name = "Firmware version for the float"; FIRMWARE_VERSION:_FillValue = " "; | The firmware version. This is specified as per the format on the manufacturer's manual. Example: 072804 The dimension STRING32 instead of STRING64 remains accepted. |
| MANUAL_VERSION | char MANUAL_VERSION(STRING16); MANUAL_VERSION:long_name = "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(STRING32); FLOAT_SERIAL_NO:long_name = "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); STANDARD_FORMAT_ID:long_name = "Standard format number to describe the data format type for each float"; STANDARD_FORMAT_ID:_FillValue = " "; | Standardized format number as described in the online reference table: https://docs.google.com/spreadsheets/d/16ITji2i9tLh2PfiIMItfhaQybQB1NLAhj4s0mDML5yY/edit#gid=9 This table cross references to individual DAC format numbers. Example: 1010151 |
| DAC_FORMAT_ID | char DAC_FORMAT_ID(STRING16); DAC_FORMAT_ID:long_name = "Format number used by the DAC to describe the data format type for each float"; DAC_FORMAT_ID:_FillValue = " "; | Format numbers used by individual DACs to describe each float type. This is cross-referenced to a standard format id as outlined in the online reference table: https://docs.google.com/spreadsheets/d/16ITji2i9tLh2PfiIMItfhaQybQB1NLAhj4s0mDML5yY/edit#gid=9 Example: A9. |

| | | |
|---------------|--|--|
| 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 (https://vocab.nerc.ac.uk/collection/R08/). Example : 846 : Webb Research float, Seabird sensor |
| PROJECT_NAME | char PROJECT_NAME(String64); PROJECT_NAME:long_name = "Name of the project"; PROJECT_NAME:_FillValue = " "; | Name of the project that operates the float. Multiple projects can be separated by commas. Example: "GYROSCOPE, GMMC" |
| PROGRAM_NAME | char PROGRAM_NAME(String64); PROGRAM_NAME:long_name = "Name of the program"; PROGRAM_NAME:_FillValue = " "; | This is an optional variable. The overarching program(s) of which the float is a part. A program consists of a set of related and possibly interdependent projects (PROJECT_NAME) that meet an overarching objective. A program defines a group of floats managed by the same lead agency. It materializes the implementing, operating, and responsible team. PROGRAM_NAME are managed by OceanOPS. Detailed definition of PROGRAM_NAME is available here: https://www.ocean-ops.org/metadata/#concept-definitions . Example : "Argo India" or "Argo GO-BGC, UW". This is a constrained vocabulary (R41). |
| 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 (https://vocab.nerc.ac.uk/collection/R04/). 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 responsible for the profiling float. Example: "Julia UITZ". Valid PI names are listed in reference table 40 (https://vocab.nerc.ac.uk/collection/R40/). Multiple names can be concatenated, separated by commas. |
| 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 = "Type of battery packs in the float"; BATTERY_TYPE:_FillValue = " "; | Describes the type of battery packs in the float. BATTERY_TYPE = "Manufacturer Alkaline x V" or "Manufacturer Lithium x V + Manufacturer Alkaline x V" Example: "ELECTROCHEM Lithium 12 V" Use reference table 29 (not in the NVS) |
| BATTERY_PACKS | char BATTERY_PACKS(String64); BATTERY_PACKS:long_name = "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" Use reference table 30 (not in the NVS) |

| | | |
|--------------------------------------|--|--|
| CONTROLLER_BOARD_TYPE_PRIMARY | char CONTROLLER_BOARD_TYPE_PRIMARY(String32); CONTROLLER_BOARD_TYPE_PRIMARY:long_name = "Type of primary controller board"; CONTROLLER_BOARD_TYPE_PRIMARY:_FillValue = " "; | Describes the type of controller board. first part of string: see reference table 28 (https://vocab.nerc.ac.uk/collection/R28/). Remainder of string free text and contained in square brackets Example: "APF9 [iridium version xyz]" Use reference table 28 |
| CONTROLLER_BOARD_TYPE_SECONDARY | char CONTROLLER_BOARD_TYPE_SECONDARY(String32); CONTROLLER_BOARD_TYPE_SECONDARY:long_name = "Type of secondary controller board"; CONTROLLER_BOARD_TYPE_SECONDARY:_FillValue = " "; | This is an optional variable. Only applicable if there is more than one controller board in the float. Describes the second type of controller board. first part of string: see reference table 28 (https://vocab.nerc.ac.uk/collection/R28/). Remainder of string free text and contained in square brackets Example: "APF9 [iridium version xyz]" Use reference table 28 |
| CONTROLLER_BOARD_SERIAL_NO_PRIMARY | char CONTROLLER_BOARD_SERIAL_NO_PRIMARY(String32); CONTROLLER_BOARD_SERIAL_NO_PRIMARY:long_name = "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 = "Serial number of the secondary controller board"; CONTROLLER_BOARD_SERIAL_NO_SECONDARY:_FillValue = " "; | This is an optional variable. Only applicable if there is more than one controller board in the float. The serial number for the secondary controller board. Example: 4571 |
| SPECIAL_FEATURES | char SPECIAL_FEATURES(String1024); SPECIAL_FEATURES:long_name = "Extra features of the float (algorithms, compresses etc.)"; SPECIAL_FEATURES:_FillValue = " "; | Additional float features can be specified here such as algorithms used by the float. Example: Ice Sensing Algorithm, Interim Storage Algorithm, grounding avoidance or additional hardware such as a compresses (buoyancy compensator). See reference table 32 (not in the NVS) |
| FLOAT_OWNER | char FLOAT_OWNER (String64); FLOAT_OWNER:long_name = "Float owner"; FLOAT_OWNER:_FillValue = " "; | The owner of the float (may be different from the data centre and operating institution). Example: Joe Blogg |
| OPERATING_INSTITUTION | char OPERATING_INSTITUTION(String64); OPERATING_INSTITUTION:long_name = "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: Float stability collar removed for deployment in ice. |

2.4.5 Float deployment and mission information

| Name | Definition | Comment |
|------|------------|---------|
|------|------------|---------|

| | | |
|------------------|---|--|
| LAUNCH_DATE | char LAUNCH_DATE(DATE_TIME); LAUNCH_DATE:long_name = "Date (UTC) of the deployment"; LAUNCH_DATE:conventions = "YYYYMMDDHHMISS"; LAUNCH_DATE:_FillValue = " "; | Date and time (UTC) of launch of the float. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 th 2001 03:05:00 |
| LAUNCH_LATITUDE | double LAUNCH_LATITUDE; LAUNCH_LATITUDE:long_name = "Latitude of the float when deployed"; LAUNCH_LATITUDE:units = "degree_north"; LAUNCH_LATITUDE:_FillValue = 99999.; LAUNCH_LATITUDE:valid_min = -90.; LAUNCH_LATITUDE:valid_max = 90.; | Latitude of the launch. Unit : degree north. Example : 44.4991 : 44° 29' 56.76" N |
| LAUNCH_LONGITUDE | double LAUNCH_LONGITUDE; LAUNCH_LONGITUDE:long_name = "Longitude of the float when deployed"; LAUNCH_LONGITUDE:units = "degree_east"; LAUNCH_LONGITUDE:_FillValue = 99999.; LAUNCH_LONGITUDE:valid_min = -180.; LAUNCH_LONGITUDE:valid_max = 180.; | Longitude of the launch. Unit : degree east. Example : 16.7222 : 16° 43' 19.92" E |
| LAUNCH_QC | char LAUNCH_QC; LAUNCH_QC:long_name = "Quality on launch date, time and location"; LAUNCH_QC:conventions = "Argo reference table 2"; LAUNCH_QC:_FillValue = " "; | Quality flag on launch date, time and location. The flag scale is described in the reference tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed mode). Example : '1' : launch location seems correct. |
| START_DATE | char START_DATE(DATE_TIME); START_DATE:long_name = "Date (UTC) of the first descent of the float"; START_DATE:conventions = "YYYYMMDDHHMISS"; START_DATE:_FillValue = " "; | Date and time (UTC) of the first descent of the float. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 th 2001 06 :05 :00 |
| START_DATE_QC | char START_DATE_QC; START_DATE_QC:long_name = "Quality on start date"; START_DATE_QC:conventions = "Argo reference table 2"; START_DATE_QC:_FillValue = " "; | Quality flag on start date. The flag scale is described in the reference tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed mode). Example : '1' : start date seems correct. |
| STARTUP_DATE | char STARTUP_DATE(DATE_TIME); STARTUP_DATE:long_name = "Date (UTC) of the activation of the float"; STARTUP_DATE:conventions = "YYYYMMDDHHMISS"; STARTUP_DATE:_FillValue = " "; | Date and time (UTC) of the activation of the float before or just after deployment. This may include automatic startup during pressure activation. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 th 2001 06 :05 :00 |
| STARTUP_DATE_QC | char STARTUP_DATE_QC; STARTUP_DATE_QC:long_name = "Quality on startup date"; STARTUP_DATE_QC:conventions = "Argo reference table 2"; STARTUP_DATE_QC:_FillValue = " "; | Quality flag on startup date. The flag scale is described in the reference tables 2 (https://vocab.nerc.ac.uk/collection/RR2/ for real time and https://vocab.nerc.ac.uk/collection/RD2/ for delayed mode). Example : '1' : start date seems correct. |

| | | |
|---------------------------------|---|--|
| DEPLOYMENT_PLATFORM | char DEPLOYMENT_PLATFORM(STRING128); DEPLOYMENT_PLATFORM:long_name = "Identifier of the deployment platform"; DEPLOYMENT_PLATFORM:_FillValue = " "; | Identifier of the deployment platform, i.e. vessel or ship name. Example : "Ronald H. Brown" uri: http://vocab.nerc.ac.uk/collection/C17/current/33RO/ " char DEPLOYMENT_PLATFORM(STRING32) is also accepted. |
| 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"; DEPLOYMENT_CRUISE_ID:_FillValue = " "; | Identification number or reference number of the cruise used to deploy the platform. Example : POMME2 |
| DEPLOYMENT_REFERENCE_STATION_ID | char DEPLOYMENT_REFERENCE_STATION_ID(STRING256); DEPLOYMENT_REFERENCE_STATION_ID:long_name = "Identifier or reference number of co-located stations used to verify the first profile"; DEPLOYMENT_REFERENCE_STATION_ID:_FillValue = " "; | Identifier or reference number of co-located CTD or XBT stations used to verify the first profile. Example : 58776, 58777 |
| END_MISSION_DATE | char END_MISSION_DATE(DATE_TIME); END_MISSION_DATE:long_name = "Date (UTC) of the end of mission of the float"; END_MISSION_DATE:conventions = "YYYYMMDDHHMISS"; END_MISSION_DATE:_FillValue = " "; | Date (UTC) of the float's deployment. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 th 2001 03:05:00 |
| END_MISSION_STATUS | char END_MISSION_STATUS; END_MISSION_STATUS:long_name = "Status of the end of mission of the float"; END_MISSION_STATUS:conventions = "T:No more transmission received, R:Retrieved"; END_MISSION_STATUS:_FillValue = " "; | Status of the end of the float's deployment. 'T':No more transmission received, 'R':Retrieved |

