NWP SAF

AAPP Version 8 Test Plan

Version 1.2

December 2017

AAPP Version 8 Test Plan

Version	Date	Author	Remarks
1.0	06/09/17	N C Atkinson	Initial
1.1	31/10/17	N C Atkinson	Add test results
1.2	14/12/17	N C Atkinson	Add results of additional tests recommended by DRI

Table of Contents

1. INTRODUCTION

This document defines the test plan for Version 8 of the ATOVS and AVHRR Processing Package (AAPP).

AAPP is one of the key deliverables of the NWP SAF project. It is a software package used for the pre-processing of polar-orbiter satellite data, for the purposes of:

- Preparing data for input to NWP assimilation systems
- Imagery generation

The package was first released by EUMETSAT in the late 1990s in order to support the NOAA KLM series of satellites. The primary motivation was to allow users to process direct readout data from these satellites, but the package may also be used to process level 1 data.

From 2004 the package has been an NWP SAF deliverable, and the functionality has been extended include support for MetOp and other satellites (as detailed in AD-1). The main enhancements in the initial release of AAPP v8 are:

- Updated MAIA4 to be used for both AVHRR and VIIRS, replacing MAIA2.1 and MAIA3.
- Interfaces to the ecCodes package for BUFR encoding and decoding.

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.

1.1 Applicable documents

[AD-1] NWP SAF AAPP Version 8 Product Specification: NWPSAF-MO-DS-033
[AD-2] NWP SAF Development Procedures for Software Deliverables: NWPSAF-MO-SW-002
[AD-3] NWP SAF AAPP Version 8 Top Level Design: NWPSAF-MO-DS-034

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 corresponding testing procedures with respect to the AAPP software are outlined below.

2.1 Coding and compilation testing

Coding guidelines are outlined in [AD-2]. Details are beyond the scope of this document, however it should be noted that parts of AAPP were written before the guidelines were put in place, and therefore may not conform strictly to the standards (e.g. headers may not be complete). Also, note that much of AAPP is written in Fortran 77, and therefore coding guidelines for Fortran 90 (referenced in AD-2) are not applicable. Nevertheless, code should be readable and fit for purpose.

The code will be inspected by someone other than the author, who is familiar with AAPP 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 Installation Guide, to run the "configure" and "make" scripts in the AAPP_8 top directory. This will be performed on a number of different platforms/compilers, including Linux (gfortran, ifort, pgf90) and macOS. External libraries (e.g. BUFRDC, HDF5, ecCodes) should be linked in where available.

Whilst the release of AAPP v8 does not necessarily coincide with a new release of OPS-LRS, the compilation testing shall verify that the current version of OPS-LRS can be built with AAPP v8.

2.2 Module testing

In AAPP, all modules (subroutines) are linked to a top-level program, and the user would not normally be expected to link AAPP libraries directly with their source code. Therefore 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 required to agree to the terms of a beta-tester license. They will be expected to review the documentation (see 2.4), run test cases and report on their experiences to the development team.

The different tests are outlined below.

Note that some of the test cases currently available on the NWPSAF web site are several years old, and do not reflect the current status of polar satellites. Therefore a new set of test cases will be prepared.

2.3.1 NOAA HRPT

A NOAA-19 test case shall be run, and output files (HIRS 1d, AMSU 1c, MHS 1c, HIRS 1c) compared with the equivalent output of the latest release of AAPP v7, using the ATOVSCompare utility. There shall be no unexpected differences. Unmapped brightness temperatures would normally be expected to be consistent (with respect to different versions and compilers) to with 0.05K; derived products (e.g. cloud top temperature and mapped AVHRR BTs) to within 1K.

The test will be run in 3 ways:

1. Without AVHRR
2. With AVHRR (AVHRR radiances are mapped to HIRS but MAIA is not run)
3. With AVHRR and MAIA (environment variable RUN_MAIA=yes). GFS forecast files used.

Comparisons will be made with AAPP v7 (running MAIA2.1).

In addition, a MAIA cloud mask will be generated at the AVHRR resolution, using RUN_MAIA.

2.3.2 MetOp AHRPT for ATOVS and AVHRR

As for 2.3.1, but using a Metop-B test case

2.3.3 MetOp AHRPT for IASI

The Metop test case of 2.3.2 will be re-run, this time including IASI, using the current version of OPS-LRS. The test should include generation of PC scores. The output will be examined using tools external to AAPP, to verify that IASI has been correctly processed.

2.3.4 BUFR encode/decode, comparing BUFRDC with ecCodes

The tests are as follows:

1. An AAPP level 1c or 1d dataset (for instruments defined below) is to be BUFR encoded using both *aapp_encodebufr_1c* and *eccodes_encodebufr_1c*. Each BUFR file will be decoded using *bufr_dump* (an ecCodes) tool, and the ASCII decode files compared.
2. The BUFR files from step 1 will be decoded, back to AAPP format, using aapp_decodebufr_1c and eccodes_decodebufr_1c (in all cases where the relevant decoders are implemented in AAPP). The l1c files will be converted to hdf5 using *convert_to_hdf5* and compared using *h5diff*. A 3-way comparison will be made between:
    a. The original l1c file
    b. The BUFRDC decode
    c. The ecCodes decode

    Some differences are to be expected, because not all variables are represented in the BUFR, but there should be no unexpected differences.

The instruments to be tested include: 'HIRS', 'HIRS1D' , 'AMSU-A', 'MHS', 'IASI', 'PCIASI', 'MWHS2', 'IRAS', 'MWRI', 'ATMS', 'CRIS'

2.3.5 Suomi-NPP tests (ATMS, CrIS, VIIRS)

Sensor data records from Suomi-NPP direct readout processing (CSPP) are to be ingested into AAPP, creating level 1c datasets (ATMS, CrIS), and processed onward to CRIS level 1d (with mapped ATMS).

A VIIRS cloud mask shall be generated (MAIA4), using GFS forecast files.

VIIRS to CrIS mapping shall be performed.

2.3.6 FY-3C ingest

Sensor data records from FY-3C direct readout processing (fy3cl1db) are to be ingested into AAPP, creating level 1c datasets for MWHS-2, IRAS and MWRI. (This test could be part of the BUFR test case).

