Argo data management

DOI: http://dx.doi.org/xxx

Description of the Argo GDAC File Checks: Data Format and Consistency Checks

Version 2.5.2

October 2017





Argo data management

Description of the Argo GDAC File Checks

Authors: Mark Ignaszewski / FNMOC

How to cite this document

Mark Ignaszewski / FNMOC, **Description of the** **Argo GDAC File Checks: Data Format and Consistency Checks**. <http://dx.doi.org/xxx>

Table of contents

[Table of contents 3](#_Toc461544201)

[History of the document 5](#_Toc461544202)

[1 Introduction 6](#_Toc461544203)

[1.1 Changes from last revision 6](#_Toc461544204)

[1.2 Items to Note/Discuss 6](#_Toc461544205)

[1.3 Proposed Changes 6](#_Toc461544206)

[2 GDAC File Name Validation 6](#_Toc461544207)

[3 Format Checks 7](#_Toc461544208)

[3.1 Physical Parameter Variables 7](#_Toc461544209)

[3.2 Exceptions 8](#_Toc461544210)

[4 Meta-data Data Consistency Checks 8](#_Toc461544211)

[4.1 Reasonable Date Checks 9](#_Toc461544212)

[4.2 Highly-desirable Parameter Checks (file-version 2.3 and earlier) 9](#_Toc461544216)

[4.3 Mandatory Parameter Checks (file version 2.4 and later) 9](#_Toc461544217)

[4.4 Configuration Parameter Name Checks 10](#_Toc461544218)

[5 Profile Data Checks 10](#_Toc461544219)

[5.1 Meta-data (within a profile file) Checks 10](#_Toc461544220)

[5.2 Reasonable Date Checks 11](#_Toc461544221)

[5.3 Physical Parameter Checks 11](#_Toc461544222)

[5.4 Additional D-mode File Checks 13](#_Toc461544223)

[6 Technical Data Checks 14](#_Toc461544224)

[6.1 Meta-data Checks (within technical file) 14](#_Toc461544225)

[6.2 Reasonable Date Checks 14](#_Toc461544226)

[6.3 Technical Parameter Names (format-version 2.4 and later) 14](#_Toc461544230)

[6.4 \*\*\*FOR FUTURE IMPLEMENTATION\*\*\* Technical Parameter Values 15](#_Toc461544231)

[7 Trajectory Data Checks 15](#_Toc461544232)

[7.1 Meta-data Checks 15](#_Toc461544233)

[7.2 Reasonable Date Checks 16](#_Toc461544234)

[7.3 N\_MEASUREMENT Variable Group Checks 16](#_Toc461544235)

[7.4 N\_CYCLE variable group checks 20](#_Toc461544236)

[7.5 N\_MEASUREMENT / N\_CYCLE Inter-dependence 21](#_Toc461544237)

[Appendix A. Allowed exceptions 23](#_Toc461544239)

[DATA\_TYPE settings: 23](#_Toc461544240)

[Pre-v3.1 Files 23](#_Toc461544241)

[V3.1 Files 23](#_Toc461544242)

History of the document

|  |  |  |
| --- | --- | --- |
| Version | Date | Comment |
| 1.0 | November 2014 | Original version sent around for comment |
| 1.1 | March 2015 | **Update D-mode checks when all values for a parameter are missing.**  **Allow <\*> in config and tech templates to match “empty”** |
| 1.3 | October 2015 | Update for profile checks (including bio-profile checks)  Add “N\_VALUES” description  Add naming rules for “\*2” variables  Add proposed technical parameter value checks.  Add proposed trajectory file consistency checks.  Put document in standard Argo format. |
| 1.4 | December 2015 | Incorporate comments from reviewers and ADMT-16 decisions, including…  Bio-profile, intermediate parameter: <PARAM> cannot exist by itself (PROF\_\*\_QC, and \*\_QC) must always be present.  Bio-profile data checks are accepted (not proposed).  DATA\_STATE\_INDICATOR checks are accepted (not proposed). |
| 2.4 | April 2016 | File name validation has been added to the FileChecker.  Add check of units for CONFIGURATION\_PARAMETER\_NAME settings.  Add check that <PARAM>\_ADJUSTED and <PARAM>\_ADJUSTED\_ERROR are set at me levels for D-mode parameters.  Allow <PARAM>\_ADJUSTED\_ERROR to be set for A-mode bio-parameters  Remove comparison of DATE\_UPDATE to GDAC receipt time in all files.  Allow JULD and JULD\_LOCATION to differ by 2 days (instead of 1)  LAUNCH\_DATE mandatory in meta-data files.  Trajectory files: Meta-data, Reasonable Date, and TRAJECTORY\_PARAMETERS checks have been implemented.  Adjusted version number of the document to match the software version number. |
| 2.4.1 | May 2016 | Add LAUNCH\_CONFIG\_PARAMETER\_NAME to meta-data data checks. |
| 2.4.2 | June 2016 | Add the capability to accept variables with “alternate dimensions”: variables that can have different dimensions depending on need. |
| 2.5.0 | October 2016 | Transition “v2.,5” code to production   * Add initial version of full trajectory data consistency checks (turned off in the production version. On in the “TEST” version. |
| 2.5.1 | January 2017 | Allow the “<\*>” in all “string-based” reference tables  Handle “PRES<n>” the same as PRES in all tests. |
| 2.5.2 | October 2017 | Update CONFIG\_MISSION\_NUMBER checks  Revisions to trajectory file data consistency checks  Add handling of “status” for google-doc reference tables. |

# Introduction

Every Argo data file submitted by a DAC for distribution on the GDAC has its format and data consistency checked by the Argo FileChecker. Two types of checks are applied:

1. Format checks. Ensures the file formats match the Argo standards precisely.
2. Data consistency checks.

Additional data consistency checks are performed on a file after it **passes** the format checks. These checks do not duplicate any of the quality control checks performed elsewhere. These checks can be thought of as “sanity checks” to ensure that the data are consistent with each other.

