JRadar

A Java application to transform HF Radar data files to the European standard

This project has received funding from the European Commission's Horizon 2020 Research and Innovation programme under grant agreements No 871153 and 951799.

AZTI
MEMBER OF
BASQUE RESEARCH
& TECHNOLOGY ALLIANCE

Date:

October 18, 2023

About this software

THE SOFTWARE IS PROVIDED UNDER THE LICENSE

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

AND "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

You will find more information about the license in the next URL.

http://creativecommons.org/licenses/by-nc-sa/4.0/

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

You are free to:

- **Share —** copy and redistribute the material in any medium or format.
- **Adapt** — remix, transform, and build upon the material.

  The licensor cannot revoke these freedoms if you follow the license terms.

Under the following term:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **NonCommercial** — You may not use the material for commercial purposes.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

- **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Dependencies

JRadar application has been developed using Java 8 as programming language. It is mandatory to install a Java Runtime Environment compatible with it in the computer running the application.

https://www.oracle.com/java/

Adding to that, JRadar requires the 'NetCDF-C' software library developed by Unidata to be installed in the computer to work properly. The library can be downloaded and installed from the next URL.

https://www.unidata.ucar.edu/downloads/netcdf/index.jsp

As it is developed using Java, it can be executed under Windows, Linux or Mac OS, but you must consider that the Unidata library is platform dependent. The same JRadar application works in any platform, but you must install a different Unidata C library in each platform.

Index

About this software 3
Dependencies 4
Introduction 6
Current status and future work 7
Installation, execution and description of the files 8
Description of JRadar 9
Main window 10
Select CODAR File 12
View CODAR Data 13
Run Quality Tests 17
Load and display metadata 22
Load and save a profile 23
Save as NetCDF 26
Console mode; running JRadar using commands 27
Credits 29

Introduction

JRadar is an application developed to transform raw text HF Radar files to the European NetCDF classic format v.4 defined under the INCREASE directive. This document is a user guide to explain the functionality of the software.

JRadar provides an easy-to-use user interface to edit the mandatory and some recommended parameters and attributes needed to create the NetCDF file and transform the data to NetCDF v.4 format.

Adding to that, it can be run with no user interface using the console mode, providing a way to automatize the transformation of multiple files with no user interaction.

For the information does not present in the document related to the metadata, data or format, you can consult the INCREASE, JERICO-Next, CMEMS-INSTAC & EUROSEA projects.

For information about the HF radar input file, look up CODAR radar system data format.

For information about the software itself, you can check the credits at the end of the document to contact.

Current status and future work

The JRadar software tool is now on 2.0.0 version. Now, it transforms completely CODAR Radial files to European Standard. CODAR Total files are transformed with some limitations related to the information CODAR provide and WERA files are not supported yet.

We are working to success in the next points soon.

- Complete WERA radar files transformation.
- Functionality to add and manipulate attributes and parameters do not present in the standard if the user wants to.
- Read and transform NetCDF files to European standard.
- Create total NetCDF files using radial file information.

Installation, execution and description of the files

The software does not require an installation itself. Once you have the compressed archive containing all the files that compose the JRadar application, you must decompress it in any place you have reading, writing and execution permissions. Once the file has been decompressed, it is ready to be executed.

After decompression, a JRadar folder with the next structure will be created:

- **examples**: directory with some example files of CODAR text files for radial and total data.
- **transform**: directory with some transformation example files. All of them have been created using JRadar.
- **README.pdf**: this file.
- **codar.properties**: file containing the properties used by JRadar to read the CODAR text files. It is a mapping between the data within the CODAR file and the properties of the Java objects we create during the JRadar application execution.
- **logoVentana.png**: icon for the UI application.
- **JRadar.jar**: executable file, the application.
- **JRadar.log**: application log. If something goes wrong, the information is written down in the file.
- **example.radial**: a radial profile example
- **example.total**: a total profile example

To run the JRadar software, double click in JRadar.jar file and the main window will appear. If there are problems with the paths, it can be run using the console, once positioned on the JRadar folder with the command:

java –jar JRadar.jar

To run with no user interface, please refer to the chapter describing the console mode.

Description of JRadar

The Java Radar data transformer software –JRadar- can be executed in two different ways:

- **User Interface mode (UI):** the JRadar.jar file must be executed, double clicking it. UI mode is composed of different windows.
- **Console mode:** commands can be executed, typing instructions in a console.

Using the UI mode, JRadar will show the user a main window with the different options to load, save, modify, and check the radial and total radar files before trying the transformation to the NetCDF European common standard for HFR data and metadata.

Using the console mode JRadar runs the application and makes all needed checks and transformation to convert the input file passed as parameter to a NetCDF file. If something does not work properly, the output file is not created, and a warning or error message will be show in the console as in the log file.

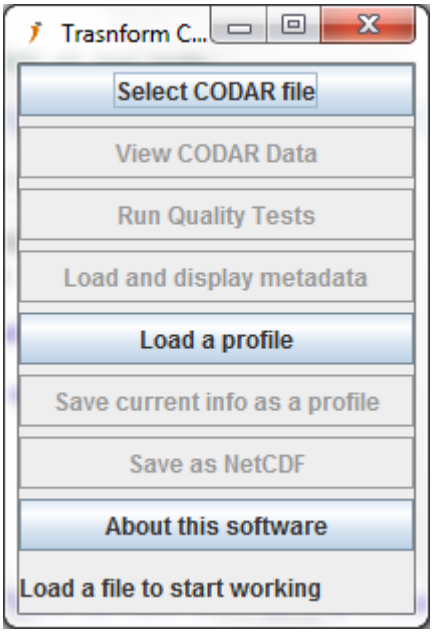
Adding to that, to avoid problems due to missing information in the input file needed in the standard, the JRadar software provides a functionality to save different profiles for radial and total transformations. The profiles are used as default values when the information we want to save in the NetCDF file is not present in the origin radar text file. In this way, we can transform multiple files with a missing mandatory parameter, using a profile to load the missing value from it for all the files.

The profile must be created using the UI mode of the JRadar tool, and it is mandatory to have a profile created to run the console mode of the JRadar application.

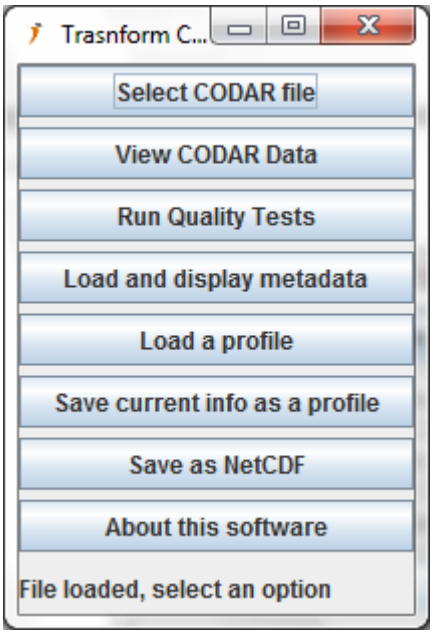
All these functionalities are described in depth in the next chapters of the document.

Main window

To execute JRadar application in the UI mode, we can double click the JRadar.jar file in our file system. It can be run using a command too, via console. Type 'java –jar JRadar.jar' once located in the JRadar folder with access to the JRAdar.jar file.



