Framework Service Contract EEA/DIS/R0/20/001 Lot 1 for Services supporting the European Environment Agency's (EEA) cross-cutting coordination of the Copernicus programme's in situ data activities – Observational data

**C-RAID/MCO - CORIOLIS 2022**
**C-RAID Drifter NetCDF Format Manual**
**Jean-Philippe RANNOU and Patricia ZUNINO**

NetCDF conventions and
Reference Tables

Version 1.4
October 12th, 2023
Doi: 10.17882/77184

**Issue: 5**
**Date: 20/10/2023**

# [DOCUMENT RELEASE]

| | **Name(s)** | **Affiliation** |
|---|---|---|
| Coordinated by: | Thierry Carval | IFREMER |
| Contributions: | Jean-Philippe RANNOU<br>Patricia ZUNINO RODRIGUEZ | Capgemini Engineering<br>Capgemini Engineering |
| Approval: | Henrik Steen Andersen | European Environment Agency |

| | |
|---|---|
| Prepared for: | European Environment Agency (EEA) |
| Represented by: (Project Manager) | Henrik Steen Andersen |
| Contract No. | EEA/DIS/R0/20/001 Lot 1 |

# [Change Record]

| **Doc Version** | **Format Version** | **Date** | **Changes** |
|---|---|---|---|
| 0.9 | 1.0 | 15/05/2020 | ALTRAN/JPR: initialization of the document |
| 0.99 | 1.0 | 15/07/2020 | ALTRAN/JPR: Greylist information added.<br>ALTRAN/JPR: Dimension N_PARAM added to PARAMETERE_DATA_MODE variable.<br>ALTRAN/JPR: Initialization of Reference Table 3: parameter list. |
| 0.999 | 1.0 | 21/08/2020 | ALTRAN/JPR: few additional comments |
| 0.999 | 1.0 | 28/09/2020 | ALTRAN/JPR: review of the document |
| 1.0 | 1.0 | 22/10/2020 | ALTRAN/PZR: review of the document |
| 1.1 | 1.1 | 15/02/2022 | Capgemini Engineering/JPR: added OPERATING_COUNTRY<br>Capgemini Engineering /JPR: updated Ref Table 21: DRIFTER_MAKER |
| 1.2 | 1.1 | 30/05/2022 | Capgemini Engineering PZR: added parameter AIR_TEMP in section 3.3.1 |
| 1.3 | 1.1 | 20/07/2023 | Capgemini Engineering/JPR: updated Ref Table 17: DATA_STATUS, Ref Table 20: DRIFTER_TYPE and Ref Table 21: DRIFTER_MAKER |
| 1.4 | 1.2 | 12/10/2023 | Capgemini Engineering/JPR: removed unlimited dimension (N_TIME) so that NetCDF-4 compression will be efficient. |

# Table of Contents

**Preface**

This document is still in progress. As such, there are highlighted sections of text throughout that need to be addressed.

Yellow highlighting means this is a topic open to discussion - some things are known about this topic, but agreement needs to be reached.

Red highlighting means the issue needs a solution and nothing has been suggested yet.

Green highlighting is used for sections that have been modified since the previous version of the document.

The following temporary acronyms are also used:

- TBD: To Be Done or To Be Defined depending of the context.

# 1 Introduction

This document specifies the NetCDF file format used to store the drifter data and meta-data processed in the framework of the C-RAID project.

## 1.1 Main principles

### 1.1.1 NetCDF file name and contents

The **mission** of a drifter is the time period which starts at drifter deployment date and ends when it is recovered (or considered as lost). Each drifter mission has a unique C-RAID drifter identifier (stored in DRIFTER_NUMBER variable).

If a drifter is recovered, reconditioned and deployed again. It is a new mission (with a new C-RAID drifter identifier).

One NetCDF file is generated for each drifter mission. It contains the meta-data of the drifter and all data sampled during the mission. **When available, additional data (received from the drifter transmission identifier but sampled outside the mission time period) are also stored in the NetCDF file (and affected with a QC flag 4).**

The file naming convention of C-RAID NetCDF file is *TYYNNNN*.nc.

Where *TYYNNNN* is the C-RAID drifter identifier for which:

- *T*: refers to the drifter transmission type (1: for Argos, 2: for Iridium);
- *YY*: refers to the drifter deployment date (00: if deployed before 1979, deployed in year YY+1978 otherwise);
- *NNNN*: is a number designed so that the C-RAID drifter identifier is unique.

### 1.1.2 Drifter meta-data

Two types of meta-data are reported in the NetCDF file:

- **Static meta-data** are set before the mission start time and cannot be modified during the mission, they concern:
  - Persons and institutions involved in drifter mission;
  - Drifter characteristics;
  - Drifter deployment and mission information;
  - Drifter initial configuration;
  - Sensor mounted on the drifter, sampled parameters and associated pre-deployment calibration information.
- **Dynamic meta-data** can change during the mission:

- Drifter configuration;
- Presence of drogue;
- Sensor status;
- Real-time adjustment and scientific calibration information.

### 1.1.3 Parameter measurements

**Parameter measurements** are sampled by the sensors mounted on the drifter. A time is affected to each measurement.

Some sensors provide parameter measurements synchronously sampled at different depth levels.

**Surface locations** are provided by the drifter positioning system (GPS) and/or by the drifter transmission system (Argos, Iridium). A time is affected to each surface location.

Depending on the drifter configuration, multiple surface locations types can be available during the mission.

Parameter measurements, surface locations and associated times are stored in a group of N_TIME indexed NetCDF variables.

When multiple measurements are available from a given timestamp, the N_LEVEL<n> dimension is used.

#### 1.1.3.1 Parameter list

The list of managed parameters is provided in Reference Table 3. It gathers ocean state parameters (even provided as intermediate parameters) and drifter technical parameters.

Some of these parameters have been decided to be 'adjustable'. This means that a new set of measurements, derived from the original ones, can be generated for them.

Non 'adjustable' parameters are:

- Intermediate parameters (the original "raw" sensor output used to compute ocean state parameters);
- Technical parameters.

#### 1.1.3.2 Parameter data modes

In this document <PARAM> stands for any parameter name listed in Reference Table 3.

<PARAM> contains the values telemetered from the drifter or computed with pre-deployment calibration information (from intermediate parameters).
The values in <PARAM> **are never altered**.

<PARAM>_QC contains QC flags that pertain to the values in <PARAM>. Values in <PARAM>_QC are set initially in Real-time ('R') and real-time with Adjusted values ('A') modes by the automatic real-time tests.
They are later modified in Delayed mode ('D') at indices where the QC flags are set incorrectly by the real-time procedures, and where erroneous data are not detected by the real-time procedures.
The mode (‘R’, ‘A’ or ‘D’) of each parameter for each measurement is stored in the PARAMETER_DATA_MODE variable.

#### 1.1.3.3 Parameter data adjustment

Some parameters (the ‘adjustable’ ones) 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 parameter can be adjusted, the adjusted variables (<PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC and <PARAM>_ADJUSTED_ERROR) are present in the NetCDF file.
When a parameter is in ‘R’ mode, no adjusted data are available. Hence the adjusted variables are filled with FillValue.
When a parameter is in ‘A’ mode, <PARAM>_ADJUSTED and <PARAM>_ADJUSTED_QC are set; <PARAM>_ADJUSTED_ERROR is set or not, depending on the real-time adjustment specifications.
When a parameter is in ‘D’ mode, all adjusted variables are set.

#### 1.1.3.4 Access to parameter data

To get the list of parameters stored in the NetCDF file, the user should read the PARAMETER variable content.

Then, to access the ‘best’ existing version of each parameter (<PARAM>) data, the user should:

1. Retrieve the data mode of the <PARAM> parameter from the PARAMETER_DATA_MODE variable;
2. Access the data:
    - If the data mode is ‘R’: in <PARAM> and <PARAM>_QC variables;
    - If the data mode is ‘A’ or ‘D’: in <PARAM>_ADJUSTED, <PARAM>_ADJUSTED_QC and <PARAM>_ADJUSTED_ERROR variables.

# 2 C-RAID drifters NetCDF data format version 1.2

The mandatory variables or attributes are in **bold characters in the following tables**

## 2.1 Global attributes

The global attribute section is used for data discovery.

| Name | Example | Comment |
|---|---|---|
| **title** | title = "Drifter meta-data and data file" | A succinct description of what is in the dataset. |
| **institution** | institution = "Coriolis" | Specifies where the original data was produced. |
| **source** | institution = "C-RAID drifter" | 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** | history = "2011-04-22T06:00:00Z creation" | 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. |
| **data_mode** | data_mode="R" | Indicates if the file contains real-time, delayed mode or mixed data.<br>The list of valid data modes is in Reference Table 14.<br>Examples:<br>- data_mode = "R" means that all parameters of the file have a PARAMETER_DATA_MODE = 'R'<br>- data_mode = "A" means that all parameters of the file have a PARAMETER_DATA_MODE = 'R', or 'A' (with at least one of them with a PARAMETER_DATA_MODE = 'A')<br>- etc… |
| **references** | reference = "http://www.xxx" | Published or web-based references that describe the data or methods used to produce it. |
| comment | comment = "free text" | Miscellaneous information about the data or methods used to produce it. |
| **user_manual_version** | user_manual_version = "1.4" | The version number of the user manual |
| **Conventions** | Convention = "CF-1.6 C-RAID-1.2" | The conventions supported by this file, blank separated |
| netcdf_version | netcdf_version = "3.6" | Netcdf version used for the data file. |
| doi | doi = "http://doi.org/XX" | List of Data Object Identifiers (DOI) related to this data file (blank separated). |
| citation | Citation = "XX" | The citation to be used in publications using the data file. |
| qc_manual | qc_manual = "http://doi.org/XX" | Contains the name of the manual that describes the quality control procedure. |
| time_coverage_start | time_coverage_start = "2010-07-01T00:00:00Z" | Start date of the data in UTC. See note on time format below. |
| time_coverage_end | time_coverage_end = "2010-09-18T23:59:29Z" | Final date of the data in UTC. See note on time format below. |

| | | |
|---|---|---|
| geospatial_lat_min | geospatial_lat_min="34.7" | Southernmost valid latitude, a value between -90 and 90 decimal degrees.<br>This is calculated from the valid latitudes in the file. |
| geospatial_lat_max | geospatial_lat_max="59.8" | Northernmost valid latitude, a value between -90 and 90 decimal degrees.<br>This is calculated from the valid latitudes in the file. |
| geospatial_lon_min | geospatial_lon_min="-41.2" | Westernmost valid longitude, a value between -180 and 180 decimal degrees.<br>This is calculated from the valid longitudes in the file. |
| geospatial_lon_max | geospatial_lon_max="-33.5" | Easternmost valid longitude, a value between -180 and 180 decimal degrees.<br>This is calculated from the valid longitudes in the file. |
| geospatial_lat_start | geospatial_lat_start="34.7" | First valid latitude, a value between -90 and 90 decimal degrees.<br>This is calculated from the valid latitudes in the file. |
| geospatial_lat_end | geospatial_lat_end="59.8" | Last valid latitude, a value between -90 and 90 decimal degrees.<br>This is calculated from the valid latitudes in the file. |
| geospatial_lon_start | geospatial_lon_start="-41.2" | First valid longitude, a value between -180 and 180 decimal degrees.<br>This is calculated from the valid longitudes in the file. |
| geospatial_lon_end | geospatial_lon_end="-33.5" | Last valid longitude, a value between -180 and 180 decimal degrees.<br>This is calculated from the valid longitudes in the file. |
| drogue_off_time | drogue_off_time = "2010-07-01T00:00:00Z" | Drogue off date in UTC. See note on time format below. |
| drogue_off_latitude | drogue_off_latitude ="59.8" | Drogue off latitude, a value between -90 and 90 decimal degrees. |
| drogue_off_longitude | drogue_off_longitude ="-41.2" | Drogue off longitude, a value between -180 and 180 decimal degrees. |

## Note on time formats

Whenever time information is given in the global attributes, it ought to be a string of the format:

"YYYY-MM-DDThh:mm:ssZ" (i.e. year - month - day T hour : minute : second Z)

If higher resolution than seconds is needed, any number of decimal digits (".s") for the seconds is acceptable:

"YYYY-MM-DDThh:mm:ss.sZ"

These formats are two (of many) described by ISO8601.

In any case:

- The time must be in UTC;
- A capital "T" separates the date and the hour information;
- The string must end with a capital "Z", an old indication of UTC.