2.4.6 Configuration parameters

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

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

Configuration parameter names 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 |
|-------------------------------|--|---|
| LAUNCH_CONFIG_PARAMETER_NAME | char LAUNCH_CONFIG_PARAMETER_NAME(N_LAUNCH_CONFIG_PARAM, STRING128) LAUNCH_CONFIG_PARAMETER_NAME:long_name = "Name of configuration parameter at launch"; LAUNCH_CONFIG_PARAMETER_NAME:_FillValue = " "; | Name of the configuration parameter; pre-deployment or launch settings. See reference tables 18a and 18b (https://vocab.nerc.ac.uk/collection/R18/) for standard configuration parameter names. Example : "CONFIG_ParkPressure_dbar" |
| LAUNCH_CONFIG_PARAMETER_VALUE | float (for standard Argo floats) or can use double if applicable (e.g. BGC-Argo float) LAUNCH_CONFIG_PARAMETER_VALUE(N_LAUNCH_CONFIG_PARAM) LAUNCH_CONFIG_PARAMETER_VALUE:long_name = "Value of configuration parameter at launch"; LAUNCH_CONFIG_PARAMETER_VALUE:_FillValue = 99999.f; | Value of the configuration parameter; either pre-deployment or launch settings. Example : 1500 |

| | | |
|------------------------|--|--|
| CONFIG_PARAMETER_NAME | char CONFIG_PARAMETER_NAME(N_CONFIG_PARAM, STRING128) CONFIG_PARAMETER_NAME:long_name = "Name of configuration parameter"; CONFIG_PARAMETER_NAME:_FillValue = " "; | Name of the configuration parameter; mission settings. See reference tables 18a and 18b (https://vocab.nerc.ac.uk/collection/R18/) for standard configuration parameter names. Example : "CONFIG_ParkPressure_dbar" |
| CONFIG_PARAMETER_VALUE | float (for standard Argo floats) or can use double if applicable (e.g. BGC-Argo float) CONFIG_PARAMETER_VALUE(N_MISSIONS, N_CONFIG_PARAM) CONFIG_PARAMETER_VALUE:long_name = "Value of configuration parameter"; CONFIG_PARAMETER_VALUE:_FillValue = 99999.f; | Value of the configuration parameter; mission settings. Example : 1500 |
| CONFIG_MISSION_NUMBER | int CONFIG_MISSION_NUMBER(N_MISSIONS); CONFIG_MISSION_NUMBER:long_name = "Unique number denoting the missions performed by the float"; CONFIG_MISSION_NUMBER:conventions = "1...N, 1 : first complete mission"; CONFIG_MISSION_NUMBER:_FillValue = 99999; | Unique number of the mission to which this profile belongs. See note on floats with multiple configurations §2.4.6.1. The number 0 (zero) can be used to denote the float's mission prelude, if it exists. |
| CONFIG_MISSION_COMMENT | char CONFIG_MISSION_COMMENT(N_MISSIONS, STRING256) CONFIG_MISSION_COMMENT:long_name = "Comment on configuration"; CONFIG_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 tables 18a and 18b - <https://vocab.nerc.ac.uk/collection/R18/>). 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 the online reference tables 18a and 18b: <http://www.argodatamgt.org/Documentation>

There are two configuration tables: one for Core Argo configuration parameters (reference table 18a) and one for floats with BGC-Argo sensors (reference table 18b). If you are a new user or have only standard Argo floats you should first check table 18a which has the basic set of configuration parameters that caters for most of the standard float types. If your float carries additional BGC-Argo sensors you will also need to refer to table 18b. Please send requests for new configuration parameter names for Core Argo floats to John Gilson (jgilson@ucsd.edu). Requests for configuration parameter names for BGC-Argo variables should be sent to Catherine Schmechtig (catherine.schmechtig@imev-mer.fr).

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.

Configuration parameters are separated into two types.

1). Pre-deployment or launch configuration parameters that are the 'configured' start settings of the float and the initial mission configuration parameters for the first cycle. These parameters with the dimension N_LAUNCH_CONFIG_PARAM, are stored in LAUNCH_CONFIG_PARAMETER_NAME and LAUNCH_CONFIG_PARAMETER_VALUE.

2). After deployment, configuration parameters define float behavior for each mission, including all applicable mandatory and highly-desirable parameters and any other parameters that change during the life of the float are reported as mission settings. These parameters with the dimension `N_CONFIG_PARAM`, need to be reported in the `CONFIG_PARAMETER_NAME` and `CONFIG_PARAMETER_VALUE` variables. The mission parameters for the first cycle must also be reported in the launch section.

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).

2.4.6.1 Note on floats with multiple missions

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

However, some floats may be configured to change their behavior 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 a set of pre-deployment/launch configuration information that contains all the start settings for the float. So for a float with one basic mission, it will have launch configuration parameters and mission 1 configuration parameters (typically just the mission critical subset).

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: `LAUNCH_CONFIG_PARAMETER_NAME` and `LAUNCH_CONFIG_PARAMETER_VALUE`.

Argo best practice, and our recommendation to users, is a minimum of configuration missions. If there is a change to one or more of the configuration parameters that does not repeat a previous configuration then a new mission number must 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.

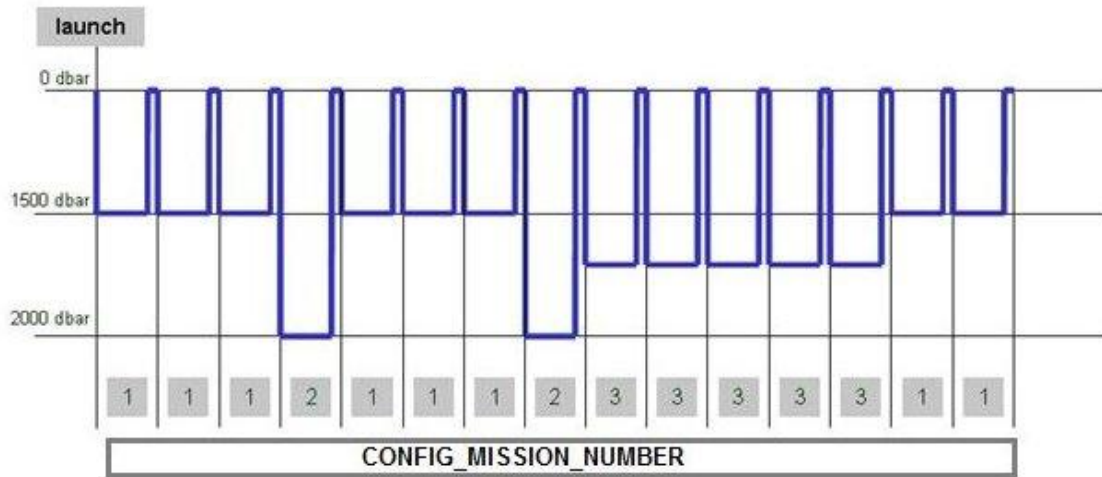
Some floats conduct a pressure-activated test mission after deployment (i.e. some SOLO floats) where they essentially ‘wake-up’ at depth and return to the surface. This is not a normal mission in any way, and the floats do not use their configuration parameters in this first ‘test’ cycle. In this case you would assign a fill value to `CONFIG_MISSION_NUMBER`.

2.4.6.2 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 in the trajectory files, and the `CONFIG_MISSION_NUMBER(N_PROF)` variable in the profile files.

See section “§2.2.4 General information for each profile” and “§2.3.6 `N_CYCLE` dimension variable group”).