Some options are only available once a data file has been loaded (we cannot view the data if we have not loaded a data file for example). Once we have loaded one data file, the same panel looks like:

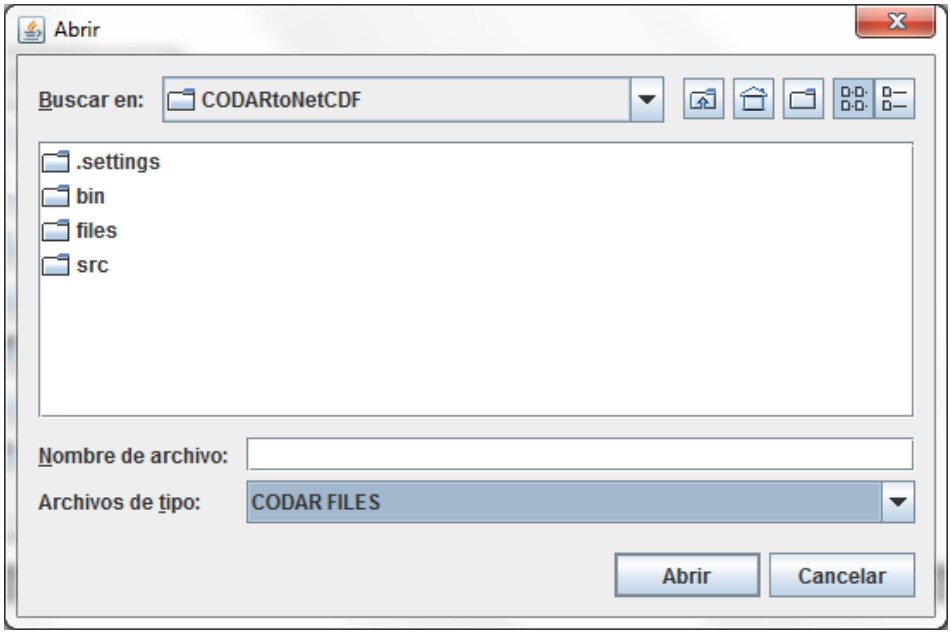


Within the main window we will find 8 buttons, and a little display. The description of the different options is:

- **Select CODAR file**: this option opens a new window and in the new window the user can navigate to select a total or radial CODAR file.
- **View CODAR Data**: Once the CODAR file has been selected, this option allows the user displaying the information loaded.
- **Run Quality Tests**: This option opens a new window with different test available to run. The tests are different for radial files and total files.
- **Load and display metadata**: This option asks for Network and Station id information to load data from the European Node database.
- **Load a profile**: We can load a previously saved profile information to have it available if necessary.
- **Save current info as a profile**: If we have all the information ready, with no missing mandatory data to create a NetCDF file, we can save the information as a profile to have it available for other transformations.
- **Save as NetCDF**: Stores the current information as a NetCDF using the European standard.
- **About this software**: Credits and documentation of the software
- **Last row display**: It shows an info message related to the last user action.

Select CODAR File

Once we have the main window visible in our computer, the first button allows us to load a CODAR file.



We can choose between CODAR radial files (*.ruv files) or CODAR total files (*.tuv files), the same menu allows us to load any of them, only one type each time.

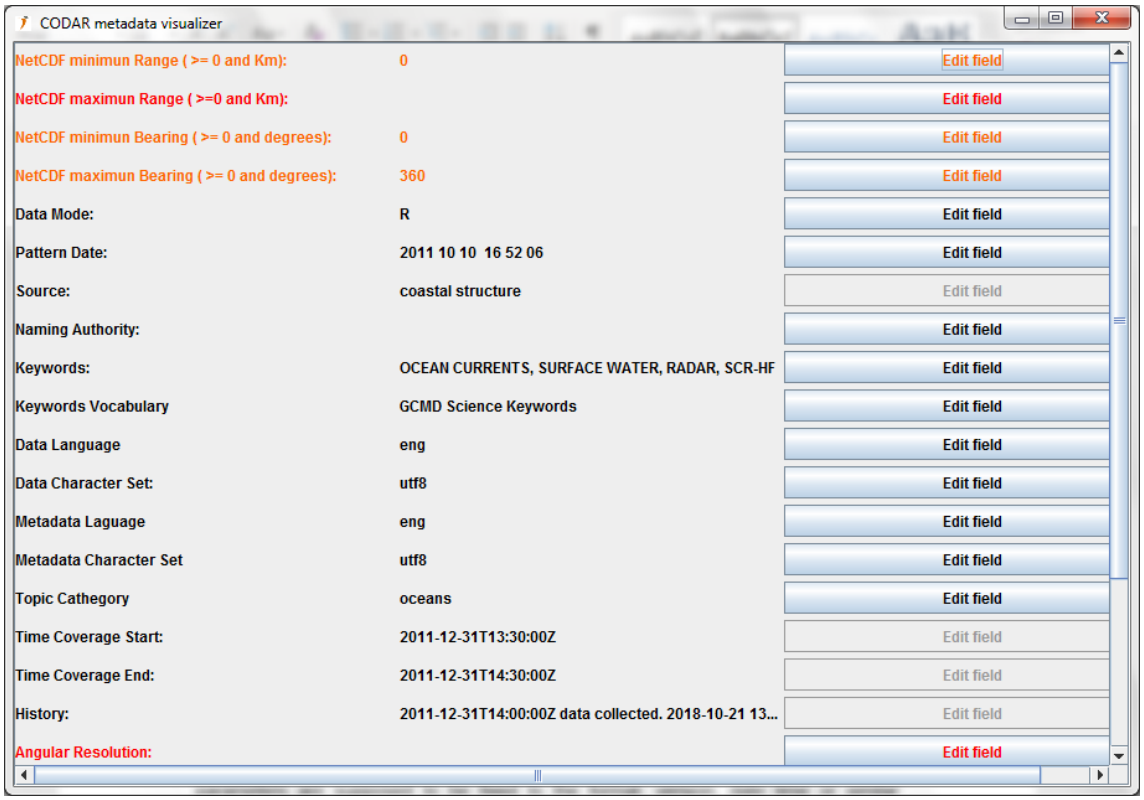
Once we have loaded a file, the info panel at the bottom of the main window, will type the next message: “File loaded, select an option” that means that the file is correctly parsed and loaded.

We can click again the same button to load another one, but we will lose the information about the first one. We can only load one file each time.

View CODAR Data

The second option of the menu is a display of the information present in the loaded CODAR file. The display is similar but different for radial files and total files.

Example of radial file



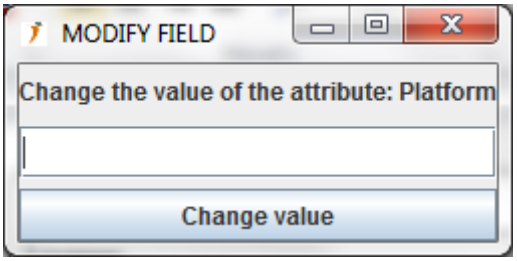
The user will find a long table with parameters. The first column is the name of the attribute. The second column is the value of the attribute. The third and last column is a button to edit, modify or delete the value.

The data is printed in three colors depending on the status.

- **Black**: the data is correct, or it is not mandatory.
- **Red**: there is a problem with the data, and it is mandatory.
- **Orange**: warning, there is a problem with a mandatory data, but we can fix it using a loaded profile. This will be explained in the profile functionality.

