NWP SAF

AAPP-SG v1 Test Plan

Version 1.0

May 2026

AAPP-SG v1 Test Plan

This documentation was developed within the context of the EUMETSAT Satellite Application Facility on Numerical Weather Prediction (NWP SAF), under the Cooperation Agreement dated 7 September 2021, between EUMETSAT and the Met Office, UK, by one or more partners within the NWP SAF. The partners in the NWP SAF are the Met Office, ECMWF, DWD and Météo France.

Version	Date	Author	Remarks
0.1	11/08/25	N C Atkinson	Initial
0.2	27/01/26	N C Atkinson	With test results
0.3	10/03/26	N C Atkinson and E C Nolland	Feedback from beta testing and resulting changes; Release Candidate 5 test results
0.4	13/04/26	N C Atkinson	For external review
1.0	07/05/26	E C Nolland	From external review

Table of Contents

1. INTRODUCTION

1.1 Applicable documents

[AD-1] NWP SAF AAPP-SG v1 Product Specification: NWPSAF-MO-DS-057
[AD-2] NWP SAF Development Procedures for Software Deliverables: NWPSAF-MO-SW-002
[AD-3] NWP SAF AAPP-SG v1 Top Level Design: NWPSAF-MO-DS-056

1.2 Purpose

This document defines the test plan for Version 1 of the ATOVS and AVHRR Processing Package Second Generation (AAPP-SG)

AAPP-SG is a deliverable of the NWP SAF project. It is a software package used for the pre-processing of data from the EPS-SG series of satellites (and in the future for other new-generation polar-orbiting satellites), primarily for the purpose of preparing data for input to NWP assimilation systems.

NWP SAF software products are developed according to the guidelines of [AD-2].

The purpose of the testing described in this document is to ensure that the requirements of the Product Specification [AD-1] are met. The document will be upgraded to v1.0 when approved by Delivery Readiness Review (DRR).

2. TESTED ITEMS

Software testing can be generally divided into the following items:

- Coding and compilation testing
- Module testing
- Integration testing
- Validation testing
- Portability testing
- Timing testing
- Documentation testing

The testing procedures with respect to the AAPP-SG software are outlined below.

2.1 Coding and compilation testing

Coding guidelines are outlined in [AD-2]. The code will be inspected by someone other than the author, who is familiar with AAPP-SG principles. The inspection will be to sign off that the code

- Is written to the guidelines of AD-2, where applicable
- Correctly implements the design of AD-3

The second test is that the code compiles without error and that there are no unexpected warnings. The tester is to follow the instructions in the User Guide, to run the “configure” and “make” scripts in the AAPP-SG top directory. This will be performed on a number of different linux platforms (e.g. those hosted by Met Office and ECMWF) and compilers (GNU and Intel). External libraries (e.g. netCDF, HDF5, ecCodes) should be linked in.

2.2 Module testing

In AAPP-SG, all modules (subroutines) are linked to a top-level program, and the user would not normally be expected to link AAPP-SG libraries directly with their source code. Strict module testing is not required to be performed, and all modules will be tested in an integrated environment, as described in the following section.

2.3 Integration and verification testing

This section covers both integration testing by the development team and subsequent verification testing by beta testers. Beta testers will be expected to review the documentation (see Section 2.4), run test cases and report on their experiences to the development team.

The different tests are summarised in Table 1 and discussed in more detail in the following subsections. A sub-set of the test cases is to be made available on the NWPSAF web site, and instructions are to be provided to allow users to run them.

All tests should be run with the locally compiled code (in the order shown below). In addition, a subset of the tests should be run with the executables that are to be provided for the users (requirement 7.5 of AD-1).

Table 1: AAPP-SG test cases (table taken from the AAPP-SG User Manual)

No.	Test Case	Top-level namelist (where applicable)	Comment
1	MWS_netcdf_1b_to_bufr_1b	MWS_bufr_1b.nl	MWS conversion from level 1b netcdf to a standard level 1b BUFR at full spatial resolution.
2	MWS_netcdf_1b_to_bufr_1d	MWS_main.nl	MWS conversion from level 1b netcdf to a level 1d BUFR file. The MWS BUFR output is spatially thinned and channel BTs have been spatially averaged.
3	MWS_bufr_1b_to_bufr_1d	MWS_main.nl	As above, but for 1b BUFR input.
4	IASI_NG_netcdf_1c_to_bufr_1d_IASI_SRF	NG_main_1c_to_1d.nl	IASI-NG conversion from level 1c netCDF to level 1d BUFR at IASI first generation spectral resolution. Outputs come from the raw radiances.
5	IASI_NG_netcdf_1c_to_bufr_1c	NG_bufr_1c.nl	IASI-NG conversion from level 1c netCDF to a standard level 1c BUFR at full spatial resolution.
6	IASI_NG_netcdf_1d_to_bufr_1d	NG_main_1d.nl	IASI-NG conversion from level 1d netCDF (PC scores) to a standard level 1d BUFR at full spatial resolution.
7	IASI_NG_netcdf_1d_to_bufr_1d_IASI_SRF	NG_main_iasi314.nl	IASI-NG processing from level 1d netCDF (PC scores) to “enhanced” level 1d BUFR, including spatial thinning and conversion to IASI first generation spectral resolution. Outputs are also fed back to the input netCDF file.
8	IASI_NG_netcdf_1d_to_bufr_1d_with_MWS	NG_main_with_mws.nl	IASI-NG processing from level 1d netCDF (PC scores) to “enhanced” level 1d BUFR, including mapped MWS.
9	IASI_NG_1c_to_1d_netcdf	-	IASI-NG conversion from level 1c netCDF (full spectrum) to level 1d netCDF (PC scores). This can be used in as a follow-on step to the Direct Broadcast package which

			generates level 1c as its output.
10	IASI_NG_bufr_1c_to_bufr_1d	NG_main_1c_to_1d.nl	IASI-NG conversion from level 1c BUFR to level 1d BUFR at IASI first generation spectral resolution. Outputs come from the raw radiances.
11	IASI_NG_bufr_1d_to_bufr_1d	NG_main_1d_to_1d.nl	As for test 10, but for level 1d BUFR input. Outputs are reconstructed radiances.
12	IASI_NG_bufr_1d_to_bufr_1d_with_MWS	NG_main_with_mws.nl	As for test 11, but the inputs are BUFR.

2.3.1 Test Case 1. MWS netCDF to BUFR level 1b

Input: MWS netCDF level 1b
Output: MWS BUFR level 1b
Main usage: Preparation of level 1b data for international dissemination e.g. within the WMO DBNet programme.

Test requirement:

- output BUFR is consistent with the equivalent EUMETSAT BUFR product, i.e. same BUFR sequence and consistent values. Use the ecCodes tool *bufr_compare*.

2.3.2 Test Case 2. MWS netCDF to BUFR level 1d

Input: MWS netCDF level 1b
Output: MWS BUFR level 1d
Main usage: Usage of MWS in NWP. Provides spatial thinning and spatial averaging capability, also scattering/surface tests.

Test requirement:

- exercise the thinning and averaging options
- demonstrate the surface and scattering tests (qualitative – e.g. using plots)

2.3.3 Test Case 3. MWS BUFR 1b to BUFR 1d

As for Test Case 2 (Section 2.3.2) but for BUFR input. Verify that the products are compatible.

2.3.4 Test Case 4. IASI-NG netCDF 1c (full spectra) to BUFR 1d (at IASI first generation resolution)

Input: IASI-NG netCDF level 1c
Output: IASI-NG level 1d BUFR at IASI first generation spectral resolution
Main usage: NWP centres

Test requirement:
Compare the product with that from Test Case 11 (Section 2.3.11). The radiance values should be similar (to within instrument noise).