Example



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

```
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 and changes to multiple configuration parameters is shown below.

In this example there are 11 pre-deployment/launch configuration parameters for this float which are set before launch. Of this set, the last four are mission critical parameters that change and control float behavior; these are reported in subsequent missions (missions 1 to n). For this float, the depth at which the float parks, changes in missions 2 and 3 (with changes in two configuration parameters). For mission 4, the park depth stays the same as for mission 3 but now the park sampling period has changed. For mission 5, the park sampling period changes again. All configuration parameters from mission 1 must still be reported for each subsequent mission, even those that do not change.

| Configuration parameter names from reference table 18a or 18b LAUNCH_CONFIG_PARAMETER_NAME(N_LAUNCH_CONFIG_PARAM,STRING128) * N_LAUNCH_CONFIG_PARAM = int value | Pre-deployment or launch configuration settings LAUNCH_CONFIG_PARAMETER_VALUE(N_LAUNCH_CONFIG_PARAM) * float or double depending on data type |
|---|---|
| CONFIG_PistonPositionPressureActivation_COUNT | 100 |
| CONFIG_Direction_NUMBER | 1* |
| CONFIG_AscentToSurfaceTimeout_hours | 3 |
| CONFIG_MissionPreludeTime_hours | 3 |
| CONFIG_MeasureBattery_LOGICAL | 0** |

| | |
|---------------------------------|-------|
| CONFIG_IceDetection_DegC | -1.78 |
| CONFIG_CycleTime_hours | 240 |
| CONFIG_ParkPressure_dbar | 1000 |
| CONFIG_ProfilePressure_dbar | 2000 |
| CONFIG_ParkSamplingPeriod_hours | 10 |
| CONFIG_PistonPark_COUNT | 111 |
| ... | ... |

(*): 1 = Ascending, 2 = Descending

(**): 0 = No, 1 = Yes

| Configuration parameter names (from reference table 18a or 18b) CONFIG_PARAMETER_NAME(N_CONFIG_PARAM,STRING128) | Mission configuration settings at sea CONFIG_PARAMETER_VALUE(N_MISSIONS,N_CONFIG_PARAM) | | | | | |
|--|--|------|------|------|------|-----|
| CONFIG_MISSION_NUMBER | 1 | 2 | 3 | 4 | 5 | ... |
| CONFIG_ParkPressure_dbar | 1000 | 1500 | 1700 | 1700 | 1000 | ... |
| CONFIG_ProfilePressure_dbar | 2000 | 2000 | 2000 | 2000 | 2000 | ... |
| CONFIG_ParkSamplingPeriod_hours | 10 | 10 | 10 | 15 | 17 | ... |
| CONFIG_PistonPark_COUNT | 111 | 75 | 53 | 53 | 45 | ... |
| ... | ... | ... | ... | ... | ... | ... |

2.4.6.3 Highly desirable configuration parameters

Highly desirable configuration parameters should be correctly filled according to the following table.

| Highly desirable configuration parameter | Format | Example |
|---|---------------------|---------|
| CONFIG_CycleTime_hours CONFIG_CycleTime_days | Duration of a cycle | 240 |
| CONFIG_ParkPressure_dbar | Park pressure | 1900 |
| CONFIG_ProfilePressure_dbar | Profile pressure | 2000 |

2.4.7 Float sensor and parameter information

A **sensor** is a device used to measure a physical parameter. Sensor outputs are provided in parameter counts and need to be converted into parameter physical units using a calibration equation. This conversion can be done onboard the float or during the decoding process.

A **parameter** is a measurement of a physical phenomenon; it can be provided by a sensor (in sensor counts or in physical units) or computed (derived) from other parameters.

A sensor can measure 1 to N parameter(s). A parameter can be measured by 1 or N sensor(s).

2.4.7.1 Float sensor information

This section contains information about the sensors of the profiler.

A list of standardized Argo sensor names is given in reference table 25

(<https://vocab.nerc.ac.uk/collection/R25/>).

Each ocean state **parameter** recorded in an Argo profile or trajectory file must be considered its own sensor in this context. This is necessary to record the sensor serial numbers, etc. Thus a CTD which records PRES, TEMP, and PSAL should be considered 3 separate sensors (CTD_PRES, CTD_TEMP, CTD_CNDC respectively), as all 3 parameters are reported in the Argo profile and trajectory files. In comparison, an OPTODE only has DOXY reported in Argo NetCDF files, thus it should be considered a single sensor (OXYGEN_OPTODE or OXYGEN_ELECTROCHEMICAL) even though the sensor also

reports TEMP_DOXY. All intermediate variables that do not appear in the core Argo files, even though they may appear in the 'raw' form (i.e. in the BGC-Argo B-file), do not have to be listed as a separate sensor. This is especially true if there is not a unique serial number for the sensor.

| Name | Definition | Comment |
|-------------------------|---|--|
| SENSOR | char SENSOR(N_SENSOR, STRING32); SENSOR:long_name = "Name of the sensor mounted on the float"; SENSOR:conventions = "Argo reference table 25"; SENSOR:_FillValue = " "; | Name of the sensor mounted on the float Example: CTD_PREC, CTD_TEMP, CTD_CNDC, OPTODE_DOXY. Valid SENSOR are listed in reference table R25 (https://vocab.nerc.ac.uk/collection/R25/). |
| SENSOR_MAKER | char SENSOR_MAKER(N_SENSOR, STRING256); SENSOR_MAKER:long_name = "Name of the sensor manufacturer"; SENSOR_MAKER:conventions = "Argo reference table 26"; SENSOR_MAKER:_FillValue = " "; | Name of the manufacturer of the sensor. Example : DRUCK, SBE, AANDERAA. Valid SENSOR_MAKER are listed in reference table R26 (https://vocab.nerc.ac.uk/collection/R26/). |
| SENSOR_MODEL | char SENSOR_MODEL(N_SENSOR, STRING256); SENSOR_MODEL:long_name = "Type of sensor"; SENSOR_MODEL:conventions = "Argo reference table 27"; SENSOR_MODEL:_FillValue = " "; | Name of the model of the sensor, as defined by the sensor manufacturer. Example: DRUCK, SBE41CP, AANDERAA_OPTODE_3930. Valid SENSOR_MODEL are listed in reference table R27 (https://vocab.nerc.ac.uk/collection/R27/). |
| SENSOR_SERIAL_NO | char SENSOR_SERIAL_NO(N_SENSOR, STRING16); SENSOR_SERIAL_NO:long_name = "Serial number of the sensor"; SENSOR_SERIAL_NO:_FillValue = " "; | Serial number of the sensor. Example : 2646 036 073 |
| SENSOR_FIRMWARE_VERSION | char SENSOR_FIRMWARE_VERSION(N_SENSOR, STRING32); SENSOR_FIRMWARE_VERSION:long_name = "Firmware version of the sensor"; SENSOR_FIRMWARE_VERSION:_FillValue = " "; | This is an optional variable. Firmware version of the sensor. Example: 1.141 |

2.4.7.2 Float parameter information

This section contains information about the parameters measured by the profiler or derived from profiler measurements.

| Name | Definition | Comment |
|------------------|---|---|
| PARAMETER | char PARAMETER(N_PARAM, STRING64); PARAMETER:long_name = "Name of parameter computed from float measurements"; PARAMETER:conventions = "Argo reference table 3"; PARAMETER:_FillValue = " "; | Names of the parameters measured by float sensors or derived from float measurements. The parameter names are listed in reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). Examples : TEMP, PSAL, CNDC TEMP : temperature in Celsius PSAL : practical salinity in psu CNDC : conductivity in mhos/m |
| PARAMETER_SENSOR | char PARAMETER_SENSOR(N_PARAM, STRING128); PARAMETER:long_name = "Name of the sensor that measures this parameter"; PARAMETER:conventions = "Argo reference table 25"; | Names of the sensors that measured the float parameters. See Argo reference table 25 (https://vocab.nerc.ac.uk/collection/R25/). Example: CTD_PREC, CTD_TEMP, CTD_CNDC, OXYGEN_OPTODE. |

| | | |
|----------------------|---|--|
| | PARAMETER:_FillValue = " "; | |
| PARAMETER_UNITS | char PARAMETER_UNITS(N_PARAM, STRING32); PARAMETER_UNITS:long_name = "Units of accuracy and resolution of the parameter"; PARAMETER_UNITS:_FillValue = " "; | Units of accuracy and resolution of the parameter. Example : psu |
| PARAMETER_ACCURACY | char PARAMETER_ACCURACY(N_PARAM, STRING32); PARAMETER_ACCURACY:long_name = "Accuracy of the parameter"; PARAMETER_ACCURACY:_FillValue = " "; | Accuracy of the parameter. Example: "8 micromole/l or 5%" |
| PARAMETER_RESOLUTION | char PARAMETER_RESOLUTION(N_PARAM, STRING32); PARAMETER_RESOLUTION:long_name = "Resolution of the parameter"; PARAMETER_RESOLUTION:_FillValue = " "; | Resolution of the parameter returned by the sensor (note that this is not necessarily equivalent to the resolution of the parameter returned by the float through telemetry). Example : 0.001 micromole/l |

2.4.8 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.

The PREDEPLOYMENT_CALIB_* parameters in the table below link to the PARAMETER variable listed in 1.1.7.2. It is critical that these are ordered in the same way so that calibration information is assigned to the correct parameter.

| Name | Definition | Comment |
|---------------------------------|--|--|
| PREDEPLOYMENT_CALIB_EQUATION | char PREDEPLOYMENT_CALIB_EQUATION(N_PARAM, STRING1024); PREDEPLOYMENT_CALIB_EQUATION:long_name = "Calibration equation for this parameter"; PREDEPLOYMENT_CALIB_EQUATION:_FillValue = " "; | Calibration equation for this parameter. Example : $T_c = a_1 * T + a_0$ |
| PREDEPLOYMENT_CALIB_COEFFICIENT | char PREDEPLOYMENT_CALIB_COEFFICIENT(N_PARAM, STRING1024); PREDEPLOYMENT_CALIB_COEFFICIENT:long_name = "Calibration coefficients for this equation"; PREDEPLOYMENT_CALIB_COEFFICIENT:_FillValue = " "; | Calibration coefficients for this equation. Example : $a_1=0.99997$, $a_0=0.0021$ |
| PREDEPLOYMENT_CALIB_COMMENT | char PREDEPLOYMENT_CALIB_COMMENT(N_PARAM, STRING1024); PREDEPLOYMENT_CALIB_COMMENT:long_name = "Comment applying to this parameter calibration"; PREDEPLOYMENT_CALIB_COMMENT:_FillValue = " "; | Comments applying to this parameter calibration. |

2.4.9 Mandatory metadata parameters

Part of the highly desirable metadata were made mandatory and are listed under this section.

| Mandatory metadata | Mandatory format | Example |
|------------------------------------|--|--|
| BATTERY_TYPE | <i>Manufacturer</i> from reference table 29 (not yet on the NVS). Type to include "Alkaline" or "Lithium" or "Hybrid" and total voltage | BATTERY_TYPE = " <i>Manufacturer</i> Alkaline x V" or " <i>Manufacturer</i> Lithium x V + <i>Manufacturer</i> Alkaline x V"; |
| CONTROLLER_BOARD_SERIAL_NO_PRIMARY | not empty | CONTROLLER_BOARD_SERIAL_NO_PRIMARY = "4567" |
| CONTROLLER_BOARD_TYPE_PRIMARY | first part of string: see reference table 28 (https://vocab.nerc.ac.uk/collektion/R28/). Remainder of string free text and contained in square brackets []. | CONTROLLER_BOARD_TYPE_PRIMARY = "APF9 [iridium version xyz]"; |
| DAC_FORMAT_ID | not empty | DAC_FORMAT_ID = "11"; |
| DATA_CENTRE | see reference table 4 (https://vocab.nerc.ac.uk/collektion/R04/). | 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 | if exists then not empty otherwise default value = "n/a" | 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 (https://vocab.nerc.ac.uk/collektion/RR2/ , https://vocab.nerc.ac.uk/collektion/RD2/). | LAUNCH_QC = "1"; |
| MANUAL_VERSION | if exists then not empty otherwise default value = "n/a" | MANUAL_VERSION = "004" or "041708" |
| | | |
| PI_NAME | not empty | PI_NAME = "Julia UITZ"; |
| PLATFORM_FAMILY | see reference table 22 (https://vocab.nerc.ac.uk/collektion/R22/). | PLATFORM_FAMILY = "subsurface profiling float", "ITP", "POPS"; |
| PLATFORM_MAKER | see reference table 24 (https://vocab.nerc.ac.uk/collektion/R24/). | PLATFORM_MAKER = "Optimare"; |
| PLATFORM_NUMBER | IIIII or A9IIIII | PLATFORM_NUMBER = "5900077"; |
| PLATFORM_TYPE | see reference table 23 (https://vocab.nerc.ac.uk/collektion/R23/). | PLATFORM_TYPE = "SOLO" or "APEX" or "PROVOR"; |
| POSITIONING_SYSTEM | see reference table 9 (https://vocab.nerc.ac.uk/collektion/R09/). | POSITIONING_SYSTEM = "ARGOS"; |
| PREDEPLOYMENT_CALIB_COEFFICIENT | if exists then not empty otherwise default value = "n/a" | PREDEPLOYMENT_CALIB_COEFFICIENT = "ser# = 3016 temperature coeffs: A0 = -0.0000 A1 = 0.0003 A2 = -0.0000 A3 = 0.0000"; |
| PREDEPLOYMENT_CALIB_EQUATION | if exists then not empty otherwise default value = "n/a" | 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 | if exists then not empty otherwise default value = "n/a" | PTT = "23978"; |
| SENSOR | see reference table 25 (https://vocab.nerc.ac.uk/collektion/R25/). | SENSOR = "CTD_TEMP", "CTD PRES", "CTD_CNDC", "OXYGEN_OPTODE"; |

| | | |
|--------------------|--|---|
| SENSOR_MAKER | see reference table 26 (https://vocab.nerc.ac.uk/collection/R26/). | SENSOR_MAKER = "SEABIRD"; |
| SENSOR_MODEL | see reference table 27 (https://vocab.nerc.ac.uk/collection/R27/). | SENSOR_MODEL = "SBE41" |
| SENSOR_SERIAL_NO | not empty | SENSOR_SERIAL_NO = "6785"; |
| PARAMETER | not empty, see reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). | PARAMETER = "TEMP", "PRES", "CNDC"; |
| PARAMETER_UNITS | not empty | PARAMETER_UNITS = "degree_Celsius", "decibar", "mhos/m"; |
| PARAMETER_SENSOR | not empty | PARAMETER_SENSOR = "CTD_TEMP", "OXYGEN_OPTODE"; |
| STANDARD_FORMAT_ID | reference table available online https://docs.google.com/spreadsheets/d/16iTji2i9tLh2PfiiMIthaQybQB1NLAhj4s0mDML5yY/edit#gid=9 | STANDARD_FORMAT_ID = "1010151"; |
| TRANS_FREQUENCY | if exists then not empty otherwise default value = "n/a" | TRANS_FREQUENCY = "1/44"; |
| TRANS_SYSTEM | see reference table 10 (https://vocab.nerc.ac.uk/collection/R10/). | TRANS_SYSTEM = "ARGOS"; |
| TRANS_SYSTEM_ID | If exists not empty, otherwise = "n/a" | TRANS_SYSTEM_ID = "14281"; |
| WMO_INST_TYPE | not empty | WMO_INST_TYPE = "846"; |

2.5 Technical information format version 3.1

The format version 3.1 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 3.1 format, all its technical files must be provided in version 3.1.

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

The number and the type of technical information is different from one float model to another. To be flexible, for each cycle, the name of the parameters and their values are recorded. The name of the parameters recorded may therefore change from one model of float to another. See Argo reference table 14 (<https://vocab.nerc.ac.uk/collection/R14/>).

For file naming conventions, see §4.1.6.

2.5.1 Global attributes

The global attributes section is used for data discovery. It complies with NetCDF Climate and Forecast (CF) Metadata Conventions (version 1.6).

- <http://cfconventions.org/>

// 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 = "3.4";
:Conventions = "Argo-3.1 CF-1.6";
```