2.3.7 Additional FY-3 ingest tests

AAPP is also required to ingest historical data for the MWHS, MWTS and IRAS instruments on FY-3B and MWTS-2 on FY-3C. These instruments are no longer working, therefore archived data will be used.

2.3.8 General test case requirements

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.

Note that there is not a one to one correspondence between test cases and user requirements: a given test may be used to verify more than one user requirement.

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 v8 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.

2.5 Portability testing

Portability testing will be performed in-house (Met Office and Météo-France), 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.

Tolerances for the product differences obtained using different compilers are the same as those given in 2.3.1.

2.6 Timing testing

Examples of run times will be given in the Test Log. Timing testing serves only as a guideline to other users of AAPP, it is not a benchmark test. For existing AAPP v8 functionality, the testing should determine whether there are any significant timing differences between v8 and v7, and if any significant differences are found they should be explained.

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 Log.

2.8 Documentation testing

AAPP will include a set of updated documents (based on those currently available on the AAPP web page), together with new documents specific to AAPP v8. These will be reviewed internally and by the beta testers, 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 MATRIX

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

Requirement	Testing method	Test plan reference	Comment
7.1 Release note completeness	Inspection	2.4	
7.2 Building and linking	Test	2.1	Includes coding and compilation testing
7.3 Compile and run on specified platforms	Test	2.5	
7.4 Direct readout test cases	Test	2.3.1, 2.3.2, 2.3.3	
7.5 MetOp BUFR and principal components	Test	2.3.3, 2.3.4	
7.6 MAIA4	Test	2.3.1, 2.3.2, 2.3.5	
7.7 Imager clusters	N/A	N/A	Postponed to future update release of AAPP
7.8 BUFR test cases	Test	2.3.4	
7.9 Generation of required output formats	Inspection	2.3.1, 2.3.2, 2.3.3	
7.10 Test cases available on NWPSAF web site	Inspection	2.3.8	Usability of the test cases assessed during validation testing

4. TEST RESULTS

4.1 AAPP v8 beta releases under test

8.01: 25 Sept 2017

Initial release provided to beta testers

8.02: 03 Oct 2017

Change 10.8um channel from 1125 to 1121, i.e. use one of the channels in the channel selection.
For ifort, add flag -heap-arrays, to avoid exceeding the default stack size.
Fix some compilation errors in the sections of code that are used when external libraries are *not* used.
Fix a compliation error when GRIB_API is used. (Normally recommend to use ecCodes).
Correction to IRAS BUFR decode using ecCodes.
Allow the possibility of re-calibrating NESDIS-supplied AMSU-A data (a temporary issue that arose recently with NOAA-15).
Routine update of amsua_clparams.dat.
Some updates to MAIA4, including changes to the satellite IDs.

8.03: 04 Oct 2017

In hdf5 outputs, change all instances of attribute name "period_minutes" to "period_seconds" (values are unchanged).

8.04: 10 Oct 2017

Make EUMETSAT TLEs the default (in ATOVS_ENV8)
Set default USE_OB_TIME=Y in aapp_encodebufr_1c, as this is needed for DBNet.
Issue a build script install_aapp.sh.

8.05: 30 Oct 2017 (submitted for DRI)

Update the satellite list in "configure".
Update atms_sdr_out.F so that only scans with a valid end time are reported (required for CSPP SDR v3.0)
Update usage of eccodes_encodebufr_1c.ksh.
New tool gtsheaders_bufr.pl (already used by some DBNet stations).
Note: the above changes have no impact on test case results.

8.1 release candidate: 14 Dec 2017

Add satellite IDs for JPSS-1 (NOAA-20) and FY-3D, now that both have been launched.
For CrIS FSR BUFR encode/decode, use the NOAA sequence rather than the test sequence used previously. The order of certain descriptors had been swapped swapped to get around an issue in ecCodes, but now we need to support the actual sequence that will be used by NOAA.
For ATMS and CrIS ingest, use attribute Platform_Short_Name instead of Mission_Name because it is better defined in the JPSS documentation.
In atovpp and cris_channels, use the CrIS 431-channel selection if the input data is FSR, and the 399-channel selection if it is NSR.
In aapp_encodebufr_1c, set default ENHANCED_IASI=Y for IASI l1c. This does not affect DBNet (which uses the PC sequence), but ensures that any IASI full-channel BUFR products match EUMETSAT's products.
Change default for BUFR encode so that if no options are specified by the user then the NEDT and software version are encoded by default for ATOVS, but not for other data types. This reflects operational practice since early Dec 2017. Existing options can be used.
In aapp_encodebufr_1c, for ATOVS store originating sub-centre in section 4. This was missing.
In amsua_main.c, correct handling of missing MDRs at the start of a L0 file, which occasionally resulted in bad geolocation for Metop AMSU-A.

4.2 Beta test reports

Developer responses are in blue.

4.2.1 Liam Gumley - CIMSS

03/10/2017 – testing v8.02.

"My build process is identical to the AAPP build process I documented for the DBNet coordination group, where I showed how to build AAPP 7.15 inside a Docker container back in March." Using GRIB-API, not ecCodes.

Compilation error in Maia_Read_PrevConst.F90. Need to define CODES_SUCCESS.

Compilation error in eccodes_decodebufr_1c.F: "Unterminated character constant"

Patches were provided and included in v8.02 "After this patch, the build runs to completion."

The AAPP v8 ATMS/CrIS test case ran to completion.

Metop-B ATOVS test case ran to completion

Metop-B IASI test case ran to completion

NOAA-19 test case ran to completion

In general I thought that the AAPP v8 installation went very much as expected, and there were really no surprises.

Here are some suggestions for your consideration.

1. Include HDF5 products in the test datasets, and use h5dump to permit the end user to compare the locally produced HDF5 files to the reference HDF5 files to verify that the local installation is correct.

This is a good idea. Propose to add reference directories containing l1c* and l1d* files, for the metopb and noaa19 test cases. Don't include AVHRR because these are relatively large.