2.3.5 Test Case 5. IASI-NG netCDF 1c (full spectra) to BUFR 1c

Input: IASI-NG netCDF level 1c
Output: IASI-NG BUFR level 1c

Main usage: Preparation of level 1c data for international dissemination e.g. within the WMO DBNet programme.

Test requirement:
output BUFR is consistent with the equivalent EUMETSAT BUFR product, i.e. same BUFR sequence and consistent values. Use *bufr_compare*.

2.3.6 Test Case 6. IASI-NG netCDF 1d (PC scores) to BUFR 1d (PC scores)

Input: IASI-NG netCDF level 1d
Output: IASI-NG BUFR level 1d (standard format, full spatial resolution)
Main usage: Preparation of level 1d data for international dissemination e.g. within the WMO DBNet programme.

Test requirement:
output BUFR is consistent with the equivalent EUMETSAT BUFR product, i.e. same BUFR sequence and consistent values. Use *bufr_compare*.

2.3.7 Test Case 7. IASI-NG netCDF 1d (PC scores) to BUFR 1d (enhanced)

Input: IASI-NG netCDF level 1d
Output: IASI-NG BUFR level 1d (enhanced)
Ancillary files: IASI-NG eigenvectors
Main usage: Preparation of IASI-NG data for use in NWP.

This test exercises the option to convert IASI-NG to the spectral resolution of IASI first generation, then computes the radiance for selected channels and writes the result (i) to a BUFR file, and (ii) back to the input netCDF file.

Test requirement:
Check that the output BUFR contains reconstructed radiances, and also that the “derived” group is written to the input netCDF dataset. Plot an output spectrum and verify that the turning on of the IASI first generation conversion ("convert_NG_to_IASI" set to true in NG_ancil.nl) changes the spectral response.

2.3.8 Test Case 8. IASI-NG netCDF 1d (PC scores) to BUFR 1d (enhanced with MWS)

Input: (i) IASI-NG netCDF level 1d, (ii) MWS netCDF level 1b
Output: IASI-NG BUFR level 1d (enhanced with MWS)
Ancillary files: IASI-NG eigenvectors
Main usage: Preparation of IASI-NG data for use in NWP.

Test requirement:
Check that the output BUFR contains reconstructed radiances and mapped MWS. Modify the script to use the “-io” option instead of “-i” and verify that the mapped MWS data are written back to the input file in the “derived” group.

2.3.9 Test Case 9. IASI-NG netCDF 1c (full spectrum) to netCDF 1d (PC scores)

Input: IASI-NG netCDF level 1c
Output: IASI-NG netCDF level 1d
Ancillary files: IASI-NG eigenvectors
Main usage: Direct broadcast users who are running the IASI-NG level 1 processor and require a level 1d output compatible with EUMETSAT's product.

Test requirement:
Use a tool such as ncdiff to verify that the generated level 1d product is compatible with the equivalent EUMETSAT level 1d product.

2.3.10 Test Case 10. IASI-NG BUFR 1c to BUFR 1d (at IASI first generation resolution)

Input: IASI-NG level 1c BUFR (output of Test Case 5)
Output: IASI-NG level 1d BUFR at IASI first generation spectral resolution
Main usage: This is expected to be used by some NWP centres during instrument commissioning, when the PC product is not yet available

Test requirement:
Compare the product with that from Test Case 11 (Section 2.3.11). The radiance values should be similar (to within instrument noise).

2.3.11 Test Case 11. IASI-NG BUFR 1d (standard) to BUFR 1d (IASI first generation resolution)

Input: IASI-NG level 1d BUFR (output of Test Case 6)
Output: IASI-NG level 1d BUFR at IASI first generation spectral resolution
Main usage: NWP centres

Test requirement:
Compare the product with that from Test Case 10 (Section 2.3.10). The radiance values should be similar (to within instrument noise).

2.3.12 Test Case 12. IASI-NG BUFR 1d (PC scores) to BUFR 1d (enhanced with MWS)

Input: IASI-NG level 1d BUFR (output of Test Case 6)
Output: IASI-NG level 1d BUFR with mapped MWS
Main usage: NWP centres

Test requirement:
The IASI-NG radiances should be as in Test Case 8 (Section 2.3.8). In addition, the output should contain mapped MWS.

2.4 Validation testing

The purpose of validation is to provide a high degree of assurance that a product, service, or system accomplishes its intended requirements. For AAPP-SG v1 this will be achieved through the activities of beta testers. Beta testers will be asked not only to verify the technical aspects of AAPP installation and running, but also to review the scientific documentation and the Release Note, to assess to what extent the package meets the needs of users.

Note that all the test cases of AAPP-SG v1 use *simulated* Metop-SG-A1 data. Post-launch validation using in-orbit data will not be possible until the instruments have been commissioned and level 1 datasets approved.

2.5 Portability testing

Portability testing will be performed in-house (Met Office), using a range of operating systems and Fortran compilers. Actual compilers will be listed in the Test Report.

Portability testing will include:

- Building AAPP on the platform
- Running a selection of the integration tests

It may not be possible to run all the integration tests on all platforms, but as many as possible should be attempted, within the constraints of available time, external libraries, etc.

2.6 Timing testing

Examples of run times will be given in the test results (section 4 onwards of this document). Timing testing serves as a guideline to other users of AAPP-SG.

2.7 Regression testing

If any changes are made to the software after the start of formal testing, it may be necessary to re-run some or all of the previous tests. The impact of any changes between release versions shall be carefully monitored and reported in the test results.

2.8 Documentation testing

AAPP-SG will include a User Manual. This will be reviewed internally and by the beta testers and reviewers, who will provide feedback about their experiences. The developers will address any recommendations for improvement before release of the package to general users.

3. REQUIREMENTS TRACEABILITY TABLES

This section demonstrates how the requirements listed in AD-1 relate to the test plan.

Table 2: Relationship between Test Plan and General Requirements in AD-1 Section 7

Requirement	Testing method	Test plan reference	Comment
7.1 Release note completeness	Inspection	2.4	
7.2 User Manual	Inspection	2.8	
7.3 Building and linking	Test	2.1	Includes coding and compilation testing
7.4 Compile and run on Linux PC or server	Test	2.5	
7.5 Provide AAPP-SG as executables	Test	2.3	Verify that the executables run
7.6 Test Cases for MWS and IASI-NG	Test	2.3	See Table 1 and subsequent subsections

7.7 Test Cases available on NWPSAF web site	Inspection	-	Assessed during release approval

Please see Appendix X for a more detailed software requirements traceability matrix mapping the software processing requirements to the relevant test cases. The matrix covers the five supported data processing paths, indicating the processing options available for each path.

4. TEST RESULTS

This section describes testing carried out by the PT at the Met Office.

4.1 Building the package from source

- Date: 22/1/2026
- Version: v1.0.0.rc3
- Operating system: RHEL9.7
- Dependency libraries (as specified in ***AAPP-SG/setup/requirements.txt***):
  - eccodes==2.44.0 (noting that this is the minimum version that supports IASI-NG BUFR encoding)
  - hdf5==1.14.6
  - netCDF-C==4.9.3
  - netCDF-Fortran==4.6.2

  and central utilities/libraries
  - zlib==1.2.11
  - cmake==3.26.5
- The build was executed using the Test Harness (available in the NWPSAF-AAPP-SG GitHub repository), running on the Met Office Scientific Desktop with GCC11.5.0 (gfortran, gcc, g++). Result: pass.
- The build was also executed in a Conda environment using GCC15.2.0 and the additional packages:
  - cmake==4.1.2
  - zlib==1.3.1

  Result: pass.