| 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. |
| user_manual_version | The version number of the user manual |
| Conventions | The conventions supported by this file, blank separated |

2.5.2 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 5 th 2001 17:28:34 19971217000000 : December 17 th 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. |
| N_TECH_MEASUREMENT | N_TECH_MEASUREMENT | This optional dimension reports the number of recorded technical dated information of the file. |

2.5.3 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 : A9IIII"; 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:conventions = "Argo reference table 1"; 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 (https://vocab.nerc.ac.uk/collection/R01/). Example : "Argo technical data" |
| FORMAT_VERSION | char FORMAT_VERSION(STRING4); FORMAT_VERSION:long_name = "File format version"; FORMAT_VERSION:_FillValue = " "; | File format version Example : «3.1» |
| 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 (https://vocab.nerc.ac.uk/collection/R04/). 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 29 th 2001 16 :17 :00 |
| DATE_UPDATE | char DATE_UPDATE(DATE_TIME); DATE_UPDATE:long_name = "Date of update of this file"; DATE_UPDATE:conventions = "YYYYMMDDHHMISS"; DATE_UPDATE:_FillValue = " "; | Date and time (UTC) of update of this file. Format : YYYYMMDDHHMISS Example : 20011230090500 : December 30 th 2001 09 :05 :00 |

2.5.4 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

(<https://vocab.nerc.ac.uk/collection/R14/>).

| 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 14a (https://vocab.nerc.ac.uk/collection/R14/) 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 |

Timeseries of technical data (optional)

Some technical data are transmitted with time stamps, such as battery voltage during parking drift (Apex APF11, Arvor). Some technical data are sent in distinct parts of a cycle, such as SOLOII battery voltage on drift, ascent and surface.

This optional section is for technical data reported with a timestamp or a Measurement Code (MC).

| Name | Definition | Comment |
|-------------------|---|---|
| JULD | double JULD(N_TECH_MEASUREMENT) JULD:long_name = "Julian day (UTC) of each measurement"; 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 measurement. The integer part represents the day, the decimal part represents the time of the measurement. Date and time are in universal time coordinates. Example : 18833.8013889885 : July 25 2001 19:14:00 |
| CYCLE_NUMBER_MEAS | int CYCLE_NUMBER_MEAS(N_TECH_MEASUREMENT) | Cycle number of the measurement. Example: 4 |
| MEASUREMENT_CODE | int MEASUREMENT_CODE(N_TECH_MEASUREMENT) MEASUREMENT_CODE:long_name = "Flag referring to a measurement event in the cycle"; MEASUREMENT_CODE:conventions = "Argo reference table 15"; MEASUREMENT_CODE:_FillValue = 99999; | Flag for each event in the cycle which corresponds to Argo reference table 15 (https://vocab.nerc.ac.uk/collection/R15/). Example: 100 : any measurements made at start of descent to drift pressure |
| <TECH_PARAM> | float <TECH_PARAM>(N_TECH_MEASUREMENT) <TECH_PARAM>:long_name="<X>"; <TECH_PARAM>:units="<Y>"; | Name of the technical parameter. All names of the reference table 14a are allowed (https://vocab.nerc.ac.uk/collection/R14/). <X> : use the description provided in the |

| | | |
|--|--|--|
| | | <p>reference table 14a.</p> <p><Y> : use the units part of the technical parameter name (units allowed list is http://www.argodatamgt.org/content/download/27456/187267/file/Table%20Tech%20and%20Conf%20Units%20V2.3.xlsx).</p> <p>Example :</p> <p>To store "VOLTAGE_Battery_volts" data, use</p> <p>float VOLTAGE_Battery(N_TECH_MEASUREMENT) <TECH_PARAM>:long_name="battery voltage when battery capacity is unknown"; <TECH_PARAM>:units="volts";</p> |
|--|--|--|

2.6 B-Argo profile format additional features

A B-Argo profile/trajectory file contains all the parameters from a float, except the core-Argo parameters temperature, salinity, conductivity (TEMP, PSAL, CNDC). A float that performs only CTD measurements does not have B-Argo data files.

To accommodate non-core parameters, a series of optional additions to core-Argo profile/trajectory formats are listed here.

2.6.1 Pressure axis management in core-Argo and b-Argo profile files

This section contains information on pressure axis management for core-Argo and b-Argo profile and trajectory files.

The vertical pressure levels PRES is the simple and unambiguous link between the parameters in the core- and b- profiles. The same PRES is recorded in the core-Argo and b-Argo profile files.

PRES is the only parameter in Reference Table 3 (<https://vocab.nerc.ac.uk/collection/R03/>) duplicated in core-Argo and b-Argo profile files.

The adjusted pressure parameter PRES_ADJUSTED is only available in the core-Argo profile files. The variables PROFILE_PRES_QC, PRES_QC, PRES_ADJUSTED and PRES_ADJUSTED_ERROR are not duplicated in the b-Argo profile files.

Some single-cycle profile files will contain multiple profiles with different vertical sampling schemes ($N_PROF > 1$). In these cases, all parameters in the core-Argo and the b-Argo profile files will have the same N_PROF dimension, listed in the same order. The same set of vertical pressure levels (PRES with $N_PROF > 1$) is duplicated in the core-Argo and b-Argo profile files. Other parameters (e.g. TEMP, PSAL) that are not measured in all vertical sampling schemes will be filled with FillValue in the respective N_PROF dimension.

Users are reminded that the profile with $N_PROF = 1$ is required to be the Primary sampling profile. Please refer to Table 16 (<https://vocab.nerc.ac.uk/collection/R16/>) for detailed descriptions of the various vertical sampling schemes.

Example

Suppose a hypothetical float carries a high-resolution CTD sensor and a low-resolution nitrate sensor. In each single-cycle, this hypothetical float is configured to return a 2-dbar bin-averaged CTD profile to 1000 dbar with no corresponding nitrate measurements, and a discrete 250-dbar interval nitrate profile to 1000 dbar with no corresponding temperature and salinity measurements. The parameters in the resulting core-Argo and b-Argo profile files are formatted as follows for the two profiles in each cycle:

In the core-Argo profile file, N_PROF = 2, N_LEVELS = 500.

```
PRES      = [2, 4, 6, .....1000];
           = [250, 500, 750, 1000, FillValue, .....].
TEMP      = [T2, T4, T6, .....T1000];
           = [FillValue, .....].
PSAL      = [S2, S4, S6, .....S1000];
           = [FillValue, .....].
```

In the b-Argo profile file, N_PROF = 2, N_LEVELS = 500.

```
PRES      = [2, 4, 6, .....1000];
           = [250, 500, 750, 1000, FillValue, .....].
NITRATE   = [FillValue, .....];
           = [N250, N500, N750, N1000, FillValue, ...].
```

Other intermediate nitrate variables in the b-file are omitted in this example.

2.6.3 Management of multi-dimensional parameters

Observations are usually one dimension variables, such as temperature or salinity. However, some sensors provide multi-dimensional variables.

For example, an optical sensor for Nitrate reports a spectrum of up to 41 values for each measurement, one per wavelength.

When needed, an additional dimension is added to report the N sub-levels of spectrum observation performed on each level. The associated PARAM_QC dimension remains as (N_PROF, N_LEVELS).