2. Provide a directory containing the support library tarfiles (e.g., HDF5, BUFRDC, SZIP etc) used at the NWP SAF to build and test the reference version of AAPP and OPS-LRS.

For the external libraries, we would prefer not to put copies on the NWPSAF web site, as the libraries are maintained externally (e.g. ECMWF for BUFRDC and ecCodes) and are readily available. But instead, we have created a build script (see below) that can automatically download appropriate versions of all these libraries.

3. Provide a script showing exactly how the supporting libraries are built at NWP SAF.

4. Provide a script showing exactly how AAPP and MAIA are built at NWP SAF.

5. Provide a script showing how OPS-LRS is built at NWP SAF.

For items 2-5, I know that all the relevant information can be found in the online documentation. However, I think it would help end users to quickly and confidently install and test a new version of AAPP if these items were available.

3 to 5. We have created a build script *install_aapp.sh*. This takes the user through the process of building the core AAPP and the OPS-LRS, together with the required external libraries. This script will be uploaded to the NWPSAF web site and linked to the AAPP Installation Guide.

4.2.2 Sebastian Sanfilippo (CONAE, Argentina)

29/09/2017 – testing v8.01.

Here is the feedback about the AAPP8 installation:

• Are the Release Note and installation instructions easy to follow?

Yes, absolutely, in particular your additional notes.

• Did the code compile OK? Which compiler/OS did you use?

Yes, the code was compiled just fine following the instructions, only a few warnings came up but nothing relevant. Here in ETC I used Centos 6.7, gcc 4.4.7, gfortran 4.4.7, python 2.6.6, perl 5.10.1. In EBM will different : Ubuntu 16.04.3, gcc 5.4.0, gfortran 5.4.0. python 2.7.12, perl 5.22, this station will only process NO18 and NO19, we have plans to add M01 in the near future.

• Were you able to run the test cases? (We would not expect all users to run all the test cases, as different users are interested in different aspects.)

Yes, all the cases were executed successfully, included the MAIA4 cases.

• Were you able to run with your own direct readout data, or other local requirements?

Yes, I did the same test with our real time data from NOAA18/19, Metop-B and NPP, and the result was the expected.

• Is the documentation complete and understandable?

Yes, the documentation was complete and quite detailed.

• Any other suggestions for changes before the public release (planned for late October)?

I suggest you that should mention that download the IASI auxiliary data is mandatory in order to run OPS.

This is now clear from the new build script *install_aapp.sh* which has a section dedicated to installing the IASI auxiliary data.

Also maybe is useful to set the variable USE_OB_TIME to Y in the source code, if that is the expected value.

Agreed. We have set the default USE_OB_TIME=Y in script aapp_encodebufr_1c.ksh which is used with the BUFRDC interface. For the ecCodes interface, it is already the default.

Maybe for someone that is starting from scratch will be useful that the installation guide also include the software needed to process NPP, (nagg, rt-stps, cspp), and Metop (rt-stps, metopizer)

These pieces of external software are already mentioned in the AAPP User Guide, which we think is a better place than the AAPP Installation Guide. See

https://nwpsaf.eu/site/software/aapp/documentation/aapp-v8-userguide/#Metop_satellites

https://nwpsaf.eu/site/software/aapp/documentation/aapp-v8-userguide/#Suomi-NPP_and_JPSS_satellites

In the release notes you mention that "The software and data files require about 9GB of disk space. We recommend that the system has at least 8GB of available memory." those values are too low if you plan to process METOP (IASI) and NPP.

In the release notes for v8.02, this paragraph has been modified to say

> The software and data files require about 9GB of disk space, and you will need space for your own data. We recommend that the system has at least 8GB of available memory, but more is desirable especially if you are planning to run CSPP.

4.2.3 Stephan Zinke (EUMETSAT)

It would be good if the documents NWPSAF-MO-UD-005 (AAPP Installation Guide) and NWPSAF-MO-UD-036 (AAPP User Guide) were available as PDF documents from the website or the ftp site.

We don't have an easy way to create nice-looking PDFs from the web pages. All we can suggest for now is to use your web browser to print to file. These documents have always been html.

AAPP8 uses ecCode, so I need to build ecCodes first. ecCodes is built with CMake, which is not (yet) available on our build/dev machine.

- I need to have CMake installed first by our sysadmins .
- ECMWF is silent about which cmake version to use/install (I will need to find out).
- Installation guide of AAPP8 is silent about which version of ecCodes to build (I will take the latest (2.5.0))

We have cmake version 2.8.12.2 installed on our desktop, but for testing on our dev machine I downloaded the binaries cmake-3.8.0-rc1-Linux-x86_64.tar.gz. The *install_aapp.sh* script calls for ecCodes 2.5.0. We recommend always to use the latest because they have made some significant performance enhancements recently, and there is still a bug that we are waiting for them to fix (affecting encoding of CrIS in the new BUFR sequence).

I have a problem in decoding CrIS and IASI bufr products created with AAPP8.

This was traced to a bug in the gtsheaders_bufr.pl script that EUMETSAT use to add GTS headers. The bug was already known, but EUMETSAT were using an old version. *Solution is to include gtsheaders_bufr.pl in AAPP v8.1, so that it a properly version-controlled tool is distributed.*

One more thing: When I looked at eccodes_encodebufr_1c, I couldn't find "CRIS" in the helptext, although the SW seems to handle it.

Agreed. The "usage" has been updated (for v8.05)

I have now successfully built the ecCodes (2.5.0) and AAPP 8.04.
When looking at ATOVS_ENV8 I saw "M04" (presumably for MetC). Shouldn't this be M03? Or am I mistaken?

M04 was originally used for Metop Simulator, before the launch of Metop-A. Since this is no longer relevant, the identifier has been changed to M03, ready for Metop-C. Also, have removed the obsolete noaa16 and noaa17. (For re-processing historical data, users have always understood that they should set up their own PAR_NAVIGATION_DEFAULT_LISTESAT).

Problem with BUFR tables when generating the HIRS and IASI level 1d BUFR products using BUFRDC.