4.2 Running test cases using build from source

- Date: 22/1/2026
- Version: v1.0.0.rc3
- Operating system: RHEL9.7

The results of the standard tests described in Section 2.3 are given below. Note that the test names have been modified slightly compared with Table 1.

The automated tests do not take account of comparisons with EUMETSAT BUFR products: this was done manually afterwards (see below).

Table 3: Testing summary for standard AAPP-SG test cases

No.	Test name	Runs without error?	Comment
1	MWS_netcdf_to_bufr_1b	Pass	MWS conversion to a standard level 1b BUFR at full spatial resolution. See 4.2.1 below for analysis of BUFR product. See

			Figure 1
2	MWS_netcdf_to_bufr_1d	Pass	The MWS BUFR output is spatially thinned, and channel BTs have been spatially averaged. See Figure 2 and Figure 3.
3	MWS_bufr_1b_to_bufr_1d	Pass	As above, but for 1b BUFR input
4	IASI_NG_netcdf_to_bufr_1c	Pass	IASI-NG conversion from level 1c netCDF to a standard level 1c BUFR at full spatial resolution. See 4.2.2 for discussion of BUFR products. See Figure 4
5	IASI_NG_netcdf_to_bufr_1d	Pass	IASI-NG conversion from level 1d netCDF (PC scores) to a standard level 1d BUFR at full spatial resolution. See 4.2.2 for discussion of BUFR products.
6	IASI_NG_netcdf_to_bufr_1d_IASI_SRF	Pass	IASI-NG processing from level 1d netCDF (PC scores) to "enhanced" level 1d BUFR, including spatial thinning and conversion to IASI first generation spectral resolution. Outputs are also fed back to the input netCDF file. See Figure 5
7	IASI_NG_netcdf_to_bufr_1d_with_MWS	Pass	IASI-NG processing from level 1d netCDF (PC scores) to "enhanced" level 1d BUFR, including mapped MWS. See Figure 6 and Figure 7
8	IASI_NG_1c_to_1d_netcdf	Pass	IASI-NG conversion from level 1c netCDF (full spectrum) to level 1d netCDF (PC scores). This can be used in as a follow-on step to the Direct Broadcast package which generates level 1c as its output.
9	IASI_NG_bufr_1c_to_bufr_1d	Pass	IASI-NG conversion from level 1c BUFR to level 1d BUFR at IASI first generation spectral resolution. Outputs come from the raw radiances.
10	IASI_NG_bufr_1d_to_bufr_1d	Pass	As for test 9, but for level 1d BUFR input. Outputs are reconstructed radiances.
11	IASI_NG_bufr_1d_to_bufr_1d_with_MWS	Pass	As for test 7, but the inputs are BUFR.

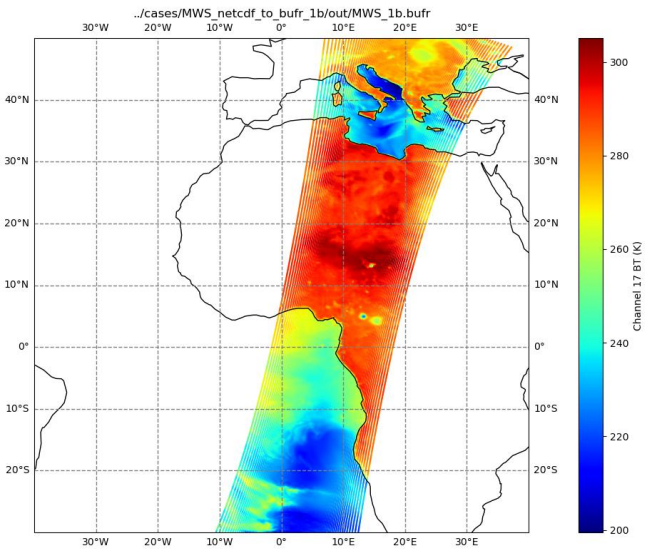

Figure 1: MWS brightness temperature, level 1b full resolution (zoomed in). Test 1.

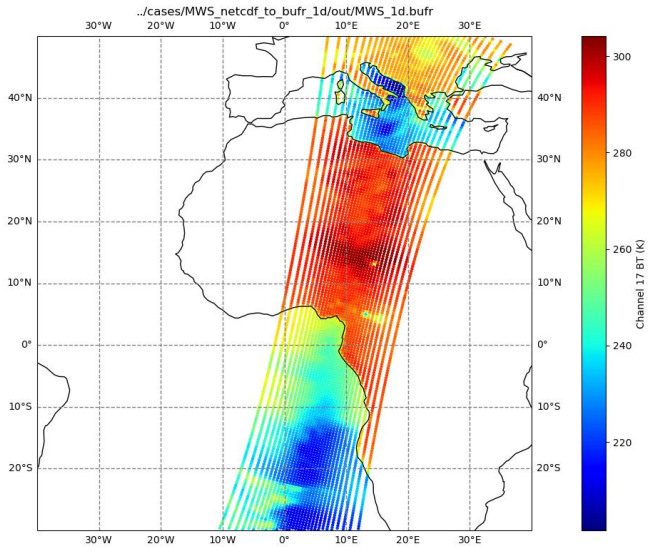

Figure 2: MWS channel 17 brightness temperature, level 1d (thinned and zoomed in). Test 2.

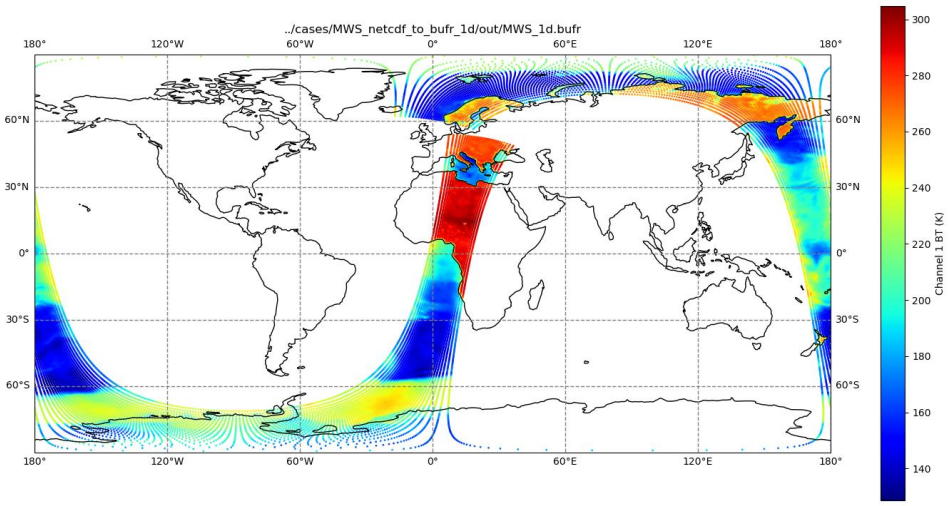

Figure 3: MWS channel 1 brightness temperature, level 1d (thinned). Test 2.