- float <PARAM>(N_PROF, N_LEVELS, N_VALUES);
- char <PARAM>_QC(N_PROF, N_LEVELS);

Example of 60 measurements of “RAW_DOWNWELLING_IRRADIANCE” parameter performed at each 41 wavelengths of an individual profile.

- NPROF = 1
- N_LEVELS = 60
- N_VALUES41 = 41

The N_VALUES## dimension is used only when it is necessary: if there is more than one value for each level (N_VALUES > 1) and the ## equals the number of values present for a given variable. You can add more than one of these, with each variable referencing a different N_VALUES## if required.

| | | |
|------------|-------------------------------|---|
| N_VALUES41 | N_VALUES41 = <int value> ; | Maximum number of parameter measurements sampled at a given pressure level. This dimension depends on the data set. Example : N_VALUES41 = 41 |
|------------|-------------------------------|---|

To describe wavelengths of the sensor (41 in the example), an attribute of the variable called `wave_length_nanometer` provides the list

```
double RAW_DOWNWELLING_IRRADIANCE(N_PROF, N_LEVELS, N_VALUES41);
    RAW_DOWNWELLING_IRRADIANCE:long_name = "IRRADIANCE COUNTS FROM OCR SENSOR" ;
    RAW_DOWNWELLING_IRRADIANCE:standard_name = "TBD" ;
    RAW_DOWNWELLING_IRRADIANCE:_FillValue = 99999. ;
    RAW_DOWNWELLING_IRRADIANCE:units = "counts" ;
    RAW_DOWNWELLING_IRRADIANCE:valid_min = "TBD" ;
    RAW_DOWNWELLING_IRRADIANCE:valid_max = "TBD" ;
    RAW_DOWNWELLING_IRRADIANCE:C_format = "%10.0f" ;
    RAW_DOWNWELLING_IRRADIANCE:FORTRAN_format = "F10.0" ;
    RAW_DOWNWELLING_IRRADIANCE:resolution = 1. ;
    RAW_DOWNWELLING_IRRADIANCE:wave_length_nanometer = "115 132 149 166 183 200 217 234 251 268 285 302
319 336 353 370 387 404 421 438 455 472 489 506 523 540 557 574 591 608 625 642 659 676 693 710 727 744 761 778 795" ;
```

2.6.4 Parameter variable type

Parameter values may be float or double. Some sensors provide values that cannot be stored as float, but have to be stored as double. In that case, a variable with the type “double” needs to be used instead of a “float” variable.

These parameters are precursors to calculated parameters. They will not be adjusted or quality controlled (no record in history or calibration sections).

Concerned variables

Concerned variables can occur in profile and trajectory files are: `<PARAM>` and `HISTORY_PREVIOUS_VALUE`.

Example: `MTIME`: Fractional day of the individual measurement relative to `JULD` of the station.

2.6.5 PARAMETER_DATA_MODE

In both the core- and b- profile files, the variable `DATA_MODE(N_PROF)` is not related to a specific parameter. The value of `DATA_MODE(N_PROF)` is set to ‘D’ when adjusted values for one or more `PARAM` in each `N_PROF` become available.

In b-Argo profile files, there are additional biogeochemical parameters which can receive delayed-mode adjustments at different times. Therefore the variable `PARAMETER_DATA_MODE(N_PROF, N_PARAM)` is added to b-Argo profile files to indicate the data mode of each `PARAM` in each `N_PROF`.

The adjusted section (`<PARAM>_ADJUSTED`, `<PARAM>_ADJUSTED_QC` and `<PARAM>_ADJUSTED_ERROR`) for each `PARAM` in each `N_PROF` should then be filled independently according to its `PARAMETER_DATA_MODE`.

- For example, in a b-Argo profile file with `DOXY` and `NITRATE`, it is possible that `PARAMETER_DATA_MODE = 'D'` for `DOXY`, and
- `PARAMETER_DATA_MODE = 'R'` for `NITRATE`.

In this case:

- the adjusted section for `DOXY` should be filled with their adjusted values;
- the adjusted section for `NITRATE` should be filled with `FillValues`.

| | | |
|---------------------|---|---|
| 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 |
|---------------------|---|---|

2.6.6 N_PARAM management in b-Argo profile files

For floats that measure multiple biogeochemical parameters ($N_PARAM > 1$) with multiple vertical sampling schemes ($N_PROF > 1$), users are reminded that N_PARAM in the b-Argo profile files does not necessarily equal to the total count of all unique parameters in a single cycle.

The definition of N_PARAM is the “Maximum number of parameters measured or calculated for a pressure sample.”

The same definition applies when $N_PROF > 1$.

For example, a single-cycle b-Argo profile file has 3 vertical sampling schemes:

- at $N_PROF = 1$, $STATION_PARAMETERS = [PRES]$: $N_PARAM1 = 1$;
- at $N_PROF = 2$, $STATION_PARAMETERS = [PRES, DOXY, NITRATE]$: $N_PARAM2 = 3$;
- at $N_PROF = 3$, $STATION_PARAMETERS = [PRES, CHLA, BBP700, CDOM]$: $N_PARAM3 = 4$.

In this example, there are 6 unique parameters (PRES, DOXY, NITRATE, CHLA, BBP700, CDOM), but:

- $N_PARAM = \text{maximum of } (N_PARAM1, N_PARAM2, N_PARAM3) = 4$.

2.6.7 QC and ADJUSTED variables in b-Argo profile files

In core-Argo profile files, where $PARAM = PRES, TEMP, PSAL$ (and sometimes $CNDC$), each $PARAM$ has 5 associated variables in qc and adjusted variables that are used to record real-time qc test results and delayed-mode adjustment information:

$PARAM_QC$, $PROFILE_PARAM_QC$, $PARAM_ADJUSTED$, $PARAM_ADJUSTED_QC$, and $PARAM_ADJUSTED_ERROR$.

In b-Argo profile files, $PARAM$ can be classified into 3 groups:

- (a). B-Argo $PARAM$: these are the ocean state biogeochemical variables that will receive real-time qc tests and delayed-mode adjustments. They are stored in both the b-files and the GDAC merged files.
- (b). I-Argo $PARAM$: these are the intermediate biogeochemical variables that are only stored in the b-files.
- (c). PRES: this is the stand-alone vertical index that links the core- and b-files.

The following are some clarification on what qc and adjusted variables to include in the b-files:

- (a). B-Argo $PARAM$: All 5 associated qc and adjusted variables are mandatory for B-Argo $PARAM$ in the b-files.
- (b). I-Argo $PARAM$: $PARAM_QC$ and $PROFILE_PARAM_QC$ are mandatory for I-Argo $PARAM$. $PARAM_ADJUSTED$, $PARAM_ADJUSTED_QC$ and $PARAM_ADJUSTED_ERROR$ are optional as a group; they must either all be present or all be absent.
- (c). PRES: the b-files do not contain any qc or adjusted variables for PRES. (See Section 2.6.1.)

2.6.8 PARAMETER and PARAMETER_SENSOR need expansion

In B-Argo files text strings for PARAMETER and PARAMETER_SENSOR are extended from 16 to 64 or 128 characters to allow for more explicit names.

Applicable variables

In profile files: STATION_PARAMETERS, PARAMETER, HISTORY_PARAMETER

In trajectory files: TRAJECTORY_PARAMETERS, HISTORY_PARAMETER

In metadata files: PARAMETER size is set to 64, PARAMETER_SENSOR size is set to 128

2.6.9 DATA_TYPE dimension extended from 16 to 32 characters

B-Argo and Merged files DATA_TYPE dimension are extended from 16 to 32 characters.

2.6.10 Predeployment calibration variable dimension

The PREDEPLOYMENT_CALIB_EQUATION, PREDEPLOYMENT_CALIB_COEFFICIENT, PREDEPLOYMENT_CALIB_COMMENT string lengths are by default set to STRING1024.

When necessary (such as for the Nitrate sensor), the string length is set to 4096.

2.7 GDAC FTP directory file format

2.7.1 Profile directory file format 2.0

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.

It is located at the root of the GDACs ftp servers.

- <ftp://usgodae.org/pub/outgoing/argo/>
- <ftp://ftp.ifremer.fr/ifremer/argo/>

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.

Index file naming convention

- ./ar_index_global_prof.txt
- ./ar_index_global_prof.txt.gz

Profile directory format definition : ar_index_global_prof.txt

```
# 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://usgoda.usgoda.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 contains 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://usgoda.usgoda.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
```

Note on date or position with a bad quality control flag in index files

When the POSITION_QC or JULD_QC of the profile is 3 or 4 (probably bad, bad), the position is reported as missing in the "profile directory file". Note that in this case, the position in core-argo file differs from the position in the directory file."

In the index file, a missing date or position is set to “,” (empty element).

2.7.2 Profile directory file format version 2.1

The profile directory file describes all individual profile files of the GDAC ftp site. Its format is an 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 the 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 the user to check that the file she/he collected through FTP is identical to the original file.

Index file naming convention

- etc/argo_profile_detailed_index.txt.gz
- etc/argo_profile_detailed_index.txt.gz.md5

Detailed profile directory format definition

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

- file: path and file name on the ftp site. The file name contains 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 oxygen 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 : creation 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_ps
al_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
...

```

2.7.3 Trajectory index file format 2.0

The trajectory directory file describes all trajectory files of the GDAC ftp site. Its format is an auto-descriptive 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, parameters, date_update

file : path and file name on the ftp site
Fill value : none, this field is mandatory

latitude_max, latitude_min, longitude_max, longitude_min : extreme locations of the float
Fill values : 99999.

profiler_type : type of profiling float as described in reference table 8
Fill value : " " (blank)

institution : institution of the profiling float described in reference table 4
Fill value : " " (blank)

parameters: blank separated list of parameters
Fill value : " " (blank)

date_update : date of last update of the file, YYYYMMDDHHMISS
Fill value : " " (blank)

```

Trajectory directory format example

```

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