This problem was traced to the fact that EUMETSAT had modified their ATOVS_ENV8 to use a central directory of BUFR tables, so the test script was unable to use the ones supplied with AAPP. With the default ATOVS_ENV8, it should be OK. Note that level 1d BUFR products are not intended for international exchange, as they use Met Office tables.

Conclusion: "I would then consider our beta-testing successful".

4.3 Met Office testing

4.3.1 AAPP v8.01

The following test cases were run on "radsat" machine:

MAIA4_test
atmscris_test
bufr_test
metopb_test
noaa19_test
viirs_to_cris

The following issues were raised: Developer response in blue

- For gfortran and ifort, in the compare_decoded_data test, in the .diff_bufrdc file, there is an error message <iras_FY3C_20071_07-AUG-2017_20:37:40.l1c.h5>: unable to open file. File seems fine; instruments should all be processed in a similar way so unsure why iras is giving this error.

The iras*.h5 was not present in eccodes_decoded, because convert_to hdf5 had failed. Traced to a typo in eccodes_get_1c_iras.F. Instrument should be 17 not 19. *Code corrected 02/10/2017.*

- For gfortran compiler testing, xxdiff eccodes_bufr/hirsl1c_M01_20170725_1004_25171.bufr.txt ../bufrdc_bufr/hirsl1c_M01_20170725_1004_25171.bufr.txt does show a minor difference in instrument temps
    - radiometerIdentifier, value 0; instrumentTemperature 286.3 vs an array of values 286.3 and 286.4.
    - radiometerIdentifier, value 0; instrumentTemperature 286.4 vs an array of values 286.3 and 286.4.

These are due to the way the different BUFR packages round number sequences like [286.34,286.35], which are only specified to a precision of 0.1K in the BUFR. BUFRDC appears to round up, so sees a change in mid-message. *No action needed.*

- Unable to build MAIA with g95 compiler

Agreed with Météo-France to remove g95 from the list of supported compilers and to advise Linux users to use gfortran or ifort (or pgf90if not running OPS-LRS). Noted also that there are problems building some of the external packages with g95. Support for gfortran is much better these days than it is for g95.

- Some #ifdef HAS_LIBHDF5 statements are missing in MAIA

Meteo-France provided code updates. We will re-run the compilation test and the MAIA4 test (see later).

- Some of the file sizes are incorrect in the 00README files

Test cases will be re-generated using the final AAPP beta release.

- Installing OPS-LRS: Make install should ideally be performed for both FFTW and XERCES (is only suggested in manual for XERCES).

This is clear in the new build script *install_aapp.sh*. Propose to update the OPS-LRS User Manual next time there is an OPS-LRS release.

- ATMS/CrIS test case: there are two l1d files generated (not mentioned in the README)

Agreed. Have modified the README to mention these files.

- BUFRDC decode not implemented in AAPP for MWRI - should be made more explicit in 00README.txt: Missing mwri in bufrdc_decoded and therefore mwri_FY3C_20071_07-AUG-2017_20:37:40.l1c.h5.diff_bufrdc.

Agreed. Have modified the README to say "Note that the decode using BUFRDC is NOT implemented for MWRI, so you will see a warning message."

- For the VIIRS to CrIS test, to run MAIA4, you need to link GRIB_API. However, this section has been removed from the installation guide. The test could be modified/a comment provided?

I have updated the readme file, changing this sentence to "To run MAIA4, you also need to link ecCodes or GRIB_API (see the AAPP Installation Guide)."

4.3.2 AAPP v8.02

All tests ran to completion. The only difference is that the iras.diff_bufrdc and iras.diff_orig files are correctly populated. Compiler used: gfortran.

4.3.3 AAPP v8.04

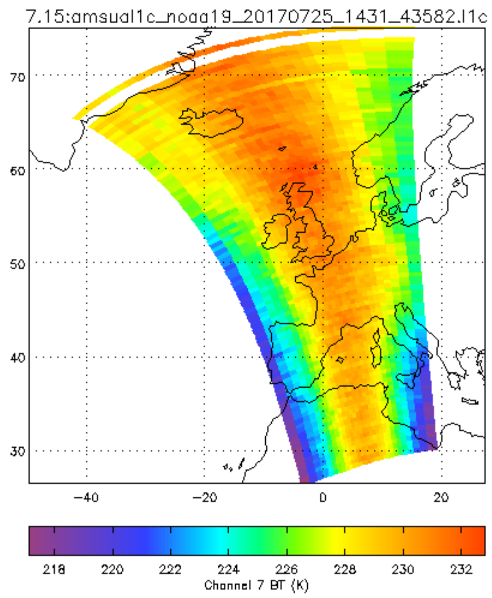
All tests ran to completion (using gfortran).

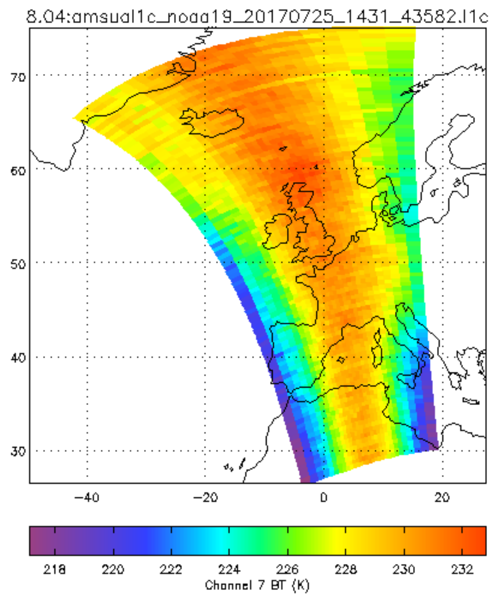
The run times for each test were as follows:

atms_cris: 55 sec
noaa19_test: 4 sec ATOVS, 1 min 10s with MAIA4, 1 min 24s maia_avhrr
metopb_test: 6 sec ATOVS, 3 min 40s IASI, 1 min 33s with MAIA4

viirs_to_cris with MAIA4: 1 min 6s
bufr_test: 9s ecCodes, 5s BUFRDC, 1 min 19s to compare products
MAIA4 VIIRS: 42s

Comparisons were made with AAPP v7.15, for the noaa19 and metopb cases. For noaa19 AMSU-A there was a difference in channel 7: v7.15 had a data gap over part of the image, due to over-stringent quality control on the cal counts, whereas v8.04 correctly calibrated these lines.