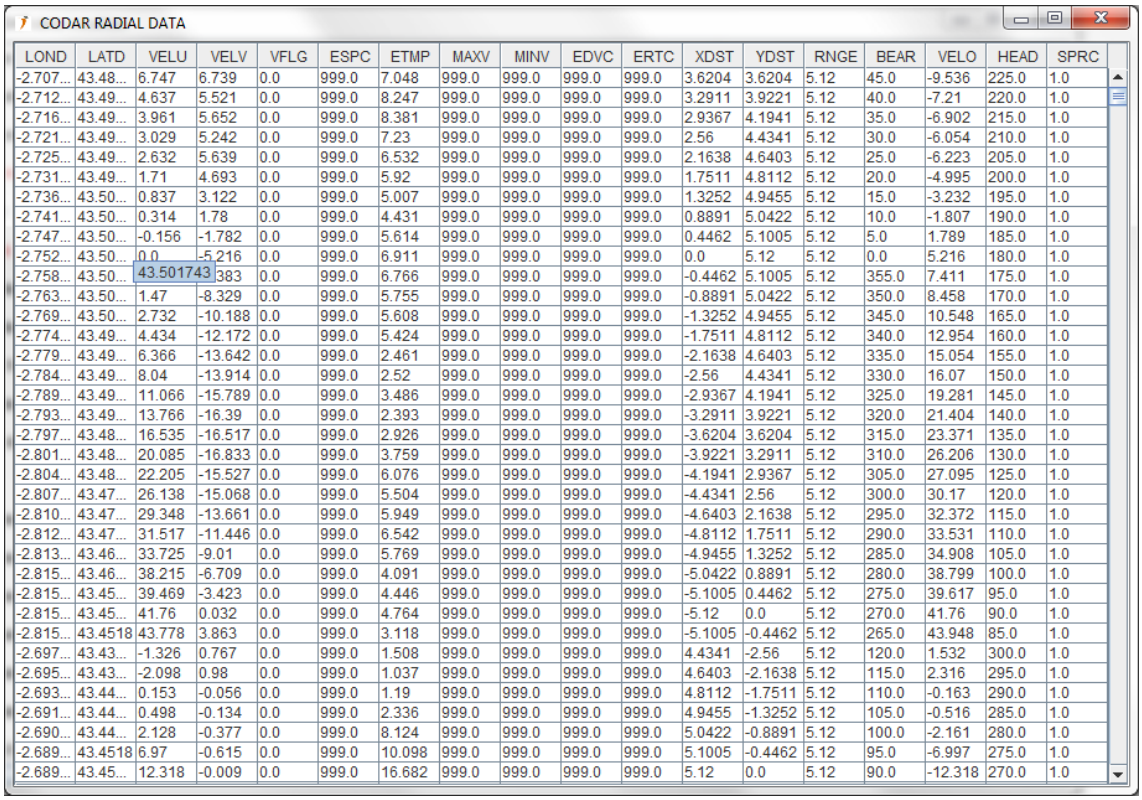
The first four parameters are related to the grid we want to create within the NetCDF radial file. They define the limits of bearing and range for radials and x and y for totals.

If we click the “Edit field” button, a new window will help us changing the value. Once we type the information and click the “Change Value” button, the information is updated in the data visualizer, and the color will change if needed.



If the button is disabled, the value is fixed, and it is not possible to modify it.

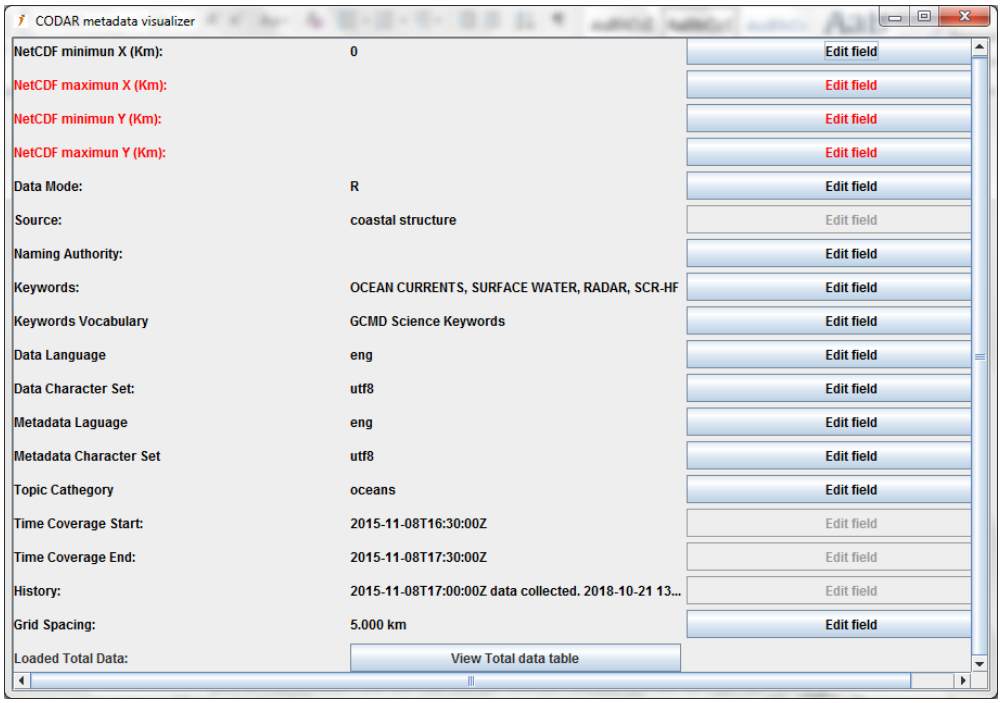
For radial files, the last row button named “Radial Data table”, is a little bit different. It doesn’t allow users to modify information but shows the radial data within the CODAR file. Clicking it the next window will appear.

