

ROPP v10 Beta Testing Report

Doc.No. : EUM/RSP/REP/20/1163206
Issue : v1
Date : 28 April 2020
WBS/DBS :

EUMETSAT
Eumetsat-Allee 1, D-64295 Darmstadt, Germany
Tel: +49 6151 807-7
Fax: +49 6151 807 555
<http://www.eumetsat.int>

Signature Table

	<i>Name</i>	<i>Function</i>	<i>Signature</i>	<i>Date</i>
Prepared by:	Christian Marquardt			
	Leonid Butenko			
Reviewed by:				
Approved by:				

Distribution List

<i>Name</i> (enter email address in case of external distribution)	<i>Organisation</i>
ROM SAF Project Team	

Change Record

<i>Version</i>	<i>Date</i>	<i>DCR* No.</i>	<i>Description of Changes</i>
v1 Draft	22/04/2020		Initial (draft) version.
v1	28/04/2020		Typos fixed, introduction and conclusions added.

****DCR = Document Change Request***

Table of Contents

1	INTRODUCTION	5
1.1	Scope	5
1.2	Links into EUMETSAT's GitLab repository	5
2	DOWNLOAD, INSTALLATION AND TESTS	6
2.1	Platforms	6
2.2	Download	6
2.3	Installation	6
2.4	ECCodes support	7
2.4.1	Patches for the ECCodes library	7
2.4.2	ECCodes environment variables	9
2.4.3	ECCodes binaries	9
2.5	Version string	10
2.6	AIX support	10
2.7	Darwin/macOS support	11
3	DOCUMENTATION.....	12
4	SOFTWARE COMPONENTS	13
4.1	ropp_io	13
5	CONCLUSIONS	14

1 INTRODUCTION

This report summarises observations and suggestions based on the testing of the ROPP v10-beta software package of the ROM SAF. It thus contains feedback to the ROM SAF development team on the usability and documentation of the software.

Our approach to testing the ROPP software – which we note already is a very mature software package – was to download the ROPP v10 pre-release from the ROM SAF website, and then to unpack and compile the software in EUMETSAT’s development and operational hardware and software environments. The test cases provided as part of the ROPP software suite were then executed with the expectation that they should all pass successfully. Unfortunately, no scientific testing of newly developed functionality in ROPP v10 could be undertaken due to time constraints.

During this testing, we found that the compilation of the ROPP software required a small number of modifications to either the source code itself or to settings in the ROPP build system. These changes – which in parts may be specific to the EUMETSAT environment – are documented in Section 2. Most of them are related to the newly introduced support for ECMWF’s eccodes software library which acts as a replacement of the (by now no longer supported) bufrdc software collection previously provided by ECMWF. In this context, we also provide some recommendations for an even better documentation of some aspects of the installation process, e.g. on the use of certain environment variables. Other suggestions for the overall documentation (which are very few) are collected in Section 3.

With EUMETSAT preparing the rollout of a new release of the operational GRAS processing with updated level 1b data formats, we have also looked into the option to adapt the ROPP data conversion tools to the upcoming level 1b data format. A short overview and links to the corresponding software patches are provided in Section 4.

1.1 Scope

The report is provided to the ROM SAF project team for further deliberation.

1.2 Links into EUMETSAT’s GitLab repository

This document contains hyperlinks to commits and/or merge requests in EUMETSAT’s GitLab source code repository server, where the RO team maintains a git repository of the locally used ROPP software. EUMETSAT’s ROPP repository is regularly updated with formal releases of the ROPP software including the v10 beta release reviewed in this document, and all patches discussed in this document are available from there via the links provided.

At present, members of the ROM SAF Project Team do not have access to this repository, but this can be changed on short notice. We would prefer to provide such an access to the GitLab repository to provide the ROM SAF team with our proposed software patches. Alternatively, we can provide the relevant patches as text files (to be applied with the command line tool patch) or the modified source code files themselves.