Examples:

- 2005-10-24T08:00:00Z
- 2008-01-01T22:50:02.031Z

## 2.2 Dimensions and definitions

Drifter data are recorded as time-series; the N_TIME dimension is the main dimension the file.

| Name | Example | Comment |
|---|---|---|
| N_TIME | N_TIME = <int value> | Number of time steps. |
| N_LEVEL<n> | N_LEVEL10 = 10 | Maximum number of parameter measurements that can be simultaneously sampled by a sensor. |
| STRING<N> | STRING256 = 256 | String dimensions. |
| DATE_TIME | DATE_TIME = 14 | This dimension is the length of an ASCII date and time value.<br>DATE_TIME convention is: YYYYMMDDHHMISS<br>YYYY: year<br>MM: month<br>DD: day<br>HH: hour of the day (as 0 to 23)<br>MI: minutes (as 0 to 59)<br>SS: seconds (as 0 to 59)<br>Date and time values are always in universal time coordinates (UTC).<br>Examples:<br>20010105172834 for January 5th 2001 17:28:34<br>19971217000000 for December 17th 1997 00:00:00 |
| DATE | DATE = 8 | This dimension is the length of an ASCII date value.<br>DATE convention is: YYYYMMDD<br>YYYY: year<br>MM: month<br>DD: day<br>Examples:<br>20010105 for January 5th 2001<br>19971217 for December 17th 1997 |
| N_TRANS_SYSTEM | N_TRANS_SYSTEM = <int value> | Number of transmission systems. |
| N_POSITIONING_SYSTEM | N_POSITIONING_SYSTEM = <int value> | Number of positioning systems. |
| N_LAUNCH_CONFIG_PARAM | N_LAUNCH_CONFIG_PARAM = <int value> | Number of pre-deployment or launch configuration parameters. |
| N_CONFIG_PARAM | N_CONFIG_PARAM = <int value> | Number of configuration parameters. |
| N_MISSION | N_MISSION = <int value> | Number of missions. |
| N_SENSOR | N_SENSOR = <int value> | Number of sensors mounted on the drifter and used to measure the parameters. |
| N_PARAM | N_PARAM = <int value> | Number of parameters measured by the drifter or calculated from transmitted parameters. |

| | | |
|---|---|---|
| N_HISTORY | N_HISTORY = <int value>; | Number of history records. |
| N_GREYLIST | N_GREYLIST = <int value>; | Number of greylist records. |

## 2.3 General information on the file

| Name | Definition | Comment |
|---|---|---|
| **DATA_TYPE** | char DATA_TYPE(STRING32);<br>DATA_TYPE:long_name = "Data type";<br>DATA_TYPE:conventions = "Reference table 1";<br>DATA_TYPE:_FillValue = " "; | This field contains the type of data contained in the file.<br>The list of acceptable data types is in the Reference Table 1.<br>Example : "C-RAID drifter time-series data" |
| **FORMAT_VERSION** | char FORMAT_VERSION(STRING4);<br>FORMAT_VERSION:long_name = "File format version";<br>FORMAT_VERSION:_FillValue = " "; | File format version.<br>Example : "1.1" |
| **REFERENCE_DATE_TIME** | char REFERENCE_DATE_TIME(DATE_TIME);<br>REFERENCE_DATE_TIME:long_name = "Date of reference for Julian days";<br>REFERENCE_DATE_TIME:conventions = "YYYYMMDDHHMISS";<br>REFERENCE_DATE_TIME:_FillValue = " "; | Date of reference for Julian days.<br>The recommended reference date time is "19500101000000" : January 1st 1950 00:00:00 |
| **DATE_CREATION** | char DATE_CREATION(DATE_TIME);<br>DATE_CREATION:long_name = "Date of file creation";<br>DATE_CREATION:conventions = "YYYYMMDDHHMISS";<br>DATE_CREATION:_FillValue = " "; | Date and time (UTC) of creation of this file.<br>Format : YYYYMMDDHHMISS<br>Example : 20011229161700 : December 29th 2001 16:17:00 |
| **DATE_UPDATE** | char DATE_UPDATE(DATE_TIME);<br>DATE_UPDATE:long_name = "Date of update of this file";<br>DATE_UPDATE:conventions = "YYYYMMDDHHMISS";<br>DATE_UPDATE:_FillValue = " "; | Date and time (UTC) of update of this file.<br>Format : YYYYMMDDHHMISS<br>Example : 20011230090500 : December 30th 2001 09:05:00 |

2.4.1

## 2.4 Drifter and mission meta-data

### Persons and institutions involved in drifter mission

| Name | Definition | Comment |
|---|---|---|
| PI_NAME | char PI_NAME(STRING64);<br>PI_NAME:long_name = "Name of the principal investigator";<br>PI_NAME:_FillValue = " "; | Name of the principal investigator in charge of the drifter.<br>Example: TBD. |

| | | |
|---|---|---|
| PI_EMAIL | char PI_EMAIL(STRING64);<br>PI_EMAIL:long_name = "Email address of the principal investigator";<br>PI_EMAIL:_FillValue = " "; | Email address of the principal investigator in charge of the drifter.<br>Example: TBD. |
| INSTITUTION | char INSTITUTION(STRING64);<br>INSTITUTION:long_name = "Preferably institution of the principal investigator";<br>INSTITUTION:_FillValue = " "; | Preferably institution of the principal investigator in charge of the drifter.<br>Example: TBD. |
| INSTITUTION_REFERENCES | char INSTITUTION_REFERENCES(STRING64);<br>INSTITUTION_REFERENCES:long_name = "References to preferably institution of the principal investigator";<br>INSTITUTION:_FillValue = " "; | References to preferably institution of the principal investigator in charge of the drifter, the place to find all information on the data file (web-base, i.e. give URLs)<br>Example: "http://www.nocs.uk". |
| PROJECT_NAME | char PROJECT_NAME(STRING64);<br>PROJECT_NAME:long_name = "Program under which the drifter was deployed";<br>PROJECT_NAME:_FillValue = " "; | Name of the project which operates the drifter.<br>Example: TBD. |
| DRIFTER_OWNER | char DRIFTER_OWNER (STRING64);<br>DRIFTER_OWNER:long_name = "Drifter owner";<br>DRIFTER_OWNER:_FillValue = " "; | The owner of the drifter (may be different from the data centre and operating institution).<br>Example: TBD. |
| OPERATING_INSTITUTION | char OPERATING_INSTITUTION(STRING64);<br>OPERATING_INSTITUTION:long_name = "Operating institution of the drifter";<br>OPERATING_INSTITUTION:_FillValue = " "; | The operating institution of the drifter (may be different from the drifter owner and data centre).<br>Example: TBD. |
| OPERATING_COUNTRY | char OPERATING_COUNTRY(STRING64);<br>OPERATING_COUNTRY:long_name = "Operating country of the drifter";<br>OPERATING_COUNTRY:_FillValue = " "; | The operating country of the drifter.<br>Example: TBD. |
| AUTHORS | char AUTHORS(STRING64);<br>AUTHORS:long_name = "List of relevant persons involved in the creation of the data file (comma separated)";<br>AUTHORS:_FillValue = " "; | List of relevant persons involved in the creation of the data file (comma separated).<br>Example: TBD. |

| | | |
|---|---|---|
| DATA_CENTRE | char DATA_CENTRE(STRING2);<br>DATA_CENTRE:long_name = "Data centre in charge of drifter real-time processing";<br>DATA_CENTRE:conventions = "Reference table 4";<br>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.<br>Example: "IF" for Ifremer. |
| DAC_FORMAT_ID | char DAC_FORMAT_ID(STRING16);<br>DAC_FORMAT_ID:long_name = "Format number used by the DAC to describe the data format type for each drifter";<br>DAC_FORMAT_ID:_FillValue = " "; | Format numbers used by individual DACs to describe each drifter type.<br>Example: TBD. |

## 2.4. Drifter characteristics

| Name | Definition | Comment |
|---|---|---|
| PLATFORM_FAMILY | char PLATFORM_FAMILY(STRING256);<br>PLATFORM_FAMILY:long_name = "Category of instrument";<br>PLATFORM_FAMILY:conventions = "Reference table 19";<br>PLATFORM_FAMILY:_FillValue = " "; | Category of instrument.<br>See Reference Table 19.<br>Example: TBD. |
| DRIFTER_TYPE | char DRIFTER_TYPE(STRING32);<br>DRIFTER_TYPE:long_name = "Type of drifter";<br>DRIFTER_TYPE:conventions = "Reference table 20";<br>DRIFTER_TYPE:_FillValue = " "; | Type of drifter.<br>See Reference Table 20.<br>Example: TBD. |
| DRIFTER_MAKER | char DRIFTER_MAKER(STRING256);<br>DRIFTER_MAKER:long_name = "Name of the manufacturer";<br>DRIFTER_MAKER:conventions = "Reference table 21";<br>DRIFTER_MAKER:_FillValue = " "; | Name of the drifter manufacturer.<br>See Reference Table 21.<br>Example: TBD. |
| DRIFTER_MANUFACTURE_DATE | char DRIFTER_MANUFACTURE_DATE(DATE);<br>DRIFTER_MANUFACTURE_DATE:long_name = "Date of the drifter manufacture";<br>DRIFTER_MANUFACTURE_DATE:conventions = "YYYYMMDD";<br>DRIFTER_MANUFACTURE_DATE:_FillValue = " "; | Date of the drifter manufacture.<br>Format : YYYYMMDD<br>Example: "20011230" for December 30th 2001. |
| **DRIFTER_SERIAL_NO** | char DRIFTER_SERIAL_NO(STRING32);<br>DRIFTER_SERIAL_NO:long_name = "Serial number of the drifter";<br>DRIFTER_SERIAL_NO:_FillValue = " "; | This field should contain only the serial number of the drifter.<br>Example: TBD. |
| **DRIFTER_NUMBER** | char DRIFTER_NUMBER(STRING7);<br>DRIFTER_NUMBER:long_name = "Drifter unique identifier";<br>DRIFTER_NUMBER:conventions = "TBD";<br>DRIFTER_NUMBER:_FillValue = " "; | C-RAID drifter unique identifier.<br>Example: "1190171" |

| | | |
|---|---|---|
| DRIFTER_DAC_NUMBER | char DRIFTER_DAC_NUMBER(STRING8);<br>DRIFTER_DAC_NUMBER:long_name = "DAC drifter unique identifier";<br>DRIFTER_DAC_NUMBER:conventions = "TBD";<br>DRIFTER_DAC_NUMBER:_FillValue = " "; | DAC drifter unique identifier.<br>Example: "9706311" |
| DRIFTER_WMO_NUMBER | char DRIFTER_WMO_NUMBER(STRING8);<br>DRIFTER_WMO_NUMBER:long_name = "Drifter WMO number";<br>DRIFTER_WMO_NUMBER:conventions = "WMO drifter identifier : A9IIIII";<br>DRIFTER_WMO_NUMBER:_FillValue = " "; | WMO drifter identifier. WMO is the World Meteorological Organization. This platform number is unique.<br>Example: "3200552" |
| WMO_INST_TYPE | char WMO_INST_TYPE(STRING4);<br>WMO_INST_TYPE:long_name = "Coded instrument type";<br>WMO_INST_TYPE:conventions = "Reference table 8";<br>WMO_INST_TYPE:_FillValue = " "; | Instrument type from WMO code table 1770. A subset of WMO table 1770 is documented in the Reference Table 8.<br>Example: TBD. |
| PTT | char PTT(STRING256);<br>PTT:long_name = "Transmission identifier (Argos, Iridium, etc.)";<br>PTT:_FillValue = " "; | Transmission identifier of the drifter.<br>Example: "11195" |
| TRANS_SYSTEM | char TRANS_SYSTEM(N_TRANS_SYSTEM, STRING16);<br>TRANS_SYSTEM:long_name = "Telecommunication system used";<br>TRANS_SYSTEM:conventions = "Reference table 10";<br>TRANS_SYSTEM:_FillValue = " "; | Name of the telecommunication system from Reference Table 10.<br>Example: "ARGOS" |
| TRANS_SYSTEM_ID | char TRANS_SYSTEM_ID(N_TRANS_SYSTEM, STRING32);<br>TRANS_SYSTEM_ID:long_name = "Program identifier used by the transmission system";<br>TRANS_SYSTEM_ID:_FillValue = " "; | Program identifier of the telecommunication subscription. DACS can use N/A or alternative of their choice when not applicable.<br>Example: "221" is a program number for all the beacons of an Argos customer. |
| TRANS_SYSTEM_FORMAT | char TRANS_SYSTEM_FORMAT(N_TRANS_SYSTEM, STRING32);<br>TRANS_SYSTEM_FORMAT:long_name = "Format Id of drifter transmission data";<br>TRANS_SYSTEM_FORMAT:_FillValue = " "; | Binary format used to relay the data to shore. This entry may point to a table where each entry links to external documents (with DOIs). For example, for Iridium SBD data formats for drifters, there is this document: doi: 10.5281/zenodo.1305119. A similar document for ARGOS drifter data formats would be ideal. |

| | | |
|---|---|---|
| POSITIONING_SYSTEM | char POSITIONING_SYSTEM(N_POSITIONING_SYSTEM, STRING8);<br>POSITIONING_SYSTEM:long_name = "Positioning system";<br>POSITIONING_SYSTEM:conventions = "Reference table 9";<br>POSITIONING_SYSTEM:_FillValue = " "; | Position system from Reference Table 9.<br>Example: "ARGOS" |
| CONTROLLER_BOARD_TYPE | char CONTROLLER_BOARD_TYPE(STRING32);<br>CONTROLLER_BOARD_TYPE:long_name = "Type of controller board";<br>CONTROLLER_BOARD_TYPE:_FillValue = " "; | Describes the type of controller board.<br>Example: TBD. |
| CONTROLLER_BOARD_SERIAL_NO | char CONTROLLER_BOARD_SERIAL_NO(STRING32);<br>CONTROLLER_BOARD_SERIAL_NO:long_name = "Serial number of the controller board";<br>CONTROLLER_BOARD_SERIAL_NO:_FillValue = " "; | The serial number for the primary controller board.<br>Example: TBD. |
| FIRMWARE_VERSION | char FIRMWARE_VERSION(STRING64);<br>FIRMWARE_VERSION:long_name = "Firmware version for the drifter";<br>FIRMWARE_VERSION:_FillValue = " "; | The drifter firmware version. This is specified as per the format on the manufacturer's manual.<br>Example: TBD. |
| FLOAT_DIAMETER | float FLOAT_DIAMETER;<br>FLOAT_DIAMETER:long_name = "Diameter of the float";<br>FLOAT_DIAMETER:units = "cm";<br>FLOAT_DIAMETER:_FillValue = 99999.; | Diameter of the float. |
| SUBSURFACE_FLOAT_PRESENCE | char SUBSURFACE_FLOAT_PRESENCE(STRING16);<br>SUBSURFACE_FLOAT_PRESENCE:long_name = "Is the drifter equipped with a subsurface float?";<br>SUBSURFACE_FLOAT_PRESENCE:_FillValue = " "; | Is the drifter equipped with a subsurface float?<br>Example: TBD. |
| DROGUE_TYPE | char DROGUE_TYPE(STRING16);<br>DROGUE_TYPE:long_name = "Type of the drogue";<br>DROGUE_TYPE:_FillValue = " "; | The type of the drogue.<br>Example: TBD. |
| DROGUE_DIAMETER | float DROGUE_DIAMETER;<br>DROGUE_DIAMETER:long_name = "Diameter of the drogue";<br>DROGUE_DIAMETER:units = "m";<br>DROGUE_DIAMETER:_FillValue = 99999.; | Diameter of the drogue. |
| DROGUE_LENGTH | float DROGUE_LENGTH;<br>DROGUE_LENGTH:long_name = "Length of the drogue";<br>DROGUE_LENGTH:units = "m";<br>DROGUE_LENGTH:_FillValue = 99999.; | Length of the drogue. |
| DROGUE_CENTER_DEPTH | float DROGUE_CENTER_DEPTH;<br>DROGUE_CENTER_DEPTH:long_name = "Depth of the center of the drogue";<br>DROGUE_CENTER_DEPTH:units = "m";<br>DROGUE_CENTER_DEPTH:_FillValue = 99999.; | Depth of the center of the drogue. |
| DRAG_AREA_ABOVE_DROGUE | float DRAG_AREA_ABOVE_DROGUE;<br>DRAG_AREA_ABOVE_DROGUE:long_name = "Drag area of all drifter components except the drogue";<br>DRAG_AREA_ABOVE_DROGUE:_FillValue = 99999.; | Drag area of all drifter components except the drogue. |

