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

Version 2.6.2
October 2020

ARGO
part of the integrated global observation strategy

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/10.13155/46120

Table of contents

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.
2.5.3	Never released	Version incremented to stay synchronized with software configuration management.
2.5.4	April 2018	VERTICAL_SAMPLING_SCHEME tests.
		DATA_MODE/PARAMETER_DATA_MODE consistency checks
		Bio-profile: PARAMETER_DATA_MODE = 'R' for PRES
		Variable attribute data type checks.
		Highlight tests that should be ERRORS but that are still generating WARNINGS to allow time

		for DACs to transition.
2.5.5	Jan 2019	CONFIG_MISSION_NUMBER is allowed to be 0 (ADMT-19 decision)
2.6.1	Oct 2020	Add metadata cross-reference checks • PLATFORM_TYPE / PLATFORM_MAKER • PLATFORM_TYPE / WMO_INST_TYPE • SENSOR_MODEL / SENSOR_MAKER • SENSOR_MODEL / SENSOR Append PLATFORM_MAKER to SENSOR_MAKER table Add additional "template" tests to CONFIG and TECH parameter name tests Add full Trajectory file tests
2.6.2	March 2021	Allow "intermediate parameters" in both core- and bio-profile files Document parameter names with <parameter_sensor_name>

1   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.

1.1   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) approved	Valid, accepted value.
deprecated	Previously "active" value. Still allowed while DACs transition. WARNING message generated. Will become obsolete at a future time.
publication underway	Values not yet "active". REJECTED by FileChecker.
creation underway	Rejection message will indicate the Status Value.
obsolete	Invalid value. REJECTED by FileChecker.

1.2   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.

1.3   Items to Note/Discuss

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

1.4   Proposed Changes

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

2   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.)

3   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_VALUESn 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.

3. 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.)

4. Variable attributes: The variable attributes specified in the User's Manual must be present in the data file.

NOTES:
- Data Types: The data type of the value specified for an attribute is checked. If a "string" is expected, the value must be a string. If a numeric value is expected, the value must be numeric and, in most cases, be the same numeric type as the associated variable.
- Attribute Values: Where the User's Manual specifies the value of the attribute, the value is checked for conformance. Some attribute values are unspecified, in which case the value is not checked. (Example: JULD:resolution)
- Attribute existence: There are a few attributes that are not allowed in specific cases. In these cases, the existance of these attributes is an error. (Example: <PARAM>:valid_min when it is not specified in the "physical parameter list".)
- 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]

3.1 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 specifies the name, data type, fill value, and approved attribute settings for each variable (long_name, standard_name, unit, valid_min, valid_max, and category (bio-, core-, or intermediate).

If an attribute that is in this table has an unspecified value, the attribute is not allowed to exist. Specifically, standard_name, valid_min, valid_max.

There are additional attributes that are required for each <PARAM> that are not part of this table. These are: C_format, Fortran_format.

There are 3 categories of physical parameters: core, bio, and intermediate. Intermediate variables are allow in both core-profile and bio-profile files. The physical parameter list referred to above includes a "category" column that defines the category for each physical parameter name.

- 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.

---

[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.

Physical Parameters that include the \<short_sensor_name\> template:

There are some physical parameter variables that include a \<parameter_sensor_name\> "template"; for example, NB_SAMPLE_\<parameter_sensor_name\>. These variables can be defined for any of the \<parameter_sensor_name\> values defined in the corresponding column of Reference Table 27.

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]

3.2 Exceptions

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

Allowed exceptions are described in the appendices.

4 Meta-data Data Consistency Checks

---

[2] The change to the naming rules for duplicate sensors was discussed and approved at ADMT-16.

