Copernicus Climate Change Service Global Land and Marine Observations Database

First version cdm. Data sources

**Issued by:** NUIM / Peter Thorne

**Date:** 31/12/2018

**Ref:**C3S\_D311a\_Lot2.3.4.4-2018\_201812\_ Marine\_code\_v1

**Official reference number service contract:** 2017/C3S\_311a\_Lot2\_NUIM/SC1

Contributors

natural environment research council, National oceanography centre (NERC noc)

1. David I. Berry

2. Elizabeth C. Kent

natIONAL UNIVERSITY OF IRELAND MAYNOOTH

1. Peter Thorne

2. Corinne Voces

National oceanic and atmospheric administration’s national centers for environmental information

1. Eric Freeman

Table of Contents

[2 Introduction 5](#_Toc16060161)

[3 References 6](#_Toc16060162)

[4 Data sources mapping 7](#_Toc16060163)

[4.1 ICOADS R3.0.0.0T generic (icoads\_r3000) 7](#_Toc16060164)

[4.1.1 Header table 7](#_Toc16060165)

[4.1.2 Observations tables 8](#_Toc16060166)

[4.2 ICOADS R3.0.0T with supplementals 9](#_Toc16060167)

[4.2.1 MEDS - Canadian drifting buoys (icoads\_r3000\_d714) 9](#_Toc16060168)

Executive Summary

The C3S 311a Lot 2 (Global Land and Marine Observations Database) service is concerned with the provision of globally available land and marine surface meteorological records. The service includes inventorying of, and brokering access to, data sources, their harmonization (via conversion to a Common Data Model (CDM), merging, and quality assurance) and their provision via the Copernicus Climate Change Service (C3S) Climate Data Store (CDS).

The version history is given below:

|  |  |  |
| --- | --- | --- |
| **Version** | **Release Date** | **Release notes** |
| 1.0 |  |  |
| 2.0 |  |  |

# Introduction

The Copernicus Climate Change Service (C3S) Global Land and Marine Observations Database service provides brokered access to global historical holdings of surface meteorological observations. It builds upon existing national, regional and global efforts to create an augmented set of quality assured holdings that can be used to create datasets, products and services.

**This document will contain relevant detailed information on the mappings available in the CDM tool**

# Data sources mapping

The following mappings reflect the **initial** mapping from a specific data source. Merging with additional data sources is understood to happen independently from the initial mapping.

## ICOADS R3.0.0.0T generic (icoads\_r3000)

Model status: built from beta release following last CDM definition available in GLAMOD GitHub, august-sept19 and adding corrections to beta release following main.py and main\_sst\_only.py in /gws/nopw/j04/c3s311a\_lot2/data/level1/marine/sub\_daily\_data/IMMA1\_R3.0.0T/working/python/.

All keys: if they were included in beta release mapping but nothing was defined for them, that CDM element is deleted from the mapping file in this version, as it is understood not to be directly applicable from the source….(the corresponding <table>.psv file will have the field as null) and might be afterwards fill in by external meta data (pub47….)

ALL fields (numeric): check that the decimal\_places are configured according to the CDM field specifications. This is used to set the printed table version number of decimal places.

### Header table

Nulls in beta release:

* region
* sub\_region
* station\_name
* location\_method
* location\_quality: null in beta release. Now set to default not checked(3)
* station\_heading
* height\_of\_station\_above\_sea\_level\_accuracy
* sea\_level\_datum
* report\_time\_quality: null in beta release. Now set to default not checked (2)
* report\_time\_reference
* profile\_id
* events\_at\_station
* duplicates
* processing\_codes
* processing\_level

Other issues and fields that require special attention:

* **platform\_type**: in beta release, we did imma fixing via CDM mapping, using a fill\_value = 2. Now that we fix imma pt prior to mapping, and then select, we delete the fill\_value, so that changes are controlled by dck in the fixing module.
* **platform\_sub\_type**: code mapping only defined for icoads platform type *7*, with no default value. cdm\_latest.pdf has a code table for this, and subtype 69, unspecified drifting buoy, is for platform\_type 5, which I suppose is a typo in the document.
* **primary\_station\_id\_scheme**: code mapping only defined for platform type *7*, with default value to 5, that I have removed for the buoys mapping.

Should we apply here the per sid-dck general classification in deck\_list\_1946\_onwards.txt. Can this be potentially changed after id fixing, linkage however and where ever that is applied?