```


2.7.4 Metadata directory file format 2.0

The metadata directory file describes all metadata files of the GDAC ftp site. Its format is an auto descriptive 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
```


3 Reference tables

The Argo reference tables are managed on Argo Vocabulary Server hosted on Nerc Vocabulary server (NVS). The Argo tables are prefixed with “R”:

- <https://vocab.nerc.ac.uk/collection>

3.1 Reference table 1: data type

<https://vocab.nerc.ac.uk/collection/R01/current>

This table contains the list of acceptable values for the DATA_TYPE field.

3.2 Reference table 2: Argo quality control flag scale

3.2.1 Reference table 2: quality control flag scale

<https://vocab.nerc.ac.uk/collection/RR2/current>

<https://vocab.nerc.ac.uk/collection/RD2/current>

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 (applicable to <PARAM>_QC in 'R' mode and <PARAM>_ADJUSTED_QC in 'A' mode) | Delayed-mode comment (applicable to <PARAM>_ADJUSTED_QC in 'D' mode) |
|---|---|---|--|
| 0 | No QC is performed | No QC is performed. | No QC is performed. |
| 1 | Good data | Good data. All Argo real-time QC tests passed. These measurements are good within the limits of the Argo real-time QC tests. | Good data. No adjustment is needed, or the adjusted value is statistically consistent with good quality reference data. An error estimate is supplied. |
| 2 | Probably good data | Probably good data. These measurements are to be used with caution. | Probably good data. Delayed-mode evaluation is based on insufficient information. An error estimate is supplied. |
| 3 | Probably bad data that are potentially adjustable | Probably bad data. These measurements are not to be used without scientific adjustment, e.g. data affected by sensor drift but may be adjusted in delayed-mode. | Probably bad data. An adjustment may (or may not) have been applied, but the value may still be bad. An error estimate is supplied. |
| 4 | Bad data | Bad data. These measurements are not to be used. A flag '4' indicates that a relevant real-time qc test has failed. A flag '4' may also be assigned for bad measurements that are known to be not adjustable, e.g. due to sensor failure. | Bad data. Not adjustable. Adjusted data are replaced by FillValue. |
| 5 | Value changed | Value changed | Value changed |
| 6 | Not used | Not used | Not used |
| 7 | Not used | Not used | Not used |
| 8 | Estimated value | Estimated value (interpolated, extrapolated or other estimation). | Estimated value (interpolated, extrapolated or other estimation). |
| 9 | Missing value | Missing value. Data parameter will record FillValue. | Missing value. Data parameter will record FillValue. |

| | | | |
|----|-----------|-----------------------------|------------------------------|
| '' | FillValue | Empty space in NetCDF file. | Empty space in NetCDF file*. |
|----|-----------|-----------------------------|------------------------------|

) this case can occur, e.g., in aggregated {wmo}_prof.nc and {wmo}_Sprof.nc files when N_LEVELS (max of level number from single cycles) is not reached by a cycle, in synthetic single cycle files for levels not concerning a specific sensor, in BGC B files for sampling schemes not concerning a sensor, etc.

3.2.2 Reference table 2a: overall profile quality flag

<https://vocab.nerc.ac.uk/collection/RP2/current>

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

- QC flag values of 1, 2, 5, 8 (only used in S-files) are GOOD data
- QC flag values of 0 (no QC), 9 (missing) or "" (FillValue) are NOT USED in the computation
- QC flag values of 3, 4 are BAD data

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

| n | Meaning |
|----|---|
| "" | No QC is performed or no usable flag values are present |
| A | $N = 100\%$; All profile levels contain good data. |
| B | $75\% \leq N < 100\%$ |
| C | $50\% \leq N < 75\%$ |
| D | $25\% \leq N < 50\%$ |
| 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
- 5 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";

3.3 Reference table 3: parameter code table

<https://vocab.nerc.ac.uk/collection/R03/current>

The following table describes the parameter codes used for Argo data management. The detailed parameter codes tables is available on Argo data-management web site:

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

Parameter attributes

- The C_Format, Fortran_Format and Format_resolution attributes are float/sensor dependents. They are set by the DAC (Data Assembly Centre).

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 for approval and inclusion.

Note on resolution

For each parameter, the resolution attribute is mandatory. However, the resolution value is sensor dependent.

3.3.1 How to indicate multiple sensors for the same parameter

Some floats are equipped with multiple sensors, measuring the same physical parameter. In that case, add the integer with an underscore "_n" at the end of the code of the parameter.

The integer "_n" starts with "_2" and increases by 1 when there are more than two sensors measuring the same physical parameter.

Example

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

- Use DOXY and TEMP_DOXY for Optode
- Use DOXY_2 for SBE

If a float has two Optode oxygen sensors:

- Use DOXY and TEMP_DOXY, and DOXY_2 and TEMP_DOXY_2

If a float has four SBE oxygen sensors:

- Use DOXY, DOXY_2, DOXY_3, DOXY_4

3.3.2 Oxygen related parameters

Some Argo floats perform Oxygen observation from different types of sensors, such as the Aanderaa 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>, Cookbook documents, "Processing Argo oxygen data at the DAC level"

3.4 Reference table 4: data centres and institutions codes

<https://vocab.nerc.ac.uk/collection/R04/current>

3.5 Reference table 5: location classes

<https://vocab.nerc.ac.uk/collection/R05/current>

3.6 Reference table 6: data state indicators

<https://vocab.nerc.ac.uk/collection/R06/current>

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 10dbar 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 sub-sampling. This is usually done to exploit oversampling by the instrument, and to ensure good measurements are achieved. These are considered to be part of the geospatial and temporal referencing process.

3.7 Reference table 7: history action codes

<https://vocab.nerc.ac.uk/collection/R07/current>

3.8 Reference table 8: instrument types

<https://vocab.nerc.ac.uk/collection/R08/current>

The instrument type codes come from WMO table 1770.

3.9 Reference table 9: positioning system

<https://vocab.nerc.ac.uk/collection/R09/current>

3.10 Reference table 10: transmission system

<https://vocab.nerc.ac.uk/collection/R10/current>

3.11 Reference table 11: QC test binary IDs

<https://vocab.nerc.ac.uk/collection/R11/current>

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.

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> .

3.12 Reference table 12: history steps codes

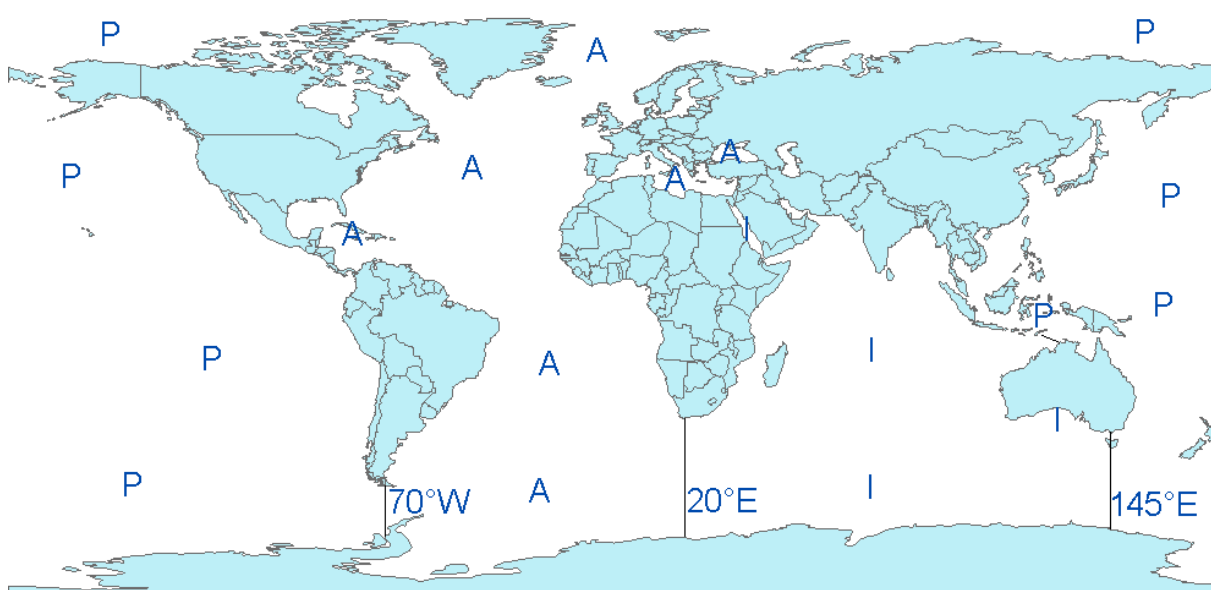
<https://vocab.nerc.ac.uk/collection/R12/current>

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

3.13 Reference table 13: ocean codes

<https://vocab.nerc.ac.uk/collection/R13/current>

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



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

3.14 Reference table 14: technical parameter names

<https://vocab.nerc.ac.uk/collection/R14/current>

All technical parameter names are standardized.

The list of technical parameter names (14a) 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 (14b) 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.

Requests for new names can be sent 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

3.15 Reference Table 15: codes of trajectory measurements performed within a cycle

<https://vocab.nerc.ac.uk/collection/R15/current>

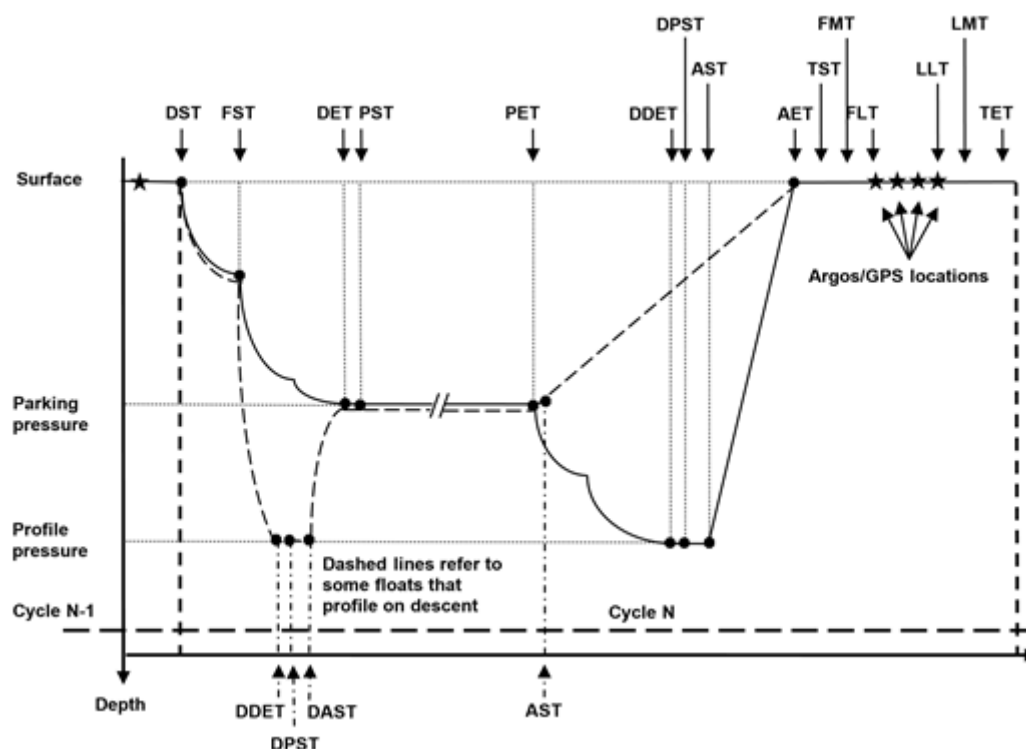


Figure 1: Figure showing float cycle and the cycle timing variables. Floats can profile either on descent or ascent. Most floats profile on ascent. Their float path is shown with a solid black line. Some floats profile on descent. One such float, the new SOLO-II Deep float, has a cycle as shown by the dashed line.

Floats that profile on ascent would have the following mandatory cycle timings:

- DST, DET, PET, DDET, AST, AET and all surface times

Floats that profile on descent might have the following cycle timings:

- DST, DDET, DAST, DET, PET, AST, AET, and all surface times

General Measurement Code Table Key

| Measurement code type | Definition |
|--|---|
| Any code evenly divisible by 100 (e.g. 100, 200, 300, etc.) | Primary Measurement Codes (MC). Each marks a mandatory-to-fill cycle timing variable. These are very important for determining trajectory estimates. All are found in both the N_MEASUREMENT and N_CYCLE data arrays. |
| Any code evenly divisible by 50 but not evenly divisible by 100 (e.g. 150, 250, 450, et) | Secondary Measurement Codes (MC). Each marks a suggested-to-fill cycle timing variable. Secondary MC are not always applicable to all floats, but are very useful in determining trajectory estimates. |
| Any code that falls in between any Primary or Secondary Measurement Code (span of 50 values). These codes describe data that are important cycle timing information but are not as important as the primary or secondary timing variables. The value span is subdivided into two halves. Measurement codes in this section will be described relative to the values of the Primary and Secondary codes. | <p>Relative Generic Codes. Values spanning from MC minus 24 to MC minus 1: Measurement codes that have lower value and within 24 of a Primary or Secondary Measurement Code. These code definitions are phrased generally, so can be attached to data from many different floats. These code values (MC minus 24 to MC minus 1) are assigned when a float records a measurement while transitioning TOWARDS the MC. The definitions of the MC from MC minus 24 to MC minus 1 are repeated for all Primary and Secondary MC. An example, most floats record pressure/temperature/salinity during drift. The float is transitioning towards PET (MC=300) during this period. Thus the pressure/temperature/salinity measurements will have an MC between MC minus 24 and MC minus 1 where MC=300 (thus between MC=276 and MC=299). Which value is chosen is determined by the measurement itself (See table below).</p> <p>Relative Specific Codes. Values spanning from MC plus 1 to MC plus 25: These are specific measurements that are generally NOT recorded by multiple float types. They are believed to be valuable enough in trajectory estimation that they are defined here, and not within the generically defined MC minus 24 to MC minus 1 span. MC codes in this span will be specific to the MC code, and will NOT be repeated for other Primary and Secondary MCs. An example, APEX floats report the "Down-time end date", which is important in determining the start of ascent (MC=500). The MC for "Down-time end date" is recorded with MC plus 1 (MC=501).</p> |

