NWP SAF

AAPP Version 7 Test Plan

Version 1.0

21st October 2011

AAPP Version 7 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 1 December 2006, 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, KNMI and Météo France.

Change record				
Version	**Date**	**Author**	**Approved**	**Remarks**
0.1	28.07.11	N C Atkinson	-	Initial draft
1.0	21.10.11	N C Atkinson	B.J.Conway	

Table of Contents

1. INTRODUCTION

This document defines the test plan for Version 7 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 enhancement in AAPP v7 is to support the pre-processing of instruments on the NPP and JPSS platforms.

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 7 Product Specification: NWPSAF-MO-DS-014
[AD-2] NWP SAF Development Procedures for Software Deliverables: NWPSAF-MO-SW-002
[AD-3] NWP SAF AAPP Version 7 Top Level Design: NWPSAF-MO-DS-015
[AD-4] Annex to AAPP scientific documentation: Pre-processing of ATMS and CrIS: NWPSAF-MO-UD-027

2. TESTED ITEMS

The required 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 with 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 first test will be a visual inspection of the code 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_7 top directory. This will be performed on a number of different platforms/compilers, including Linux (gfortran, ifort, pgf90), Sun, AIX and HPUX (if available). External libraries (e.g. BUFR, HDF5, GRIB API) should be linked in where available.
Ref: AD1 section 8.2.

Whilst the release of AAPP v7 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 v7.

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. References to specific user requirements in AD1 are given in *red*. The relevant test cases, which will also be distributed to AAPP users on release, are given in *blue*.

2.3.1 NOAA HRPT

Existing NOAA test cases (available on the AAPP ftp server) 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 v6, using the ATOVSCompare utility. There shall be no unexpected differences. Use a TBUS test case and a TLE test case. 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.
Ref: AD1 section 8.4.
*Test case: **noaa16_test** (TBUS); **noaa18_test** (TLE)*

2.3.2 MetOp AHRPT for ATOVS

An existing MetOp test case (available on the AAPP ftp server) 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 v6. There shall be no unexpected differences.
Ref: AD1 section 8.4.

Test case: ***metopa_20100128****, script* ***run_atovs_avhrr.sh***

2.3.3 MetOp AHRPT for IASI

An existing MetOp test case (available on the AAPP ftp server) shall be run, including OPS-LRS for processing IASI. Output shall be examined using IDL tools (external to AAPP). There shall be no significant differences between the output from AAPP v7 and the latest release of AAPP v6.
Ref: AD1 section 8.4.
Test case: ***metopa_20100128****, script* ***run_iasi_OPS.sh***

2.3.4 BUFR encode/decode for ATOVS and IASI

An existing MetOp test case (available on the AAPP ftp server), or equivalent, shall be run to exercise the BUFR encode/decode capabilities in AAPP v7 for AMSU, MHS, HIRS and IASI, and to exercise the IASI Principal Component (PC) routines (i.e. generation of reconstructed radiances and BUFR encode of the PCs). There shall be no run-time warnings.
Ref: AD1 section 8.5.
Test case: ***metopa_20100128****, script* ***run_atovs_bufr.sh***

Additionally, the encode facility for HIRS and IASI level 1d shall be run, with a standard inspection tool (e.g. decode_bufr in the ECMWF library) being used to check the output. This is new for AAPP v7.
Test case: ***metopa_20100128****, scripts* ***run_atovs_bufr.sh*** *and* ***run_iasi_PC_bufr.sh***

2.3.5 BUFR encode/decode for ATMS and CrIS

This capability is new for AAPP v7. Simulated ATMS and CrIS BUFR data from the NOAA ftp server ftp://ftp2.orbit.nesdis.noaa.gov/smcd/czhang/ shall be used. The aapp_decodebufr_1c and aapp_encodebufr_1c tools shall be exercised using this data. Level 1c files shall be examined using an IDL reader (external to AAPP), and output BUFR files examined using decode_bufr (or a similar tool).
Ref: AD1 section 8.8.
Test case: ***ATMS_CrIS****, scripts* ***atms_cris_bufr.sh*** *and* ***process_atms_cris_1c.sh***

Level 1d files (generated by test 2.3.7) will also be BUFR encoded and the BUFR files examined.
Ref: AD1 section 8.12.
Test case: ***ATMS_CrIS****, script* ***process_atms_cris_1c.sh***

2.3.6 ATMS beam width modification

The input is an ATMS l1c file. For the initial release of AAPP v7, simulated data (from test 2.3.5) shall be used; after launch of NPP the test should be repeated using real data. The *atms_beamwidth.dat* file shall be set up to provide an appropriate transformation to selected channels, e.g. to broaden the beam width of channels 3-15 and to narrow the beam width of channels channels 1-2 (as described in AD-4). The utility *atms_beamwidth* shall be run and the output displayed using a suitable tool (e.g. using IDL), to verify qualitatively that the image appears smoothed or enhanced.