The data consistency checks enforce data standards and ensure that certain data values are reasonable and/or consistent with other information in the files. Examples of the “data standard” checks are the “mandatory parameters” defined for meta-data files and the technical parameter names in technical data files.

Files with format or consistency errors are rejected by the GDAC and are not distributed. Less serious problems will generate warnings and the file will still be distributed on the GDAC.

## Reference Tables and Data Standards:

Many of the consistency checks involve comparing the data to the published reference tables and data standards. These tables are documented in the User’s Manual. (The FileChecker implements “text versions” of these tables.)

Most the reference tables allow for a “Status” to be defined for each entry. This capability is still evolving but the FileChecker has already implemented some of the features as discussed at ADMT-17:

|  |  |
| --- | --- |
| Status Value | FileChecker Response |
| active  (or blank) | Valid, accepted value. |
| approved |
| deprecated | Previously “active” value. Still allowed while DACs transition.  WARNING message generated.  Will become obsolete at a future time. |
| publication unmderway | Values not yet “active”. REJECTED by FileChecker.  Rejection message will indicate the Status Value. |
| creation underway |
| obsolete | Invalid value. REJECTED by FileChecker. |

## Changes from last revision

Changes to the tests from the last revision will be highlighted as shown: Changes to tests. Editorial changes are not highlighted.

## Items to Note/Discuss

Items that should be discussed (in my opinion) have been highlighted as shown: Items to be noted and/or discussed.

## Proposed Changes

There are proposed changes included in this document. These are highlighted with as shown: Proposed changes.

# GDAC File Name Validation

The FileChecker validates that the name of the file submitted to the GDAC conforms to the standard documented in the Users Manual based on the data contained within the data file. If the submitted file name does not match the expected file name, the file is rejected. (In practice, this check is performed after the format and data checks documented below.)

# Format Checks

The FileChecker compares the netCDF file structure of submitted files with the format specification documented in the Argo Data Management User’s Manual. The format checking process is very strict about compliance with the documented specification.

NOTE: “Format Version” versus “Manual Version”.

The term “format-version” refers specifically to the FORMAT\_VERSION documented for each file type in the User’s Manual. When referring to a specific version of the User’s Manual the term “manual-version” will be used.

Multiple format-versions of a given file type (meta-data, profile, etc) are accepted by the FileChecker. This is done to provide DACs sufficient time to transition from one format-version to another. The ADMT will determine when an older format-version should no longer be accepted and the FileChecker will be re-configured to reject any subsequent files received in that version.

The format checking process is the same for each type of data file. The process compares:

1. Global attributes
2. Dimensions: The dimensions specified in the User’s Manual, and only those dimensions, must be present in the data file.

For dimensions that represent constants, such as the STRING\* dimensions, the values of the dimensions are also checked.

In the case of the N\_VALUES*n* dimensions allowed in the bio-argo files, any numeric digits will be accepted for “*n*” and the value will be checked to ensure it matches “*n*”.

A missing dimension, an extra dimension, or a dimension with an incorrect value will cause a file to be rejected.

1. Variables: The variables specified in the User’s Manual, and only those variables, must be present in the data file. Additionally, the variable’s data type and dimensions must match the specification. (Special rules apply to “physical parameter” - <PARAM> - variables, as described below.)
2. Variable attributes: The variable attributes specified in the User’s Manual must be present in the data file. Further, the value of the attributes must match the specification.

NOTE on additional attributes:

The FileChecker only ensures that all of the specified variable attributes exist in the file. Extra attributes are accepted by the FileChecker (and are not checked). [[1]](#footnote-2)

## Physical Parameter Variables

Physical parameters are implemented as netCDF variables (with attributes) in the profile and trajectory files. The list of allowed physical parameter variable names (the <PARAM> variables) and their associated attributes are documented in the User’s Manual prior to manual-version 3.1. Starting with manual-version 3.1 the list of allowed physical parameter variables is maintained in a separate document; see User’s Manual v3.1 (or later) for the link to the list. This list also specifies the approved attribute settings for each variable.

There are 3 categories of physical parameters: core, bio, and intermediate.

* Core and bio parameters: All six of the *<PARAM>*-related variables must exist in the file: PROFILE\_*<PARAM>*\_QC, *<PARAM>*, *<PARAM>*\_QC, *<PARAM>*\_ADJUSTED, *<PARAM>*\_ADJUSTED\_QC, *<PARAM>*\_ADJUSTED\_ERROR for each parameter.
* Intermediate parameters: For each parameter, the following combinations of <PARAM>-related variables are allowed:
  + *The “QC group”:* The three variables can be present together: PROFILE\_<PARAM>\_QC, <PARAM>, <PARAM>\_QC.
  + *The “full group”:* All six variables (as detailed above) exist in the file.

**PRES / PRESn Special Rules:**

PRES and PRES*n* (where n is a digit) have special rules. In a core-file, these parameters conform to the rules for core-parameters.

In a bio-file, these parameters exist without *any other related variables*: no “QC group” variables and no “\*\_ADJUSTED\*” variables. Furthermore, the data-mode for these parameters is always “R” (in a bio-file) since there can not be any adjusted values.