2 DOWNLOAD, INSTALLATION AND TESTS

2.1 Platforms

We performed the beta testing of the ROPP v10 pre-release on the following 64-bit platforms:

- Linux: Ubuntu 16.04.06 LTS, with GNU gcc and gfortran v7.3.0 compilers;
- AIX: AIX v7.1, with IBM xlc and xlf v16.1.0 compilers;
- macOS 10.14.6 (Mojave), with GNU gcc and gfortran v9.3.0 compilers.

On all platforms, the software libraries required for the ROPP compilation were not taken from the ROM SAF website, but already existing installations were used instead. The reason for this is that the software development environment at EUMETSAT is used for multiple software applications and teams, and the installation of dedicated versions of software libraries for individual tools is not practical. With the exception of the problems we encountered regarding the newly introduced `eccodes` library, however, no issues arose from this setting.

Note that we did not test a build ROPP against the MetOffice's BUFR library as the latter is not available within EUMETSAT. We also tried building ROPP against the `eccodes` library only, i.e. without building against the by-now outdated `bufcdc` software. The reason for the latter is that we had difficulties porting `bufcdc` to some modern 64-bit environments and isn't actively supported anymore.

Our testing concentrated on compiling the ROPP software in these environments and the subsequent testing of the various tools offered by ROPP through the tests delivered with the ROPP software. In this process, we experienced some difficulties in both steps, but could fix all of them by either setting environment variables or by patching the build system and/or software of ROPP.

After the modifications described in the following sections, all ROPP tests passed successfully.

2.2 Download

We experienced no issues in downloading the ROPP software. The description of the downloadable items on the ROM SAF website was sufficiently clear to understand what to download, and where further information can be found.

2.3 Installation

The installation could be performed without problems for as long as the (by now no longer supported) `bufcdc` BUFR library previously available from ECMWF was used. When instead building against ECMWF's `eccodes` library, we experienced the following problems:

- The only explanation which environment variables must be set for a successful installation are in the Release Notes; they should be more easily available, e.g. in the README files of each sub-package.
- The file `roppbufrcodes.nl` is installed into the directory defined by one of the environment variables `BUFR_TABLES` (for the ECMWF BUFR library) or `BUFR_LIBRARY` (for the MetOffice BUFR library). These environment variables are also

used by the respective library to point data files and bufr tables relevant for these libraries. The namelist file, however, is not relevant for these libraries; it is only used by a small number of ROPP tools. Therefore, they shouldn't be installed into the directory trees of these libraries. A much better option would be to put them into a path like `<prefix>/share/ropp` which would also be more in line with the usual directory layout for automake/autoconf projects as well as Unix software in general.

- If `eccodes` is used as replacement for other BUFR libraries (in particular `bufrdc`), the file `roppbufrcodes.nl` is not installed automatically, even if `BUFR_TABLES` is set.
- The same place (`<prefix>/share/ropp`) could also be used to install the EGM96 coefficient files rather than having the user to have to sort those files him- or herself. We did not immediately find any reference to these files, actually, in any README.
- It is nowhere mentioned that `ECCODES_BUFR_SET_TO_MISSING_IF_OUT_OF_RANGE` (an environment variable) must be set (all tests and setup shell scripts use it) - see the detailed discussion below.
- The Release Notes contain incorrect (in our view) advice of hacking the ECCodes source code – see the discussion in the next section as well.

2.4 ECCodes support

When running the test cases provided with the ROPP v10-beta software, most tests of tools linked against `eccodes` did not pass initially.