The other significant difference, as expected, is the MAIA cloud mask on the HIRS grid, with MAIA2.1 being used for AAPP v7.15 and MAIA4 being used for v8.04.

The AAPP v8 Product Specification lists certain requirements on output products (section 7.9). It was verified by inspection that these products are present:
For IAPP:
hirsl1d_M01_20170725_1004_25171.l1d
hirsl1d_noaa19_20170725_1431_43582.l1d
For MIRS:
amsual1b_noaa19_20170725_1431_43582.l1b
mhsl1b_noaa19_20170725_1431_43582.l1b
amsual1b_M01_20170725_1004_25171.l1b
mhsl1b_M01_20170725_1004_25171.l1b
For UW Hyperspectral:
IASI_xxx_1C_M01_20170725100425Z_20170725101705Z_V_T_20171011150900Z

Portability testing

- gfortran 4.4.7 – used for majority of tests
- ifort 12.0.4 – full functionality. Test cases ran correctly with no significant differences wrt gfortran.
- pgf90 and g95 – only able to build "core" AAPP with no external libraries, due to issues building the external libraries. For pgf90, ran those parts of the test cases that were compatible with the build.

Testing the compatibility with CSPP SDR v3.0 (released by SSEC/CIMSS 13th Oct 2017)
It was found that CSPP SDR v3.0 can report ATMS scans for which the start time is valid but the end time is missing. This causes problems downstream in AAPP. Solution is to modify atms_sdr_out.F to check the scan end time.

4.3.4 AAPP v8.1 release candidate

This section addresses issues raised during the DRI review.

NOAA ingest (see Section 3 of Product Specification)

A test was created to check the ingest of NOAA level 1b files, current and historical.

Inputs:

```
-rw-r--r-- 1 frna satsound  4537951 Dec 13 16:13 NSS.HIRX.NJ.D01162.S0402.E0556.B3322829.WI
-rw-r--r-- 1 frna satsound 74148000 Dec 13 16:10 NSS.HRPT.NJ.D01162.S2059.E2113.B3323838.WI
-rw-r--r-- 1 frna satsound  2091520 Dec 13 15:56 NSS_AMAX.NP_D17347_S1245_E1434_B4559798_GC
-rw-r--r-- 1 frna satsound  4704768 Dec 13 15:56 NSS_HIRX.NP_D17347_S1245_E1434_B4559798_GC
-rw-r--r-- 1 frna satsound  7523328 Dec 13 15:56 NSS_MHSX.NP_D17347_S1245_E1434_B4559798_GC
```

Outputs generated:

```
-rw-r--r-- 1 frna satsound   4916736 Dec 13 16:25 NSS.HIRX.NJ.D01162.S0402.E0556.B3322829.WI.l1b
-rw-r--r-- 1 frna satsound   7101952 Dec 13 16:25 NSS.HIRX.NJ.D01162.S0402.E0556.B3322829.WI.l1c
-rw-r--r-- 1 frna satsound 110300160 Dec 13 16:25 NSS.HRPT.NJ.D01162.S2059.E2113.B3323838.WI.l1b
-rw-r--r-- 1 frna satsound   2509824 Dec 13 16:25 NSS_AMAX.NP_D17347_S1245_E1434_B4559798_GC.l1c
-rw-r--r-- 1 frna satsound   6795776 Dec 13 16:25 NSS_HIRX.NP_D17347_S1245_E1434_B4559798_GC.l1c
-rw-r--r-- 1 frna satsound  11284992 Dec 13 16:25 NSS_MHSX.NP_D17347_S1245_E1434_B4559798_GC.l1c
```

The test was successful. Not proposed for release to users, since the method is well described in the AAPP User Guide.

FY-3 ingest and BUFR (see Section 3 of Product Specification)

Requirement to test ingest and BUFR encode/decode of instruments on FY-3B, and also the failed MWTS-2 on FY-3C. Note: at this stage we are not claiming that AAPP v8.1 is ready for FY-3D, since CMA often introduces format changes for new satellites that are not publicised in advance.

Inputs:

```
-rw-r--r-- 1 frna satsound 13640100 Dec 14 09:34 FY3B_IRASX_GBAL_L1_20130129_1356_017KM_MS.HDF
-rw-r--r-- 1 frna satsound 14995336 Dec 14 09:36 FY3B_MWHSX_GBAL_L1_20130710_0203_015KM_MS.HDF
-rw-r--r-- 1 frna satsound   466448 Dec 14 09:35 FY3B_MWTSX_GBAL_L1_20130710_0203_060KM_MS.HDF
-rw-r--r-- 1 frna satsound 13052916 Dec 14 09:41 FY3C_IRASX_GBAL_L1_20140910_1203_017KM_MS.HDF
-rw-r--r-- 1 frna satsound 18786264 Dec 14 09:36 FY3C_MWHSX_GBAL_L1_20141006_2226_015KM_MS.HDF
-rw-r--r-- 1 frna satsound  5405400 Dec 14 09:35 FY3C_MWTSX_GBAL_L1_20150115_0219_033KM_MS.HDF
```

Outputs: For each input file:
1) Level 1c file generated from the HDF
2) BUFR file generated using BUFRDC
3) BUFR file generated using ecCodes
4) Decoded l1c file from the BUFRDC BUFR
5) Decoded l1c file from the ecCodes BUFR

Images were generated from each l1c file and were consistent. The test was successful.

Metop AHRPT problems at the start of pass

2 cases were provided by EUMETSAT that were incorrectly geolocated in the operational EARS system:
Maspalomas: AMSA_HRP_00_M01_20171115204818Z_20171115205850Z_N_O_20171115205743Z
Svalbard: AMSA_HRP_00_M01_20171118213825Z_20171118215226Z_N_O_20171118215242Z

These were correctly processed by AAPP v8.1 (see below for the Mas pass).