**“Statistics variables” in bio-argo files:**

For each <PARAM> in a float’s core- and bio-file, two additional variables are allowed *in the bio-profile file*: *<PARAM>*\_STD and *<PARAM>*\_MED.

**NOTE on “C\_format”, “FORTRAN\_format” and “resolution” attributes:**

These attributes are *required* for each physical parameter however their settings are sensor dependent and are, therefore, not checked by the FileChecker.

**NOTE on “Parameters from duplicate sensors”:**

The User’s Manual (version 3.1, November 7th, 2014) states:

**3.3.1 Parameters from duplicate sensors**

Some floats are equipped with 2 different sensors, measuring the same physical parameter. In that case, add the integer "2" at the end of the code of the duplicate parameter (e.g. DOXY2).

If more sensors that measure the same physical parameter are added, then the integer will simply increase by 1 (i.e. DOXY3, DOXY4, and so on).

The names of some of the bio-argo parameters end with a numeric digit. In order to avoid confusion, the following standard has been adopted: When a parameter name ends with a numeric digit, insert an underscore (“\_”) between the name and the integer suffix when creating the parameter name for the duplicate sensor. (For example, “BBP700\_2”)[[2]](#footnote-3)

## Exceptions

Exceptions to format rules are discouraged. However, some exceptions are allowed for historical reasons.

Allowed exceptions are described in the appendices.

# Meta-data Data Consistency Checks

## Reasonable Date Checks



“Reasonable date checks” are applied in an attempt to ensure that all of the dates in the file are consistent. Subtle problems cannot be detected but hopefully large errors can be caught and corrected.

Note that cross-file checks – comparing data in meta-data files with profile files, etc – are *not* performed.

|  |  |  |
| --- | --- | --- |
| Variable | Required Conditions | Result (not met) |
| DATE\_CREATION | Set and valid date.  After 1997-01-01.  Before time file submitted to GDAC. | Reject  Reject  Reject |
| DATE\_UPDATE | Set and valid date.  Not before DATE\_CREATION  Before time submitted to GDAC. | Reject  Reject  Reject |
| LAUNCH\_DATE | Valid date.  After 1997-01-01. | Reject  Reject |
| START\_DATE | If set   * Valid date * Not before LAUNCH\_DATE | Reject  Warning |
| STARTUP\_DATE | If set   * Valid date | Reject |
| END\_MISSION\_DATE | If set   * Valid date * Not before LAUNCH\_DATE * LAUNCH\_DATE set | Reject  Warning  Warning |

## Highly-desirable Parameter Checks (file-version 2.3 and earlier)

The User’s Manual specifies a list of highly-desirable parameters for format-versions up to v2.3. Each of the variables specified in the table are checked for compliance to the stated requirement. Any non-compliant variables generate warnings and do *not* prevent the file from being distributed on the GDAC.

## Mandatory Parameter Checks (file version 2.4 and later)

Format-version 2.4 and later meta-data files have mandatory parameters specified in the User’s Manual. Any non-compliant variables generate an ***error*** and the file is rejected by the GDAC.

Variables with multiple entries (array variables such as the PARAMETER\_\* and SENSOR\_\* variables) have all elements of the arrays checked for compliance. These are: PARAMETER, PARAMETER\_UNITS, PARAMETER\_SENSOR, PREDEPLOYMENT\_CALIB\_COEFFICIENT, PREDEPLOYMENT\_CALIB\_EQUATION, SENSOR, SENSOR\_MAKER, SENSOR\_MODEL, POSITIONING\_SYSTEM, and TRANS\_SYSTEM.

NOTE: Format-version v3.1 also has one highly-desirable parameter (BATTERY\_PACKS) that will only generate a *warning* if it is non-compliant.

## Configuration Parameters

**CONFIG\_MISSION\_NUMBER / CONFIG\_MISSION\_COMMENT:**

* Not 0
* Not FillValue (WARNING)
* CONFIG\_MISSION\_COMMENT is not checked

**CONFIG\_PARAMETER\_NAME / LAUNCH\_CONFIG\_PARAMETER\_NAME:**

Configuration parameters were added in format-version 2.4. The FileChecker compares the configuration parameter name entries (LAUNCH\_CONFIG\_PARAMETER\_NAME and CONFIG\_PARAMETER\_NAME) to the list of approved names.

The configuration parameter entries are composed of two parts: the parameter name (everything up to the last “\_”) and the units (everything after the last “\_”). The FileChecker “decomposes” the entries into the name and unit parts and compares the each part to the currently approved lists. The allowed parameter names are documented in reference tables 18. The approved units are the same as those used for the TECHNICAL\_PARAMETER\_NAME and are documented in reference table 14b.

The configuration parameter specifications can be a “template” similar to:

* CONFIG\_<PARAM>Offset
* CONFIG\_Ocr<param>Bandwidth<I>
* CONFIG\_<short\_sensor\_name>BetaAngle
* CONFIG\_<short\_sensor\_name>DepthZone<N>PowerAcquisitionMode
* CONFIG\_<short\_sensor\_name><param>FluorescenceEmissionBandwidth

In these cases, the “<\*>” section can be replaced with anything or, in fact, nothing (it can be left out).

\*\*\*FOR FUTURE IMPLEMENTATION\*\*\* A future enhancement of the FileChecker will apply checks to some or all of these “templates”. For instance, a list of approved “short\_sensor\_names” have been developed (and will be published soon) that the settings can be checked against.