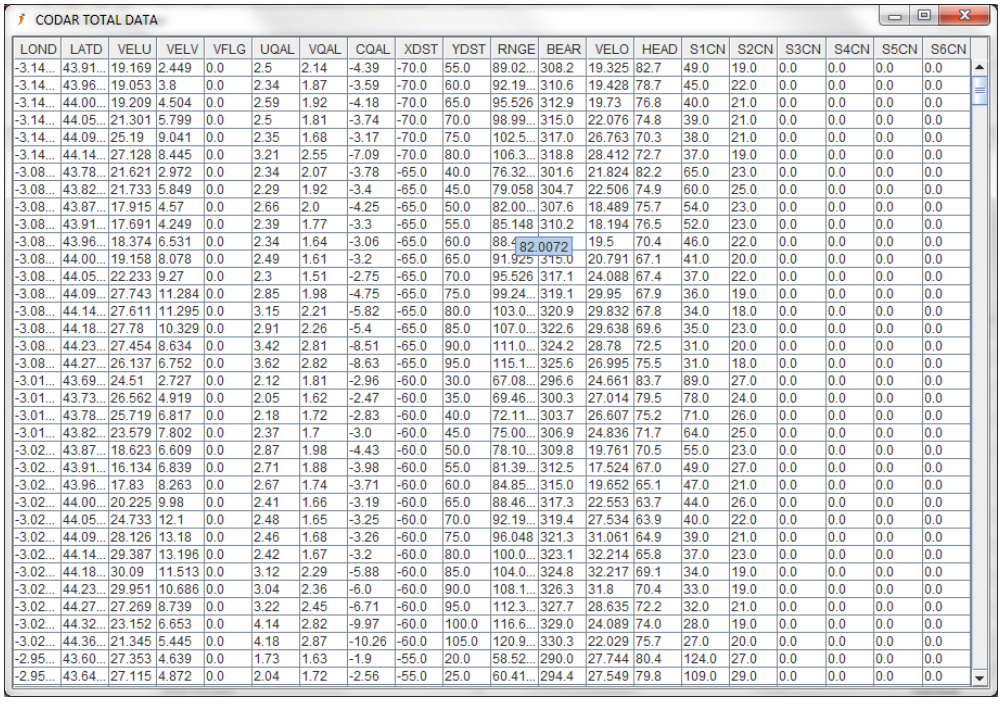

CODAR RADIAL DATA

LOND	LATD	VELU	VELV	VFLG	ESPC	ETMP	MAXV	MINV	EDVC	ERTC	XDST	YDST	RNGE	BEAR	VELO	HEAD	SPRC
-2.707...	43.48...	6.747	6.739	0.0	999.0	7.048	999.0	999.0	999.0	999.0	3.6204	3.6204	5.12	45.0	-9.536	225.0	1.0
-2.712...	43.49...	4.637	5.521	0.0	999.0	8.247	999.0	999.0	999.0	999.0	3.2911	3.9221	5.12	40.0	-7.21	220.0	1.0
-2.716...	43.49...	3.961	5.652	0.0	999.0	8.381	999.0	999.0	999.0	999.0	2.9367	4.1941	5.12	35.0	-6.902	215.0	1.0
-2.721...	43.49...	3.029	5.242	0.0	999.0	7.23	999.0	999.0	999.0	999.0	2.56	4.4341	5.12	30.0	-6.054	210.0	1.0
-2.725...	43.49...	2.632	5.639	0.0	999.0	6.532	999.0	999.0	999.0	999.0	2.1638	4.6403	5.12	25.0	-6.223	205.0	1.0
-2.731...	43.49...	1.71	4.693	0.0	999.0	5.92	999.0	999.0	999.0	999.0	1.7511	4.8112	5.12	20.0	-4.995	200.0	1.0
-2.736...	43.50...	0.837	3.122	0.0	999.0	5.007	999.0	999.0	999.0	999.0	1.3252	4.9455	5.12	15.0	-3.232	195.0	1.0
-2.741...	43.50...	0.314	1.78	0.0	999.0	4.431	999.0	999.0	999.0	999.0	0.8891	5.0422	5.12	10.0	-1.807	190.0	1.0
-2.747...	43.50...	-0.156	-1.782	0.0	999.0	5.614	999.0	999.0	999.0	999.0	0.4462	5.1005	5.12	5.0	1.789	185.0	1.0
-2.752...	43.50...	0.0	-5.216	0.0	999.0	6.911	999.0	999.0	999.0	999.0	0.0	5.12	5.12	0.0	5.216	180.0	1.0
-2.758...	43.50...	43.501743	383	0.0	999.0	6.766	999.0	999.0	999.0	999.0	-0.4462	5.1005	5.12	355.0	7.411	175.0	1.0
-2.763...	43.50...	1.47	-8.329	0.0	999.0	5.755	999.0	999.0	999.0	999.0	-0.8891	5.0422	5.12	350.0	8.458	170.0	1.0
-2.769...	43.50...	2.732	-10.188	0.0	999.0	5.608	999.0	999.0	999.0	999.0	-1.3252	4.9455	5.12	345.0	10.548	165.0	1.0
-2.774...	43.49...	4.434	-12.172	0.0	999.0	5.424	999.0	999.0	999.0	999.0	-1.7511	4.8112	5.12	340.0	12.954	160.0	1.0
-2.779...	43.49...	6.366	-13.642	0.0	999.0	2.461	999.0	999.0	999.0	999.0	-2.1638	4.6403	5.12	335.0	15.054	155.0	1.0
-2.784...	43.49...	8.04	-13.914	0.0	999.0	2.52	999.0	999.0	999.0	999.0	-2.56	4.4341	5.12	330.0	16.07	150.0	1.0
-2.789...	43.49...	11.066	-15.789	0.0	999.0	3.486	999.0	999.0	999.0	999.0	-2.9367	4.1941	5.12	325.0	19.281	145.0	1.0
-2.793...	43.49...	13.766	-16.39	0.0	999.0	2.393	999.0	999.0	999.0	999.0	-3.2911	3.9221	5.12	320.0	21.404	140.0	1.0
-2.797...	43.48...	16.535	-16.517	0.0	999.0	2.926	999.0	999.0	999.0	999.0	-3.6204	3.6204	5.12	315.0	23.371	135.0	1.0
-2.801...	43.48...	20.085	-16.833	0.0	999.0	3.759	999.0	999.0	999.0	999.0	-3.9221	3.2911	5.12	310.0	26.206	130.0	1.0
-2.804...	43.48...	22.205	-15.527	0.0	999.0	6.076	999.0	999.0	999.0	999.0	-4.1941	2.9367	5.12	305.0	27.095	125.0	1.0
-2.807...	43.47...	26.138	-15.068	0.0	999.0	5.504	999.0	999.0	999.0	999.0	-4.4341	2.56	5.12	300.0	30.17	120.0	1.0
-2.810...	43.47...	29.348	-13.661	0.0	999.0	5.949	999.0	999.0	999.0	999.0	-4.6403	2.1638	5.12	295.0	32.372	115.0	1.0
-2.812...	43.47...	31.517	-11.446	0.0	999.0	6.542	999.0	999.0	999.0	999.0	-4.8112	1.7511	5.12	290.0	33.531	110.0	1.0
-2.813...	43.46...	33.725	-9.01	0.0	999.0	5.769	999.0	999.0	999.0	999.0	-4.9455	1.3252	5.12	285.0	34.908	105.0	1.0
-2.815...	43.46...	38.215	-6.709	0.0	999.0	4.091	999.0	999.0	999.0	999.0	-5.0422	0.8891	5.12	280.0	38.799	100.0	1.0
-2.815...	43.45...	39.469	-3.423	0.0	999.0	4.446	999.0	999.0	999.0	999.0	-5.1005	0.4462	5.12	275.0	39.617	95.0	1.0
-2.815...	43.45...	41.76	0.032	0.0	999.0	4.764	999.0	999.0	999.0	999.0	-5.12	0.0	5.12	270.0	41.76	90.0	1.0
-2.815...	43.4518	43.778	3.863	0.0	999.0	3.118	999.0	999.0	999.0	999.0	-5.1005	-0.4462	5.12	265.0	43.948	85.0	1.0
-2.697...	43.43...	-1.326	0.767	0.0	999.0	1.508	999.0	999.0	999.0	999.0	4.4341	-2.56	5.12	120.0	1.532	300.0	1.0
-2.695...	43.43...	-2.098	0.98	0.0	999.0	1.037	999.0	999.0	999.0	999.0	4.6403	-2.1638	5.12	115.0	2.316	295.0	1.0
-2.693...	43.44...	0.153	-0.056	0.0	999.0	1.19	999.0	999.0	999.0	999.0	4.8112	-1.7511	5.12	110.0	-0.163	290.0	1.0
-2.691...	43.44...	0.498	-0.134	0.0	999.0	2.336	999.0	999.0	999.0	999.0	4.9455	-1.3252	5.12	105.0	-0.516	285.0	1.0
-2.690...	43.44...	2.128	-0.377	0.0	999.0	8.124	999.0	999.0	999.0	999.0	5.0422	-0.8891	5.12	100.0	-2.161	280.0	1.0
-2.689...	43.4518	6.97	-0.615	0.0	999.0	10.098	999.0	999.0	999.0	999.0	5.1005	-0.4462	5.12	95.0	-6.997	275.0	1.0
-2.689...	43.45...	12.318	-0.009	0.0	999.0	16.682	999.0	999.0	999.0	999.0	5.12	0.0	5.12	90.0	-12.318	270.0	1.0

Moving the mouse over the cells, it shows the complete value of the cell as shown in the figure above. This information cannot be modified, just visualized. This table is different too for Total files.