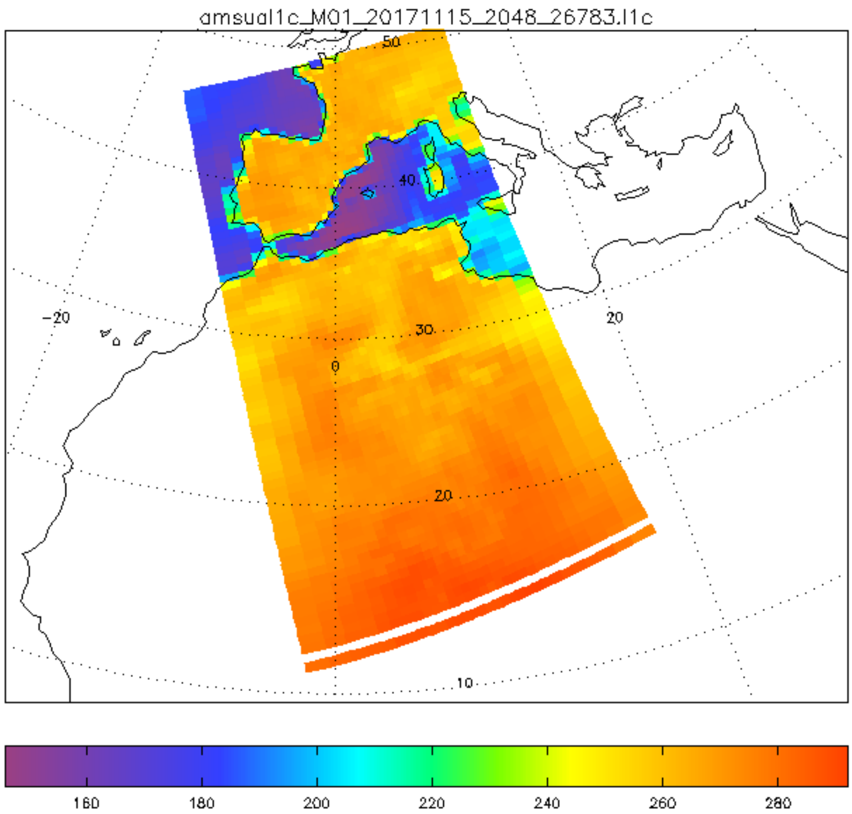


Figure 1: AMSU-A channel 1 image for Maspalomas on 15th Nov 2017

Also re-tested a 2014 case that was previously tested for AAPP v7.8:
AMSA_xxx_00_M01_20141117015641Z_20141117021105Z_N_O_20141117015641Z. This file has an awkward gap in mid-pass. Results were the same as for v7.8, i.e. OK.

<u>BUFR encoding – sub-centre</u>

Re-ran the BUFR tests with

```
export ORIGINATING_CENTRE=74
export SUB_CENTRE=1
```

Confirmed that for ATOVS the sub-centre is present in Section 1 and that NEDTs are included by default. Test successful.

<u>Handling of CrIS FSR data</u>

This is part of the required JPSS-1 functionality. We test that:

- CrIS FSR SDR files (from CSPP) can be ingested successfully
- CrIS FSR BUFR files can be decoded
- CrIS channel selection (cris_channels script) correctly picks up the 431-channel selection.

Inputs:

```
107544 Dec 14 11:45 GCRSO_npp_d20150411_t0836019_e0836317_b17890_c20160516103433966300_cspp_dev_fsr.h5
1013336 Dec 14 11:44 NUCAPS-C0431_v2r0_npp_s201706040107039_e201706040107337_c201706050150070.bufr
4607400 Dec 14 11:44 NUCAPS-C2211_v2r0_npp_s201706021049439_e201706021050137_c201706021319020.bufr
```

28970348 Dec 14 11:45 SCRIS_npp_d20150411_t0836019_e0836317_b17890_c20160516103433851782_cspp_dev_fsr.h5

Outputs:

2501560 Dec 14 11:51 NUCAPS-C0431_v2r0_npp_s201706040107039_e201706040107337_c201706050150070.l1c
2501560 Dec 14 11:51 NUCAPS-C2211-431_v2r0_npp_s201706021049439_e201706021050137_c201706021319020.l1c
12113560 Dec 14 11:51 NUCAPS-C2211_v2r0_npp_s201706021049439_e201706021050137_c201706021319020.l1c
12113560 Dec 14 11:51 SCRIS_npp_d20150411_t0836019_e0836317_b17890_c20160516103433851782_cspp_dev_fsr.l1c
2501560 Dec 14 12:23 SCRIS-431_npp_d20150411_t0836019_e0836317_b17890_c20160516103433851782_cspp_dev_fsr.l1c

It was verified (by IDL) that the output of cris_channels (in both cases – SDR and BUFR) contains the same channel set as the NUCAPS selection.

For the remaining aspects of JPSS-1 functionality – these were tested by inspection: WMO satid is set in cris_sdr_out.F and atms_sdr_out.F. Also there is an entry in satid.txt:

```
SATELLITE JPSS-1
noaa            satellite serie
20              satellite number in satellite serie
2017 073A       s/c international designation (TWO lines catalog 43013)
jpss-1          pre-launch name
noaa20          first alias
noaa-20         second alias
jpss1           third alias
18/11/17        operational start date
01/01/49        operational end   date
0               number of instruments described in file for this satellite
```

4.4 Météo-France testing

See the following documents detailing the testing of MAIA4:

maia4-cibles-metop-78-dev.pdf
maia4-target-npp-7-vs-8-dev.pdf
maia4_software_tests_report_CEP.pdf
maia4_software_tests_report_avh2hirs_atovs_new.pdf
maia4_software_tests_report_avhrr.pdf
maia4_software_tests_report_maia2pfs.pdf
maia4_software_tests_report_newSSTformula.pdf
maia4_software_tests_report_viirs.pdf