The transition from “old names” to “new names” is now supported: Names and units that are deprecated can be defined. Files that contain the deprecated name/unit will still be accepted, but a WARNING will be issued. At the end of the transition period, files with the deprecated name/unit witll be REJECTED.

# Profile Data Checks

Both core-Argo and bio-Argo profile files are subject to the following checks. (The core-argo profile files have been checked since the FileChecker was implemented in April 2015. The bio-Argo profile files were added in December 2015.)

## Meta-data (within a profile file) Checks

The profile meta-data that is contained within the file is checked for valid settings.

|  |  |  |
| --- | --- | --- |
| Variable | Required Conditions | Result (not met) |

|  |  |  |
| --- | --- | --- |
| CYCLE\_NUMBER | Set.  All the same in a single-cycle file. | Reject  Reject |
| DATA\_MODE | ‘A’, ‘D’, or ‘R’ | Reject |
| DIRECTION | ‘A’ or ‘D’. | Reject |
| DATA\_STATE\_INDICATOR | Reference table 6.  R-mode file: Not “2C” or “2C+”  D-mode file: “2C” or “2C+” | Reject  Reject  Reject |
| DATA\_CENTRE | Reference table 4.  Valid for DAC submitting file. | Reject  Reject |
| INST\_REFERENCE | Set. | Warning  (V2.3, earlier) |
| PARAMETER\_DATA\_MODE | ‘A’, ‘D’, ‘R’, or ‘ ‘   * For PRES/PRESn in bio-profile file: Must be ‘R’ | Reject |
| PLATFORM\_NUMBER | 5- or 7-digit number.  All the same in a single-cycle file. | Reject  Reject |
| POSITIONING\_SYSTEM | Reference table 9. | Warning  (V2.3, earlier) |
| WMO\_INST\_TYPE | Set.  Reference table 8. | Reject  Reject |
| JULD\_QC | Reference table 2. | Reject |
| POSITION\_QC | Reference table 2. | Reject |
| CONFIG\_MISSION\_NUMBER | R-mode file: Not 0 (zero)  D-mode file: Not FillValue or 0 (zero)   * Except CYCLE\_NUMBER = 0; may be FillValue | Reject  Reject |

## Reasonable Date Checks

“Reasonable date checks” are applied in an attempt to ensure that all of the dates in the file are consistent. Subtle problems cannot be detected but hopefully large errors can be caught and corrected.

Note that cross-file checks – meta-data dates with profile dates, etc – are *not* performed.

|  |  |  |
| --- | --- | --- |
| Variable | Required Conditions | Result (not met) |

|  |  |  |
| --- | --- | --- |
| REFERENCE\_DATE\_TIME | Set and matches Argo standard. | Reject |
| DATE\_CREATION | Set and valid date.  After 1997-01-01.  Before time file submitted to GDAC. | Reject  Reject  Reject |
| DATE\_UPDATE | Set and valid date.  Not before DATE\_CREATION  Before time submitted to GDAC. | Reject  Reject  Reject |
| JULD | If missing, JULD\_QC = 3, 4, or 9  After 1997-01-01.  Not after DATE\_CREATION  Not after GDAC receipt time | Reject  Reject  Reject  Reject |
| JULD\_LOCATION | If set, within 2 days of JULD.  If FillValue, position FillValue too. | Warning  Reject |
| HISTORY\_DATE | If set, valid date.  Not after DATE\_UPDATE. | Reject  Reject |
| CALIBRATION\_DATE | If set, valid date.  Before DATE\_UPDATE. | Reject  Reject |

## Physical Parameter Checks

The following checks are performed on the variables associated with the physical parameter data. Failed checks result in *rejection* unless otherwise noted. See section 2.1 for a description of the rules regarding which of the physical parameter variables are allowed/required for each physical parameter.

**STATION\_PARAMETERS:**

* All valid parameter names. Reference table 3.
* For each parameter, <PARAM> variable exists in file.
* For each <PARAM> variables (with data), parameter name included in STATION\_PARAMETERS
* Core-file / profile #1: PRES and TEMP present
* No duplicate names in list.
* Blank entries within the sequence of names. (For example, “PRES”, “ “, “TEMP”). Warning

**<PARAM> / <PARAM>\_QC:**

* No NaNs
* Valid QC flags: Reference table 2 or blank (“ “; not measured).
* Where data is FillValue: QC flag of 9 or “ “.
* Where QC is 9 or “ “ (not measured), <PARAM> must be FillValue.
* Where data not FillValue: 1 <= QC flag <= 4
* Exception: 0 allowed for variables without defined quality control checks (e.g. bio-Argo parameters)

*Note: Some bio-argo parameters have an “extra dimension” - <param>(N\_PROF, N\_LEVELS, N\_VALUESnn). For these variables, a level is only considered to be “FillValue” if all of the “extra dimension” values are FillValue.*

**<PARAM>\_ADJUSTED / <PARAM>\_ADJUSTED\_QC:**

* DATA\_MODE = ‘R’: All FillValue (including \*\_QC and \*\_ERROR)
* DATA\_MODE = ‘A’
  + No NaNs (including \*\_ERROR)
  + Valid QC flags: Reference table 2 or blank (“ “; not measured).
  + Where <PARAM>\_ADJUSTED\_QC = “ “ (not measured)

<PARAM>\_QC = “ “ and <PARAM>\_ADJUSTED = FillValue

* + <PARAM>\_ADJUSTED\_ERROR set to FillValue for core-parameters. (<PARAM>\_ADJUSTED\_ERROR may be set for bio-parameters.
  + Where <PARAM> is FillValue