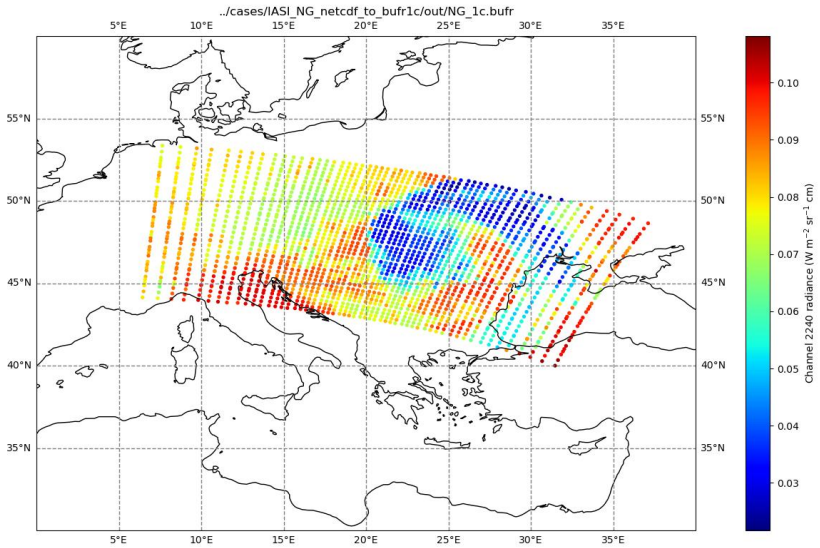

Figure 4: IASI-NG radiance from level 1c. Test 4.

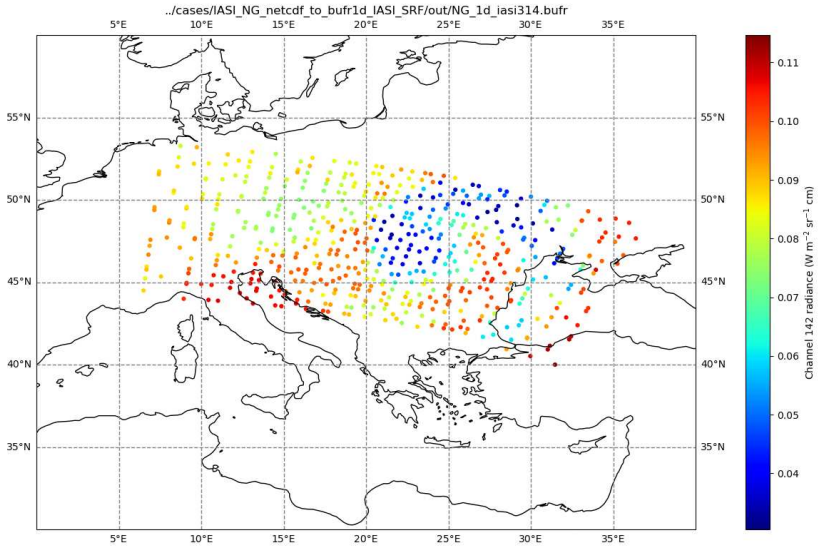

Figure 5: IASI-NG radiance level 1d, thinned with IASI first generation spectral response. Test 6.

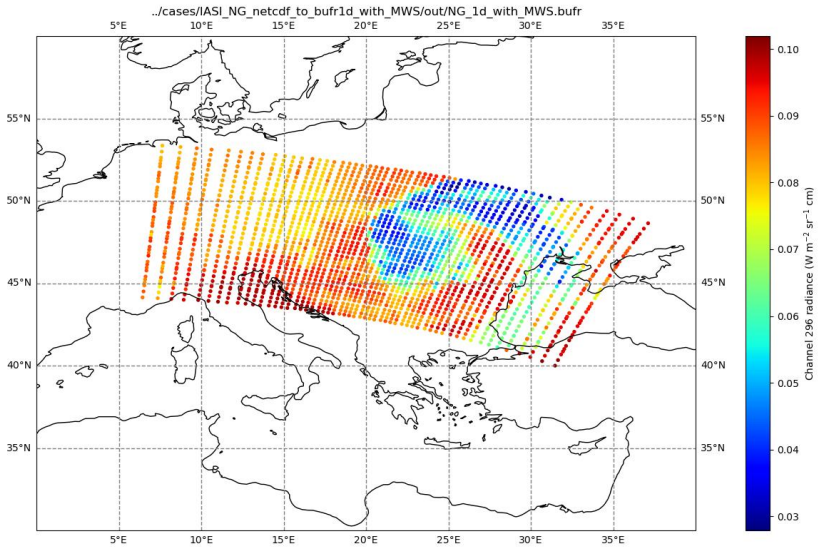

Figure 6: IASI-NG level 1d radiance. Test 7.

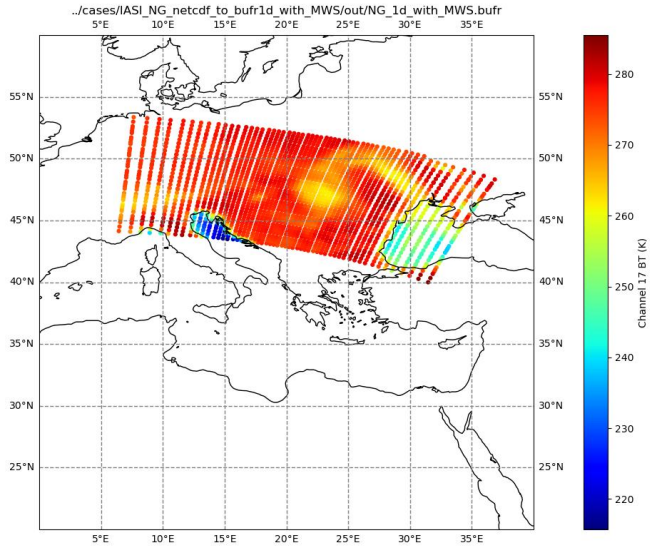

Figure 7: MWS channel 17 mapped to IASI-NG. Test 7.

4.2.1 MWS 1b comparison with EUMETSAT product

EUMETSAT reference file: W_XX-EUMETSAT-Darmstadt,SAT,SGA1-MWS-1B-RAD_C_EUMT_20241111000000_G_D_20240224182105_20070912084621_T_N____.bin

In order to obtain a meaningful comparison, the number of scans per message was temporarily set to 1 in AAPP-SG namelist MWS_bufr_1b.nl, in order to match EUMETSAT product.

Differences, using *bufr_compare*, are as follows:

Variable	EUMETSAT	AAPP-SG	Comment
bufrHeaderCentre	254	255	Configurable via namelist
dataSubCategory	3	0	Configurable via namelist

softwareVersionNumber	v1_Global	v1_AAPP	Difference is intended
#2#bandNumber	2	2147483647 (i.e. missing)	See note
#2#satelliteChannelBandWidth	0	MISSING	See note
#3#bandNumber , etc.			

Note: There is a minor difference of approach in defining unused sub-bands: EUMETSAT numbers the non-existent sub-bands and sets the bandwidths to zero; AAPP-SG sets the values to missing (since the sub-bands do not physically exist in the instrument). This is of no consequence for the user.

Result: PASS. AAPP-SG MWS 1b BUFR is consistent with EUMETSAT product.

4.2.2 IASI-NG 1c and 1d BUFR products

Unfortunately, the IASI-NG 1d and 1d BUFR sequences, defined by EUMETSAT, are not yet stable. We have used the latest available sequences (email correspondence with A Merle, EUMETSAT, Nov 2025), but to date no reference products with these sequences have been published.

Therefore, validation of tests 4 and 5 (consistency of IASI-NG 1c and 1d with EUMETSAT products) cannot yet be performed. Sections 6.1 and 6.2 of the User Manual make clear that the IASI-NG BUFR sequences are provisional.

4.2.3 Testing on RHEL7 with Apptainer

Date: 27/01/2026
Apptainer environment: Rocky Linux 9.3 (see file /etc/os-release)

The executables file AAPP-SG_v1.0.0.rc3-exec.tar.gz was installed on RHEL7 machine *expradsatdev01*, together with test cases AAPP-SG-test_cases-v1.tar.gz.