2.4.1 Patches for the ECCodes library

When using `eccodes`, however, we observed the following errors observed in the log of `ropp_io` test run *if the `eccodes` library is compiled as downloaded from ECMWF*:

```
=====
1. eum2bufr_eccodes: test eum2bufr_eccodes by comparing against reference file
=====
```

```
Testing BUFR encode/decode [ecCodes library] ...
```

```
$NCDUMP = /opt/mettools/bin/ncdump
```

```
1. Encoding ROPP netCDF -> BUFR
```

```
=====
```

```
ECCODES ERROR : encode_double_value: softwareIdentification. Value (1.7e+38) out of range
(minAllowed=0, maxAllowed=16383). Setting it to missing value
```

```
ECCODES ERROR : encode_double_value: percentConfidence. Value (1.7e+38) out of range (minAllowed=0,
maxAllowed=127). Setting it to missing value
```

```
ECCODES ERROR : encode_double_value: DistanceFromEarthCentreInDirectionOf0DegreesLongitude. Value
(1.7e+38) out of range (minAllowed=-1.07374e+07, maxAllowed=1.07374e+07). Setting it to missing value
```

```
ECCODES ERROR : encode_double_value: DistanceFromEarthCentreInDirection90DegreesEast. Value
(1.7e+38) out of range (minAllowed=-1.07374e+07, maxAllowed=1.07374e+07). Setting it to missing value
```

ECCODES ERROR : encode_double_value: DistanceFromEarthCentreInDirectionOfNorthPole. Value (1.7e+38) out of range (minAllowed=-1.07374e+07, maxAllowed=1.07374e+07). Setting it to missing value

ECCODES ERROR : encode_double_value: absolutePlatformVelocityFirstComponent. Value (1.7e+38) out of range (minAllowed=-10737.4, maxAllowed=10737.4). Setting it to missing value

ECCODES ERROR : encode_double_value: absolutePlatformVelocitySecondComponent. Value (1.7e+38) out of range (minAllowed=-10737.4, maxAllowed=10737.4). Setting it to missing value

ECCODES ERROR : encode_double_value: absolutePlatformVelocityThirdComponent. Value (1.7e+38) out of range (minAllowed=-10737.4, maxAllowed=10737.4). Setting it to missing value

ECCODES ERROR : encode_double_value: DistanceFromEarthCentreInDirectionOf0DegreesLongitude. Value (1.7e+38) out of range (minAllowed=-1.07374e+08, maxAllowed=1.07374e+08). Setting it to missing value

ECCODES ERROR : encode_double_value: DistanceFromEarthCentreInDirection90DegreesEast. Value (1.7e+38) out of range (minAllowed=-1.07374e+08, maxAllowed=1.07374e+08). Setting it to missing value

ECCODES ERROR : encode_double_value: DistanceFromEarthCentreInDirectionOfNorthPole. Value (1.7e+38) out of range (minAllowed=-1.07374e+08, maxAllowed=1.07374e+08). Setting it to missing value

...

(several hundred lines of errors like above)

...

Ia. Converting EUM netCDF4 file -> BUFR file

../tools/eum2bufr_eccodes ../data/eum_test.n4 -o eum_test_1.bfr

ECCODES ERROR : encode_double_value: timeIncrement. Value (1.7e+38) out of range (minAllowed=-4.096, maxAllowed=258.047). Setting it to missing value

ECCODES ERROR : encode_double_value: bendingAngle. Value (1.7e+38) out of range (minAllowed=-0.001, maxAllowed=0.00948575). Setting it to missing value

ECCODES ERROR : encode_double_value: firstOrderStatistics. Value (1.7e+38) out of range (minAllowed=0, maxAllowed=63). Setting it to missing value

ECCODES ERROR : encode_double_value: bendingAngle. Value (1.7e+38) out of range (minAllowed=-0.001, maxAllowed=0.00948575). Setting it to missing value

ECCODES ERROR : encode_double_value: firstOrderStatistics. Value (1.7e+38) out of range (minAllowed=0, maxAllowed=63). Setting it to missing value

...

(several hundred lines of similar errors)

The above problem is related to the advice given on page 16 of the ROPP Release Notes, where it is recommended to patch the `eccodes` source code.