Relative Generic Code Table Key (from MC minus 24 to MC minus 1)

This table pertains to any measurement code that has lower value and within 24 of a Primary or Secondary Measurement Code (see below). These definitions apply relative to every Primary and Secondary code. For example, AST (time of ascent start, MC=500) and AET (time of ascent end, MC=600) are both Primary MCs. There exists a measurement code MC minus 4 for both AST and AET which is assigned to any averaged measurement that is taken while transitioning towards the MC. If an averaged measurement is recorded while transitioning towards AST, the correct MC=496. If an averaged measurement is recorded while transitioning towards AET, the correct MC=596.

3.16 Reference table 16: vertical sampling schemes

<https://vocab.nerc.ac.uk/collection/R16/current>

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]"

3.17 Reference table 17: obsolete

This table has been removed.

3.18 Reference table 18: metadata configuration parameter names

<https://vocab.nerc.ac.uk/collection/R18/current>

All metadata variable names and configuration parameter names are standardized.

The list of metadata variable names (18a) is available at:

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

The list of configuration parameter names (18b) 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.

Requests for new names can be sent for approval and inclusion.

3.19 Reference table 19: STATUS flags

<https://vocab.nerc.ac.uk/collection/R19/current>

3.20 Reference table 20: GROUNDED flags

<https://vocab.nerc.ac.uk/collection/R20/current>

3.21 Reference table 21: REPRESENTATIVE_PARK_PRESSURE_STATUS

<https://vocab.nerc.ac.uk/collection/R21/current>

3.22 Reference table 22: PLATFORM_FAMILY

<https://vocab.nerc.ac.uk/collection/R22/current>

Please note that this reference table is frequently updated to include new sensor and float models.

3.23 Reference table 23: PLATFORM_TYPE

<https://vocab.nerc.ac.uk/collection/R23/current>

Please note that this reference table is frequently updated to include new sensor and float models.

3.24 Reference table 24: PLATFORM_MAKER

<https://vocab.nerc.ac.uk/collection/R24/current>

Please note that this reference table is frequently updated to include new sensor and float models.

3.25 Reference table 25: SENSOR

<https://vocab.nerc.ac.uk/collection/R25/current>

Please note that this reference table is frequently updated to include new sensor and float models. You can

3.26 Reference table 26: SENSOR_MAKER

<https://vocab.nerc.ac.uk/collection/R26/current>

Please note that this reference table is frequently updated to include new sensor and float models.

3.27 Reference table 27: SENSOR_MODEL

<https://vocab.nerc.ac.uk/collection/R27/current>

The SENSOR_MODEL variable is standardised, i.e. we expect the manufacturer followed by the standard model number, i.e. SBE41CP or AANDERAA_3830. If there is a version number for a particular model then this is mentioned in the SENSOR_FIRMWARE_VERSION field.

* Note that some biogeochemical sensors have different configurations, i.e. they are either in the pumped stream or not in the pumped stream. Sensor readings from those in the pumped vs unpumped stream can be very different. Some manufacturers do not distinguish this in the sensor model name. In order to capture this information there is a configuration parameter that specifies this, i.e. CONFIG_SensorInPumpedStream_LOGICAL, (Yes =1, No = 0). For the relevant sensors this configuration parameter should be filled in the metadata file for the launch configuration settings.

3.28 Reference table 28: CONTROLLER_BOARD_TYPE_PRIMARY

<https://vocab.nerc.ac.uk/collection/R28/current>

The first part of the CONTROLLER_BOARD_TYPE_PRIMARY variable should contain one of the strings in the below table. The remainder of the string is free text and may contain more information on the controller board type to suit the float model. The free text should be delimited by square brackets '['']. For example: "APF9 [iridium version xyz]". Please note that this reference table is sometimes updated to include new sensor and float models.

3.29 Reference table 29: BATTERY_TYPE

This table has not been transferred to the NVS yet.

The BATTERY_TYPE should be coded with the following frame:

<manufacturer> <battery_type> <voltage> V [+<manufacturer> <battery_type> <voltage> V]

Where:

<manufacturer> should be one of the following controlled list of battery manufacturers:

| List of battery manufacturers |
|-------------------------------|
| ELECTROCHEM |
| TADIRAN |
| SAFT |

< battery_type > should be one of the following controlled list of battery types:

| List of battery types |
|-----------------------|
| Alkaline |
| Lithium |
| Hybrid |

<voltage> stands for the initial battery voltage.

Examples:

| |
|--|
| ELECTROCHEM Alkaline 12 V |
| TADIRAN Lithium 24 V |
| TADIRAN Alkaline 12 V + TADIRAN Lithium 12 V |
| ELECTROCHEM Lithium 24 V + ELECTROCHEM Hybrid 24 V |

3.30 Reference table 30: BATTERY_PACKS (optional field)

This table has not been transferred to the NVS yet.

Although the BATTERY_PACKS variable is not mandatory, it is controlled. The format of the field should be as shown in the table below, where x indicates the number of packs (not the number of batteries in a pack). ‘Li’, ‘Alk’ or ‘Hyb’ abbreviations should be used.

Examples:

| |
|--|
| xDD y (x = number of packs, y = Li or Alk or Hyb) |
| xC y (x = number of packs, y = Li or Alk or Hyb) |
| xD y (x = number of packs, y = Li or Alk or Hyb) |
| Any combination of above with + to join (eg 4DD Li + 1C Alk) |
| U (Unknown) |

3.31 Reference table 31: ENDING_CAUSE and ENDING_CAUSE_CATEGORY

This table has not been transferred to the NVS yet.

This vocabulary captures all the possible reasons why a float may have died.

The “ending cause category” and “ending cause” for a float are stored in the “ANOMALY” variable of its meta-data file. They are reported with a “key0:value0,key1:value1” syntax where:

- key0 = “ending_cause_category”
- key1 = “ending_cause”

The “ANOMALY” variable describes any anomalies or problems the float may have had. Anomalies or problems are comma separated statements.

Example

- ANOMALY = "from cycle 107 the immersion drift is not stable,ending_cause_category:hardware,ending_cause:ballast"

Before registration in the Argo vocabulary server, the “ending cause category” and “ending cause” terms are managed in the [ArgoVocab-FloatsEndingCauses](#) document.

3.32 Reference table 32: SPECIAL_FEATURE

This table has not been transferred to the NVS yet.

This vocabulary captures the possible SPECIAL_FEATURES in a float’s characteristics.

Additional float features can be specified here such as algorithms used by the float.

The SPECIAL_FEATURES variable in a float metadata file must be filled by the DAC, to indicate there are AUX data files (see <https://argo.ucsd.edu/data/auxiliary-directory>).

3.33 Reference table 40 : PI_NAME

<https://vocab.nerc.ac.uk/collection/R40>

This table captures the possible PI_NAME. Preferred label is composed in most cases by the first name in lower cases and the family name in upper cases. When there are several Principal Investigators, their names should be separated by commas.

4 Data access

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

The FTP addresses are:

- <ftp://ftp.ifremer.fr/ifremer/argo>

The whole Argo dataset is available in a one click download from Argo DOI (monthly updated)

- <https://doi.org/10.17882/42182>

The GDACs internet addresses are:

- <http://www.usgodac.org/argo/argo.html>
- <http://www.argodatamgt.org>

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>

4.1 File naming convention on GDACs

The GDAC ftp sites comply with the following naming conventions.

4.1.1 Core-Argo individual profile files

The individual profile files are provided by the DACs (Data Assembly Centres).

The core-Argo profile files contain the core parameters provided by a float: pressure, temperature, salinity, conductivity (PRES, TEMP, PSAL, CNDC). All additional parameters are managed in B-Argo data files (see §0).

For floats that collect no more than 1 ascending and 1 descending profile per cycle the file names for individual profiles are <R/D><FloatWmoID>_<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).

Note on <XXX> cycle numbers greater than 999

When float reports a cycle greater than 999, the cycle number appears as 4 digits in the file name.

Examples: R5902319_999.nc, R5902319_1000.nc

Note on <R/D> in profile file name

- For core-files, the file name mode follows DATA_MODE in N_PROF = 1, which is the primary Argo CTD profile. Hence a 'R'-core file becomes a 'D'-core file only when DATA_MODE(1) = 'D'. This file naming rule ensures that the integrity of the core Argo program is not affected by various miscellaneous secondary samplings.
- For b-files and GDAC merged-files, when any DATA_MODE(N_PROF) changes to 'D', the file name mode changes to 'D'. This is because b-files and GDAC merged-files contain biogeochemical data, which can receive delayed-mode adjustments at different times.

4.1.2 B-Argo data file

4.1.2.1 B-Argo individual profile file

A B-Argo profile file contains all the parameters from a float, except the core-Argo parameters temperature, salinity, conductivity (TEMP, PSAL, CNDC). A float that performs only CTD measurements does not have B-Argo data files.