In order for the test cases to run to completion in the Apptainer environment, the following changes were needed:

1. Addition of *libsz.so.2*
2. Addition of ecCodes tool *bufr_compare* (required for the test cases validation)
3. Addition of netCDF tool *nccopy* (required for the conversion from IASI-NG level 1c to level 1d)

These three files were added in the script ***package_executables.sh*** ready for the final release.

Result: PASS. AAPP-SG can be run from executables (requirement 7.5, Section 3).

5. BETA TESTING REPORTS

NWPSAF response comments in red.

5.1 CNR, Italy (Salvatore Larosa)[1] – testing Release Candidate RC2.

Email dated 22/12/2025

[1] http://www.satclop.imaa.cnr.it/index.php?lang=en

Following up on my previous message, I have completed an initial round of tests of AAPP-SG v1.0.0.rc2.

On CentOS 7, I was unable to compile ecCodes (v2.44), possibly due to issues related to the AEC library. I am attaching the build logs with AEC both enabled and disabled.

The user's log indicated that with AEC disabled, the CentOS7 build of ecCodes failed with the following error message: *Target "cmTC_82a4e" requires the language dialect "CXX17" (with compiler extensions). But the current compiler "GNU" does not support this, or CMake does not know the flags to enable it.*

This is to be expected with the GNU 4.8.5 that was used. Although ecCodes is not part of AAPP-SG, we should mention this limitation in the User Manual. ecCodes releases 2.31.0 and higher require C++17 support (i.e. at least g++ v8).

I then tested the package using a Docker-based setup on Rocky Linux 9, where the build completed successfully and all test cases ran without errors. I also tested the -io option for mapping MWS onto IASI-NG IFOVs, which worked very well.

An archive containing all test logs is attached.

I additionally ran the MWS_images_from_bufr.sh script. All steps completed successfully except for the final command (MWS mapped to IASI-NG), which fails with the attached error message. As noted in the README, this part is currently marked as a TODO, so this behaviour is likely expected at this stage.

It failed because the BUFR file does not contain brightness temperatures. The test case has an error: in namelist NG_netcdf_to_bufr1d_with_MWS/NG_bufr_1d.nl: it should say "write_channels_or_mws = .true.". When this is corrected and the test re-run, the image is generated.

The Project Team subsequently discovered problems with some of the test cases that take BUFR input. A new RC3 was generated (see below).

The Docker setup is available in a private GitHub repository (https://github.com/SatCloP/aapp-sg-docker). If you are interested in reviewing it, please let me know your GitHub username and I will grant access. If you find it useful, this Docker-based approach could also be mentioned in the User Manual as an alternative installation method, by simply pointing users to the repository.

Access has been provided. PT to consider.

I will continue testing as time allows and will report back if needed.

Update 26/01/2026:
Yes, I noticed last Friday that a new release candidate was available and I have tested RC3 using the Docker environment. I ran the run_all.sh script and the workflow completed successfully. The plotting also works perfectly now. Everything worked as expected and I did not observe any particular anomalies during the tests.

5.2 ECMWF (Chris Burrows) – testing Release Candidate RC3.

Email dated 10/02/2026

Environment:

- Atos GPIL login node (AMD EPYC Rome, 2.5GHz) at ECMWF

Compiler:

- Ifort 2021.4.0

Dependencies:

- HDF5 1.14.6
- NetCDF 4.9.3
- ecCodes 2.44
- Dependencies installed fine using "install_libraries.sh --all".

Installation:

- Configure failed because I had environment variables called $HDF5_LIB and $ECCODES_LIB from centrally-installed versions. Unsetting these variables fixed it. Have added a warning in section 2.3.1 of the User Manual.
- During the "make", this warning appeared many times: "ifort: command line warning #10006: ignoring unknown option '-fno-range-check'". For ifort and ifx, it appears that the default is not to check range. It seems that -WB does the same thing (and is easy to add to the existing script); to be checked whether it works on all compiler versions. If not, we can probably omit the flag.
- "make install" ran without errors.

Pre-running:

- In section 3.1 of the User Manual, it would be good if the definition of $AAPP_SG_INSTALL was noted before the instruction to source the config script. Have adjusted the wording in this section.
- Also in section 3.1, the name of the script has a "_" which should be a "-". After fixing this, the three PATHs look correct. Corrected.
- It looks like the bin directory is duplicated under AAPP-SG-1.0.0.rc3-build and AAPP-SG-1.0.0.rc3-install. Is this necessary? It's a consequence of the design. Software is first built in "build" then the executables are copied to "install".

Test Cases:

- ./run_all.sh was successful. A visual inspection of the log output showed no errors or warnings.
- There seem to be two sets of input files which look identical (AAPP-SG-test_cases-v1/data/input/IASI-NG/ and AAPP-SG-test_cases-v1/data/input/IASI-NG/IASI-NG-1.0). Why are there two sets? This is a bit confusing, especially when specifying the $AAPP_SG_INSTALL. Probably a mistake.
- Some of the BUFR files seemed to require non-standard BUFR tables. Nigel explained that either the provided tables would work, or the recently-released ecCodes 2.45 (both checked and confirmed to work)

Problems:

- Recreating the correct PATHs etc in a new terminal session is not necessarily straightforward as the paths to the dependencies are set during the AAPP-installation.
- When I tried to recompile, the PATH setting in the environment script (aapp-sg_env.sh) was not present. No idea why this was, and I had to add the relevant locations of the dependency binaries manually. In RC3, the path to the dependency binaries was not included, since these

binaries are not required when running the AAPP-SG executables. However, we noted that they are required when running the test cases (specifically, *bufr_compare* and *nccopy*), so this will be added for the final release.

I created a new test for my main use case (this is attached). I was trying to convert a L1C IASI-NG netcdf file to a BUFR file containing radiances only, for all channels, having been mapped onto IASI SRFs. It didn't work, so maybe I'm doing something wrong:

- the output bufr file contains no radiance data
- There are sections for satellite instruments 231, 232 and 233, whereas I only want IASI-NG (and possibly met-image). In the new test case (see below) the 233 (MWS) is included but with "delayedDescriptorReplicationFactor" equal to zero so it takes up no space.
- Despite deleting the line in the namelist pointing to the ancillary namelist, it still seems to read it, despite the eigenvectors not being needed for this use case (unless the non-truncated eigenvector set is used to do the remapping). The software needs to know which channels go in which bands, and this information is only available from the eigenvectors file. It only takes a fraction of a second to read this file.
- Until reading the documentation, I hadn't really thought in terms of the spectral remapping being considered L1D, but it makes sense. I was expecting the package to produce an IASI bufr with the IASI radiance sequence. Not a problem, just a surprise for me. IASI L1C BUFR is very different from IASI-NG L1C BUFR, so it would not have been feasible to create an output with the old sequence.
- If a working test for this use case could be provided as part of the package, that would be super-helpful (it would be running operationally so unnecessary reading of ancillary data etc should be avoided). Created a new test case based on "IASI_NG_bufr_1c_to_bufr_1d". This does what Chris was asking for. Note, however, that the operational near-real-time dissemination for IAS L1C is expected to be BUFR, not netCDF.