We believe the formulation of how to patch the eccodes library is misleading (and even outright dangerous for users). The reason is that the formulation proposes to change a call of the logging system from using an error level (which also causes the program to abort) to a debug level (which will continue silently unless debug messages are requested). At least in v2.16 of the library, the conversion of missing values is handled based on the setting of an environment variable named `ECCODES_BUFR_SET_TO_MISSING_IF_OUT_OF_RANGE` (see below). In the branch of the code which performs the conversion correctly, only an `fprintf()` statement issues a warning; the call to the logging system of the library only occurs in the branch which

should result in an error message. Editing that latter call (as described in the Release Notes) would thus disable the correct handling of values outside of validity bounds without giving the user any notice that this has happened. The correct modification would be to replace the `fprintf()` statement with a call to the logging system using the debug level. Rather than verbally describing the required change, ROPP should also come with a proper patch to avoid ambiguities in the written description.

In addition, most users will not go through the Release Note document in its entirety; we therefore suggest to mention the need for a patch in README files.

2.4.2 ECCodes environment variables

As mentioned above, the ECCodes library has an environment variable to control this behaviour (`ECCODES_BUFR_SET_TO_MISSING_IF_OUT_OF_RANGE`). If this variable is set to 1, only warnings are produced. In the test scripts as well as a shell script setting up environment variables for the compilation, this variable is indeed set. However, the user documentation does not mention this; the information should be added.

Using a plain `fprintf()` statement rather than a call to logging facility of the library is, in our view, a bug in the upstream `eccodes` library. Have you discussed the problem with user support at ECMWF? We also believe that controlling the behaviour through an environment variable is a bad design choice because different applications linked to the `eccodes` library might have different needs; requiring to set environment variables depending on which application is run isn't a workable option for ordinary users. Instead, each application should be able to set the handling of missing values (and the consistency with `bufrdc` in general) through the API. We haven't found an API function in the `eccodes` library offering such a function; we thus suggest a feature request to ECMWF to add such an API to `eccodes`.

There is another way to not require the correct setting of the environment variable. This works by setting the required environment variable within the Fortran program before the relevant `eccodes` structures are initialised. We have implemented this approach in our local version of ROPP 10 successfully; patches are available in [#8adb4a4a](#).

2.4.3 ECCodes binaries

When the binaries built with the `eccodes` library are installed, they are installed with a postfix to their name. We believe this is an excellent approach if the older versions (e.g. those build against `bufrdc`) are also present.

If binaries are only built using the `eccodes` library (given that `bufrdc` hasn't been maintained since a long time), the standard executables (e.g., `eum2bufr`) do not even exist. One way around for a user is to manually create symbolic links to the existing binaries after installation; this could be automated at installation time (see [#72b03748](#) for an actual implementation of this feature). An alternative option could be to install a user-selectable version under the default names. In case such behaviour is not implemented, the READMEs and other documentation

should explain the chosen approach and make users aware that they need to manually add symbolic links or rename some binaries when building ROPP against `ectools`.

2.5 Version string

The version of the ROPP libraries and tools is available through the `ropp_xx_version()` functions, where a version string is hardcoded (usually in `common/ropp_xx_version.f90`). In at least one place (`ropp_io/ropp/bgrasc2ropp.f90`, line 452) this version string is parsed to extract a numerical representation.

This parsing relies on the version number starting at the second position in the string and being followed by a blank (' ') character. Note that this excludes versions with patch levels or other modifications (e.g., v10.0.1, v11.0-beta), and thus violates the now common approach to semantic versioning. The potential problem is that for users modifying the version string (e.g. in order to indicate that they have rolled out an adapted version of the software) some programs will crash without giving a meaningful error message; this is actually how we became aware of the problem.

We found no documentation that the version string must follow the above convention, and prohibits use of the common semantic versioning rules. In our view, the source code should contain a comment close to the definition of the version string explaining this convention. A better solution would be to have the parsing of the version string allowing a wider range of version formats; note that for semantic versioning, appropriate regular expressions are known (see <https://semver.org/>).

Note: We fixed this in our ROPP v10 version in [#7f6433a4](#) by modifying our version string; however, better parsing would be preferred.