| | | |
|---|---|---|
| DRAG_AREA_OF_DROGUE | float DRAG_AREA_OF_DROGUE;<br>DRAG_AREA_OF_DROGUE:long_name = "Drag area of the drogue";<br>DRAG_AREA_OF_DROGUE:_FillValue = 99999.; | Drag area of the drogue. |
| DRAG_AREA_RATIO | float DRAG_AREA_RATIO;<br>DRAG_AREA_RATIO:long_name = "Ratio of the drogue cross-sectional area to the area of all other components";<br>DRAG_AREA_RATIO:_FillValue = 99999.; | Ratio of the drogue cross-sectional area to the area of all other Components. |
| DROGUE_DETECTION_METHOD | char DROGUE_DETECTION_METHOD(STRING16);<br>DROGUE_DETECTION_METHOD:long_name = "Method used to detect drogue presence";<br>DROGUE_DETECTION_METHOD:conventions = "Reference table 25";<br>DROGUE_DETECTION_METHOD:_FillValue = " "; | Method used to detect drogue presence.<br>Example: TBD. |
| DROGUE_BALLAST | float DROGUE_BALLAST;<br>DROGUE_BALLAST:long_name = "Weight of the drogue ballast";<br>DROGUE_BALLAST:units = "kg";<br>DROGUE_BALLAST:_FillValue = 99999.; | Weight of the drogue ballast. |
| BATTERY_TYPE | char BATTERY_TYPE(STRING64);<br>BATTERY_TYPE:long_name = "Type of battery packs in the drifter";<br>BATTERY_TYPE:_FillValue = " "; | Describes the type of battery packs in the drifter.<br>Example: "Alkaline", "Lithium" or "Alkaline and Lithium" |
| BATTERY_PACKS | char BATTERY_PACKS(STRING64);<br>BATTERY_PACKS:long_name = "Configuration of battery packs in the drifter";<br>BATTERY_PACKS:_FillValue = " "; | Describes the configuration of battery packs in the drifter, number and type.<br>Example: "4DD Li + 1C Alk" |
| BATTERY_CAPACITY | float BATTERY_CAPACITY;<br>BATTERY_CAPACITY:long_name = "Capacity of the battery at drifter deployment";<br>BATTERY_CAPACITY:units = " Amper-Hour";<br>BATTERY_CAPACITY:_FillValue = 99999.; | Capacity of the battery at drifter deployment. |
| SPECIAL_FEATURES | char SPECIAL_FEATURES(STRING1024);<br>SPECIAL_FEATURES:long_name = "Extra features of the drifter (algorithms, compressee etc.)";<br>SPECIAL_FEATURES:_FillValue = " "; | Additional float features can be specified here such as algorithms used by the drifter.<br>Example: TBD. |
| DRIFTER_MANUAL_VERSION | char DRIFTER_MANUAL_VERSION(STRING16);<br>DRIFTER_MANUAL_VERSION:long_name = "Manual version for the drifter";<br>DRIFTER_MANUAL_VERSION:_FillValue = " "; | The version date or number for the manual for each drifter.<br>Example: TBD. |
| ANOMALY | char ANOMALY(STRING256);<br>ANOMALY:long_name = "Describes any anomalies or problems the drifter may have had";<br>ANOMALY:_FillValue = " "; | This field describes any anomaly or problem the drifter may have had.<br>Example: TBD. |

| | | |
|---|---|---|
| CUSTOMISATION | char CUSTOMISATION(STRING1024);<br>CUSTOMISATION:long_name = "Drifter customisation, i.e. (institution and modifications)";<br>CUSTOMISATION:_FillValue = " "; | Free form field to record changes made to the drifter after manufacture and before deployment, i.e. this could be the customisation institution plus a list of modifications.<br>Example: TBD. |

## 2.4.3 Drifter deployment and mission information

| Name | Definition | Comment |
|---|---|---|
| **LAUNCH_DATE** | char LAUNCH_DATE(DATE_TIME);<br>LAUNCH_DATE:long_name = "Date (UTC) of the deployment";<br>LAUNCH_DATE:conventions = "YYYYMMDDHHMISS";<br>LAUNCH_DATE:_FillValue = " "; | Date and time (UTC) of launch of the drifter.<br>Format : YYYYMMDDHHMISS<br>Example: "20011230090500" for December 30th 2001 09:05:00. |
| **LAUNCH_LATITUDE** | double LAUNCH_LATITUDE;<br>LAUNCH_LATITUDE:long_name = "Latitude of the drifter when deployed";<br>LAUNCH_LATITUDE:units = "degree_north";<br>LAUNCH_LATITUDE:valid_min = -90.;<br>LAUNCH_LATITUDE:valid_max = 90.;<br>LAUNCH_LATITUDE:_FillValue = 99999.; | Latitude of the drifter launch.<br>Unit: decimal degree north.<br>Example: 44.4991 for 44° 29′ 56.76″ N |
| **LAUNCH_LONGITUDE** | double LAUNCH_LONGITUDE;<br>LAUNCH_LONGITUDE:long_name = "Longitude of the drifter when deployed";<br>LAUNCH_LONGITUDE:units = "degree_east";<br>LAUNCH_LONGITUDE:valid_min = -180.;<br>LAUNCH_LONGITUDE:valid_max = 180.;<br>LAUNCH_LONGITUDE:_FillValue = 99999.; | Longitude of the drifter launch.<br>Unit: decimal degree east.<br>Example: 16.7222 for 16° 43′ 19.92″ E |
| **LAUNCH_QC** | byte LAUNCH_QC;<br>LAUNCH_QC:long_name = "Quality on launch date, time and location";<br>LAUNCH_QC:conventions = "Reference table 2";<br>LAUNCH_QC:flag_values = 0, 1, 2, 3, 4, 5, 8, 9;<br>LAUNCH_QC:flag_meanings = "no_qc_performed good_data probably_good_data bad_data_that_are_potentially_correctable bad_data value_changed interpolated_value missing_value";<br>LAUNCH_QC:_FillValue = -128; | Quality flag on drifter launch date, time and location.<br>The flag scale is described in the Reference Table 2.<br>Example: 1 (launch time and location seems correct). |
| STARTUP_DATE | char STARTUP_DATE(DATE_TIME);<br>STARTUP_DATE:long_name = "Date (UTC) of the activation of the drifter";<br>STARTUP_DATE:conventions = "YYYYMMDDHHMISS";<br>STARTUP_DATE:_FillValue = " "; | Date and time (UTC) of the activation of the drifter before or just after deployment.<br>Format: YYYYMMDDHHMISS<br>Example: "20011230090500" for December 30th 2001 09:05 :00. |

| | | |
|---|---|---|
| STARTUP_DATE_QC | byte STARTUP_DATE_QC;<br>STARTUP_DATE_QC:long_name = "Quality on startup date";<br>STARTUP_DATE_QC:conventions = "Reference table 2";<br>STARTUP_DATE_QC:flag_values = 0, 1, 2, 3, 4, 5, 8, 9;<br>STARTUP_DATE_QC:flag_meanings = "no_qc_performed good_data probably_good_data bad_data_that_are_potentially_correctable bad_data value_changed interpolated_value missing_value";<br>STARTUP_DATE_QC:_FillValue = -128; | Quality flag on startup date.<br>The flag scale is described in the Reference Table 2.<br>Example: 1 (startup date seems correct). |
| DEPLOYMENT_PLATFORM | char DEPLOYMENT_PLATFORM(STRING32);<br>DEPLOYMENT_PLATFORM:long_name = "Identifier of the deployment platform";<br>DEPLOYMENT_PLATFORM:_FillValue = " "; | Identifier of the deployment platform, i.e. vessel or ship name.<br>Example: "L'ATALANTE". |
| DEPLOYMENT_CRUISE_ID | char DEPLOYMENT_CRUISE_ID(STRING32);<br>DEPLOYMENT_CRUISE_ID:long_name = "Identification number or reference number of the cruise used to deploy the drifter";<br>DEPLOYMENT_CRUISE_ID:_FillValue = " "; | Identification number or reference number of the cruise used to deploy the platform.<br>Example: "POMME2". |
| DEPLOYMENT_REFERENCE_STATION_ID | char DEPLOYMENT_REFERENCE_STATION_ID(STRING256);<br>DEPLOYMENT_REFERENCE_STATION_ID:long_name = "Identifier or reference number of co-located stations used to verify the drifter measurements";<br>DEPLOYMENT_REFERENCE_STATION_ID:_FillValue = " "; | Identifier or reference number of co-located CTD or XBT stations used to verify the drifter measurements.<br>Example: 58776, 58777 |
| DEPLOYMENT_OPERATOR | char DEPLOYMENT_OPERATOR(STRING256);<br>DEPLOYMENT_OPERATOR:long_name = "Name of the person in charge of the drifter deployment";<br>DEPLOYMENT_OPERATOR:_FillValue = " "; | Name of the person in charge of the drifter deployment. |
| DEPLOYMENT_BATTERY_VOLTAGE | float DEPLOYMENT_BATTERY_VOLTAGE;<br>DEPLOYMENT_BATTERY_VOLTAGE:long_name = "Drifter battery voltage value when deployed";<br>DEPLOYMENT_BATTERY_VOLTAGE:units = "Volt";<br>DEPLOYMENT_BATTERY_VOLTAGE:_FillValue = 99999.; | Battery voltage at deployment. |
| DROGUE_OFF_DATE | char DROGUE_OFF_DATE(DATE_TIME);<br>DROGUE_OFF_DATE:long_name = "Date (UTC) of the loss of the drogue";<br>DROGUE_OFF_DATE:conventions = "YYYYMMDDHHMISS";<br>DROGUE_OFF_DATE:_FillValue = " "; | Date (UTC) of the loss of the drogue.<br>Format: YYYYMMDDHHMISS<br>Example: "20011230090500" for December 30th 2001 09:05:00 |
| DROGUE_OFF_LATITUDE | double DROGUE_OFF_LATITUDE;<br>DROGUE_OFF_LATITUDE:long_name = "Latitude of the loss of the drogue";<br>DROGUE_OFF_LATITUDE:units = "degree_north";<br>DROGUE_OFF_LATITUDE:valid_min = -90.;<br>DROGUE_OFF_LATITUDE:valid_max = 90.;<br>DROGUE_OFF_LATITUDE:_FillValue = 99999.; | Latitude of the loss of the drogue.<br>Unit: decimal degree north.<br>Example: 44.4991 for 44° 29' 56.76" N |

| | | |
|---|---|---|
| DROGUE_OFF_LONGITUDE | double DROGUE_OFF_LONGITUDE;<br>DROGUE_OFF_LONGITUDE:long_name = "Longitude of the loss of the drogue";<br>DROGUE_OFF_LONGITUDE:units = "degree_east";<br>DROGUE_OFF_LONGITUDE:valid_min = -180.;<br>DROGUE_OFF_LONGITUDE:valid_max = 180.;<br>DROGUE_OFF_LONGITUDE:_FillValue = 99999.; | Longitude of the loss of the drogue.<br>Unit: decimal degree east.<br>Example: 16.7222 for 16° 43′ 19.92″ E |
| DROGUE_OFF_QC | byte DROGUE_OFF_QC;<br>DROGUE_OFF_QC:long_name = "Quality on drogue loss date, time and location";<br>DROGUE_OFF_QC:conventions = "Reference table 2";<br>DROGUE_OFF_QC:flag_values = 0, 1, 2, 3, 4, 5, 8, 9;<br>DROGUE_OFF_QC:flag_meanings = "no_qc_performed good_data probably_good_data bad_data_that_are_potentially_correctable bad_data value_changed interpolated_value missing_value";<br>DROGUE_OFF_QC:_FillValue = -128; | Quality flag on drogue loss date, time and location.<br>The flag scale is described in the Reference Table 2.<br>Example: 1 (drogue loss time and location seems correct). |
| END_MISSION_DATE | char END_MISSION_DATE(DATE_TIME);<br>END_MISSION_DATE:long_name = "Date (UTC) of the end of mission of the drifter";<br>END_MISSION_DATE:conventions = "YYYYMMDDHHMISS";<br>END_MISSION_DATE:_FillValue = " "; | Date (UTC) of the end of mission of the drifter.<br>Format: YYYYMMDDHHMISS<br>Example: "20011230090500" for December 30th 2001 09:05:00 |
| END_MISSION_LATITUDE | double END_MISSION_LATITUDE;<br>END_MISSION_LATITUDE:long_name = "Latitude of the drifter at the end of its mission";<br>END_MISSION_LATITUDE:units = "degree_north";<br>END_MISSION_LATITUDE:valid_min = -90.;<br>END_MISSION_LATITUDE:valid_max = 90.;<br>END_MISSION_LATITUDE:_FillValue = 99999.; | Latitude at end of the mission.<br>Unit: decimal degree north.<br>Example: 44.4991 for 44° 29′ 56.76″ N |
| END_MISSION_LONGITUDE | double END_MISSION_LONGITUDE;<br>END_MISSION_LONGITUDE:long_name = "Longitude of the drifter at the end of its mission";<br>END_MISSION_LONGITUDE:units = "degree_east";<br>END_MISSION_LONGITUDE:valid_min = -180.;<br>END_MISSION_LONGITUDE:valid_max = 180.;<br>END_MISSION_LONGITUDE:_FillValue = 99999.; | Longitude at end of the mission.<br>Unit: decimal degree east.<br>Example: 16.7222 for 16° 43′ 19.92″ E |
| END_MISSION_QC | byte END_MISSION_QC;<br>END_MISSION_QC:long_name = "Quality on end mission date, time and location";<br>END_MISSION_QC:conventions = "Reference table 2";<br>END_MISSION_QC:flag_values = 0, 1, 2, 3, 4, 5, 8, 9;<br>END_MISSION_QC:flag_meanings = "no_qc_performed good_data probably_good_data bad_data_that_are_potentially_correctable bad_data value_changed interpolated_value missing_value";<br>END_MISSION_QC:_FillValue = -128; | Quality flag on end mission date, time and location.<br>The flag scale is described in the Reference Table 2.<br>Example: 1 (launch time and location seems correct). |
| END_MISSION_STATUS | char END_MISSION_STATUS;<br>END_MISSION_STATUS:long_name = "Status of the end of mission of the drifter";<br>END_MISSION_STATUS:conventions = "Reference table 16";<br>END_MISSION_STATUS:_FillValue = " "; | Status of the end of the float's deployment.<br>See Reference Table 16.<br>Example: TBD. |

| | | |
|---|---|---|
| ODDS_AGROUND | float ODDS_AGROUND;<br>ODDS_AGROUND:long_name = "Probability that the drifter ran aground";<br>ODDS_AGROUND:_FillValue = 99999.; | Probability that the drifter ran aground. |
| ODDS_PICKUP | float ODDS_PICKUP;<br>ODDS_PICKUP:long_name = "Probability that the drifter was picked up";<br>ODDS_PICKUP:_FillValue = 99999.; | Probability that the drifter was picked up. |