<PARAM>\_ADJUSTED, <PARAM>\_ADJUSTED\_ERROR is FillValue

<PARAM>\_ADJUSTED\_QC = 9

* DATA\_MODE = ‘D’
  + No NaNs (including \*\_ERROR)
  + Valid QC flags: Reference table 2 or blank (“ “; not measured).
  + Where <PARAM>\_ADJUSTED\_QC = “ “ (not measured)

<PARAM>\_QC = “ “ and <PARAM>\_ADJUSTED = FillValue

* + Where <PARAM> is FillValue

<PARAM>\_ADJUSTED, <PARAM>\_ADJUSTED\_ERROR is FillValue

<PARAM>\_ADJUSTED\_QC = 9

* + Where <PARAM> is not FillValue and <PARAM>\_ADJUSTED is FillValue

QC = 4 or 9

* + Where <PARAM> is not FillValue and <PARAM>\_ADJUSTED is not FillValue

QC ≠ 4 or 9

* + Where QC ≠ 4 or 9: <PARAM>\_ADJUSTED\_ERROR is not FillValue
  + <PARAM>\_ADJUSTED and <PARAM>\_ADJUSTED\_ERROR set at the same levels.

**PROFILE\_<PARAM>\_QC:**

* Computed based on the value of:

DATA\_MODE = ‘R’: <PARAM>\_QC

DATA\_MODE = ‘A’ or ‘D’: <PARAM>\_ADJUSTED\_QC

* Data values: Number of levels flagged with something other than ‘9’ or ‘ ‘
* Good values: Number of levels flagged with 1, 2, 5, or 8

Reported value must match expected value according to reference table 2a.

## Additional D-mode File Checks

These additional checks are performed on D-mode files.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Required Conditions** | **Result (not met)** |
| DATA\_STATE\_INDICATOR | Must be “2C” or “2C+” | Reject |
| PARAMETER | Every parameter in STATION\_PARAMETERS must also be in PARAMETER. | Reject |
| SCIENTIFIC\_CALIB\_COMMENT | Not empty – all entries.\* | Reject |
| CALIBRATION\_DATE | Must be set to a valid date (see “Reasonable Date Checks” above). | Reject |

\* NOTE: If all of the parameter data values in a profile are missing (fill value). Then, the SCIENTIFIC\_CALIB\_COMMENT and CALIBRATION\_DATE are allowed to be empty and are not checked. The parameter name must still be present in PARAMETER.

# Technical Data Checks

## Meta-data Checks (within technical file)

The meta-data that is contained within the file is checked for valid settings.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Required Conditions** | **Result (not met)** |
| PLATFORM\_NUMBER | 5- or 7-digit number.  All the same in a single-cycle file. | Reject  Reject |
| DATA\_CENTRE | Reference table 4.  Valid for DAC submitting file. | Reject  Reject |

## Reasonable Date Checks



“Reasonable date checks” are applied in an attempt to ensure that all of the dates in the file are consistent. Subtle problems cannot be detected but hopefully large errors can be caught and corrected.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Required Conditions** | **Result (not met)** |
| DATE\_CREATION | Set and valid date.  After 1997-01-01.  Before time file submitted to GDAC. | Reject  Reject  Reject |
| DATE\_UPDATE | Set and valid date.  Not before DATE\_CREATION  Before time submitted to GDAC. | Reject  Reject  Reject |

## Technical Parameter Names (format-version 2.4 and later)

TECHNICAL\_PARAMETER\_NAME entries are composed of two parts: the name (everything up to the last “\_”) and the unit (everything after the last “\_”).

The FileChecker “decomposes” the entries into the name and unit parts and compares the each part to the currently approved lists. (reference tables 14a/b).

The technical parameter specification can be a “template” similar to:

* NUMBER\_Ascending<short\_sensor\_name>SamplesDeepAbsolute
* NUMBER\_<short\_sensor\_name>DescentSamplesDepthZone<Z>
* FLAG\_<short\_sensor\_name>Status
* VOLTAGE\_Battery<short\_sensor\_name>
* PRES\_<int>HoursIntoDesentToProfile

In these cases, the “<\*>” section can be replaced with anything or, in fact, nothing (it can be left out).

\*\*\*FOR FUTURE IMPLEMENTATION\*\*\* A future enhancement of the FileChecker will apply checks to some or all of these “templates”. For instance, a list of approved “short\_sensor\_names” have been developed (and will be published soon) that the settings can be checked against.

TECHNICAL\_PARAMETER\_VALUE entries are not checked.

## \*\*\*FOR FUTURE IMPLEMENTATION\*\*\* Technical Parameter Values

**\*\*\*This check was discussed at ADMT-16. It was generally agreed that this is useful but it is prioritized below the trajectory data checks and will be implemented at a later date\*\*\***

The currently approved list of Technical Parameter Units includes the “data type” of values associated with that unit. The TECHNICAL\_PARAMETER\_VALUE is checked to ensure that it agrees with the specified type.

|  |  |
| --- | --- |
| **Data Type** | **Valid Value** |
| date/time | Valid date/time string matching the indicated format. |
| hex | Valid hex characters: 0-9, a-f |
| integer | Optional leading sign and numeric digits |
| float | Valid floating point number |
| string | Any string of characters (no real check) |