6. CHANGES FOLLOWING BETA-TESTING

Table 4: Issues raised during and after beta-testing and how these were addressed

Issue raised	*Action taken*
To build ecCodes >= 2.31.0, C++17 support is required. Raised by: Salvatore Larosa	A warning has been included in the User Manual (Section 1.5); An additional check has been added into the `install_dependencies.sh` section to install ecCodes. The script will now abort with an appropriate error message unless the GNU C++ and Intel C++ compilers are at versions >=8 and >=19.1 (where C++17 is fully supported).
The test cases which output "enhanced level 1d" BUFR were using namelists with `write_channels_or_mws` set to `false`, when it should have been set to `true`. Raised by: Salvatore Larosa	This has been corrected. Additional plotting scripts have been incorporated into the test-harness to sanity-check test case outputs.
Executables of the dependencies used to build AAPP-SG (e.g. `h5dump`, `nccopy` etc) should be available on PATH after the	If the `bin` directory for an AAPP-SG dependency is not already present in PATH, it will be included in PATH by the runtime environment script.

runtime environment script `aapp-sg_env.sh` is sourced. Raised by: Chris Burrows	
When building with the ifort compiler, the warnings "`ifort: command line warning #10006: ignoring unknown option '-fno-range-check'`" were seen repeatedly. Raised by: Chris Burrows	Add "rangeflage" (empty for Intel compiler) instead of hard-coding "-fno-range-check". This eliminates the warnings.
Lack of test case demonstrating how to convert from level 1c netCDF to level 1d BUFR at IASI first generation spectral resolution. Raised by: Chris Burrows	Added new demonstration test case: `IASI_NG_netcdf_1c_to_bufr_1d_IASI_SRF`
Two sets of input data included in test cases tar file. Raised by: Chris Burrows	The extraneous set of input data has been removed.
Internal testing spotted a bug in orbit angle values in BUFR files: negative orbit angles are labelled as "missing". Raised by: Brett Candy, Met Office	In netCDF files, the orbit angle range in netCDF is still -180 to +180; in BUFR files, the orbit angle range has been altered to 0 to 360 degrees.
ecCodes BUFR encoding fails (and therefore test case scripts fail) if the input data contains any out-of-range values. Raised by: Met Office PT	Set default value in test cases so that this causes a warning, rather than an error: `ECCODES_BUFR_SET_TO_MISSING_IF_OUT_OF_RANGE=1`
The Met Office MetDB database - and possibly the databases used by other NWP centres - cannot ingest BUFR files encoded using the BUFR operator 203 in compressed data (which is used to change the offset for IASI-NG radiance). Raised by: Met Office PT	An additional variable `use_operator_203` has been defined for use by the NG_bufr namelist (see Table 8 of the User Manual). If this variable is set to false, the BUFR encoding will use an alternative 1d sequence with a manual offset change for principalComponentScore.
Make the software compatible with both simulated data and in-orbit data Raised by: Met Office PT	AAPP-SG was tested using the output of the EPS-SG Direct Broadcast packages (Feb 2026 version provided by EUMETSAT to the NWP SAF) using in-orbit data. It was identified that IASI-NG L1c files do not have some of the top-level attributes that were present in the test data. As these attributes are not used by AAPP-SG, they were removed from *NG_1c_to_1d.sh*. Also, some top-level attributes that had previously been fixed length strings became variable length, and vice versa. The code was modified to accept

	either.

7. PENULTIMATE RELEASE CANDIDATE – TEST RESULTS

This section describes the testing of release candidate 5 (RC5) carried out by the PT at the Met Office. This is the penultimate release candidate due to the need to fix the bug picked up in Section 7.2, which only affects running AAPP-SG from binaries.

7.1 Testing on RHEL8 and RHEL9

The following table lists the permutations of tests carried out using the release candidate, RC5. In each case: AAPP-SG was successfully built and installed (if using source code); all tests (listed in Table 6) ran and produced output which matches the known good output (KGO).

Table 5: Platforms and compilers tested

Date	Operating System	Built from source code or using binaries	Compiler versions	Dependencies built from source code or installed via conda	Core dependency versions	Supporting utilities or libraries versions
06/03/2026	RHEL9.7	Binaries	--	--	--	--
		Source code	GCC15.2.0 (gfortran, gcc, g++)	Installed via conda	As specified in ***setup/ requirements.txt:*** hdf5: 1.14.6; netCDF-C: 4.9.3; netCDF-Fortran: 4.6.2; eccodes: 2.44.0 (noting that this is the minimum version that supports IASI-NG BUFR encoding)	cmake: 4.1.2; libz.so: 1.3.1
			GCC11.5.0 (gfortran, gcc, g++)	Built from source code		cmake: 3.26.5; libz.so: 1.2.11
			ifort 19.0.0 and GCC11.5.0 (gcc, g++)	Built from source code		
10/03/2026	RHEL8.10	Source code	Intel 2021.4.0 (ifort, icc, icpc)	Built from source code		

The results of the standard tests described in section 2.3 are given below. Note that the test names have been modified slightly compared with Table 3 (Table 1 has been updated accordingly) and there is an additional test case, `IASI_NG_netcdf_1c_to_bufr_1d_IASI_SRF`. This was added at the request of Chris Burrows.

Table 6: Testing summary for standard AAPP-SG test cases run on release candidate 5

No.	Test Case	Runs without error?	Run time[2]	Comment
1	MWS_netcdf_1b_to_bufr_1b	Pass	14.3 s	MWS conversion from level 1b netcdf to a standard level 1b BUFR at full spatial resolution. See **Figure 8**.
2	MWS_netcdf_1b_to_bufr_1d	Pass	3.8 s	MWS conversion from level 1b netcdf to a level 1d BUFR file. The MWS BUFR output is spatially thinned and channel BTs have been spatially averaged. See **Figure 9** and **Figure 10**.
3	MWS_bufr_1b_to_bufr_1d	Pass	5.4 s	As above, but for 1b BUFR input.
4	IASI_NG_netcdf_1c_to_bufr_1d_IASI_SRF	Pass	12.0 s	IASI-NG conversion from level 1c netCDF to level 1d BUFR at IASI first generation spectral resolution. Outputs come from the raw radiances.
5	IASI_NG_netcdf_1c_to_bufr_1c	Pass	8.4 s	IASI-NG conversion from level 1c netCDF to a standard level 1c BUFR at full spatial resolution. See **Figure 11**.
6	IASI_NG_netcdf_1d_to_bufr_1d	Pass	0.8 s	IASI-NG conversion from level 1d netCDF (PC scores) to a standard level 1d BUFR at full spatial resolution.
7	IASI_NG_netcdf_1d_to_bufr_1d_IASI_SRF	Pass	0.9 s	IASI-NG processing from level 1d netCDF (PC scores) to "enhanced" level 1d BUFR, including spatial thinning and conversion to IASI first generation spectral resolution. Outputs are also fed back to the input netCDF file. See **Figure 12**.
8	IASI_NG_netcdf_1d_to_bufr_1d_with_MWS	Pass	1.8 s	IASI-NG processing from level 1d netCDF (PC scores) to "enhanced" level 1d BUFR, including mapped MWS. See **Figure 13** and **Figure 14**.
9	IASI_NG_1c_to_1d_netcdf	-	2.8 s	IASI-NG conversion from level 1c netCDF (full spectrum) to level 1d netCDF (PC scores). This can be used in as a follow-on step to the Direct Broadcast package which generates level 1c as its output.
10	IASI_NG_bufr_1c_to_bufr_1d	Pass	12.5 s	IASI-NG conversion from level 1c BUFR to level 1d BUFR at IASI first generation spectral resolution. Outputs come from the raw radiances.
11	IASI_NG_bufr_1d_to_bufr_1d	Pass	1.1 s	As for test 10, but for level 1d BUFR input. Outputs are reconstructed radiances.
12	IASI_NG_bufr_1d_to_bufr_1d_with_MWS	Pass	3.9 s	As for test 11, but the inputs are BUFR.

[2] Run times are illustrative of performance on a system with 16 GB of memory and 4 CPUs. The duration of the input data used by these test cases is 2.5 minutes for IASI-NG and 100 minutes for MWS.