### 2.4.4 Configuration parameter

This section describes the configuration parameters for a drifter. It is important to note that configuration parameters are drifter settings, they are not measurements reported by the drifter.
Configuration parameters may or may not be reported by a drifter.
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)<br>LAUNCH_CONFIG_PARAMETER_NAME:long_name = "Name of configuration parameter at launch";<br>LAUNCH_CONFIG_PARAMETER_NAME:_FillValue = " "; | Name of the configuration parameter; pre-deployment or launch settings.<br>See Reference Table 13 for standard configuration parameter names.<br>Example: TBD. |
| LAUNCH_CONFIG_PARAMETER_VALUE | float LAUNCH_CONFIG_PARAMETER_VALUE(N_LAUNCH_CONFIG_PARAM)<br>LAUNCH_CONFIG_ PARAMETER_VALUE:long_name = "Value of configuration parameter at launch";<br>LAUNCH_CONFIG_ PARAMETER_VALUE:_FillValue = 99999.; | Value of the configuration parameter; either pre-deployment or launch settings.<br>Example: TBD. |
| CONFIG_PARAMETER_NAME | char CONFIG_PARAMETER_NAME(N_CONFIG_PARAM, STRING128)<br>CONFIG_PARAMETER_NAME:long_name = "Name of configuration parameter";<br>CONFIG_PARAMETER_NAME:_FillValue = " "; | Name of the configuration parameter; mission settings.<br>See Reference Table 13 for standard configuration parameter names.<br>Example: TBD. |
| CONFIG_PARAMETER_VALUE | float CONFIG_PARAMETER_VALUE(N_MISSION, N_CONFIG_PARAM)<br>CONFIG_ PARAMETER_VALUE:long_name = "Value of configuration parameter";<br>CONFIG_ PARAMETER_VALUE:_FillValue = 99999.; | Value of the configuration parameter; mission settings.<br>Example: TBD. |

| | | |
|---|---|---|
| CONFIG_MISSION_NUMBER | int CONFIG_MISSION_NUMBER:long_name = "Unique number denoting the missions performed by the drifter";<br>CONFIG_MISSION_NUMBER:conventions = "1...N, 1 : first complete mission";<br>CONFIG_MISSION_NUMBER:_FillValue = 99999; | Unique number of the mission to which the measurement belongs. |
| CONFIG_MISSION_COMMENT | char CONFIG_MISSION_COMMENT(N_MISSION, STRING256)<br>CONFIG_MISSION_COMMENT:long_name = "Comment on configuration";<br>CONFIG_MISSION_COMMENT:_FillValue = " "; | Comment on this configuration mission.<br>Example: TBD. |

### 2.4.5 Drifter 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 in parameter physical units using a calibration equation. This conversion can be done onboard the drifter 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).

#### 2.4.5.1 Drifter sensor information

This section contains information about the sensors of the drifter.

A list of standardised C-RAID sensor names is given in Reference Table 22.

| Name | Definition | Comment |
|---|---|---|
| **SENSOR** | char SENSOR(N_SENSOR, STRING32);<br>SENSOR:long_name = "Name of the sensor mounted on the drifter";<br>SENSOR:conventions = "Reference table 22";<br>SENSOR:_FillValue = " "; | Names of the sensors mounted on the drifter.<br>See Reference Table 22.<br>Example: TBD. |
| SENSOR_MAKER | char SENSOR_MAKER(N_SENSOR, STRING256);<br>SENSOR_MAKER:long_name = "Name of the sensor manufacturer";<br>SENSOR_MAKER:conventions = "Reference table 23";<br>SENSOR_MAKER:_FillValue = " "; | Name of the manufacturer of the sensor.<br>See Reference Table 23.<br>Example: TBD. |
| SENSOR_MANUFACTURE_DATE | char SENSOR_MANUFACTURE_DATE(N_SENSOR, DATE);<br>SENSOR_MANUFACTURE_DATE:long_name = "Date of the sensor manufacture";<br>SENSOR_MANUFACTURE_DATE:conventions = "YYYYMMDD";<br>SENSOR_MANUFACTURE_DATE:_FillValue = " "; | Date of the sensor manufacture.<br>Format : YYYYMMDD<br>Example: "20011230" for December 30th 2001. |
| SENSOR_MODEL | char SENSOR_MODEL(N_SENSOR, STRING256);<br>SENSOR_MODEL:long_name = "Model of sensor";<br>SENSOR_MODEL:conventions = "Reference table 24";<br>SENSOR_MODEL:_FillValue = " "; | Model of sensor.<br>See Reference Table 24.<br>Example: TBD. |
| **SENSOR_SERIAL_NO** | char SENSOR_SERIAL_NO(N_SENSOR, STRING16);<br>SENSOR_SERIAL_NO:long_name = "Serial number of the sensor";<br>SENSOR_SERIAL_NO:_FillValue = " "; | Serial number of the sensor.<br>Example: TBD. |

| | | |
|---|---|---|
| SENSOR_NB_LEVEL | int SENSOR_NB_LEVEL(N_SENSOR);<br>SENSOR_NB_LEVEL:long_name = "Number of measurement levels of the sensor";<br>SENSOR_NB_LEVEL:_FillValue = 99999; | Number of measurements levels of the sensor.<br>Example: TBD. |
| SENSOR_LEVEL_DEPTH | float SENSOR_LEVEL_DEPTH(N_SENSOR, N_LEVEL<n>);<br>SENSOR_LEVEL_DEPTH:long_name = "Theoretical depth of sensor measurements";<br>SENSOR_LEVEL_DEPTH:standard_name = "depth";<br>SENSOR_LEVEL_DEPTH:conventions = "SENSOR_LEVEL_DEPTH is >0 below the surface";<br>SENSOR_LEVEL_DEPTH:units = "m";<br>SENSOR_LEVEL_DEPTH:axis = "Z";<br>SENSOR_LEVEL_DEPTH:_FillValue = 99999.; | Theoretical depth of sensor measurements.<br>Example: TBD. |
| SENSOR_INITIAL_END_DATE | char SENSOR_INITIAL_END_DATE(N_SENSOR, DATE_TIME);<br>SENSOR_INITIAL_END_DATE:long_name = "Date (UTC) of end of useful mission for the sensor, according to the initial platform operators";<br>SENSOR_INITIAL_END_DATE:conventions = "YYYYMMDDHHMISS";<br>SENSOR_INITIAL_END_DATE:_FillValue = " "; | Date (UTC) of end of useful mission for the sensor, according to the initial platform operators.<br>Format: YYYYMMDDHHMISS<br>Example: "20011230090500" for December 30th 2001 09:05:00 |

### 2.4.5.2 Drifter parameter information

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

| Name | Definition | Comment |
|---|---|---|
| **PARAMETER** | char PARAMETER(N_PARAM, STRING64);<br>PARAMETER:long_name = "Name of parameter computed from drifter measurements";<br>PARAMETER:conventions = "Reference table 3";<br>PARAMETER:_FillValue = " "; | Names of the parameters measured by drifter sensors or derived from drifter measurements.<br>The parameter names are listed in Reference Table 3.<br>Examples: TBD. |
| **PARAMETER_SENSOR** | char PARAMETER_SENSOR(N_PARAM, STRING128);<br>PARAMETER:long_name = "Name of the sensor that measures this parameter";<br>PARAMETER:conventions = "Reference table 22";<br>PARAMETER:_FillValue = " "; | Names of the sensors that measured the drifter parameters.<br>See Reference Table 22.<br>Example: TBD. |
| PARAMETER_UNITS | char PARAMETER_UNITS(N_PARAM, STRING32);<br>PARAMETER_UNITS:long_name = "Units of accuracy and resolution of the parameter";<br>PARAMETER_UNITS:_FillValue = " "; | Units of accuracy and resolution of the parameter.<br>Example: TBD. |

| | | |
|---|---|---|
| PARAMETER_ACCURACY | char PARAMETER_ACCURACY(N_PARAM, STRING32);<br>PARAMETER_ACCURACY:long_name = "Accuracy of the parameter";<br>PARAMETER_ACCURACY:_FillValue = " "; | Accuracy of the parameter.<br>Example: TBD. |
| PARAMETER_RESOLUTION | char PARAMETER_RESOLUTION(N_PARAM, STRING32);<br>PARAMETER_RESOLUTION:long_name = "Resolution of the parameter";<br>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 drifter through telemetry).<br>Example: TBD. |
| **PARAMETER_DATA_STATE_INDICATOR** | char PARAMETER_DATA_STATE_INDICATOR(N_PARAM, STRING4);<br>PARAMETER_DATA_STATE_INDICATOR:long_name = "Degree of processing the parameter data have passed through";<br>PARAMETER_DATA_STATE_INDICATOR:conventions = "Reference table 6";<br>PARAMETER_DATA_STATE_INDICATOR:_FillValue = " "; | Degree of processing the parameter data has passed through. The parameter data state indicator is described in the Reference Table 6. |

### 2.4.6 Sensor calibration information

This section contains information about the calibration of the sensor.
The PREDEPLOYMENT_CALIB_* variables concerns the instrumental calibration of the sensor.
The REAL_TIME_ADJUSTMENT_CALIB_* variables concerns the real-time adjustment calibration of the sensor.
The SCIENTIFIC_CALIB_* variables concerns the delayed mode adjustment of the sensor, based on a data analysis.
The PREDEPLOYMENT_CALIB_*, REAL_TIME_ADJUSTMENT_CALIB_* and SCIENTIFIC_CALIB_* parameters of the table below link to the PARAMETER variable (through the N_PARAM index). 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);<br>PREDEPLOYMENT_CALIB_EQUATION:long_name = "Predeployment calibration equation for this parameter";<br>PREDEPLOYMENT_CALIB_EQUATION:_FillValue = " "; | Predeployment calibration equation for this parameter.<br>Example: "Tc = a1 * T + a0". |

| | | |
|---|---|---|
| PREDEPLOYMENT_CALIB_COEFFICIENT | char PREDEPLOYMENT_CALIB_COEFFICIENT(N_PARAM, STRING1024);<br>PREDEPLOYMENT_CALIB_COEFFICIENT:long_name = "Predeployment calibration coefficients for this equation";<br>PREDEPLOYMENT_CALIB_COEFFICIENT:_FillValue = " "; | Predeployment calibration coefficients for this equation.<br>Example: "a1=0.99997 , a0=0.0021". |
| PREDEPLOYMENT_CALIB_COMMENT | char PREDEPLOYMENT_CALIB_COMMENT(N_PARAM, STRING1024);<br>PREDEPLOYMENT_CALIB_COMMENT:long_name = "Comment applying to this parameter predeployment calibration";<br>PREDEPLOYMENT_CALIB_COMMENT:_FillValue = " "; | Comments applying to this parameter predeployment calibration.<br>Example: TBD. |
| PREDEPLOYMENT_CALIB_DATE | char PREDEPLOYMENT_CALIB_DATE(N_PARAM, DATE_TIME)<br>PREDEPLOYMENT_CALIB_DATE:long_name = "Date of predeployment calibration for this parameter";<br>PREDEPLOYMENT_CALIB_DATE:conventions = "YYYYMMDDHHMISS";<br>PREDEPLOYMENT_CALIB_DATE:_FillValue = " "; | Date of the pre-deployment calibration.<br>Example: "20011217161700" for November 17th 2001 16:17:00. |
| REAL_TIME_ADJUSTMENT_CALIB_EQUATION | char REAL_TIME_ADJUSTMENT_CALIB_EQUATION(N_PARAM, STRING1024);<br>REAL_TIME_ADJUSTMENT_CALIB_EQUATION:long_name = "Real-time adjustment calibration equation for this parameter";<br>REAL_TIME_ADJUSTMENT_CALIB_EQUATION:_FillValue = " "; | Real-time adjustment calibration equation applied to the parameter.<br>Example: "Tc = a1 * T + a0". |
| REAL_TIME_ADJUSTMENT_CALIB_COEFFICIENT | char REAL_TIME_ADJUSTMENT_CALIB_COEFFICIENT(N_PARAM, STRING1024);<br>REAL_TIME_ADJUSTMENT_CALIB_COEFFICIENT:long_name = "Real-time adjustment calibration coefficients for this equation";<br>REAL_TIME_ADJUSTMENT_CALIB_COEFFICIENT:_FillValue = " "; | Real-time adjustment calibration coefficients for this equation.<br>Example: "a1=0.99997 , a0=0.0021". |
| REAL_TIME_ADJUSTMENT_CALIB_COMMENT | char REAL_TIME_ADJUSTMENT_CALIB_COMMENT(N_PARAM, STRING1024);<br>REAL_TIME_ADJUSTMENT_CALIB_COMMENT:long_name = "Comment applying to this parameter real-time adjustment calibration";<br>REAL_TIME_ADJUSTMENT_CALIB_COMMENT:_FillValue = " "; | Comment about this real-time adjustment calibration<br>Example: TBD. |
| REAL_TIME_ADJUSTMENT_CALIB_DATE | char REAL_TIME_ADJUSTMENT_CALIB_DATE(N_PARAM, DATE_TIME)<br>REAL_TIME_ADJUSTMENT_CALIB_DATE:long_name = "Date of real-time adjustment calibration for this parameter";<br>REAL_TIME_ADJUSTMENT_CALIB_DATE:conventions = "YYYYMMDDHHMISS";<br>REAL_TIME_ADJUSTMENT_CALIB_DATE:_FillValue = " "; | Date of the real-time adjustment calibration.<br>Example: "20011217161700" for November 17th 2001 16:17:00. |

| | | |
|---|---|---|
| SCIENTIFIC_CALIB_EQUATION | char SCIENTIFIC_CALIB_EQUATION(N_PARAM, STRING1024);<br>SCIENTIFIC_CALIB_EQUATION:long_name = "Scientific calibration equation for this parameter";<br>SCIENTIFIC_CALIB_EQUATION:_FillValue = " "; | Scientific calibration equation applied to the parameter. Example: "Tc = a1 * T + a0". |
| SCIENTIFIC_CALIB_COEFFICIENT | char SCIENTIFIC_CALIB_COEFFICIENT(N_PARAM, STRING1024);<br>SCIENTIFIC_CALIB_COEFFICIENT:long_name = "Scientific calibration coefficients for this equation";<br>SCIENTIFIC_CALIB_COEFFICIENT:_FillValue = " "; | Scientific calibration coefficients for this equation. Example: "a1=0.99997 , a0=0.0021". |
| SCIENTIFIC_CALIB_COMMENT | char SCIENTIFIC_CALIB_COMMENT(N_PARAM, STRING1024);<br>SCIENTIFIC_CALIB_COMMENT:long_name = "Comment applying to this parameter scientific calibration";<br>SCIENTIFIC_CALIB_COMMENT:_FillValue = " "; | Comment about this scientific calibration Example: TBD. |
| SCIENTIFIC_CALIB_DATE | char SCIENTIFIC_CALIB_DATE(N_PARAM, DATE_TIME)<br>SCIENTIFIC_CALIB_DATE:long_name = "Date of scientific calibration for this parameter";<br>SCIENTIFIC_CALIB_DATE:conventions = "YYYYMMDDHHMISS";<br>SCIENTIFIC_CALIB_DATE:_FillValue = " "; | Date of the scientific calibration. Example: "20011217161700" for November 17th 2001 16:17:00. |

### 2.4.7 Data measurements

The variables of this section are used to store parameter measurements, surface locations and associated times. They are all indexed with the N_TIME dimension.