Example for total files:





The modification of the data in the visualization panel does not modify the original loaded CODAR file. The changes are only applied to the output file.

Run Quality Tests

The mandatory tests defined for the European standard and the meaning of the result values of running them are:

For Radial files:

QC test	Meaning	QC variable type
Syntax	This test will ensure the proper formatting and the existence of all the necessary fields within the radial netCDF file. This test is performed on the netCDF files and it assesses the presence and correctness of all data and attribute fields and the correct syntax throughout the file.	N/A, it is a test on the netCDF file structure, not on data content.
Over-water	This test labels radial vectors that lie on land with a “bad data” flag and radial vectors that lie on water with a “good data” flag.	gridded
Velocity Threshold	This test labels radial velocity vectors whose module is bigger than a maximum velocity threshold with a “bad data” flag and radial vectors whose module is smaller than the threshold with a “good data” flag.	gridded
Variance Threshold	This test labels radial vectors whose temporal variance is bigger than a maximum threshold with a “bad data” flag and radial vectors whose temporal variance is smaller than the threshold with a “good data” flag. This test is applicable only to Beam Forming (BF) systems. Data files from Direction Finding (DF) systems will apply instead the “Temporal Derivative” test reporting the explanation “Test not applicable to Direction Finding systems. The Temporal Derivative test is applied.” in the comment attribute.	gridded
Temporal Derivative	For each radial bin, the current hour velocity vector is compared with the previous and next ones. If the differences are bigger than a threshold (specific for each radial bin and evaluated on the basis of the analysis of one-year-long time series), the present vector is flagged as bad_data, otherwise it is labelled with a good_data flag. Since this method implies a one-hour delay in the data provision, the current hour file should have the related QC flag set to 0 (no QC performed) until it is updated to the proper values when the next hour file is generated.	gridded
Median Filter	For each source vector, the median of all velocities within a radius of <RCLim> and whose vector bearing (angle of arrival at site) is also within an angular distance of <AngLim> degrees from the source vector's bearing is evaluated. If the difference between the vector's velocity and the median velocity is greater than a threshold, then the vector is labelled with a “bad_data” flag, otherwise it is labelled with a “good_data” flag.	gridded
Average Radial Bearing	This test labels the entire datafile with a ‘good_data” flag if the average radial bearing of all the vectors contained in the data file lies within a specified margin around the expected value of normal operation. Otherwise, the data file is labelled with a “bad_data” flag. The value of normal operation has to be defined within a time interval when the	scalar

	proper functioning of the device is assessed. The margin has to be set according site-specific properties. This test is applicable only to DF systems. Data files from BF systems will have this variable filled with "good_data" flags (1) and the explanation "Test not applicable to Beam Forming systems" in the comment attribute.	
Radial Count	Test labelling radial data having a number of velocity vectors bigger than the threshold with a "good data" flag and radial data having a number of velocity vectors smaller than the threshold with a "bad data" flag.	scalar

For Total Files:

QC test	**Meaning**	**QC variable type**
Syntax	This test will ensure the proper formatting and the existence of all the necessary fields within the total netCDF file. This test is performed on the netCDF files and it assesses the presence and correctness of all data and attribute fields and the correct syntax throughout the file.	N/A, it is a test on the netCDF file structure, not on data content.
Data Density Threshold	This test labels total velocity vectors with a number of contributing radials bigger than the threshold with a "good data" flag and total velocity vectors with a number of contributing radials smaller than the threshold with a "bad data" flag.	gridded
Velocity Threshold	This test labels total velocity vectors whose module is bigger than a maximum velocity threshold with a "bad data" flag and total vectors whose module is smaller than the threshold with a "good data" flag.	gridded
Variance Threshold	This test labels total vectors whose temporal variance is bigger than a maximum threshold with a "bad data" flag and total vectors whose temporal variance is smaller than the threshold with a "good data" flag. This test is applicable only to Beam Forming (BF) systems. Data files from Direction Finding (DF) systems will apply instead the "Temporal Derivative" test reporting the explanation "Test not applicable to Direction Finding systems. The Temporal Derivative test is applied." in the comment attribute.	gridded
Temporal Derivative	For each grid cell, the current hour velocity vector is compared with the previous and next ones. If the differences are bigger than a threshold (specific for each grid cell and evaluated on the basis of the analysis of one-year-long time series), the present vector is flagged as bad_data, otherwise it is labelled with a good_data flag. Since this method implies a one-hour delay in the data provision, the current hour file should have the related QC flag set to 0 (no QC performed) until it is updated to the proper values when the next hour file is generated.	gridded
GDOP Threshold	This test labels total velocity vectors whose GDOP is bigger than a maximum threshold with a "bad data" flag and the vectors whose GDOP is smaller than the threshold with a "good data" flag.	gridded

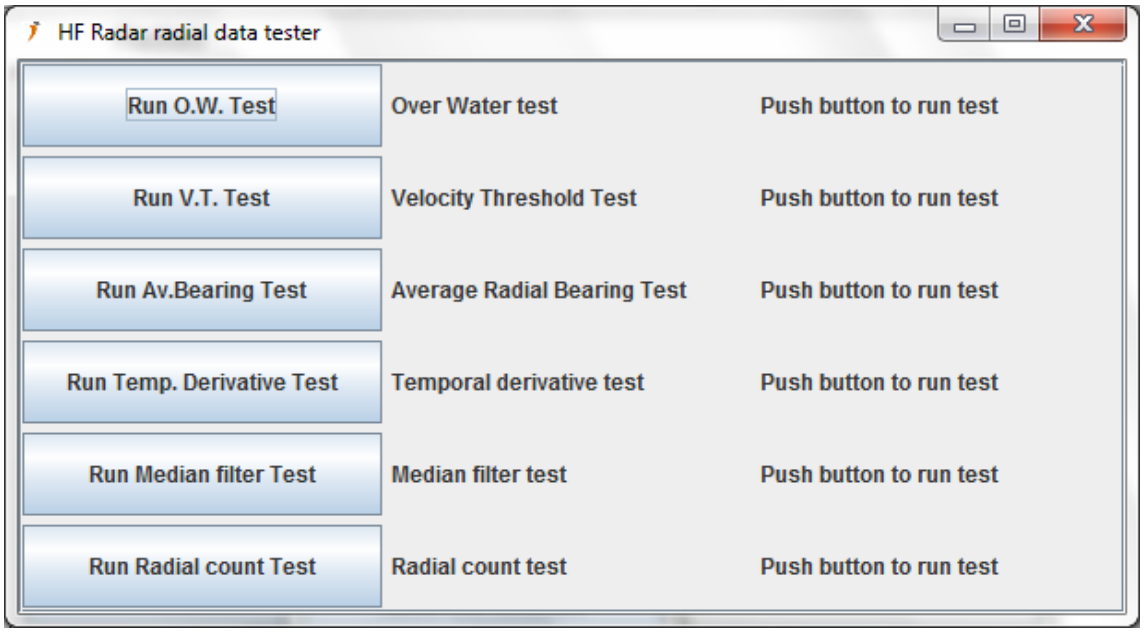
Output values:

The possible output values for the test, that are saved in the NetCDF file once the HF Radar text file is transformed are the ones of the ARGO QC flag scale:

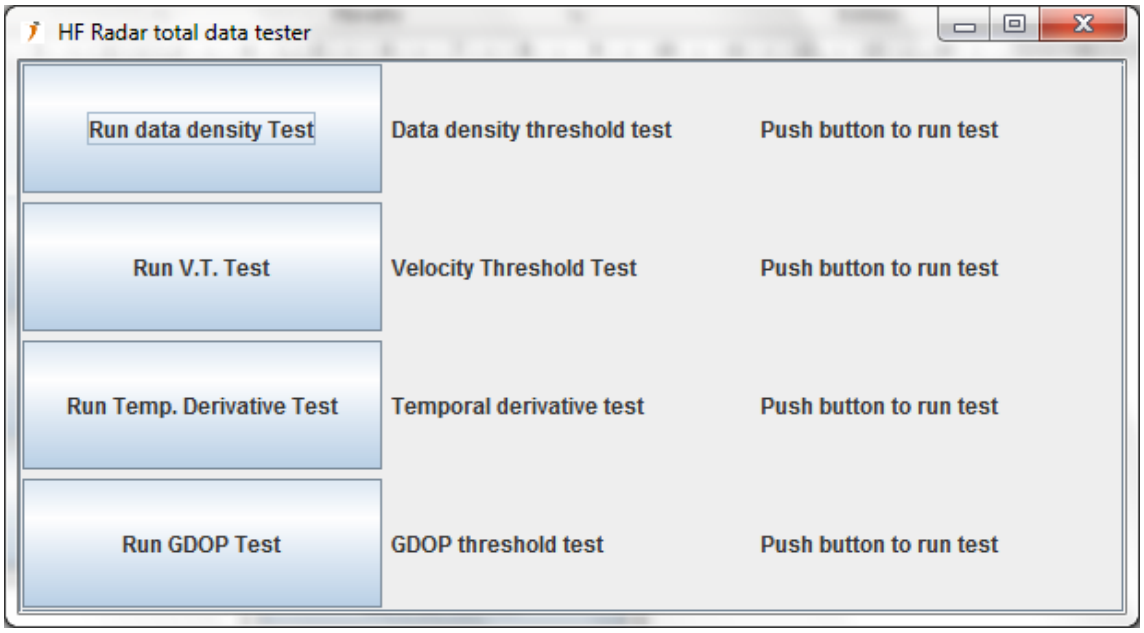
Code	Meaning	Comment
0	unknown	No QC was performed.
1	good data	All QC tests passed.
2	probably good data	
3	potentially correctable bad data	These data are not to be used without scientific correction or re-calibration.
4	bad data	Data have failed one or more QC tests.
5	-	Not used
6	-	Not used
7	nominal value	Data were not observed but reported (e.g., instrument target depth)
8	interpolated value	
9	missing value	

Clicking “Run Quality Tests” button, the application shows a menu with the different test available for each file type.

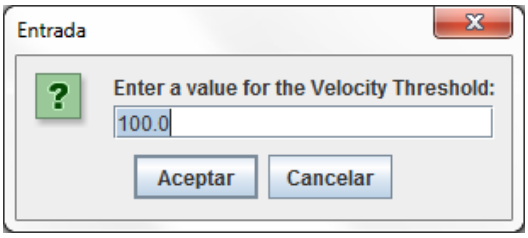
Radials



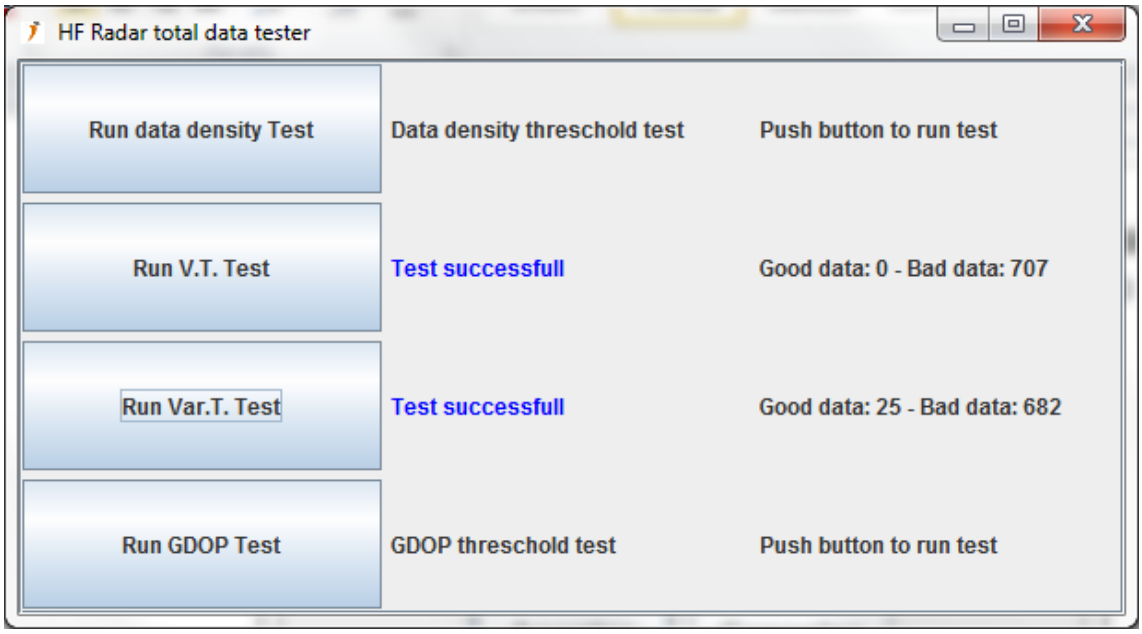
Totals



When we click a button, the test is run and if a value is needed to properly run the test, it is asked to the user. For example, when we try to run a threshold test, the maximum value is asked to the user:



Once we run the test, the result will appear in the test window, showing how many good and bad data are present according to the limits the user has typed. We can run the same tests with different threshold values, and we will have different results as expected.



The application stores always the last information the user provides. For example, if we run a velocity threshold test with a limit of 4 m/s, and then we run the same test with a limit of 5 m/s, the 5 is the only data that JRadar will remember for the file transformation.

Having bad results on a quality test does not prevent the user to save the file as NetCDF. The quality test is just an indicator of the data quality contained in the file according to the thresholds, but the transformation and data storage are always available if the minimum information needed according to the European Radar data format standard are provided.

Load and display metadata

The user can load information from the European HFR Node database. It is mandatory to register in the European HFR Node data base to run properly this software.

The data required to load the metadata are Network id and Station id. And once loaded, the information is displayed in a table, but changes are not allowed.

The parameters are related to the station and network and describe the metadata needed to create the NetCDF according to the European standard. The information is added to the loaded CODAR information, and it is used in the test quality tests and writing the NetCDF attributes.