We further wondered why the version information provided in `configure.ac` (in the `AC_INIT()` call) isn't used to update the version strings during configuration; after all, this is why `automake` requires a version string in this call. Using this feature of the build system would simplify version updates considerably.

2.6 AIX support

Several changes (mostly to the `automake` build system) have been introduced for making it possible to compile and run ROPP v10 on AIX; most of them are related to the native AIX linker not being as clever as linkers on modern Linux systems. They therefore require correctly ordered lists of all libraries to be linked. Some modifications to a small number of Fortran files were also required to make them conforming with the Fortran 9x standard; GNU compilers tend to be more relaxed about them.

The patches we required to build ROPP on AIX using the IBM's native compiler suite are available in the following merge request: https://gitlab.eumetsat.int/ro/ropp/merge_requests/6.

2.7 Darwin/macOS support

Thanks to ROPP v10 now supporting `eccodes`, the ROPP software can finally be built and successfully tested on macOS. This is indeed very nice as macOS appears to develop an increasing number of users in the scientific community.

3 DOCUMENTATION

As in the past, we found the documentation complete and well written. We'd like to make the following comments, though:

- During the IROWG 2019 meeting of the NWP group, one commercial provider asked for the BUFR format description to explicitly state that heights shall be given with respect to the WGS86 ellipsoid.
- Also see the discussion in the previous section which requests several updates in the documentation.

4 SOFTWARE COMPONENTS

4.1 ropp_io

EUMETSAT is currently working on rolling out a new version of the operational GRAS processing. Part of the development is an evolution of the existing level 1b data format, making it compatible with radio occultation data formats for future missions such as Sentinel-6/Jason-CS and EPS-SG. In particular, the updated data format supports data from multiple GNSS systems and will be used throughout EUMETSAT's radio occultation data portfolio. We therefore explored how difficult the adaptation of the EUMETSAT readers in ROPP to the upcoming data format changes will be.

We found that the required changes – at least for the converters of EUMETSAT level 1b bending angle to ROPP bending angle data formats – are relatively straightforward. To start with, the support for the new data format version in `ropp_io` requires support for variable length strings in `netCDF4`. A corresponding patch has been available to the ROM SAF for some time ([#9e94f972](#)).

Once the above is in place, the actual support for the upcoming EUMETSAT EPS-SG (v13) and generic RO (v14) level 1b data formats is straightforward, at least for the bending angle part: we are currently testing [#04c00b79](#) and [#e6933b77](#) which cover the `eum2ropp` tool. As far as we understand, additional modifications are, however, still required to also implement support for the conversion of carrier phase and amplitude data which are used by `ropp_pp`.

Given that the modifications for supporting the upcoming data format changes are minor, we recommend that they should be included in the final ROPP v10 release.

5 CONCLUSIONS

Since many years, ROPP provides a stable, well tested and well documented software for processing radio occultation data. EUMETSAT uses several ROPP tools in both its monitoring and offline processing systems as well as in many scientific analyses and internal studies.

The upcoming v10 release of the ROPP software continues in this tradition. Among the new features introduced in this release we in particular welcome the support of ECMWF's new standard library for the support of GRIB and BUFR data, `eccodes`, which allows the installation and use of ROPP on an even larger range of Unix-like platforms.

Not surprisingly, the support of an entirely new library in a mature software package such as ROPP may be accompanied with minor teething problems and inaccuracies in the corresponding documentation. We found several of those and provided both fixes and suggestions for improved documentation to support a smoother experience for the users of the final release. None of these issues is major, and we believe they all can be included in the final v10 release.

Concerning the upcoming data format changes in EUMETSAT's radio occultation products, we analysed the required code changes and found that the required modifications to the existing readers will be minor. We provide the required patches supporting the conversion of EUMETSAT bending angle data into the ROPP data format, and would like to offer support in implementing similar changes for carrier phases and Signal-to-Noise ratios, i.e. level 1a data. Given that the changes required to support the new data formats are not large, we suggest that the final release of ROPP v10 should include them, even if this should cause a (small, but still acceptable) delay in the release date.