4.5 Compliance to Test Plan

Test plan paragraph	Summary of requirement	How done	Compliant?
2.1	Coding inspection	Met Office examined Meteo-France code, and vice versa.	Yes
2.2	Module testing	All modules are tested in an integrated environment	Yes
2.3.1	NOAA HRPT test case	Run noaa19_test.	Yes
2.3.2	Metop AHRPT, ATOVS and AVHRR	Run metopb_test.	Yes
2.3.3	Metop AHRPT, IASI	Run metopb_test, using the current version of OPS-LRS	Yes
2.3.4	BUFR encode/decode	Run bufr_test. See Appendix for detailed	Yes

	using BUFRDC and ecCodes	analysis of these results.	
2.3.5	Suomi-NPP	Run (i) atmscris_test and (ii) viirs_to_cris	Yes
2.3.6	FY-3C ingest	This is part of bufr_test	Yes
2.3.7	Test cases for users available on NWPSAF web site	Test cases, including instructions, have been prepared and will be put on the NWPSAF web site when AAPP v8 is released.	Yes
2.4	Validation testing through beta testers	See the reports in this document	Yes
2.5	Portability testing	AAPP v8 has been tested on Linux PC with gfortran, ifort, pgf90 and g95. It is made clear in the documentation that gfortran and ifort are the preferred compilers. Limited testing has been done on macOS, at Meteo-France before release of v8.01; if users encounter issues on this platform it will be dealt with via the NWP SAF Helpdesk in the usual way.	Yes
2.6	Timing testing	Run times for the test case are reported above. Note that avh2hirs mapping is significantly slower with maia4 than maia2.1, but this is expected; the default for AAPP_RUN is not to run maia4.	Yes
2.7	Regression testing	Differences between releases are listed in this document.	Yes
2.8	Documentation testing	Beta testers have reviewed the documentation	Yes

Appendix: Detailed analysis of the BUFR test results

Comparison of BUFR files generated using (i) BUFRDC, (ii) ecCodes

HIRS, MHS, AMSU-A and MWHS-2

Default Master Table is 13 for BUFRDC and 16 for ecCodes.

ATMS, CrIS and PCIASI

No differences

HIRS1D

Master table set to 15 in the test scripts (because that is the version supplied in AAPP/data/bufrtables). No differences.

Comparison of original l1c files (converted to hdf5) with files that have had encode/decode cycles using (i) BUFRDC, (ii) ecCodes

MHS and AMSU-A

Variable	BUFRDC decoded	ecCodes decoded	Original value, or comment
ATTRIBUTE "site"	“254”	“254”	“UKM”
ATTRIBUTE "1bsite"	“xxx”	“000”	“UKM”
ATTRIBUTE "coldnedt_average_mK"	Zero	Zero	Correct (not represented in BUFR)
ATTRIBUTE "warmnedt_average_mK" (when –N flag is not used)	Zero	Zero	Correct.
ATTRIBUTE "warmnedt_average_mK" (with –N flag)	differences ~ 0.001K	differences ~ 0.001K	Correct.
DATASET "coldnedt"	Zero	Zero	Correct (not represented in BUFR)
DATASET "instrtemp"	To nearest 0.1K	To nearest 0.1K	Correct. Originally 0.01K.
DATASET "warmnedt" (when –N flag is not used)	Zero	Zero	Correct.
DATASET "warmnedt" (with –N flag)"	differences <0.01K	differences <0.01K	Precision is 0.01K in the BUFR
DATASET "Satellite_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	0 to 360 degrees in the original dataset.
DATASET "Solar_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	0 to 360 degrees in the original dataset.

Note: after BUFR decode, we set the “site” and “1bsite” to reflect the BUFR identifiers, rather than the local station.

HIRS

Variable	BUFRDC decoded	ecCodes decoded	Original value, or comment
ATTRIBUTE "site"	"254"	"254"	"UKM"
ATTRIBUTE "1bsite"	"xxx"	"000"	"UKM"
ATTRIBUTE "HIRS_temprad_offset"	5 decimal places	5 decimal places	Correct. Originally 6 decimal places.
ATTRIBUTE "HIRS_temprad_slope"	5 decimal places	5 decimal places	Correct. Originally 6 decimal places.
ATTRIBUTE "coldnedt_average_mK"	Zero	Zero	Correct (not represented in BUFR)
ATTRIBUTE "warmnedt_average_mK" (when –N flag is not used)	Zero	Zero	Correct.
ATTRIBUTE "warmnedt_average_mK" (with –N flag)	differences <0.07K	differences <0.07K	Correct. Larger diffs than AMSU because HIRS dataset only has 3 independent calibrations.
DATASET "btemps"	Channel 20 is to nearest 1% in albedo	Channel 20 is to nearest 1% in albedo	Correct. Originally 0.01%
DATASET "coldnedt"	Zero	Zero	Correct (not represented in BUFR)
DATASET "instrtemp"	To nearest 0.1K	To nearest 0.1K	Correct. Originally 0.01K.
DATASET "warmnedt" (when –N flag is not used)	Zero	Zero	Correct.
DATASET "warmnedt" (with –N flag)"	differences <0.01K	differences <0.01K	Precision is 0.01K in the BUFR
DATASET "Satellite_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	0 to 360 degrees in the original dataset.
DATASET "Solar_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	0 to 360 degrees in the original dataset.

Note: after BUFR decode, we set the "site" and "1bsite" to reflect the BUFR identifiers, rather than the local station.

ATMS

Variable	BUFRDC decoded	ecCodes decoded	Original value, or comment
ATTRIBUTE "site"	"254"	"254"	"None"
ATTRIBUTE "1bsite"	"xxx"	"000" (better)	"None"
ATTRIBUTE "centre"	254 (taken from BUFR Section 1)	254 (taken from BUFR Section 1)	255 (missing)
ATTRIBUTE "subcentre"	0 (taken from BUFR Section 1)	0 (taken from BUFR Section 1)	255 (missing)
DATASET "nedtcold"	To nearest 0.01K	To nearest 0.01K	Original: 0.001K
DATASET "nedtwarm"	To nearest 0.01K	To nearest 0.01K	Original: 0.001K
DATASET "scnlin"	Max value 254	Max value 254	OK

DATASET "Satellite_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 deg. Prefer 0 to 360 for consistency with other datasets
DATASET "Solar_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 deg. Prefer 0 to 360 for consistency with other datasets

Note: updated enddatatime in both decoders, to be consistent with original l1c.