# Trajectory Data Checks

## Meta-data Checks

The meta-data that is contained within the file is checked for valid settings.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Required Conditions** | **Result (not met)** |
| DATA\_STATE\_INDICATOR | Reference table 6.  R-mode file: Not “2C” or “2C+”  D-mode file: “2C” or “2C+” | Reject  Reject  Reject |
| DATA\_CENTRE | Reference table 4.  Valid for DAC submitting file. | Reject  Reject |
| FIRMWARE\_VERSION | Set. | Reject |
| FLOAT\_SERIAL\_NUMBER | Set. | Reject |
| PLATFORM\_NUMBER | 5- or 7-digit number. | Reject |
| PLATFORM\_TYPE | Reference table 23. | Reject |
| POSITIONING\_SYSTEM | Reference table 9. | Reject |
| TRAJECTORY\_PARAMETERS | All valid parameter names. Reference table 3.  For each parameter, <PARAM> variable exists in file.  For each <PARAM> variables (with data), parameter name included in TRAJECTORY\_PARAMETERS  No duplicate names in list.  Blank entries within the sequence of names. (For example, “PRES”, “ “, “TEMP”). | Reject  Reject  Reject  Reject  Reject |
| WMO\_INST\_TYPE | Set.  Reference table 8. | Reject  Reject |

## Reasonable Date Checks

“Reasonable date checks” are applied in an attempt to ensure that all of the dates in the file are consistent. Subtle problems cannot be detected but hopefully large errors can be caught and corrected.

Note that cross-file checks – meta-data dates with profile dates, etc – are *not* performed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | | **Required Conditions** | **Result (not met)** |
| REFERENCE\_DATE\_TIME | Set and matches Argo standard. | | Reject |
| DATE\_CREATION | Set and valid date.  After 1997-01-01.  Before time file submitted to GDAC. | | Reject  Reject  Reject |
| DATE\_UPDATE | Set and valid date.  Not before DATE\_CREATION  Before time submitted to GDAC. | | Reject  Reject  Reject |
| HISTORY\_DATE | If set, valid date.  Not after DATE\_UPDATE. | | Reject  Reject |

**\*\*\*None of the checks below are currently enabled in “production”. These are being routinely exercised in “TEST” mode\*\*\***

## N\_MEASUREMENT Variable Group Checks

**CYCLE\_NUMBER / CYCLE\_NUMBER\_ADJUSTED:**

*(Note: CYCLE\_NUMBER\_ADJUSTED is only in core-files.)*

***Real-time*** *(all DATA\_MODE = ‘R’ or ‘A’):*

* CYCLE\_NUMBER:
  + Launch cycle (-1) allowed only in first index
    - Otherwise, must be >= 0
  + Cannot be FillValue
* CYCLE\_NUMBER\_ADJUSTED *(Core-file only)*
  + Must be FillValue

***Delayed-mode file*** *(at least one DATA\_MODE=’D’; may include R-mode cycles)****:***

* CYCLE\_NUMBER
  + Launch cycle (-1) allowed only in first index
    - Otherwise, must be >= 0
  + R/A-mode cycle: CYCLE\_NUMBER must be set
* CYCLE\_NUMBER\_ADJUSTED *(Core-file only)*
  + Launch cycle (-1) allowed only in first index
    - Otherwise, must be >= 0
  + D-mode cycle: must be set
  + R/A-mode cycle: must be FillValue

**MEASUREMENT\_CODE:**

The FileChecker cannot ensure that the measurement codes are applied correctly, only that they are valid values.

Measurements codes cannot be FillValue.

Valid values:

* Cannot be FillValue
* Primary and secondary codes (MC): even multiple of 50
* Relative special codes: In range MC – 15 to MC – 1
* Specific special codes: One of the specified specific special codes

**JULD / JULD\_STATUS / JULD\_QC:**

* JULD\_QC: Reference Table 2 or blank (“ “)
* JULD\_STATUS: Reference Table 19 or blank (“ “)
* JULD\_QC / JULD\_STATUS
  + If JULD\_QC = ‘ ‘, JULD\_STATUS = ‘ ‘
  + If JULD\_QC = ‘9’, JULD\_STATUS = ‘9’
* JULD
  + If FillValue, JULD\_QC must be ‘ ‘ or ‘9’
  + If set, JULD\_QC must not be ‘ ‘ or ‘9’
* Date checks:
  + If JULD\_QC indicates “good”:
    - Date must be after 1997-01-01
    - Must be before DATE\_UPDATE (with a 2 day buffer to allow for clock drift)

\*\*\* See section 7.5 for the JULD[N\_MEASUREMENT] / JULD\_\*[N\_CYCLE] checks.

**JULD\_ADJUSTED / JULD\_ADJUSTED\_STATUS / JULD\_ADJUSTED\_QC:**

*(Note: Core-files only)*

* JULD\_ADJUSTED\_QC: Reference Table 2 or blank (“ “)
* JULD\_ADJUSTED\_STATUS: Reference Table 19 or blank (“ “)
* JULD\_ADJUSTED \_QC / JULD\_ADJUSTED \_STATUS
  + If JULD\_ADJUSTED \_QC = ‘ ‘, JULD\_ADJUSTED \_STATUS = ‘ ‘
  + If JULD\_ADJUSTED \_QC = ‘9’, JULD\_ADJUSTED \_STATUS = ‘9’
* JULD\_ADJUSTED
  + If FillValue, JULD\_ADJUSTED\_QC must be ‘ ‘ or ‘9’
  + If set,
    - JULD\_ADJUSTED \_QC must not be ‘ ‘ or ‘9’
    - If associated JULD is missing, this represents “an estimation” and the DATA\_MODE must be ‘A’ or ‘D’.
    - If associated JULD is set, DATA\_MODE for this cycle can be anything
* Date checks:
  + If JULD\_ADJUSTED\_QC indicates “good”:
    - Date must be after 1997-01-01
    - Must be before DATE\_UPDATE (with a 2 day buffer to allow for clock drift)

\*\*\* See section 7.5 for the JULD[N\_MEASUREMENT] / JULD\_\*[N\_CYCLE] checks.