* **location\_accuracy**: note that from the beta release we are assuming that it is an integer as we are rounding to 0 decimal places and taking a min value of 1, although in doc is stated as numeric, not int specifically. Following this, we now add a 0 number of decimal places to print.
* **location**: this was added in the very last minute to beta release and it is not in cdm\_latest.pdf, so it is not included in the current mapping. Apparently this is something that is only internal to the DB, not a CDM field. It is therefore foreseen to add this column only in level2….
* correction for beta release was: **source\_id** = "ICOADS-3-0-0T-" + sid.zfill(3) + "-" + dck.zfill(3) + "-" + str(year) + "-" + str(month) -> zfill(2) is not specified in month, but I am going to do it that way….
* added code table mapping for **station\_course**. This mapping from code-key to numeric was not included initially and only performed in the fixes afterwards.
* **station\_speed**: was initially set to the icoads code, then mapped in Dave’s corrections. But icoads code tables are different from 1968-9 and we are now crossing this point. There is a ship\_speed\_knots table in the cdm code\_tables mappings that is a range table (do not remember where it comes from…. ☺ ).

But this needs to map to m/s values and is this applicable to drifting buoys?

imma station\_speed codes mapped to CDM as the m/s mid-point equivalent. Decimal places set to 1, as 0.1 is already 0.36 km/h. Ok. tested and seems to be working!

* **report\_time\_accuracy**: as with location accuracy, set to decimal\_places:0 on printing
* **report\_time\_quality**: following ago-sept discussions for release 1, qc flags were to be initialized to not checked, so that it is clear after qc flags merging that duplicates are not checked. Is this ever checked?

We currently set this to not checked (2) as it is a quality flag. Should it be missing (3), for this field?

* **history**: following discussions on data lineage, I change this from varchar to varchar[]\* and add as first item: "yyyy-mm-dd hh:mm:ss, Initial conversion from ICOADS R3.0.0T" and "yyyy-mm-dd hh:mm:ss, Initial conversion from ICOADS R3.0.0T with supplemental data recovery" for buoys…

Should record\_timestamp be a sequence of datetimes same length as history and in the history only the corresponding description?

Anyway if this has not been discussed and the corresponding field in the DB is not like array, then we have to leave this as a plain varchar….

* **height\_of\_station\_above\*:** to 3 decimal places default (mm, to much, but…)

### Observations tables

Null in beta release and not included in the mapping:

* z\_coordinate
* observation\_height\_above\_stations\_surface
* secondary\_variable
* secondary\_value
* code\_table
* location\_method
* z\_coordinate\_method
* bbox\_min\_longitude
* bbox\_max\_longitude
* bbox\_min\_latitude
* bbox\_max\_latitude
* sensor\_id
* original\_code\_table
* processing\_code
* adjustment\_id

Other issues/notes:

* same as header table when same field
* **location\_precision / accuracy**?: currently as location\_precision
  + GitHub table\_definitions (and cdm\_latest): location\_precision
  + Equivalent in header table (GitHub): location\_accuracy
  + Beta release as location\_accuracy….

Dave says that although it is inconsistent, need to be as defined in CDM: different between header and observations, until naming is corrected / checked for consistency.

Added decimal\_places:0 as in header table

* **quality\_flag**: from default 3 (missing) to default 2 (not checked)
* **numerical\_precision, original\_precision**:

This has been “historically” derived from the imma field “scale” as 0.1 for 1 decimal places, 0.01 for two, etc….but now, it can also be dck dependent as we might want to add specific information that is deck-dependent

* + original\_precision: Original repor􏰀ng precision in units given by ’original\_units’
  + numerical\_precision: Repor􏰀ng precision of observa􏰀on in units given by ’units’ variable. E.g. 0.1 = reported to nearest tenth, 0.5 to nearest half etc.

*Last update: these will be added in a per sid-dck-yr basis do not know where and it is going to default to null. So basically, we do not declare it until we have something to do with it.*

*Decimal places to write out, same policy as before….independent from the actual value of these precissions.*

* all **units**, **original\_units**: we now declare those as strings (“varchar”) as they are codes from a code table. We might need to apply this to all codes…we change this in the table definition to be ‘varchar’ also, as opposed to ‘int’ when downloaded from github. So far, we will only change this for codes that are cero padded, like units (001,002,etc…)
* **sst original units** changed to 530 in beta release fixes, but this is hectopascals, we change that to ‘060’: Celsius
* **sst conversion\_method** changed to 7 in beta release fixes, but this applies to hectopascals, we keep that as 1. But 1-‘060’ or 1-‘005’ not in pdf conversion method table….
* **wd-original units:** meds buoys, specifically states relative to true north, should units be 320 (dregrees true) instead of 110 (degrees (angle))? Not changed from original, still 110
* **ws,wd-conversion\_flag:** 0 for beta release, changed to 2 (both in m/s!!!)

## ICOADS R3.0.0T with supplementals

### MEDS - Canadian drifting buoys (icoads\_r3000\_d714)

Model status: same comments and issues as icoads\_r3000

Now the mapping is complete, using IMMA1 core and attachments or supplemental data where required.

header.primary\_station\_id\_scheme fill\_value is 4 now