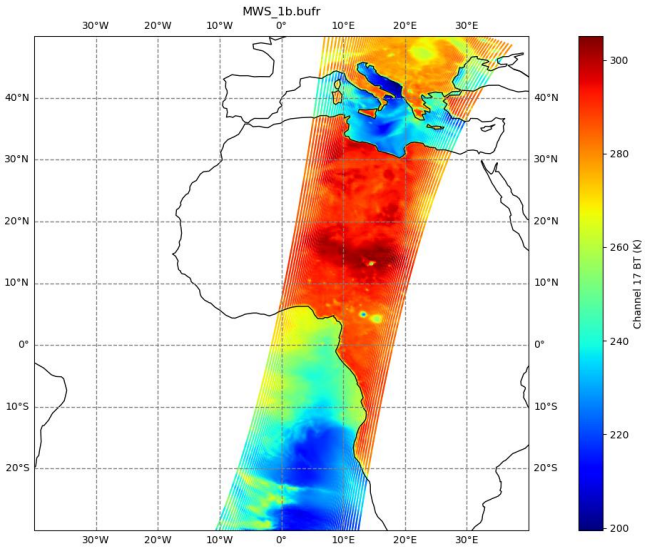

Figure 8: MWS brightness temperature, level 1b full resolution (zoomed in). Test 1.

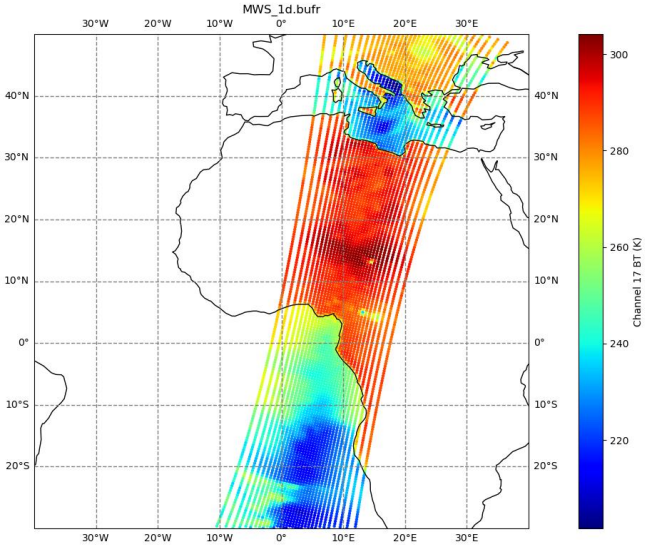

Figure 9: MWS channel 17 brightness temperature, level 1d (thinned and zoomed in). Test 2.

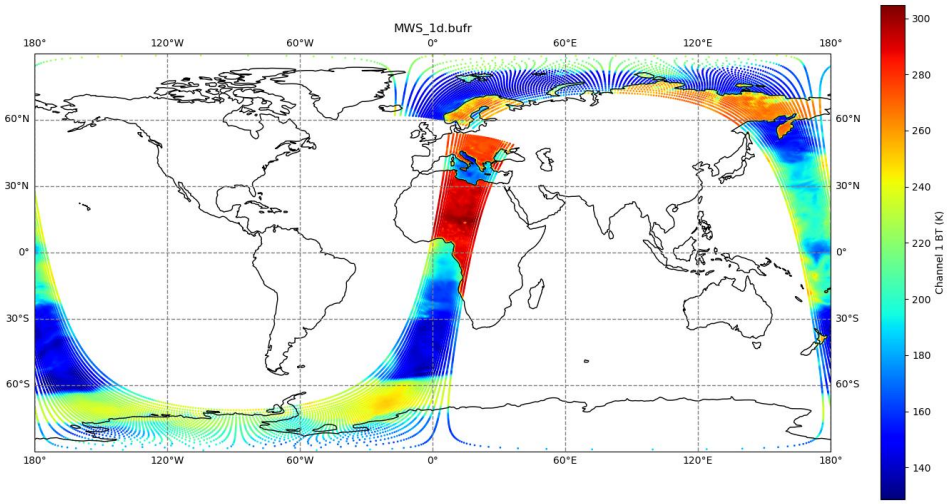

Figure 10: MWS channel 1 brightness temperature, level 1d (thinned). Test 2.

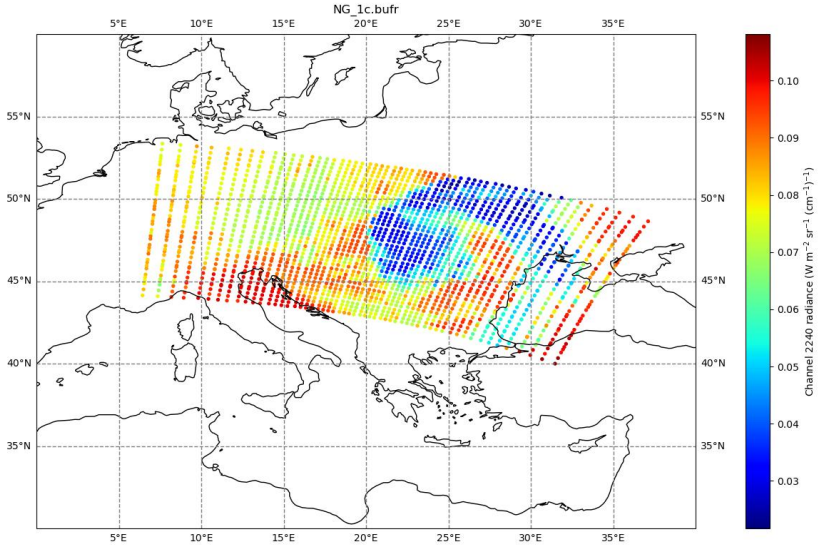

Figure 11: IASI-NG radiance from level 1c. Test 5.

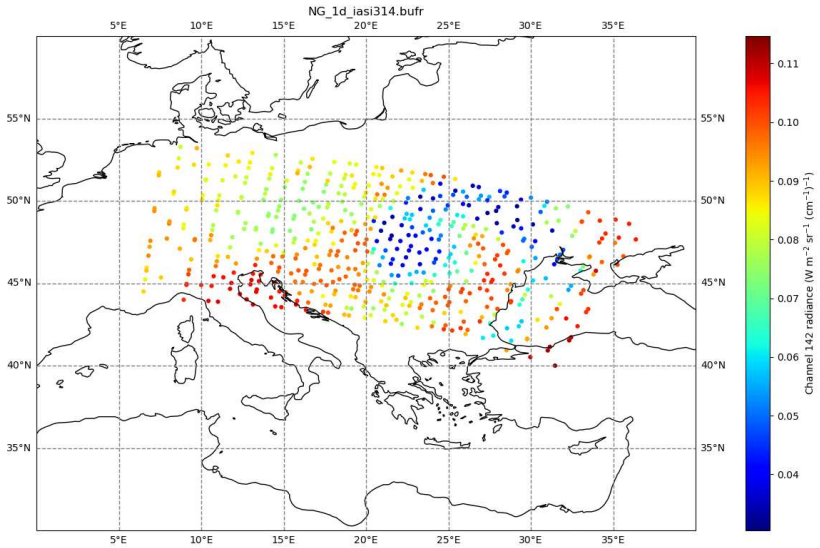

Figure 12: IASI-NG radiance level 1d, thinned with IASI first generation spectral response. Test 7.