**JULD / JULD\_ADJUSTED Sorting:**

The events in the N\_MEASUREMENT array are in the order that they occured. This is hard to check due to clock drift issues. The following rudimentary checks are applied to provide some checking that the events are in the proper order.

Ensure that successive “final JULD”‡ are ascending (or the same) among similar “types”. The “types” in this case are defined as:

* Real-time file:
  + R-mode cycle, “float-measured” JULDs (not measurement codes 702, 703, 704)
  + A-mode cycle, “float-measured” JULDs (not measurement codes 702, 703, 704)
  + R-mode cycle, “satellite-measured” JULDs (measurement codes 702, 703, 704)
  + A-mode cycle, “satellite-measured” JULDs (measurement codes 702, 703, 704)
* Delayed-mode file:
  + D-mode cycle, “float-measured” JULDs (not measurement codes 702, 703, 704)
  + D-mode cycle, “satellite-measured” JULDs (measurement codes 702, 703, 704)
  + Non D-mode cycles will be ignored

‡ Definition of “final JULD”

The “final JULD” of a measurement is the JULD\_ADJUSTED if it is set and is JULD otherwise. No consideration is given to the QC flag and/or status; those checks are performed elsewhere.

**LATITUDE / LONGITUDE / POSITION\_QC / POSITION\_ACCURACY**:

* POSITION\_QC: Reference table 2 or blank (“ “)
* Where POSITION\_QC = blank (“ “) or ‘9’: LATITUDE/LONGITUDE set to FillValue
  + Otherwise: LATITUDE and LONGITUDE must not be FillValue
* POSITION\_ACCURACY: Reference Table 5 or blank (“ “)

**<PARAM> / <PARAM>\_QC:**

* <PARAM>\_QC: Reference table 2 or blank (“ “)
* Where <PARAM>\_QC =  ‘ ‘ or ‘9’: <PARAM> set to FillValue
  + Otherwise, must not be FillValue

**<PARAM>\_ADJUSTED / <PARAM>\_ADJUSTED\_QC / <PARAM>\_ADJUSTED\_ERROR:**

*(Note: These variables are optional for intermediate parameters)*

* DATA\_MODE = ‘R’: All FillValue
  + <PARAM>\_ADJUSTED and <PARAM>\_ADJUSTED\_ERROR: All FillValue
  + <PARAM>\_ADJUSTED\_QC: ‘ ‘ or 9
* DATA\_MODE = ‘A’
  + No NaNs (including \*\_ERROR)
  + Valid QC flags: Reference table 2 or blank (“ “).
  + Where <PARAM>\_ADJUSTED\_QC = “ “ (not measured)

<PARAM>\_QC = “ “ and <PARAM>\_ADJUSTED = FillValue

* + <PARAM>\_ADJUSTED\_ERROR set to FillValue for core-parameters. (<PARAM>\_ADJUSTED\_ERROR may be set for bio-parameters)..
  + Where <PARAM> is FillValue, <PARAM>\_ADJUSTED, \*\_QC is FillValue
* DATA\_MODE = ‘D’
  + No NaNs (including \*\_ERROR)
  + Valid QC flags: Reference table 2 or blank (“ “; not measured).
  + Where <PARAM>\_ADJUSTED\_QC = “ “ (not measured)

<PARAM>\_QC = “ “ and <PARAM>\_ADJUSTED = FillValue

* + Where <PARAM> is not FillValue and <PARAM>\_ADJUSTED is FillValue

<PARAM>\_ADJUSTED\_QC = ‘4’ or ‘9’

* + Where <PARAM> is not FillValue and <PARAM>\_ADJUSTED is not FillValue

<PARAM>\_ADJUSTED\_QC ≠ ‘4’ or ‘9’

* + Where <PARAM>\_ADJUSTED\_QC ≠ ‘4’ or ‘9’: <PARAM>\_ADJUSTED\_ERROR is not FillValue

## N\_CYCLE variable group checks

**DATA\_MODE:** *(Core- and bio-files)*

* Set to 'R', 'A', or 'D'

**CYCLE\_NUMBER \_INDEX/ CYCLE\_NUMBER\_INDEX\_ADJUSTED:**

*(Note: CYCLE\_NUMBER\_INDEX are in both core- and bio-files. \*\_ADJUSTED are in core-files only.)*

***Real-time*** *(all DATA\_MODE = ‘R’ or ‘A’):*

* CYCLE\_NUMBER\_INDEX
  + Cannot be FillValue
  + Must be >= 0 (Cycle “-1” not allowed)
  + No duplicates
* CYCLE\_NUMBER\_INDEX\_ADJUSTED *(Core-file only)*
  + Must be FillValue

***Delayed-mode file*** *(at least one DATA\_MODE=’D’; may include R-mode cycles)****:***

* CYCLE\_NUMBER\_INDEX
  + Must be >= 0 (Cycle “-1” not allowed)
  + R/A-mode cycle: Must be set
  + No duplicates (ignoring any FillValues)
* CYCLE\_NUMBER\_INDEX\_ADJUSTED
  + Must be >= 0 (Cycle “-1” not allowed)
  + D-mode cycle: Must be set
  + R/A-mode cycle: Must be FillValue
  + No duplicates (ignoring any FillValues)

***Overall***

No duplicates in the “final *cycle\_number\_index*”† sequence.