When multiple measurements are available from a given timestamp, an additional N_LEVEL<n> dimension is used. This dimension is however omitted when <n> = 1.

| Name | Definition | Comment |
|---|---|---|
| | | |
| **Time information** | | |
| **JULD** | double JULD(N_TIME);<br>JULD:long_name = "Julian day (UTC) of each measurement relative to REFERENCE_DATE_TIME";<br>JULD:standard_name = "time";<br>JULD:units = "days since 1950-01-01 00:00:00 UTC";<br>JULD:conventions = "Relative Julian days with decimal part (as parts of day)";<br>JULD:resolution = XX;<br>JULD:axis = "T"; | Julian day of the location (or measurement).<br>The integer part represents the day, the decimal part represents the time of the measurement.<br>Date and time are in universal time coordinates.<br>The Julian day is relative to REFERENCE_DATE_TIME.<br>**JULD(N_TIME) should be filled and its values should be unique within the file.**<br>Example: 18833.8013889885 for July 25th 2001 19:14:00 |
| **JULD_STATUS** | char JULD_STATUS(N_TIME);<br>JULD_STATUS:long_name="Status of the date and time"<br>JULD_STATUS:conventions = "Reference table 15";<br>JULD_STATUS:_FillValue = " "; | Status flag on JULD date and time.<br>The flag scale is described in Reference Table 15.<br>Example: "2": Value is transmitted by the drifter. |
| **JULD_QC** | byte JULD_QC(N_TIME);<br>JULD_QC:long_name = "Quality on date and time";<br>JULD_QC:conventions = "Reference table 2";<br>JULD_QC:flag_values = 0, 1, 2, 3, 4, 5, 8, 9;<br>JULD_QC:flag_meanings = "no_qc_performed good_data probably_good_data bad_data_that_are_potentially_correctable bad_data value_changed interpolated_value missing_value";<br>JULD_QC:_FillValue = -128; | Quality flag on JULD date and time.<br>The flag scale is described in the Reference Table 2.<br>Example: 1 (date and time seem correct). |

| | | |
|---|---|---|
| JULD_ADJUSTED | double JULD_ADJUSTED(N_TIME);<br>JULD_ADJUSTED:long_name = "Adjusted Julian day (UTC) of each measurement relative to REFERENCE_DATE_TIME";<br>JULD_ADJUSTED:standard_name = "time";<br>JULD_ADJUSTED:units = "days since 1950-01-01 00:00:00 UTC";<br>JULD_ADJUSTED:conventions = "Relative Julian days with decimal part (as parts of day)";<br>JULD_ADJUSTED:resolution = XX;<br>JULD:axis = "T";<br>JULD_ADJUSTED:_FillValue = 999999.; | Adjusted Julian day of the location (or measurement).<br>The integer part represents the day, the decimal part represents the time of the measurement.<br>Date and time are in universal time coordinates.<br>The Julian day is relative to REFERENCE_DATE_TIME.<br><br>The date may be adjusted due to float clock drift or expert review.<br>Example: 18833.8013889885 for July 25th 2001 19:14:00 |
| JULD_ADJUSTED_STATUS | char JULD_ADJUSTED_STATUS(N_TIME);<br>JULD_ADJUSTED_STATUS:long_name="Status of the JULD_ADJUSTED date"<br>JULD_ADJUSTED_STATUS:conventions = "Reference table 15";<br>JULD_ADJUSTED_STATUS:_FillValue = " "; | Status flag on JULD date and time.<br>The flag scale is described in Reference Table 15.<br>Example: "2": Value is transmitted by the drifter. |
| JULD_ADJUSTED_QC | byte JULD_ADJUSTED_QC(N_TIME);<br>JULD_ADJUSTED_QC:long_name = "Quality on adjusted date and time";<br>JULD_ADJUSTED_QC:conventions = "Reference table 2";<br>JULD_ADJUSTED_QC:flag_values = 0, 1, 2, 3, 4, 5, 8, 9;<br>JULD_ADJUSTED_QC:flag_meanings = "no_qc_performed good_data probably_good_data bad_data_that_are_potentially_correctable bad_data value_changed interpolated_value missing_value";<br>JULD_ADJUSTED_QC:_FillValue = -128; | Quality flag on JULD_ADJUSTED date and time.<br>The flag scale is described in the Reference Table 2.<br>Example: 1 (date and time seem correct). |
| JULD_QC_FAILED | int JULD_QC_FAILED(N_TIME);<br>JULD_QC_FAILED:long_name = "Number of the QC test that set the QC failed flag";<br>JULD_QC_FAILED:_FillValue = 99999; | Number of the QC test that set the QC failed flag.<br>This information also concerns QC = 2 flags; if QC = 1, the FillValue is inserted.<br>Example: TBD. |
| **Location information** | | |
| LATITUDE | double LATITUDE(N_TIME);<br>LATITUDE:long_name = "Latitude of each location";<br>LATITUDE:standard_name = "latitude";<br>LATITUDE:units = "degree_north";<br>LATITUDE:_FillValue = 99999.;<br>LATITUDE:valid_min = -90.;<br>LATITUDE:valid_max = 90.;<br>LATITUDE:axis = "Y"; | Latitude of the location (or measurement).<br>Unit: decimal degree north.<br>Example: 44.4991 for 44° 29' 56.76" N. |

| | | |
|---|---|---|
| LONGITUDE | double LONGITUDE(N_TIME);<br>LONGITUDE:long_name = "Longitude of each location";<br>LONGITUDE:standard_name = "longitude";<br>LONGITUDE:units = "degree_east";<br>LONGITUDE:_FillValue = 99999.;<br>LONGITUDE:valid_min = -180.;<br>LONGITUDE:valid_max = 180.;<br>LONGITUDE:axis = "X"; | Longitude of the location (or measurement).<br>Unit: decimal degree east.<br>Example: 16.7222 for 16° 43' 19.92" E. |
| POSITION_ACCURACY | char POSITION_ACCURACY(N_TIME);<br>POSITION_ACCURACY:long_name = "Estimated accuracy in latitude and longitude";<br>POSITION_ACCURACY:conventions = "Reference table 5";<br>POSITION_ACCURACY:_FillValue = " "; | Position accuracy received from the positioning system.<br>The location classes from Argos are described in the Reference Table 5.<br>A "G" indicates the GPS positioning system.<br>Examples: "3" for a latitude and longitude accuracy < 250 m.<br>"G" for GPS accuracy. |
| POSITION_QC | byte POSITION_QC(N_TIME);<br>POSITION_QC:long_name = "Quality on position";<br>POSITION_QC:conventions = "Reference table 2";<br>POSITION_QC:_FillValue = -128;<br>POSITION_QC:flag_values = 0, 1, 2, 3, 4, 5, 8, 9;<br>POSITION_QC:flag_meanings = "no_qc_performed good_data probably_good_data bad_data_that_are_potentially_correctable bad_data value_changed interpolated_value missing_value"; | Quality flag on position.<br>The flag on position is set according to (JULD, LATITUDE, LONGITUDE) quality.<br>The flag scale is described in the Reference Table 2.<br>Example: 1 (position seems correct). |
| POSITION_QC_FAILED | int POSITION_QC_FAILED(N_TIME);<br>POSITION_QC_FAILED:long_name = "Number of the QC test that set the QC failed flag";<br>POSITION_QC_FAILED:_FillValue = 99999; | Number of the QC test that set the QC failed flag.<br>This information also concerns QC = 2 flags; if QC = 1, the FillValue is inserted.<br>Example: TBD. |
| AXES_ERROR_ELLIPSE_MAJOR | float AXES_ERROR_ELLIPSE_MAJOR(N_TIME);<br>AXES_ERROR_ELLIPSE_MAJOR.long_name = "Major axis of error ellipse from positioning system";<br>AXES_ERROR_ELLIPSE_MAJOR:units = "meters";<br>AXES_ERROR_ELLIPSE_MAJOR:_FillValue = 99999.; | Major axis of error ellipse reported by the positioning system. |
| AXES_ERROR_ELLIPSE_MINOR | float AXES_ERROR_ELLIPSE_MINOR(N_TIME);<br>AXES_ERROR_ELLIPSE_MINOR.long_name = "Minor axis of error ellipse from positioning system";<br>AXES_ERROR_ELLIPSE_MINOR:units = "meters";<br>AXES_ERROR_ELLIPSE_MINOR:_FillValue = 99999.; | Minor axis of error ellipse reported by the positioning system. |
| AXES_ERROR_ELLIPSE_ANGLE | float AXES_ERROR_ELLIPSE_ANGLE(N_TIME);<br>AXES_ERROR_ELLIPSE_ANGLE.long_name = "Angle of error ellipse from positioning system";<br>AXES_ERROR_ELLIPSE_ANGLE:units = "Degrees (from North when heading East)";<br>AXES_ERROR_ELLIPSE_ANGLE:_FillValue = 99999.; | Angle of error ellipse reported by the positioning system. |

| | | |
|---|---|---|
| SATELLITE_NAME | char SATELLITE_NAME(N_TIME);<br>SATELLITE_NAME.long_name = "Satellite name from positioning system";<br>SATELLITE_NAME.FillValue = " "; | Satellite name from positioning system. |
| **Measurement information** | | |
| <PARAM> | float <PARAM>(N_TIME, N_LEVEL<n>);<br><PARAM>:long_name = "<X>";<br><PARAM>:standard_name = "<X>";<br><PARAM>:units = "<X>";<br><PARAM>:valid_min = <X>;<br><PARAM>:valid_max = <X>;<br><PARAM>:C_format = "<X>";<br><PARAM>:FORTRAN_format = "<X>";<br><PARAM>:_FillValue = <X>; | <PARAM> contains the original values of a parameter listed in the "code" column of Reference Table 3.<br><X>: these fields are specified in the columns of the Reference Table 3. |
| <PARAM>_QC | byte <PARAM>_QC(N_TIME, N_LEVEL<n>);<br><PARAM>_QC:long_name = "Quality flag";<br><PARAM>_QC:conventions = "Reference table 2";<br><PARAM>_QC:flag_values = 0, 1, 2, 3, 4, 5, 8, 9;<br><PARAM>_QC:flag_meanings = "no_qc_performed good_data probably_good_data bad_data_that_are_potentially_correctable bad_data value_changed interpolated_value missing_value";<br><PARAM>_QC:_FillValue = -128; | Quality flag applied on each <PARAM> values.<br>The flag scale is specified in Reference Table 2.<br>Example: 1 (value seems correct). |
| <PARAM>_ADJUSTED | float <PARAM>_ADJUSTED(N_TIME, N_LEVEL<n>);<br><PARAM>_ADJUSTED:long_name = "<X>";<br><PARAM>_ADJUSTED:standard_name = "<X>";<br><PARAM>_ADJUSTED:units = "<X>";<br><PARAM>_ADJUSTED:valid_min = <X>;<br><PARAM>_ADJUSTED:valid_max = <X>;<br><PARAM>_ADJUSTED:comment = "<X>";<br><PARAM>_ADJUSTED:C_format = "<X>";<br><PARAM>_ADJUSTED:FORTRAN_format = "<X>";<br><PARAM>_ADJUSTED:resolution= <X>;<br><PARAM>_ADJUSTED:_FillValue = <X>; | <PARAM>_ADJUSTED contains the adjusted values derived from the original values of the parameter.<br>The list of adjustable parameters is provided in Reference Table 3.<br><X>: these fields are specified in the columns of the Reference Table 3.<br>When no adjustment is performed, the FillValue is inserted. |
| <PARAM>_ADJUSTED_QC | byte <PARAM>_ADJUSTED_QC(N_TIME, N_LEVEL<n>);<br><PARAM>_ADJUSTED_QC:long_name = "Quality flag";<br><PARAM>_ADJUSTED_QC:conventions = "Reference table 2";<br><PARAM>_ADJUSTED_QC:flag_values = 0, 1, 2, 3, 4, 5, 8, 9;<br><PARAM>_ADJUSTED_QC:flag_meanings = "no_qc_performed good_data probably_good_data bad_data_that_are_potentially_correctable bad_data value_changed interpolated_value missing_value";<br><PARAM>_ADJUSTED_QC:_FillValue = -128; | Quality flag applied on each <PARAM>_ADJUSTED values.<br>The flag scale is specified in Reference Table 2.<br>When no adjustment is performed, the FillValue is inserted.<br>Example: 1 (value seems correct). |