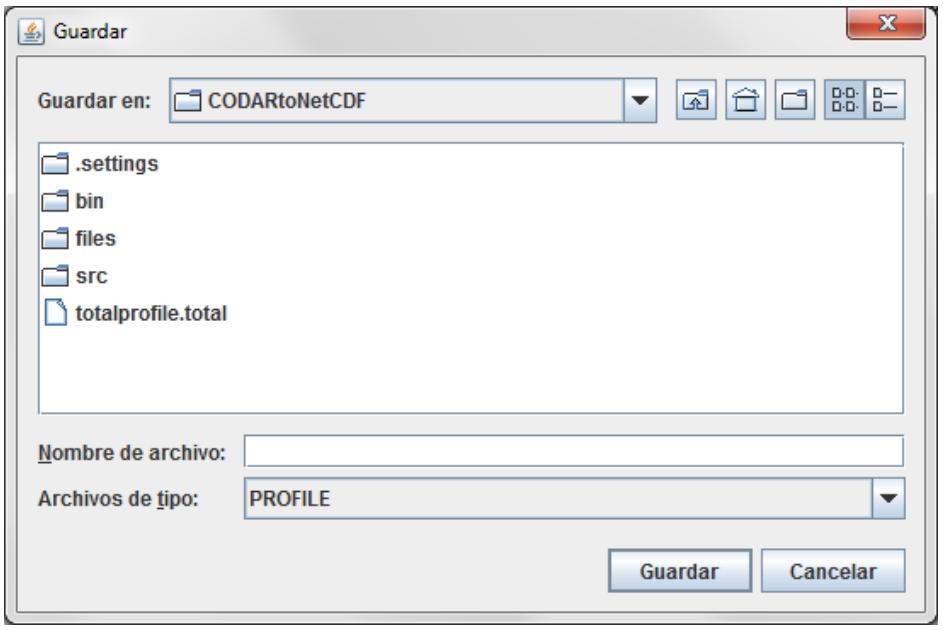
Load and save a profile.

The European standard for the HF Radar file format, defines as mandatory some information that not always is present in the original radar ascii data. In theory, users must modify the information using the edit field functionality every time they load a file in order to include the information.

Adding to that, most of the times, the information which is not included in the original ascii files are labels related to the platform names, parameters about the operators and /or other parameters which are known by the operators etc.

The profiles have been created to avoid the need to edit one by one the files we want to transform. Once we have completed all the information using JRadar window system, according to the European standard, and we have introduced threshold values, platform and site names etc., we can store all this information in a profile.

The information saved into a profile can be visualized when we click the “View CODAR Data” and the “Run Quality Tests” button. A profile file does not contain any of the data that we can visualize when clicking View Data Table button.

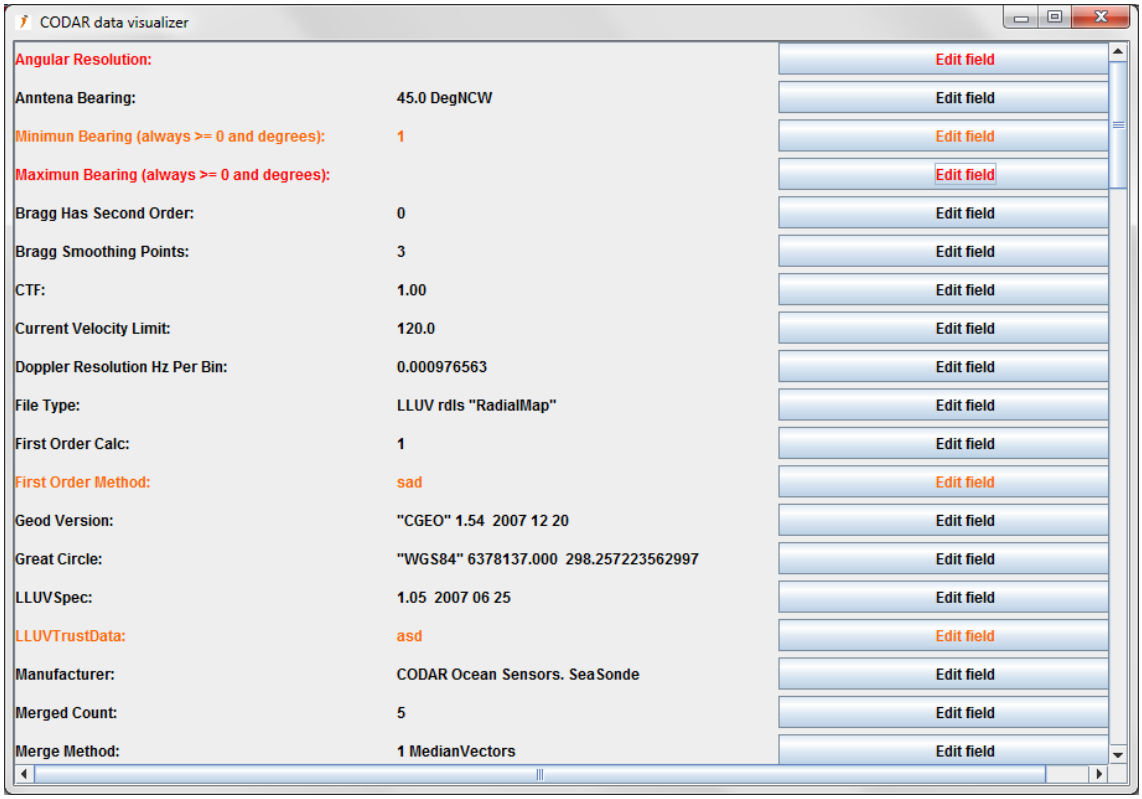


The radial radar information is stored with *.radial file extension, and total radar information with *.total file extension. The information stored is different, and the

profiles are not compatible. When we load a radial file, we can only use radial profiles. The same is applied to the totals.

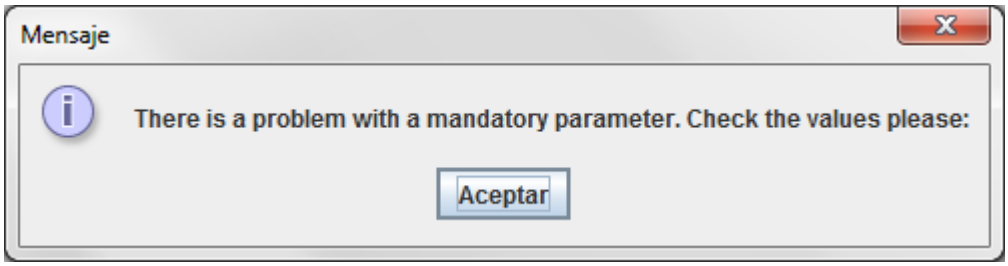
Once we have loaded a profile, if some mandatory data is missing, JRadar uses the profile info to fill it, but the information is presented in orange to the user, to realize that the information is not present in the original file.

The information of a profile can be used to create a different profile if the user wants to. For example, we can load a file, change some parameter, load a previously saved profile, and store the result as a new profile, using the old profile to full fill rest of mandatory parameters that are empty.



The profiles contain the information of the quality test too, so once we have a profile loaded, we can transform any file if compatible (radial files with radial profiles, and total files with total profiles).

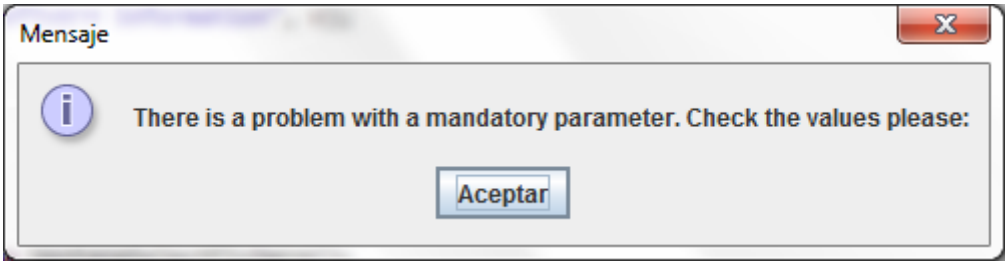
Trying to save a profile, the application runs a check in order to be sure that all the needed info is loaded or introduced. If something is missing, a warning is shown to the user and kindly asks to fulfill the mandatory parameters.