† Definition of “final *cycle\_number\_index*”:

The “final *cycle\_number*” is defined as the “best cycle\_number” for a cycle. In a real-time file, this is just the CYCLE\_NUMBER (CYCLE\_NUMBER\_INDEX) value. In a delayed-mode file – that may contain both R/A-mode and D-mode cycles – the “best value” is the CYCLE\_NUMBER\_ADJUSTED (CYCLE\_NUMBER\_INDEX\_ADJUSTED) value, if it exists, or the CYCLE\_NUMBER (CYCLE\_NUMBER\_INDEX) value otherwise.

**JULD\_\* / JULD\_\*\_STATUS:** *(Core-files only)*

* JULD\_\*
  + If FillValue, JULD\_\*\_STATUS must be ‘ ‘ or ‘9’
  + If set, JULD\_\*\_STATUS must not be ‘ ‘ or ‘9’

These variables replicate specified values in the JULD variable. The checks consist of comparing the JULD and JULD\_\* values.

* JULD\_\*\_STATUS: Reference Table 19 or blank (“ “)

\*\*\* See section 7.5 for the JULD[N\_MEASUREMENT] / JULD\_\*[N\_CYCLE] checks.

**GROUNDED:** *(Core-files only)*

* Reference table 20

**CONFIG\_MISSION\_NUMBER:** *(Core and bio-files)*

* R-mode files: Not 0 (zero)
* D-mode files: Not FillValue or 0 (zero)
  + Except CYCLE\_NUMBER = 0, which may be FillValue

## N\_MEASUREMENT / N\_CYCLE Inter-dependence

**CYCLE\_NUMBER / CYCLE\_NUMBER\_INDEX**

(Refer to the definition of a “final cycle number” above: Definition of “final *cycle\_number\_index*”) Definition of “final *cycle\_number\_index*”Definition of “final *cycle\_number\_index*”Definition of “final *cycle\_number\_index*”

* Every cycle number in CYCLE\_NUMBER (except cycle “-1”) must be in CYCLE\_NUMBER\_INDEX
* Every cycle number in CYCLE\_NUMBER\_INDEX must be in CYCLE\_NUMBER
* For each N\_CYCLE index, when CYCLE\_NUMBER, CYCLE\_NUMBER\_ADJUSTED, CYCLE\_NUMBER\_INDEX and CYCLE\_NUMBER\_INDEX\_ADJUSTED are set CYCLE\_NUMBER\_ADJUSTED-CYCLE\_NUMBER should be equal to CYCLE\_NUMBER\_INDEX\_ADJUSTED-CYCLE\_NUMBER\_INDEX

**JULD[N\_MEASUREMENT] / JULD\_\*[N\_CYCLE] Variables**

There is a group of “JULD\_\*[N\_CYCLE] variables” specified in the trajectory format that correspond to significant events during a float cycle; JULD\_DESCENT\_START, JULD\_PARK\_START, etc. The values stored in these variables are also stored in the JULD[N\_MEASUREMENT] variable. The “mapping” between JULD and JULD\_\* values is based on cycle number and MEASUREMENT\_CODE values. The measurement code mapping for JULD –to- JULD\_\* variables is documented in the Argo Trajectory Cookbook.

The following checks are performed:

* Every JULD and JULD\_STATUS value from one of the significant measurement codes is compared to the associated JULD\_\* value on a cycle-by-cycle basis.
  + If they do not match, the file is rejected.
  + One code (703) is used for a series of values within a single cycle. The associated JULD\_\* variables correspond to the first and last value within a single cycle. For the purposes of the checks, “first” and “last” are determined by the sequential position within the JULD variable.
* Every value within every JULD\_\* variable that is not associated with a JULD value must be set to FillValue (STATUS = ‘ ‘ or ‘9’).

1. Allowed exceptions

## DATA\_TYPE settings:

The approved (documented) DATA\_TYPE settings are:

* Argo meta-data
* Argo profile
* Argo profile merged
* Argo trajectory
* Argo technical data
* B-Argo profile
* B-Argo trajectory

For pre-v3.1 files, the following exceptions are currently being allowed (because DACs are using them) with warnings sent to the DACs:

* ARGO profile
* ARGO trajectory
* Argo technical
* ARGO technical data

No exceptions are allowed for v3.1 and later files.

## Pre-v3.1 Files

Several exceptions have been allowed over the years for various attributes in pre-V3.1 files. Since these files are being replaced by v3.1 files, it is not necessary to document them.

## V3.1 Files

See DATA\_TYPE above.

Global attributes:

* :user\_manual\_version = 3.*<anything>*;
* :Conventions = Argo-3.*<anything>* CF-*<anything>*;

DOXY attributes:

* valid\_min and valid\_max where changed
  + Previous values:
  + Current values
* WARNINGS are being sent to DACs that still use the old values

1. Allowing additional (unchecked) attributes was discussed and approved at ADMT-16. For future reference, the options presented were: 1) develop a list of approved “additional parameters” with the added manual overhead of maintaining the list or 2) allow any and all additional attributes. Option 2 is currently implemented. [↑](#footnote-ref-2)
2. The change to the naming rules for duplicate sensors was discussed and approved at ADMT-16. [↑](#footnote-ref-3)