Note: update the channel and band corrections in decoders, so that it's done in the same way as atms_sdr.

CrIS

Variable	**BUFRDC decoded**	**ecCodes decoded**	**Original value, or comment**
ATTRIBUTE "site"	"254"	"254"	"None" in the original.
ATTRIBUTE "1bsite"	"254"	"000" (better)	"None" in the original.
ATTRIBUTE "centre"	254 (taken from BUFR Section 1)	254 (taken from BUFR Section 1)	255 (missing) in the original
ATTRIBUTE "chanvn"	0	0	Original: 1. Not in the BUFR

Note: updated enddatatime in both decoders, to be consistent with original l1c.

Note: bug fix in cris1c_read.c-F: allocation was wrong for fovqualflgs, leading to incorrect geolocqual in the hdf5 file.

PCIASI

Variable	**BUFRDC decoded**	**ecCodes decoded**	**Original value, or comment**
ATTRIBUTE "site"	"EUM"	"254"	"XXX" in the original
ATTRIBUTE "1bsite"	"EUM"	"000"	"XXX" in the original
ATTRIBUTE "chanvn"	0	0	5 in the original. Not in the BUFR.
ATTRIBUTE "eigvn"	0	0	8 in the original. Not in the BUFR.
ATTRIBUTE "evfile1", "evfile2", "evfile3"	Not present	Not present	Not available in the BUFR
ATTRIBUTE "period_minutes"	6081	6081	6001
ATTRIBUTE "satht_hectometres"	8330	8330	8281
DATASET "GEUMAvhrr1BCldFrac"	-999999	-999999	255
DATASET "GEUMAvhrr1BLandFrac"	-999999	-999999	255
DATASET "avhrrfrac"			Precision reduced by the BUFR encoding

DATASET "GEUMAvhrr1BQual"	-999999	-999999	-1
DATASET “scalrad” attributes “startchan” and “endchan”	5 unused values are -999999	5 unused values are -999999	5 unused values are 0
DATASET "Satellite_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 degrees
DATASET "Solar_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 degrees

IASI

Variable	BUFRDC decoded	ecCodes decoded	Original value, or comment
ATTRIBUTE "site"	“EUM”	“254”	“XXX” in the original
ATTRIBUTE "1bsite"	“EUM”	“000”	“XXX” in the original
ATTRIBUTE "period_minutes"	6081	6081	6001
ATTRIBUTE "satht_hectometres"	8330	8330	8281
DATASET "GEUMAvhrr1BCldFrac"	-999999	-999999	255
DATASET "GEUMAvhrr1BLandFrac"	-999999	-999999	255
DATASET "avhrrfrac"			Precision reduced by the BUFR encoding
DATASET "GEUMAvhrr1BQual"	-999999	-999999	-1
DATASET "Satellite_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 degrees
DATASET "Solar_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 degrees
DATASET “scalrad” attributes “startchan” and “endchan”	5 unused values are -999999	5 unused values are -999999	5 unused values are 0

MWHS2

Variable	BUFRDC decoded	ecCodes decoded	Original value, or comment
ATTRIBUTE "site"	“EUM”	“254”	“None” in the original
ATTRIBUTE "1bsite"	“EUM”	“000”	“None” in the original
ATTRIBUTE "startdatatime_ms"	74160000	74160000	74159000 (limited by precision 1 sec in the BUFR)
ATTRIBUTE "vnantennacorr"	-999999	-999999	0
DATASET "instrtemp1"	0	0	-999999
DATASET "instrtemp2"	0	0	-999999
DATASET "orbitangle"	0	0	Not represented in the BUFR
DATASET "satposition"	0	0	Not represented in the BUFR
DATASET "satvelocity"	0	0	Not represented in the BUFR
DATASET "scnlintime"			Limited by precision

			of 1 sec in the BUFR
DATASET "Satellite_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 degrees
DATASET "Solar_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 degrees

MWRI

Variable	BUFRDC decoded (not implemented for MWRI)	ecCodes decoded	Original value, or comment
ATTRIBUTE "site"	N/A	"254"	"None" in the original
ATTRIBUTE "1bsite"	N/A	"000"	"None" in the original
ATTRIBUTE "enddatatime_ms"	N/A	74981000	74980656 (limited by precision 1 sec in the BUFR)
ATTRIBUTE "startdatatime_ms"	N/A	74262000	74262464 (limited by precision 1 sec in the BUFR)
ATTRIBUTE "vnantennacorr"	N/A	-999999	0
DATASET "instrtemp1"	N/A	0	-999999 in the original. Not represented in the BUFR
DATASET "instrtemp2"	N/A	0	-999999 in the original. Not represented in the BUFR
DATASET "orbitangle"	N/A	0	Not represented in the BUFR
DATASET "satposition"	N/A	0	Not represented in the BUFR
DATASET "satvelocity"	N/A	0	Not represented in the BUFR
DATASET "scnlintime"	N/A		Limited by precision of 1 sec in the BUFR
DATASET "Satellite_azimuth_angle"	N/A	0 to 360 degrees	-180 to 180 degrees
DATASET "Solar_azimuth_angle"	N/A	0 to 360 degrees	-180 to 180 degrees

IRAS

Variable	BUFRDC decoded	ecCodes decoded	Original value, or comment
ATTRIBUTE "site"	"EUM"	"254"	"" in the original
ATTRIBUTE "1bsite"	"EUM"	"000"	"" in the original
ATTRIBUTE "startdatatime_ms"	74166000	74166000	74165950 (limited by precision 1 sec in the BUFR)
ATTRIBUTE "satht_hectometres"	8360	8360	8350
ATTRIBUTE "IRAS_wavenumber"	14431.0306	14431.0273	14431.0288 Some very small

			differences. The above values are for channel 21.
DATASET "scnlintime"			Limited by precision of 1 sec in the BUFR
DATASET "Satellite_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 degrees
DATASET "Solar_azimuth_angle"	-180 to 180 degrees	0 to 360 degrees	-180 to 180 degrees