File naming convention

B<R/D><FloatWmoID>_<XXX><D>.nc

- B : B-Argo file prefix
- <R/D><FloatWmoID>_<XXX><D>.nc : identical to Core-Argo file naming convention

Examples: BR1900045_083.nc, BR1900045_083D.nc, BD1900045_003.nc

When float reports a cycle greater than 999, the cycle number appears as 4 digits in the file name.

4.1.2.2 BGC-Argo individual synthetic profile file

To facilitate the use of BGC-Argo data, the GDAC merges each BGC-Argo file with its corresponding Core-Argo data file. The BGC-Argo synthetic profiles are documented in <http://dx.doi.org/10.13155/55637>

The synthetic file contains the core-Argo and BGC-Argo parameters listed on reference table 3 (<https://vocab.nerc.ac.uk/collection/R03/>). The intermediate parameters are ignored by the synthetic files.

File naming convention

S<R/D><FloatWmoID>_<XXX><D>.nc

- S: synthetic BGC-Argo file prefix
- <R/D><FloatWmoID>_<XXX><D>.nc : identical to Core-Argo file naming convention

Examples: SR1900045_083.nc, SR1900045_083D.nc, SD1900045_003.nc

4.1.3 Argo trajectory data file

The Argo trajectory files contain the core and BGC parameters provided by a float.

<FloatWmoID>_<R/D>traj.nc

- R: real-time data
- D: delayed-mode data

Examples

1900045_Rtraj.nc: real-time trajectory from float 1900045

1900045_Dtraj.nc: delayed-mode trajectory from float 1900045

4.1.5 Metadata file

- <FloatWmoID>_meta.nc
Example : 1900045_meta.nc

4.1.6 Technical Data file

- <FloatWmoID>_tech.nc
Example : 1900045_tech.nc

4.1.7 Auxiliary data files

Auxiliary files are used for data and metadata that are not (yet) standardized among the Argo community.

They typically contain data from new types of sensors or floats.

Auxiliary data file format is NOT regulated, they are distributed on GDAC ftp server in the “aux” directory that is parallel to the “dac” directory.

The float metadata file in the ‘dac’ directory includes a text in the SPECIAL_FEATURES variable to indicate that there is additional data in the auxiliary directory and the nature of that data.

PIs are encouraged to use a file format that is easily readable – either NetCDF or ASCII that mimics the Argo data format version 3.1 if possible. A good practice (not an obligation) within DACs is to document their auxiliary files. Coriolis DAC auxiliary file formats are documented here:

- Argo auxiliary files format for Coriolis DAC <http://doi.org/10.13155/51995>

4.2 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).

Argo profiles data are distributed on GTS as BUFR messages with template n° 3-15-003.

The description of these format is available from the WMO web site (<https://public.wmo.int>)

5 Using the History section of the Argo netCDF Structure

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

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

5.1 Recording information about the Delayed Mode QC process

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

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

| Field | Example | Explanation |
|--------------------------|----------------|---|
| HISTORY_INSTITUTION | CI | Selected from the list in reference table 4 (https://vocab.nerc.ac.uk/collection/R04/) |
| HISTORY_STEP | ARSQ | Selected from the list in reference table 12 (https://vocab.nerc.ac.uk/collection/R12/). |
| 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 (https://vocab.nerc.ac.uk/collection/R07/). |
| HISTORY_PARAMETER | FillValue | This field does not apply (1) |
| HISTORY_START_PRES | FillValue | This field does not apply |
| HISTORY_STOP_PRES | FillValue | This field does not apply |
| HISTORY_PREVIOUS_VALUE | FillValue | This field does not apply |
| HISTORY_QCTEST | FillValue | This field does not apply |

Note

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

5.2 Recording processing stages

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

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

Example: History entry to record decoding of the data.

| Field | Example | Explanation |
|--------------------------|----------------|---|
| HISTORY_INSTITUTION | ME | Selected from the list in reference table 4 (https://vocab.nerc.ac.uk/collection/R04/). |
| HISTORY_STEP | ARFM | Selected from the list in reference table 12 (https://vocab.nerc.ac.uk/collection/R12/). |
| 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 (https://vocab.nerc.ac.uk/collection/R07/). |
| HISTORY_PARAMETER | FillValue | This field does not apply |
| HISTORY_START_PRES | FillValue | This field does not apply |
| HISTORY_STOP_PRES | FillValue | This field does not apply |
| HISTORY_PREVIOUS_VALUE | FillValue | This field does not apply |
| HISTORY_QCTEST | FillValue | This field does not apply |

5.3 Recording QC Tests Performed and Failed

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

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

Example: QC tests performed and failed.

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

Record 1: Documenting the tests performed

| Field | Example | Explanation |
|--------------------------|----------------|---|
| HISTORY_INSTITUTION | ME | Selected from the list in reference table 4 (https://vocab.nerc.ac.uk/collection/R04/). |
| HISTORY_STEP | ARGQ | Selected from the list in reference table 12 (https://vocab.nerc.ac.uk/collection/R12/). |
| 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 (https://vocab.nerc.ac.uk/collection/R07/). |
| 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 (https://vocab.nerc.ac.uk/collection/R11/). |
|----------------|-----|---|

Record 2: Documenting the tests that failed

| Field | Example | Explanation |
|--------------------------|----------------|--|
| HISTORY_INSTITUTION | ME | Selected from the list in reference table 4 (https://vocab.nerc.ac.uk/collection/R04/). |
| HISTORY_STEP | ARGQ | Selected from the list in reference table 12 (https://vocab.nerc.ac.uk/collection/R12/). |
| 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 (https://vocab.nerc.ac.uk/collection/R07/). |
| 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 (https://vocab.nerc.ac.uk/collection/R11/). |

5.4 Recording changes in values

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

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

Example: Changed latitude

| Field | Example | Explanation |
|--------------------------|----------------|---|
| HISTORY_INSTITUTION | CI | Selected from the list in reference table 4 (https://vocab.nerc.ac.uk/collection/R04/). |
| HISTORY_STEP | ARGQ | Selected from the list in reference table 12 (https://vocab.nerc.ac.uk/collection/R12/). |
| 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 (https://vocab.nerc.ac.uk/collection/R07/). |
| HISTORY_PARAMETER | LAT\$ | A new entry for reference table 3 (https://vocab.nerc.ac.uk/collection/R03/) 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 dbar were considered correct, after the analysis, they are considered wrong. The history entry to record this would look as follows.

Example: Changed flags

| Field | Example | Explanation |
|--------------------------|----------------|---|
| HISTORY_INSTITUTION | CI | Selected from the list in reference table 4 (https://vocab.nerc.ac.uk/collection/R04/). |
| HISTORY_STEP | ARGQ | Selected from the list in reference table 12 (https://vocab.nerc.ac.uk/collection/R12/). |
| 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 (https://vocab.nerc.ac.uk/collection/R07/). |
| HISTORY_PARAMETER | TEMP | Selected from the list in reference table 3 (https://vocab.nerc.ac.uk/collection/R03/). |
| 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.

- xxx_bgc.tar.gz xxx dac bgc-Argo files
- argo_bio-profile_index.txt.gz bgc Argo profiles index file
- argo_bio-traj_index.txt.gz bgc Argo trajectories file
- argo_merge-profile_index.txt.gz merged Argo profiles index
- argo_profile_detailed_index.txt.gz Argo profiles detailed index
- argo_synthetic-profile_index.txt.gz Synthetic Argo profile files
- ar_index_global_meta.txt.gz Index of metadata files
- ar_index_global_prof.txt.gz Index of profile files
- ar_index_global_tech.txt.gz Index of technical files
- ar_index_global_traj.txt.gz Index of trajectories
- geo directory: content of the GDAC “geo” branch
One compressed file per day and per ocean

6.4 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

7 Glossary, definitions

This chapter gives a definition for the items described in this manual.

7.1 Float

An autonomous platform deployed in the sea that performs environmental monitoring.

7.2 Sensor

A sensor is a device used to measure a physical parameter. Sensor outputs are provided in parameter counts and need to be converted into parameter physical units using a calibration equation. This conversion can be done onboard the float or during the decoding process.

7.3 Parameter measured by the sensor

A parameter is a measurement of a physical phenomenon; it can be provided by a sensor (in sensor counts or in physical units) or computed (derived) from other parameters.

7.4 Calibration of the parameter measured by the sensor

Verification of any operation measurement against independent measurements to derive a corrected value or a new parameter.

7.5 Principal Investigator (PI)

The **Principal Investigator (PI)**, typically a scientist at a research institution, maintains the observing platform and the sensors that deliver the data. He or she is responsible for providing the data and all auxiliary information to a **Data Assembly Center (DAC)**.

7.6 Global Data Assembly Centre (GDAC)

The **GDAC** distributes the best copy of the data files. When a higher quality data file (e.g. calibrated data) is available, it replaces the previous version of the data file.

7.7 Data Assembly Centre (DAC)

The **DAC** assembles Argo files and delivers these to the two **Global Data Assembly Centers (GDACs)**.

7.8 GTS

WMO's Global Telecommunication System (GTS) is the communications and data management component that allows the World Weather Watch (WWW) to operate through the collection and distribution of information critical to its processes.

WMO: World Meteorological Organization