| | | |
|---|---|---|
| <PARAM>_ADJUSTED_ERROR | float <PARAM>_ADJUSTED_ERROR(N_TIME, N_LEVEL<n>);<br><PARAM>_ADJUSTED_ERROR:long_name = "Contains the error on the adjusted values as determined by the delayed mode QC process.";<br><PARAM>_ADJUSTED_ERROR:units = "<X>";<br><PARAM>_ADJUSTED_ERROR:C_format = "<X>";<br><PARAM>_ADJUSTED_ERROR:FORTRAN_format = "<X>";<br><PARAM>_ADJUSTED_ERROR:resolution= <X>;<br><PARAM>_ADJUSTED_ERROR:_FillValue = <X>; | <PARAM>_ADJUSTED_ERROR contains the error on the adjusted values of the parameter.<br><X>: these fields are specified in the columns of the Reference Table 3.<br>**<PARAM>_ADJUSTED_ERROR is mandatory for adjustable parameters.**<br>When no adjustment is performed, the FillValue is inserted. |
| <PARAM>_QC_FAILED | int <PARAM>_QC_FAILED(N_TIME, N_LEVEL<n>);<br><PARAM>_QC_FAILED:long_name = "Number of the QC test that set the QC failed flag";<br><PARAM>_QC_FAILED:_FillValue = 99999; | Number of the QC test that set the QC failed flag (in <PARAM>_QC or <PARAM>_ADJUSTED_QC depending of the PARAMETER_DATA_MODE).<br>This information also concerns QC = 2 flags; if QC = 1, the FillValue is inserted.<br>Example: TBD. |
| PARAMETER_DATA_MODE | char PARAMETER_DATA_MODE(N_TIME, N_PARAM);<br>PARAMETER_DATA_MODE:long_name = "Delayed mode or real time data";<br>PARAMETER_DATA_MODE:conventions = "R : real time; D : delayed mode; A : real time with adjustment";<br>PARAMETER_DATA_MODE:_FillValue = " "; | Describe the data mode of the individual parameter:<br>R: real time data<br>D: delayed mode data<br>A: real time data with adjusted values |
| CONFIG_MISSION_NUMBER_INDEX | int CONFIG_MISSION_NUMBER_INDEX(N_TIME);<br>CONFIG_MISSION_NUMBER_INDEX:long_name = "Configuration mission number that corresponds to the current index";<br>CONFIG_MISSION_NUMBER_INDEX:conventions = "1...N, 1 : first complete mission";<br>CONFIG_MISSION_NUMBER_INDEX:_FillValue = 99999; | Unique number of the mission to which this measurement belongs. |
| GROUNDED | char GROUNDED(N_TIME);<br>GROUNDED:long_name = "Did the drogue touch the ground for that location?";<br>GROUNDED:conventions = "Reference table 18";<br>GROUNDED:_FillValue = " "; | GROUNDED indicates the best estimate of whether the drogue touched the ground for that location.<br>The conventions are described in Reference Table 18.<br>Example: TBD. |
| DATA_STATUS | char DATA_STATUS(N_TIME);<br>DATA_STATUS:long_name = "Status of the data associated to the current timestamp";<br>DATA_STATUS:conventions = "Reference table 17";<br>DATA_STATUS:_FillValue = " "; | Status flag on the data associated to the current timestamp.<br>The flag scale is described in Reference Table 17.<br>Example: TBD. |

## Note on measurement positioning

Parameter measurements are transmitted with or without associated location depending on the drifter transmission system.

- Iridium drifters transmit a GPS location together with each parameter measurement. The time difference between the GPS location and the parameter measurement is provided by the GPS_FIX_TIME_DELAY (minutes) parameter.
- Argos drifters transmit the parameter measurements without associated location (except for drifters equipped with a GPS receiver). The drifter trajectory is estimated in a post-processing phase by the CLS Argos center that provide (at most) one location computed from all messages received during each satellite pass.

Consequently, in the data measurement variables of this group:

- For Iridium drifters, 3 types of information are available in the timeseries:
  1. The GPS position. It is stored with POSITION_ACCURACY = ''G''. When GPS_FIX_TIME_DELAY is present in the file and set to 0, the GPS position date is correctly determined (i.e. transmitted date has been corrected from GPS_FIX_TIME_DELAY minutes), when GPS_FIX_TIME_DELAY is present in the file and set to 4095, the transmitted date has been corrected from 4094 minutes (but the GPS position date is still not correct, that's why associated TIME_QC is set to 4 by RTQC test #22), when GPS_FIX_TIME_DELAY is not present in the file (the information was not available to us) the transmitted time is used as GPS position date.
  2. The Iridium position. It is stored with POSITION_ACCURACY = ''I''. The CEP radius (when available) is stored (in meters) in AXES_ERROR_ELLIPSE_MAJOR = AXES_ERROR_ELLIPSE_MINOR, with AXES_ERROR_ELLIPSE_ANGLE = FillValue.
  3. The measurements. Their position is determined by a linear interpolation of good GPS positions.
- For Argos drifters, each N_TIME index may contain an Argos location **or** the parameter measurements. **In this later case, we provide a location for each parameter measurements; it is computed from a linear interpolation of the ‘good’ (i.e. after RTQC tests) locations of the Argos trajectory.**

## 2.4.? Greylist information

This section explains greylist information for each QC set during the DMQC performed on the time series.

Each item of this section has a N_GREYLIST (number of greylist records) dimension.

| Name | Definition | Comment |
|---|---|---|
| GREYLIST_PARAMETER | char GREYLIST_PARAMETER(N_GREYLIST, STRING64);<br>GREYLIST_PARAMETER:long_name = "Name of the greylisted parameter";<br>GREYLIST_PARAMETER:conventions = "Reference table 3";<br>GREYLIST_PARAMETER:_FillValue = " "; | Name of the parameter concerned by the greylist information.<br>Example: "SST".<br>If parameter = "MISSION", all the parameters (geophysical and location) are affected by the given QC. |
| GREYLIST_START_DATE | char GREYLIST_START_DATE(N_GREYLIST, DATE_TIME);<br>GREYLIST_START_DATE:long_name = "Start date that concerns the greylist information";<br>GREYLIST_START_DATE:conventions = "YYYYMMDDHHMISS";<br>GREYLIST_START_DATE:_FillValue = " "; | Start date that concerns the greylist information.<br>Example: "20011217160057" for December 17th 2001 16:00:57. |
| GREYLIST_END_DATE | char GREYLIST_END_DATE(N_GREYLIST, DATE_TIME);<br>GREYLIST_END_DATE:long_name = "End date that concerns the greylist information";<br>GREYLIST_END_DATE:conventions = "YYYYMMDDHHMISS";<br>GREYLIST_END_DATE:_FillValue = " "; | End date that concerns the greylist information.<br>Example: "20011217160057" for December 17th 2001 16:00:57. |
| GREYLIST_FLAG | char GREYLIST_FLAG(N_GREYLIST);<br>GREYLIST_FLAG:long_name = "Quality flag";<br>GREYLIST_FLAG:conventions = "Reference table 2";<br>GREYLIST_FLAG:_FillValue = " "; | Quality flag applied on parameter QC values.<br>Example: '1' (value seems correct). |
| GREYLIST_COMMENT | char GREYLIST_COMMENT(N_GREYLIST, STRING256)<br>GREYLIST_COMMENT:long_name = "Comment on greylist information";<br>GREYLIST_COMMENT:_FillValue = " "; | Comment on this greylist information.<br>Example: "sensor biased". |

## 2.4.9 History information

This section contains history information for each action performed on the time-series by a data center.

Each item of this section has a N_HISTORY (number of history records) dimension.

A history record is created whenever an action is performed on a part of the time-series defined by HISTORY_START_TIME_INDEX and HISTORY_STOP_TIME_INDEX.

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

| Name | Definition | Comment |
|---|---|---|
| HISTORY_INSTITUTION | char HISTORY_INSTITUTION(N_HISTORY, STRING2);<br>HISTORY_INSTITUTION:long_name = "Institution which performed action";<br>HISTORY_INSTITUTION:conventions = "Reference table 4";<br>HISTORY_INSTITUTION:_FillValue = " "; | Institution that performed the action.<br>Institution codes are described in Reference Table 4.<br>Example: "IF" for Ifremer. |
| HISTORY_STEP | char HISTORY_STEP(N_HISTORY, STRING4);<br>HISTORY_STEP:long_name = "Step in data processing";<br>HISTORY_STEP:conventions = "Reference table 12";<br>HISTORY_STEP:_FillValue = " "; | Code of the step in data processing for this history record. The step codes are described in Reference Table 12.<br>Example: TBD. |
| HISTORY_SOFTWARE | char HISTORY_SOFTWARE(N_HISTORY, STRING8);<br>HISTORY_SOFTWARE:long_name = "Name of software which performed action";<br>HISTORY_SOFTWARE:conventions = "Institution dependent";<br>HISTORY_SOFTWARE:_FillValue = " "; | Name of the software that performed the action.<br>This code is institution dependent.<br>Example: TBD. |
| HISTORY_SOFTWARE_RELEASE | char HISTORY_SOFTWARE_RELEASE(N_HISTORY, STRING4);<br>HISTORY_SOFTWARE_RELEASE:long_name = "Version/release of software which performed action";<br>HISTORY_SOFTWARE_RELEASE:conventions = "Institution dependent";<br>HISTORY_SOFTWARE_RELEASE:_FillValue = " "; | Version of the software.<br>This name is institution dependent.<br>Example: "1.0". |
| HISTORY_REFERENCE | char HISTORY_REFERENCE(N_HISTORY, STRING64);<br>HISTORY_REFERENCE:long_name = "Reference of database";<br>HISTORY_REFERENCE:conventions = "Institution dependent";<br>HISTORY_REFERENCE:_FillValue = " "; | Code of the reference database used for quality control in conjunction with the software.<br>This code is institution dependent.<br>Example: TBD. |
| HISTORY_DATE | char HISTORY_DATE(N_HISTORY, DATE_TIME);<br>HISTORY_DATE:long_name = "Date the history record was created";<br>HISTORY_DATE:conventions = "YYYYMMDDHHMISS";<br>HISTORY_DATE:_FillValue = " "; | Date of the action.<br>Example: "20011217160057" for December 17th 2001 16:00:57. |

| | | |
|---|---|---|
| HISTORY_ACTION | char HISTORY_ACTION(N_HISTORY, STRING4);<br>HISTORY_ACTION:long_name = "Action performed on data";<br>HISTORY_ACTION:conventions = "Reference table 7";<br>HISTORY_ACTION:_FillValue = " "; | Name of the action.<br>The action codes are described in Reference Table 7.<br>Example: TBD. |
| HISTORY_PARAMETER | char HISTORY_PARAMETER(N_HISTORY, STRING64);<br>HISTORY_PARAMETER:long_name = "Parameter action is performed on";<br>HISTORY_PARAMETER:conventions = "Reference table 3";<br>HISTORY_PARAMETER:_FillValue = " "; | Name of the parameter on which the action is performed.<br>Example: "PSAL". |
| HISTORY_PREVIOUS_VALUE | float HISTORY_PREVIOUS_VALUE(N_HISTORY);<br>HISTORY_PREVIOUS_VALUE:long_name = "Parameter/Flag previous value before action";<br>HISTORY_PREVIOUS_VALUE:_FillValue = 99999.; | Parameter or flag of the previous value before action.<br>Example: "2" (probably good) for a flag that was changed to "1" (good). |
| HISTORY_START_TIME_INDEX | int HISTORY_START_TIME_INDEX(N_HISTORY);<br>HISTORY_START_TIME_INDEX:long_name = "Start time index action applied on";<br>HISTORY_START_TIME_INDEX:_FillValue = 99999; | Start time index the action is applied to.<br>Example: 100 |
| HISTORY_STOP_TIME_INDEX | int HISTORY_STOP_TIME_INDEX(N_HISTORY);<br>HISTORY_STOP_TIME_INDEX:long_name = "Stop time index action applied on";<br>HISTORY_STOP_TIME_INDEX:_FillValue = 99999; | Stop time index the action is applied to.<br>Example: 150 |
| HISTORY_QCTEST | char HISTORY_QCTEST(N_HISTORY, STRING16);<br>HISTORY_QCTEST:long_name = "Documentation of tests performed, tests failed (in hex form)";<br>HISTORY_QCTEST:conventions = "Write tests performed when ACTION=QCP$; tests failed when ACTION=QCF$";<br>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).<br>The QCTEST codes are described in Reference Table 11.<br>Example: "0A" (in hexadecimal form) |

# 3 Reference Tables

This chapter gives the Reference Tables used by the C-RAID format.

## 3.1 Reference Table 1: data types

The DATA_TYPE variable should have the following value:

| DATA_TYPE |
|---|
| C-RAID drifter time-series data |

## 3.2 Reference Table 2: variable quality control flag scale

The quality control flags indicate the data quality of the data values in a file, and are normally assigned after quality control procedures have been performed.

| Code | Meaning | Comment |
|---|---|---|
| 0 | No QC was performed | - |
| 1 | Good data | All QC tests passed. |
| 2 | Probably good data | - |
| 3 | Bad data that are potentially correctable | These data are not to be used without scientific correction or re-calibration. |
| 4 | Bad data | Data have failed one or more tests. |
| 5 | Value changed | Data may be recovered after transmission error. |
| 6 | - | Not used. |
| | | |
| 8 | Estimated value | Estimated value (interpolated, extrapolated or other estimation). |
| 9 | Missing value | - |

## 3.3 Reference Table 3: C-RAID parameter list

The managed C-RAID parameters are of two types: the “geophysical” ones and the “technical” ones.

Each parameter is measured by a sensor and original ‘raw’ sensor outputs are generally transmitted. These sensor counts are stored in a “<PARAM>_COUNT” variable. The ‘usable’ parameter value is then computed from calibration equation and stored in an associated “<PARAM>” variable.

Example: received sea surface temperature is stored in SST_COUNT and SST (in degree_Celsius) is obtained as SST = SST_COUNT * SST_SLOPE + SST_OFFSET.