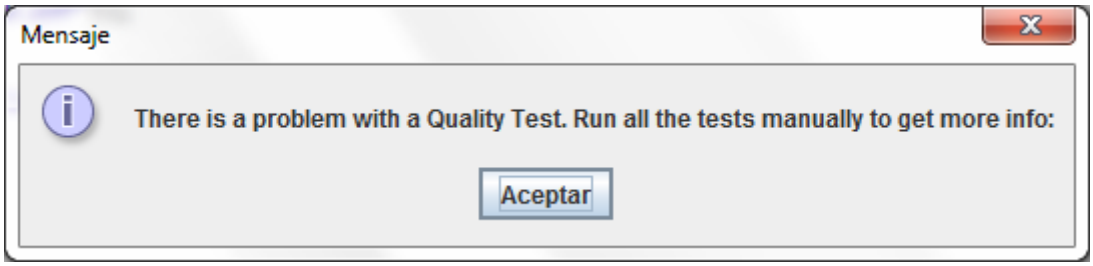

Mensaje
There is a problem with a mandatory parameter. Check the values please:
Aceptar

Save as NetCDF

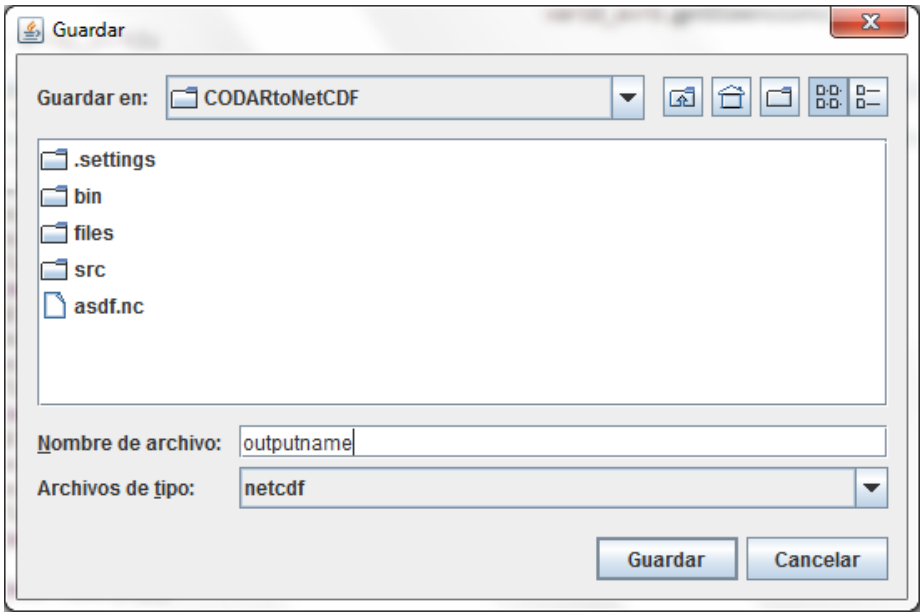
This option creates the NetCDF v4 format file containing all the information required by the European HF Radar standard.

Clicking the button runs a test that checks the mandatory parameters and quality control. If something is needed, the application asks the user to introduce missing parameters or running the needed functionalities.





If the preliminary test doesn't show any problem, the application asks a name for the NetCDF output file.



Once the name is typed, the file is saved in the path.

Console mode; running JRadar using commands

The previous description of the software is valid when we run de JRadar.jar executable file with no parameters. But the software is ready to run with no user interface to process big amounts of files automatically.

This mode of running the application (console mode) does not ask the user to fill data so all the information must be included in the command line that runs the application.

The way to run JRadar to work in console mode is typing:

java –jar JRadar.jar $attr1 $attr2 $attr3 $attr4

Where

- $attr1: must be “console”.
- $attr2: complete path to the input file or folder (input).
- $attr3: complete path to the output file or folder (output)
- $attr4: complete path to a profile (only one, radial or total).

The first parameter is a literal, an id to tell the JRadar that the execution is with no user interface. To run in background, the value must be “console”.

Parameter two can be a file or a directory. If it is a file, only that file will be transformed. If a directory, the complete directory (not recursively) will be scanned and processed. The application will look for radial (.ruv) and total (.tuv) files and if it finds any of them, will try to transform according to the profile that is in the last input parameter.

Third parameter can be a file (complete path) or a directory too. Depending on second parameter, there are 4 possibilities:

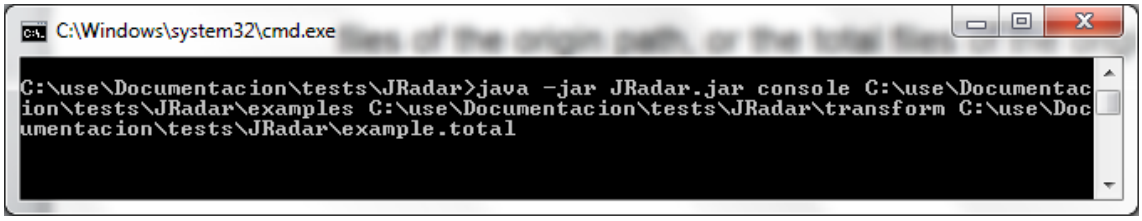
- $attr2 is a file and $attr3 is a file: the origin file is transformed, and the output file is stored with the name $attr3.

- $attr2 is a folder and $attr3 is a folder: all the origin files contained in the path described by $attr2 are transformed and the output files will be named as the original ones, with a NetCDF extension (.nc) and stored in the path described by $attr3.
- $attr2 is a file and $attr3 is a folder: the origin file is transformed, and the output file will be named as the original one, with a NetCDF extension (.nc) and stored in the path described by $attr3.
- $attr2 is a folder and $attr3 is a file: all the origin files contained in the path described by $attr2 are transformed and the output files will be named as the original ones, with a NetCDF extension (.nc) and stored in the parent folder of the file described by $attr3, at the same deep.

The fourth and last parameter must be a previously stored profile using the user interface of JRadar. This parameter as all the previous ones is mandatory. The radial and total profiles are used to transform radial or total files, and they are not interchangeable, so this parameter determines if we are transforming only the radial files of the origin path, or the total files of the origin path.

Command line example

java –jar JRadar.jar console C:\use\Documentacion\tests\JRADAR\examples
C:\use\Documentacion\tests\JRadar\transform
C:\use\Workspace\JRadar\example.total



Credits

The Java version of the software has been developed by *Jose Luis Asensio (v1.0.0) and by David Álvarez (v2.0.0)*

This software is an evolution of the **Matlab** version developed by *Lorenzo Corgnati*.

This software is the implementation of the European HF Radar Standard defined by INCREASE initiative. The contributors to this project are…

David Álvarez Peñaranda: dalvarez@azti.es
Julien Mader: jmader@azti.es
Anna Rubio: arubio@azti.es
Lohitzune Solabarrieta lsolabarrieta@azti.es
Lorenzo Corgnati: lorenzo.corgnati@sp.ismar.cnr.it
Jose Luis Asensio: txelu_ai@hotmail.com

Acknowledgments

This project has received funding from the European Commission's Horizon 2020 Research and Innovation programme under grant agreements No 871153 and 951799. Project coordinator: IFREMER, France. The information and views of this website lie entirely with the authors. The European Commission is not responsible for any use that may be made of the information it contains.