4.1 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.	Reject
	After 1997-01-01.	Reject
	Before time file submitted to GDAC.	Reject
DATE_UPDATE	Set and valid date.	Reject
	Not before DATE_CREATION	Reject
	Before time submitted to GDAC.	Reject
LAUNCH_DATE	Valid date.	Reject
	After 1997-01-01.	Reject
START_DATE	If set	
	• Valid date	Reject
	• Not before LAUNCH_DATE	Warning
STARTUP_DATE	If set	
	• Valid date	Reject
END_MISSION_DATE	If set	
	• Valid date	Reject
	• Not before LAUNCH_DATE	Warning
	• LAUNCH_DATE set	Warning

4.2 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.

4.3 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 with the Users Manual §2.4.9. 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.

4.4 Configuration Parameters

CONFIG_MISSION_NUMBER / CONFIG_MISSION_COMMENT:

- 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

The template elements (the values in <....>) are checked as shown:

Template Checks	Valid Value
<D>, <N>, <N1>, <S>, <SubS>, <Z>	One or more numeric digits.
<digit>	A single numeric digit
<I>, <int>	A valid integer
<N>, <N1>	Optional leading sign and numeric digits
<param>	A parameter name with first character upper case and other characters lower case. May consist of multiple parts with the first letter of each in upper case. [†]
<PARAM>	All upper case parameter name
<short_sensor_name>	Compared to the list of approve short sensor names[†]
<horizontal_phase_name>	Phase name constructed as one or more parts and ending in "Phase". Compared to the list of approve phase names. [†]
<vertical_phase_name>	See <horizontal_phase_name>[†]

[†] Lists of approved names: Bio-argo configuration parameters include curated lists of approved values for the indicated templates. In these cases, the template values are compared to these lists. Where a list of approved values do not exist for a specific parameter, a generic list of values is checked. The current list of generic template values:

<short_sensor_name>	<horizontal_phase_name>	<vertical_phase_name>	<param>
Crover Ctd Cyc Eco Flbb Flntu Ocr Optode Ph Sfet Stm Suna	ParkDriftPhase ProfDriftPhase SurfaceDriftPhase	AscentPhase BuoyancyReductionPhase DescentToParkPhase DescentToProfPhase	Cdom Chla DownIrr DownRad Irradiance Radiance UpIrr UpRad

4.5   Cross-Reference Checks

The SENSOR_MODEL reference table (Reference Table 27) includes columns for SENSOR_MAKER and SENSOR. The values of SENSOR_MAKER and SENSOR are cross-referenced with the SENSOR_MODEL.

The PLATFORM_TYPE reference table (Reference Table 23) includes columns for WMO_INST_TYPE and PLATFORM_MAKER. The values of WMO_INST_TYPE and PLATFORM_MAKER are cross-referenced with PLATFORM_TYPE.

5   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.)

5.1 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)
CONFIG_MISSION_NUMBER	D-mode file: Not FillValue • Except CYCLE_NUMBER = 0; may be FillValue	Reject
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+" NOTE : Allowed to be FillValue, if no data exists for the profile.	Reject Reject Reject
DATA_CENTRE	Reference table 4. Valid for DAC submitting file.	Reject Reject

INST_REFERENCE	Set.	Warning (V2.3, earlier)
JULD_QC	Reference table 2.	Reject
PARAMETER_DATA_MODE	'A', 'D', 'R', or ' ' Consistent with DATA_MODE (the DATA_MODE of a profile must be the "highest" mode of any parameter reported in the profile) Bio-profile file: PRES/PRESn: Must be 'R'	Reject (temporarily a Warning) Erreur ! Signet non défini. (temporarily a Warning) Erreur ! Signet non défini.
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)
POSITION_QC	Reference table 2.	Reject
VERTICAL_SAMPLING_ SCHEME	Reference table 16. Must start with pre-defined values; free-form after that N_PROF = 1: Must be "Primary sampling" N_PROF > 1: Must NOT be "Primary sampling" NOTE : Allowed to be FillValue, if no data exists for a profile.	Reject (temporarily a Warning) Erreur ! Signet non défini.
WMO_INST_TYPE	Set. Reference table 8.	Reject Reject

5.2 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.	Reject

	Not after DATE_UPDATE.	Reject
CALIBRATION_DATE	If set, valid date.	Reject
	Before DATE_UPDATE.	Reject