### 3.3.1 Geophysical parameter list

The following table describes the geophysical parameter codes used for C-RAID data management.

| Code | long name | standard name | unit | valid_ min | valid_ max |
|---|---|---|---|---|---|
| ATMS | Air pressure | air_pressure | hPa | | |
| ATMS_COUNT | Air pressure | | count | | |
| SST | Sea surface temperature | sea_surface_temperature | degree_Celsius | | |
| SST_COUNT | Sea surface temperature | | count | | |
| | | | | | |
| AIR_TEMP | Air temperature | Air temperature | degree Celsius | | |
| | | | | | |
| ATPT | Air pressure tendency | | hPa | | |
| ATPT_COUNT | Air pressure tendency | | count | | |
| | | | | | |

### 3.3.2 Technical parameter list

The following table describes the technical parameter codes used for C-RAID data management.

| Code | long name | standard name | unit | valid_ min | valid_ max |
|---|---|---|---|---|---|
| BATTERY | Battery voltage | | Volts | | |
| BATTERY_COUNT | Battery voltage | | count | | |
| DROGUE | Drogue detection sensor | | | | |
| DROGUE_COUNT | Drogue detection sensor | | count | | |
| MSG_REDUNDANCY | Redundancy of received message | | count | | |
| | | | | | |

## 3.4 Reference Table 4: DAC and institution codes

| Code | Data Assembly Centers and institutions |
|---|---|
| AO | AOML, USA |
| IF | Ifremer, France |

## 3.5 Reference Table 5: location classes

| Value | Estimated accuracy in latitude and longitude |
|---|---|
| 0 | Argos accuracy estimation over 1500m radius |
| 1 | Argos accuracy estimation better than 1500m radius |
| 2 | Argos accuracy estimation better than 500 m radius |
| 3 | Argos accuracy estimation better than 250 m radius |
| A | Argos no accuracy estimation (3 messages received) |
| B | Argos no accuracy estimation (1 or 2 messages received) |
| Z | Argos invalid location |
| G | GPS positioning accuracy (better than 10 m) |
| I | Iridium accuracy (better than 5 km) |
| D | Beidou accuracy (better than 10 m) |

| | |
|---|---|
| E | GLONASS accuracy (better than 10 m) |
| F | GALILEO accuracy (better than 10 m) |
| H | GNSS accuracy (better than 10 m) |
| U | Estimated position. An estimated accuracy may be in AXES_ERROR_ELLIPSE_ |

## 3.6 Reference Table 6: data state indicators

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

| Class | Descriptor | Subclass |
|---|---|---|
| A | No scrutiny, value judgements or intercomparisons are performed on the data. The records are derived directly from the input with no filtering, or subsampling. | - Some reductions or subsampling has been performed, but the original record is available.<br><br>+ Geospatial and temporal properties are checked. Geophysical values are validated. If not validated, this is clearly indicated. |
| B | Data have been scrutinized and evaluated against a defined and documented set of measures. The process is often automated (i.e. has no human intervention) and the measures are published and widely available. | - Measures are completely automated, or documentation is not widely available.<br><br>+ The measures have been tested on independent data sets for completeness and robustness and are widely accepted. |
| C | Data have been scrutinized fully including intra-record and intra-dataset comparison and consistency checks. Scientists have been involved in the evaluation and brought latest knowledge to bear. The procedures are published, widely available and widely accepted. | - Procedures are not published or widely available. Procedures have not undergone full scrutiny and testing.<br><br>+ Data are fully quality controlled, peer reviewed and are widely accepted as valid. Documentation is complete and widely available. |

### Data state indicator recommended use

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

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

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

~~Notes~~

1. ~~We need to have a pragmatic approach to what constitutes "original" or "raw" data. Despite the fact that an instrument may be capable of high sampling rates, what is reported from the instrument defines what is considered "raw". For example, Argo floats can certainly sample at finer scales than every 10 db, but because of communications, all we see for now is data at that (or worse) vertical resolution. Therefore the data "coming from the instrument" is "raw" output at 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.~~

~~The transition from class 2 to 3 occurs when assumptions of scales of variability are applied. During the course of normal data processing it is common to carry out some averaging and subsampling. This is usually done to exploit oversampling by the instrument, and to ensure good measurements are achieved. These are considered to be part of the geospatial and temporal referencing process.~~

## 3.7 Reference Table 7: history action codes

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

## 3.8 Reference Table 8: instrument types

The instrument type codes come from WMO table 1770.

| Code number | Instrument |
|---|---|
| ~~831~~ | ~~P-Alace float~~ |
| ~~837~~ | ~~Arvor-C float~~ |
| ~~838~~ | ~~Arvor-D float~~ |
| ~~839~~ | ~~Provor-II float~~ |
| ~~840~~ | ~~Provor, no conductivity~~ |
| ~~841~~ | ~~Provor, Seabird conductivity sensor~~ |
| ~~842~~ | ~~Provor, FSI conductivity sensor~~ |
| ~~843~~ | ~~POPS ice Buoy/Float~~ |
| ~~844~~ | ~~Arvor, Seabird conductivity sensor~~ |
| ~~845~~ | ~~Webb Research, no conductivity~~ |
| ~~846~~ | ~~Webb Research, Seabird sensor~~ |
| ~~847~~ | ~~Webb Research, FSI sensor~~ |
| ~~848~~ | ~~Apex-EM float~~ |
| ~~849~~ | ~~Apex-D deep float~~ |
| ~~850~~ | ~~Solo, no conductivity~~ |
| ~~851~~ | ~~Solo, Seabird conductivity sensor~~ |
| ~~852~~ | ~~Solo, FSI conductivity sensor~~ |
| ~~853~~ | ~~Solo2, Seabird conductivity sensor~~ |
| ~~854~~ | ~~S2A float~~ |
| ~~855~~ | ~~Ninja, no conductivity sensor~~ |
| ~~856~~ | ~~Ninja, SBE conductivity sensor~~ |
| ~~857~~ | ~~Ninja, FSI conductivity sensor~~ |
| ~~858~~ | ~~Ninja, TSK conductivity sensor~~ |
| ~~859~~ | ~~Profiling Float, NEMO, no conductivity~~ |
| ~~860~~ | ~~Profiling Float, NEMO, SBE conductivity sensor~~ |

| | |
|---|---|
| 861 | Profiling Float, NEMO, FSI conductivity sensor |
| 862 | Solo-D deep float |
| 863 | Navis-A Float |
| 864 | Ninja-D deep float |
| 865 | Nova float |

## 3.9 Reference Table 9: positioning systems

| Code | Description |
|---|---|
| ARGOS | ARGOS positioning system |
| GPS | GPS positioning system |
| IRIDIUM | Iridium positioning system |
| BEIDOU | Beidou navigation satellite system |
| GLONASS | GLONASS navigation satellite system |
| GALILEO | Galileo navigation satellite system |
| GNSS | Global Navigation Satellite System |

## 3.10 Reference Table 10: transmission systems

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

## 3.11 Reference Table 11: QC test binary IDs

This table is used to record the result of the quality control tests in the history section.
The binary IDs of the QC tests are used to define the history variable HISTORY_QCTEST, whose value is computed by adding the binary ID together, then translating to a hexadecimal number.
The test numbers and the test names are listed in the C-RAID Quality Control Manual.

| Test number | QC test binary ID | Test name |
|---|---|---|
| 1 | 2 | Platform Identification test |
| 2 | 4 | Impossible Date test |
| 3 | 8 | Impossible Location test |
| 4 | 16 | Position on Land test |
| 5 | 32 | Impossible Speed test |
| 6 | 64 | Global Range test |

| | | |
|---|---|---|
| ~~7~~ | ~~128~~ | ~~Regional Global Parameter test~~ |
| ~~8~~ | ~~256~~ | ~~Pressure Increasing test~~ |
| ~~9~~ | ~~512~~ | ~~Spike test~~ |
| ~~*10*~~ | ~~*1024*~~ | ~~*Top and Bottom Spike test (obsolete)*~~ |
| ~~11~~ | ~~2048~~ | ~~Gradient test~~ |
| ~~12~~ | ~~4096~~ | ~~Digit Rollover test~~ |
| ~~13~~ | ~~8192~~ | ~~Stuck Value test~~ |
| ~~14~~ | ~~16384~~ | ~~Density Inversion test~~ |
| ~~15~~ | ~~32768~~ | ~~Grey List test~~ |
| ~~16~~ | ~~65536~~ | ~~Gross Salinity or Temperature Sensor Drift test~~ |
| ~~17~~ | ~~131072~~ | ~~Visual QC test~~ |
| ~~18~~ | ~~261144~~ | ~~Frozen profile test~~ |
| ~~19~~ | ~~524288~~ | ~~Deepest pressure test~~ |
| ~~20~~ | ~~1044576~~ | ~~Questionable Argos position test~~ |

## 3.12 Reference Table 12: history steps codes

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

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

## 3.13 Reference Table 13: meta-data configuration parameter names

TBD.

## 3.14 Reference Table 14: data modes

The values for the global attribute "data_mode" is defined as follows:

| Value | Meaning |
|---|---|
| R | Real-time data. Data coming from the (typically remote) platform through a communication channel without physical access to the instruments, disassembly or recovery of the platform. Example: for a glider with a radio communication, this would be data obtained through the radio. |
| A | Rea-time adjusted data. Real-time or provisional data that have been adjusted by real-time automatic procedures. |
| D | Delayed-mode data. Data published after all calibrations and quality control procedures have been applied on the internally recorded or best available original data. This is the best possible version of processed data. |
| M | Mixed. This value indicates that the file contains data in more than one of the above states. |

## 3.15 Reference Table 15: JULD_STATUS flags

| Flag | Meaning |
|---|---|
| 0 | Value is determined by satellite |
| 1 | Value is transmitted by the float |

## 3.16 Reference Table 16: END_MISSION_STATUS flags

| Flag | Meaning |
|---|---|
| 0 | Drifter still reporting as of last update |
| 1 | Drifter ran aground |
| 2 | Drifter was picked up |
| 3 | Drifter quit transmitting |
| 4 | Unreliable transmissions at end of trajectory |
| 5 | Bad battery voltage |
| 6 | Place in inactive status while transmitting good position |

## 3.17 Reference Table 17: DATA_STATUS flags

| Flag | Meaning |
|---|---|
| 0 | Data have been generated from original (Argos or Iridium) data transmitted by the drifter. |
| 1 | Data come from already decoded data provided by drifter institution or PI. |
| 2 | Data have been generated from original (Argos or Iridium) data transmitted by the drifter and some minor information have been recovered from already decoded data provided by drifter institution or PI. |

## 3.18 Reference Table 18: GROUNDED flags

| Flag | Meaning |
|---|---|
| Y | Yes, the drogue touched the ground |
| B | Yes, the drogue touched the ground after bathymetry check with an outside database |
| N | No, the drogue did not touch the ground |
| U | Unknown |

## 3.19 Reference Table 19: PLATFORM_FAMILY

| Code | Description |
|---|---|
| DRIFTING_BUOY | Drifting Buoy |

## 3.20 Reference Table 20: DRIFTER_TYPE

| DRIFTER_TYPE | PLAFTORM_TYPE_KEY | IXIXIX (1770) | Manufacturer | Description |
|---|---|---|---|---|
| UNKNOWN | | | | To use when necessary |
| | | | | |
| SVP | | | | Standard Argos drifter carrying only SST, drogue and voltage sensors |
| SVP-no SST | | | | SVP without SST sensor |
| SVPB | | | | SVP with barometer |
| SVPBS | | | | SVP with barometer and salinity sensors |
| SVPBW | | | | SVP with barometer and wind sensors |
| SVPG | | | | SVP with GPS |
| SVPGB | | | | SVP with GPS and barometer |
| SVPGBW | | | | SVP with GPS, barometer and wind sensor |
| SVPGS | | | | SVP with GPS and salinity sensor |
| SVPC | | | | SVP with conductivity sensor |
| SVPO | | | | SVP with optical sensor |
| SVPS | | | | SVP with salinity sensor |
| SVPW | | | | SVP with wind sensor |
| SVPV | | | | SVP with wave sensor |
| SVPBV | | | | SVP with barometer and wave sensor |
| | | | | |
| SVPI | | | | Standard Iridium drifter with GPS receiver carrying only SST, drogue and voltage sensors |
| SVPIB | | | | SVPI with barometer |

| | | | | |
|---|---|---|---|---|
| SVPIBS | | | | SVPI with barometer and salinity sensor |
| SVPIBHI | | | | SVPI with barometer and hull humidity sensor |
| SVPIBWHI | | | | SVPI with barometer, wind and hull humidity sensor |
| SVPIHI | | | | SVPI with hull humidity sensor |
| SVPISHI | | | | SVPI with salinity and hull humidity sensor |
| SVPIVHI | | | | SVPI with wave and hull humidity sensor |

## 3.21 Reference Table 21: DRIFTER_MAKER

| DRIFTER_MAKER | Description |
|---|---|
| UNKNOWN | To use when necessary |
| | |
| AANDERAA | |
| AOML | |
| BLANK | |
| CICESE | |
| CLEAWATER | |
| DATONG | DATONG ELECTRONICS LIMITED |
| DBI | |
| DRAPER_LAB | |
| DTC | DOMO TACTICAL COMMUNICATIONS |
| ELTA | |
| GDC | |
| IESM | |
| INPE | |
| JVC_KENWOOD | JVC KENWOOD CORPORATION - KANAGAWA |
| KORDI | |
| MARLIN_YUG | |
| METOCEAN | METOCEAN DATA SYSTEMS LTD. |
| MOONRAKER | |

| | |
|---|---|
| NEURON_ELECTRONICS | |
| NIO_RA | |
| NKE | NKE INSTRUMENTATION |
| OROLIA | OROLIA SAS |
| PACIFIC_GYRE | |
| POLAR_RESEARCH_LAB | |
| SCRIPPS_UCSD | SCRIPPS INSTITUTION OF OCEANOGRAPHY/UCSD |
| SEIMAC | |
| SERCEL | SERCEL BREST |
| SIO | |
| TAIWAN | |
| TECHNOCEAN | |
| TELONICS | |
| TOYOCOM | |
| TURO_TECHNOLOGY | |
| TWR | TELEDYNE WEBB RESEARCH |

## 3.22 Reference Table 22: SENSOR

| SENSOR | Comment |
|---|---|
| ~~UNKNOWN~~ | ~~to use when necessary~~ |
| | |
| ~~ACOUSTIC~~ | |
| ~~ACOUSTIC_GEOLOCATION~~ | |
| ~~CTD_PRES~~ | |
| ~~CTD_TEMP~~ | |
| ~~CTD_CNDC~~ | |
| ~~EM~~ | ~~The EM sensor measures U, V, and turbulence.~~ |
| ~~FLUOROMETER_CDOM~~ | |
| ~~FLUOROMETER_CHLA~~ | |
| ~~IDO_DOXY~~ | |
| ~~OPTODE_DOXY~~ | |

| | |
|---|---|
| RADIOMETER_DOWN_IRR<nnn> | Radiometer measuring downwelling irradiance at wavelength <nnn>. |
| RADIOMETER_PAR | |
| RADIOMETER_UP_RAD<nnn> | Radiometer measuring upwelling radiance at wavelength <nnn>. |
| BACKSCATTERINGMETER_BBP<nnn> | Backscattering meter measuring backscattering at wavelength <nnn>. |
| BACKSCATTERINGMETER_TURBIDITY | |
| SPECTROPHOTOMETER_NITRATE | |
| SPECTROPHOTOMETER_BISULFIDE | |
| STS_CNDC | |
| STS_TEMP | |
| TRANSISTOR_PH | |
| TRANSMISSOMETER_CP<nnn> | Transmissometer measuring attenuation at wavelength <nnn> |
| FLOATCLOCK_MTIME | |