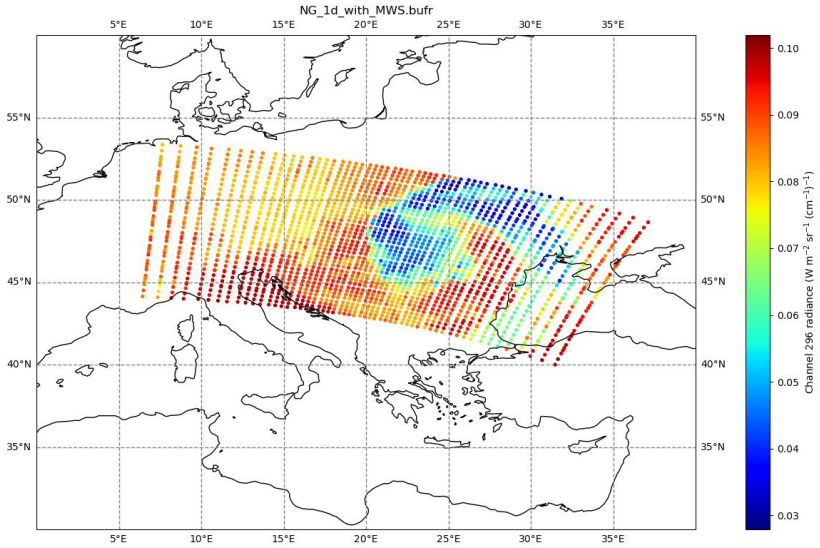

Figure 13: IASI-NG level 1d radiance. Test 8.

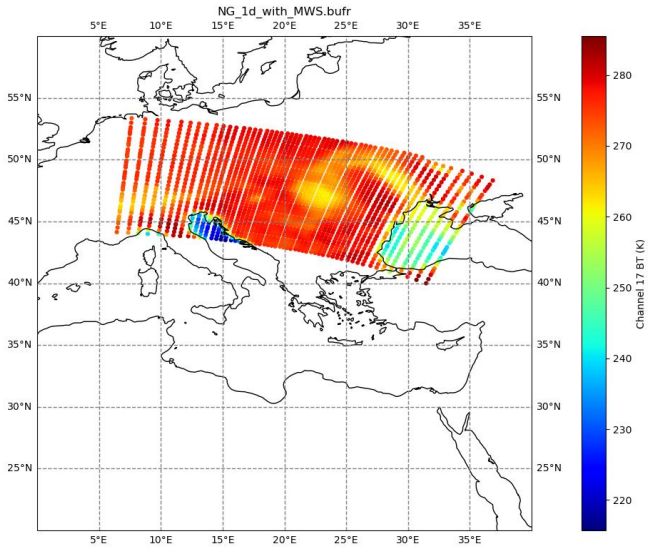

Figure 14: MWS channel 17 mapped to IASI-NG. Test 8.

7.2 Testing on RHEL7 with Apptainer

The AAPP-SG executables were installed on the Met Office *expradsatdev01* server and the test cases run on 9/3/2026 using Apptainer, as per section 2.4.2 of the AAPP-SG User Manual. This server is running RHEL7.9.

One test initially failed in the *bufr_compare* step: it was found that the test script *MWS_bufr_1b_to_bufr_1d.sh* incorrectly defines ECCODES_DEFINITION_PATH, preventing the comparison of generated products with reference products. This was corrected and following that all tests ran successfully.

7.3 Verifying required functionality of install and configure scripts

The functionality of the main AAPP-SG installation scripts (`install_dependencies.sh` and `configure_aapp-sg.sh`) was further verified by carrying out the following set of installations, building AAPP-SG from source code against centrally installed dependencies, existing local dependencies, and/or freshly built local dependencies. The set of tests in Table 6 was then run for each installation: all tests ran successfully; output data content is identical to known good output (although netcdf output metadata differs slightly depending on the versions of hdf5 and netCDF-C used to generate it).

Where new or existing local dependencies are referenced below, the dependency versions are:

- hdf5: 1.14.6;
- netCDF-C: 4.9.3;
- netCDF-Fortran: 4.6.2;

- eccodes: 2.44.0

Table 7: dependency tests

Date	Operating System	Compiler versions	Centrally installed dependencies used (if any)	New local dependencies built	Existing locally installed dependencies used (if any)
06/03/2026	RHEL9.7	GCC 11.5.0 (gfortran, gcc, g++)		hdf5 netCDF-C netCDF-Fortran eccodes	
			hdf5: 1.12.1		netCDF-C netCDF-Fortran eccodes
			hdf5: 1.12.1 netCDF-C: 4.8.1		netCDF-Fortran eccodes
				netCDF-C netCDF-Fortran	hdf5 eccodes
				netCDF-Fortran eccodes	hdf5 netCDF-C
					hdf5 netCDF-C netCDF-Fortran eccodes
		ifort 19.0.0 and GCC 11.5.0 (gcc, g++)		hdf5 netCDF-C netCDF-Fortran eccodes	
				netCDF-C netCDF-Fortran	hdf5 eccodes
				netCDF-Fortran eccodes	hdf5 netCDF-C
					hdf5 netCDF-C netCDF-Fortran eccodes

7.4 Compliance Matrix

Table 8 shows the status of the compliance matrix at the end of testing. The technical testing was all successful. The documents are subject to approval at the DRR.

Table 8: as for Table 2 but including status at the end of testing

Requirement	Testing method	Test plan reference	Comment	Date Completed
7.1 Release note completeness	Inspection	2.4	To be assessed during DRR	
7.2 User Manual	Inspection	2.8	To be assessed during DRR	
7.3 Building and linking	Test	2.1	Includes coding and compilation testing	6/3/2026
7.4 Compile and run on Linux PC or server	Test	2.5		6/3/2026
7.5 Provide AAPP-SG as executables	Test	2.3	Verify that the executables run	9/3/2026
7.6 Test Cases for MWS and IASI-NG	Test	2.3	See Table 1 and subsequent subsections	6/3/2026
7.7 Test Cases available on NWPSAF web site	Inspection	-	To be assessed during release approval	

8. APPENDIX

8.1 Software processing requirements traceability matrix

		Testcases demonstrating functionality											
Processing Requirement	**Processing Options Supported**	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**T11**	**T12**
MWS l1b netCDF → **MWS l1b BUFR**		✓											
MWS l1b netCDF or BUFR → **MWS l1d BUFR**			✓	✓									
	Input datatype: netCDF or BUFR		netCDF	BUFR									
	MWS spatial thinning		✓	✓									
	MWS spatial averaging		✓	✓									
	Run scattering & surface tests		✓	✓									
	Append radiances to input netCDF		✓										
IASI-NG l1c netCDF → **IASI-NG l1d netCDF**										✓			
IASI-NG l1c netCDF → **IASI-NG l1c BUFR**						✓							
IASI-NG l1c/l1d netCDF or BUFR → **IASI-NG l1d BUFR**					✓		✓	✓	✓		✓	✓	✓
	Input datatype: l1c netCDF, l1d netCDF, l1c netCDF, or l1d BUFR				l1c netCDF		l1d netCDF				l1c BUFR	l1d BUFR	
	IASI-NG spatial thinning							✓					
	Output **standard** l1d BUFR						✓						
	Output **enhanced** l1d BUFR (i.e. include IASI-NG radiances)				✓			✓	✓		✓	✓	✓
	↳ Include PC scores in output							✓	✓				✓
	↳ Convert to IASI 1st-gen spectral response				✓			✓			✓	✓	
	↳ Channel selection				IASI Full set			IASI 314	IASI-NG 500		IASI 314	IASI 314	IASI-NG 500
	↳ Remap MWS onto IASI-NG spatial grid								✓				✓
	↳ MWS spatial averaging								✓				✓
	↳ Append radiances to input netCDF							✓					