5.3 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

- o <PARAM>_ADJUSTED_ERROR set to FillValue for core-parameters.
  (<PARAM>_ADJUSTED_ERROR may be set for bio-parameters.
- o Where <PARAM> is FillValue

  <PARAM>_ADJUSTED, <PARAM>_ADJUSTED_ERROR is FillValue

  <PARAM>_ADJUSTED_QC = 9

- DATA_MODE = 'D'
  - o No NaNs (including *_ERROR)
  - o Valid QC flags: Reference table 2 or blank (" "; not measured).
  - o Where <PARAM>_ADJUSTED_QC = " " (not measured)

    <PARAM>_QC = " " and <PARAM>_ADJUSTED = FillValue

  - o Where <PARAM> is FillValue

    <PARAM>_ADJUSTED, <PARAM>_ADJUSTED_ERROR is FillValue

    <PARAM>_ADJUSTED_QC = 9

  - o Where <PARAM> is not FillValue and <PARAM>_ADJUSTED is FillValue

    QC = 4 or 9

  - o Where <PARAM> is not FillValue and <PARAM>_ADJUSTED is not FillValue

    QC ≠ 4 or 9

  - o Where QC ≠ 4 or 9: <PARAM>_ADJUSTED_ERROR is not FillValue
  - o <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.

5.4 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.

6 Technical Data Checks

6.1 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.	Reject
	All the same in a single-cycle file.	Reject
DATA_CENTRE	Reference table 4.	Reject
	Valid for DAC submitting file.	Reject

6.2 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.	Reject
	After 1997-01-01.	Reject
	Before time file submitted to GDAC.	Reject
DATE_UPDATE	Set and valid date.	Reject
	Not before DATE_CREATION	Reject
	Before time submitted to GDAC.	Reject

6.3 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

See the Template Checks table in section 4.4.

6.4 Technical Parameter Values

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 digits: 0-9, a-f (case-insensitive)
integer	Optional leading sign and numeric decimal digits
float	Valid floating point number
logical	One of "true", "false", "yes", "no", "1", "0"
string	Any string of characters (no real check)

7 Trajectory Data Checks

7.1 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.	Reject
DATA_CENTRE	Reference table 4.	Reject
	Valid for DAC submitting file.	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.	Reject
	For each parameter, <PARAM> variable exists in file.	Reject
	For each <PARAM> variables (with data), parameter name included in TRAJECTORY_PARAMETERS	Reject
	No duplicate names in list.	Reject
	Blank entries within the sequence of names. (For example, "PRES", " ", "TEMP").	Reject

| WMO_INST_TYPE | Set. | Reject |
| | Reference table 8. | Reject |

7.2 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.	Reject
	After 1997-01-01.	Reject
	Before time file submitted to GDAC.	Reject
DATE_UPDATE	Set and valid date.	Reject
	Not before DATE_CREATION	Reject
	Before time submitted to GDAC.	Reject
HISTORY_DATE	If set, valid date.	Reject
	Not after DATE_UPDATE.	Reject

7.3 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: Users Manual Reference Table 15

- Relative Generic Codes MC-1 to MC-15 are allowed (MC-16 to MC-24 are allowed for in the standard but are currently unassigned.)

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 1 day buffer to allow for clock drift)

*** See section 7.5 for the JULD[N_MEASUREMENT] / JULD_*[N_CYCLE] checks.

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

7.4 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 **Erreur ! Source du renvoi introuvable.** for the JULD[N_MEASUREMENT] / JULD_*[N_CYCLE] checks.

GROUNDED: *(Core-files only)*

- Reference table 20

CONFIG_MISSION_NUMBER: *(Core and bio-files)*

- D-mode files: Not FillValue
  - Except CYCLE_NUMBER = 0, which may be FillValue

7.5 N_MEASUREMENT / N_CYCLE Inter-dependence

CYCLE_NUMBER / CYCLE_NUMBER_INDEX

(Refer to the definition of a "final cycle number" above)

- 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').

Appendix A.   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