## 3.23 Reference Table 23: SENSOR_MAKER

| SENSOR_MAKER | Description | Comment |
|---|---|---|
| UNKNOWN | Unknown sensor maker | The sensor maker should not be unknown, the use of this value should be exceptional |
| | | |
| AANDERAA | | |
| AMETEK | | |
| DRUCK | | |
| FSI | | |
| KISTLER | | |
| PAINE | | |
| SBE | | |
| SEASCAN | | |
| WETLABS | Wetlabs Inc. | |
| MBARI | Monterey Bay Aquarium Research Institute | |
| SATLANTIC | | |
| JAC | JFE Advantech Co., Ltd | |
| APL_UW | University of Washington, Applied Physics Laboratory | |

| | | |
|---|---|---|
| TSK | Tsurumi-Seiki Co., Ltd | |
| RBR | RBR Ltd | |
| KELLER | | |
| MICRON | | |
| SEAPOINT | | |
| TURNER_DESIGN | | |

## 3.24 Reference Table 24: SENSOR_MODEL

| SENSOR_MODEL | Comment | Associated SENSOR |
|---|---|---|
| UNKNOWN | to use when necessary | UNKNOWN |
| **Conductivity/temperature sensors** | | |
| FSI | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE37 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41_V2.5 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41_V2.6 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41_V3 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41_IDO_V1.0c | Seabird CTD module with IDO oxygen sensor | CTD_PRES, CTD_TEMP, CTD_CNDC,IDO_DOXY |
| SBE41_IDO_V2.0 | Seabird CTD module with IDO oxygen sensor | CTD_PRES, CTD_TEMP, CTD_CNDC,IDO_DOXY |
| SBE41CP_IDO_V2.0b | Seabird CTD module with IDO oxygen sensor | CTD_PRES, CTD_TEMP, CTD_CNDC,IDO_DOXY |
| SBE41_IDO_V3.0 | Seabird CTD module with IDO oxygen sensor | CTD_PRES, CTD_TEMP, CTD_CNDC,IDO_DOXY |
| SBE41CP_V1 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.1 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.2 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.2a | | CTD_PRES, CTD_TEMP, CTD_CNDC |

| | | |
|---|---|---|
| SBE41CP_V1.3 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.3b | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.4 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.5 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.7 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.8 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.9 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V1.9a | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V2 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V3 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V3.0a | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V3.0c | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V4.4.0 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V5.0.1 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V5.3.0 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V7.2.3 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41CP_V7.2.5 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41N | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41N_V5.3.0 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE41N_V5.4.0 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE61_V4.5.2 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE61_V4.5.3 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE61_V5.0.0 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE61_V5.0.1 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| SBE61 | | CTD_PRES, CTD_TEMP, CTD_CNDC |
| CTD_F01 | TSK model | CTD_PRES, CTD_TEMP, CTD_CNDC |
| RBR | RBR model | CTD_PRES, CTD_TEMP, CTD_CNDC |
| RBRoem_V1.16 | RBR model | CTD_PRES, CTD_TEMP, CTD_CNDC |
| **Oxygen sensors** | | |
| SBE43_IDO | Seabird Electrochemical Dissolved Oxygen IDO sensor (volt output) | IDO_DOXY |

| | | |
|---|---|---|
| ~~SBE43I~~ | ~~configuration option~~ | ~~IDO_DOXY~~ |
| ~~SBE43F_IDO~~ | ~~Seabird Electrochemical Dissolved Oxygen IDO sensor (frequency output)~~ | ~~IDO_DOXY~~ |
| ~~SBE63_OPTODE~~ | ~~Seabird Optical Dissolved Oxygen Sensor~~ | ~~OPTODE_DOXY~~ |
| ~~AANDERAA_OPTODE~~ | | ~~OPTODE_DOXY~~ |
| ~~AANDERAA_OPTODE_3830~~ | | ~~OPTODE_DOXY~~ |
| ~~AANDERAA_OPTODE_3835~~ | | ~~OPTODE_DOXY~~ |
| ~~AANDERAA_OPTODE_3930~~ | | ~~OPTODE_DOXY~~ |
| ~~AANDERAA_OPTODE_4330~~ | | ~~OPTODE_DOXY~~ |
| ~~AANDERAA_OPTODE_4330F~~ | | ~~OPTODE_DOXY~~ |
| ~~AANDERAA_OPTODE_4831~~ | ~~Similar to AANDERAA_OPTODE_4330 for DO processing~~ | ~~OPTODE_DOXY~~ |
| ~~AANDERAA_OPTODE_4831F~~ | ~~Similar to AANDERAA_OPTODE_4330 for DO processing~~ | ~~OPTODE_DOXY~~ |
| ~~ARO_FT~~ | ~~JAC RINKO~~ | ~~OPTODE_DOXY~~ |
| ~~AROD_FT~~ | ~~deep housing~~ | ~~OPTODE_DOXY~~ |
| **~~Pressure sensors~~** | | |
| ~~DRUCK_2900PSIA~~ | | ~~CTD_PRES~~ |
| ~~DRUCK~~ | | ~~CTD_PRES~~ |
| ~~DRUCK_10153PSIA~~ | | ~~CTD_PRES~~ |
| ~~PAINE~~ | | ~~CTD_PRES~~ |
| ~~PAINE_1500PSIA~~ | | ~~CTD_PRES~~ |
| ~~PAINE_1600PSIA~~ | | ~~CTD_PRES~~ |
| ~~PAINE_2000PSIA~~ | | ~~CTD_PRES~~ |
| ~~PAINE_2900PSIA~~ | | ~~CTD_PRES~~ |
| ~~PAINE_3000PSIA~~ | | ~~CTD_PRES~~ |
| ~~AMETEK~~ | | ~~CTD_PRES~~ |
| ~~AMETEK_3000PSIA~~ | | ~~CTD_PRES~~ |
| ~~KISTLER~~ | | ~~CTD_PRES~~ |
| ~~KISTLER_2900PSIA~~ | | ~~CTD_PRES~~ |

| | | |
|---|---|---|
| KISTLER_10153PSIA | | CTD_PRES |
| KELLER_PA8 | | CTD_PRES |
| SEASCAN_SSTD | | CTD_PRES |
| MP40_C_2000_G | MICRON | CTD_PRES |
| **Near Surface conductivity and temperature sensors** | | |
| SBE_STS | SBE Near Surface Conductivity and Temperature Module | STS_CNDC, STS_TEMP |
| **Biogeochemical sensors (*)** | | |
| ***Spectrophotometers*** | | |
| SUNA | UV absorption to derive nitrate and bisulfide (MBARI) | SPECTROPHOTOMETER_NITRATE, SPECTROPHOTOMETER_BISULFIDE |
| SUNA_V2 | UV absorption to derive nitrate and bisulfide (SATLANTIC) | SPECTROPHOTOMETER_NITRATE, SPECTROPHOTOMETER_BISULFIDE |
| ISUS | Nitrate (MBARI) | SPECTROPHOTOMETER_NITRATE |
| ISUS_V3 | Nitrate (SATLANTIC) | SPECTROPHOTOMETER_NITRATE |
| ***Transmissometers*** | | |
| C_ROVER | Transmissometer (WETLABS) | TRANSMISSOMETER_CP<nnn> |
| ***pH Sensors*** | | |
| DURA | pH (MBARI) | TRANSISTOR_PH |
| SEAFET | pH (SEABIRD) | TRANSISTOR_PH |
| ***Radiometers*** | | |
| SATLANTIC_OCR504_ICSW | Multispectral radiometer 4 channels (SATLANTIC) with cosine detector to measure irradiance in water (at wavelength <nnn>) | RADIOMETER_DOWN_IRR<nnn>, RADIOMETER_PAR |
| SATLANTIC_OCR504_R10W | Multispectral radiometer 4 channels (SATLANTIC) with a 10° half angle field of view to measure radiance in water (at wavelength <nnn>) | RADIOMETER_UP_RAD<nnn> |

| | | |
|---|---|---|
| ~~SATLANTIC_OCR507_ICSW~~ | ~~Multispectral radiometer 7 channels (SATLANTIC) with cosine detector to measure irradiance in water (at wavelength <nnn>)~~ | ~~RADIOMETER_DOWN_IRR<nnn>, RADIOMETER_PAR~~ |
| ~~SATLANTIC_OCR507_R10W~~ | ~~Multispectral radiometer 7 channels (SATLANTIC) with a 10° half-angle field of view to measure radiance in water (at wavelength <nnn>)~~ | ~~RADIOMETER_UP_RAD<nnn>~~ |
| ~~SATLANTIC_OCR507_ICSWR10W~~ | ~~Multispectral radiometer 7 channels (SATLANTIC) combination of cosine detector to measure irradiance and a 10° half-angle field of view to measure radiance in water (at wavelength <nnn>)~~ | ~~RADIOMETER_DOWN_IRR<nnn> , RADIOMETER_PAR, RADIOMETER_UP_RAD<nnn>~~ |
| ~~SATLANTIC_PAR~~ | | ~~RADIOMETER_PAR~~ |
| ~~***Backscatteringmeters and Fluorometers combination***~~ | | |
| ~~ECO_BB~~ | ~~Wetlabs Eco optical sensor packages with one backscattering meter (at wavelength <nnn>)~~ | ~~BACKSCATTERINGMETER_BBP<nnn>~~ |
| ~~ECO_FL~~ | ~~Wetlabs Eco optical sensor packages with one fluorometer (type specified in SENSOR)~~ | ~~FLUOROMETER_CHLA~~ |
| ~~ECO_NTU~~ | ~~Wetlabs Eco optical sensor packages with one backscattering meter measuring turbidity~~ | ~~BACKSCATTERINGMETER_TURBIDITY~~ |
| ~~ECO_FLBB~~ | ~~Wetlabs Eco optical sensor packages with one fluorometer (type specified in SENSOR) and one backscattering meter (at wavelength <nnn>)~~ | ~~FLUOROMETER_CHLA, BACKSCATTERINGMETER_BBP<nnn>~~ |

| | | |
|---|---|---|
| ~~ECO_FLBB_AP2~~ | ~~Wetlabs Eco optical sensor packages with one fluorometer (type specified in SENSOR) and one backscattering meter (wavelength specified in SENSOR) mounted on an Apex float~~ | ~~FLUOROMETER_CHLA, BACKSCATTERINGMETER_BBP<nnn>~~ |
| ~~ECO_FLBB_2K~~ | ~~Wetlabs Eco optical sensor packages with one fluorometer (type specified in SENSOR) and one backscattering meter (at wavelength <nnn>) certified for applications down to 2000m~~ | ~~FLUOROMETER_CHLA, BACKSCATTERINGMETER_BBP<nnn>~~ |
| ~~ECO_FLNTU~~ | ~~Wetlabs Eco optical sensor packages with one fluorometer (type specified in SENSOR) and one backscattering meter measuring turbidity~~ | ~~FLUOROMETER_CHLA, BACKSCATTERINGMETER_TURBIDITY~~ |
| ~~ECO_BB2~~ | ~~Wetlabs Eco optical sensor packages with two backscatteringmeters (at wavelength <nnn>)~~ | ~~BACKSCATTERINGMETER_BBP<nnn>~~ |
| ~~ECO_FLBBCD~~ | ~~Wetlabs Eco optical sensor packages with two fluorometers (type specified in SENSOR) and one backscatteringmeter (at wavelength <nnn>)~~ | ~~FLUOROMETER_CHLA, FLUOROMETER_CDOM, BACKSCATTERINGMETER_BBP<nnn>~~ |
| ~~ECO_FLBB2~~ | ~~Wetlabs Eco optical sensor packages with one fluorometer (type specified in SENSOR) and two backscatteringmeters (at wavelength <nnn>)~~ | ~~FLUOROMETER_CHLA, BACKSCATTERINGMETER_BBP<nnn>~~ |
| ~~ECO_BB3~~ | ~~Wetlabs Eco optical sensor packages with 3 backscatteringmeters (at wavelength <nnn>)~~ | ~~BACKSCATTERINGMETER_BBP<nnn>~~ |

| | | |
|---|---|---|
| ~~MCOMS_FLBBCD~~ | ~~Wetlabs MCOMS optical sensor packages with two fluorometers CHLA, CDOM and one backscatteringmeter (at wavelength <nnn>)~~ | ~~FLUOROMETER_CHLA, FLUOROMETER_CDOM, BACKSCATTERINGMETER_BBP<nnn>~~ |
| ~~MCOMS_FLBB2~~ | ~~Wetlabs MCOMS optical sensor packages with one fluorometer CHLA and two backscatteringmeters (wavelengths specified in SENSOR)~~ | ~~FLUOROMETER_CHLA, BACKSCATTERINGMETER_BBP<nnn>~~ |
| ~~CYCLOPS_7_FLUOROMETER~~ | ~~Turner designs Optical sensor~~ | ~~FLUOROMETER_CHLA, BACKSCATTERINGMETER_TURBIDITY~~ |
| ~~SEAPOINT_TURBIDITY_METER~~ | ~~Seapoint turbidity meter~~ | ~~BACKSCATTERINGMETER_TURBIDITY~~ |
| **~~Other sensors~~** | | |
| ~~RAFOS~~ | ~~Receiver mounted on some floats for geopositioning under ice using RAFOS sound sources in the array in the Weddell Sea (Location derivation is done in Delayed Mode for under ice floats equipped with RAFOS receivers).~~ | ~~ACOUSTIC_GEOLOCATION~~ |
| ~~PAL_UW~~ | ~~UW Passive acoustic listener~~ | ~~ACOUSTIC~~ |
| ~~EM~~ | ~~Electromagnetic sensor package to measure velocity~~ | ~~EM~~ |
| ~~FLOATCLOCK~~ | | ~~FLOATCLOCK_MTIME~~ |

## 3.25 Reference Table 25: DROGUE_DETECTION_METHOD

| DROGUE_DETECTION_METHOD | Description |
|---|---|
| UNKNOWN | To use when necessary |
| | |
| NONE | |
| SUBMERGENCE_SENSOR | |
| STRAIN_GAUGE | |