The simulated ATMS data referenced in 2.3.5 does not have realistic instrument beam widths, therefore an IDL tool will be available within the test case to make these more realistic and (optionally) to add representative instrument noise. By comparing the *atms_beamwidth* output from a noisy input file with that from a noise-free input, the tester will be able to verify that the noise in the output file is in accordance with the theoretical predictions (AD-4).

Both the Fourier filtering and the sample averaging modes should be exercised (via the *atms_beamwidth.dat* file).

Ref: AD1 section 8.9.
Test case: ***ATMS_CrIS****, script* ***process_atms_cris_1c.sh*** *and* ***atms_noise.sh***

2.3.7 ATMS to CrIS pre-processing and mapping (atovpp)

The input is an ATMS 1c file and a CrIS 1c file (from test 2.3.5). Link these to atms.l1c and cris.l1c respectively, then run the following command:

```
atovpp -i "ATMS CRIS" -g CRIS
```

Inspect the log file atovpp.log. Some warning messages are to be expected (due to scan line numbering in the original BUFR file), but there should be no errors.

Using a suitable tool (e.g. IDL) to read the output file (cris.l1d), verify that it contains CrIS data with mapped ATMS.

The CrIS channel selection and thinning requirements are specified in CRIS.fdf. Verify that the output file correctly implements the specified channel selection and thinning. Also, run the test using the different available thinning options (i.e. no thinning, fixed detector, warmest FOV) and verify that these are correctly implemented.
Ref: AD1 section 8.11.
Test case: ***ATMS_CrIS****, script* ***process_atms_cris_1c.sh***

2.3.8 ATMS and CrIS ingest of HDF5 Sensor Data Record files

This tests the reading of ATMS and CrIS Sensor Data Record (SDR) data, in HDF5 format. Simulated data have been made available by NOAA and by EUMETSAT. The AAPP tools atms_sdr and cris_sdr shall be run to convert the data to AAPP l1c format. The contents of these should be examined using suitable IDL readers and verified against the contents of the HDF5 files (viewed with h5dump).
Ref: AD1 section 8.7.
Test case: ***ATMS_CrIS****, script* ***atms_cris_sdr.sh***

2.3.9 MAIA3 using GRIB API

The MAIA3 cloud mask utility is designed to operate with forecast files in GRIB format. For AAPP v7, calls to the obsolescent GRIBEX library (from ECMWF) have been replaced with calls to the GRIB API library (which handles GRIB2 format).

The MAIA3 routine, built with GRIB API, shall be run using a test case that includes GRIB2 forecast files. Reference cloud mask output shall be generated by Météo-France. The test shall run to completion and shall generate a cloud mask that is consistent with the reference output. Small differences are permitted due to compiler differences.
Ref: AD1 section 8.6.
Test case: MAIA_noaa19, scripts ***run_atovs_avhrr.sh*** *and* ***run_maia.sh***

The old MAIA3 test case (which did not include a forecast file) shall also be run and should give comparable output to AAPP v6.
Ref: AD1 section 8.5.
Test case: MAIA_noaa19, scripts ***run_atovs_avhrr.sh*** *and* ***run_maia.sh***

2.3.10 MWTS and MWHS ingest and pre-processing

Sensor Data Record files for the Microwave Temperature Sounder (MWTS) and Microwave Humidity Sounder (MWHS) on FengYun-3A are distributed in near real time on EUMETCast. They are in HDF5 format. AAPP v7 includes the facility to ingest these files, perform additional quality control, map MWHS to MWTS and store in AAPP format.

The routines, mwts_sdr, mwhs_sdr and mwhs_to_mwts shall be run, using samples of data from EUMETCast. The output product shall be examined visually (using an IDL utility).

Since this functionality is already in routine use the Met Office (using a locally-modified version of AAPP), the test shall be deemed to have been passed if the processing is reproducible on different platforms / compilers, and is consistent with the routine Met Office processing.
Ref: AD1 section 8.14
Test case: ***FY3A****, script* ***MWTS_MWHS.sh***

2.3.11 VIIRS processing

The facility to ingest VIIRS SDRs, generate a VIIRS cloud mask and map VIIRS to CrIS will be developed after NPP launch and integrated in a future update of AAPP v7. The testing to be preformed will be defined in a future update of this document.

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 v7 this will be achieved in several ways:

1. Through the activities of beta testers, discussed in 2.3. 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 to assess to what extent the package meets the needs of users.
2. In the particular case of NPP pre-processing, the tests of 2.3.6 and 2.3.7 provide radiance products that are representative of those that will be used for NWP assimilation. Beta testers should examine them in that light.
3. After the routine receipt of NPP data has commenced, further validation testing will take place in the context of routine operation at NWP centres. It is anticipated that this will result in AAPP updates which will be subject to the normal development procedures described in AD-2.
4. Much of the functionality of AAPP v7 is unchanged from that of AAPP v6, and in use operationally at NWP centres. For example, the ability to process data from the current generation of NOAA and MetOp satellites. Therefore these aspects are considered already validated, and the technical checks described in section 2.3 are sufficient.

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, with 32-bit and 64-bit platforms. Actual compilers will be listed in the Test Report. Portability will also be tested by the beta testers.

It is expected that the following platforms will be available for testing: Linux PC, IBM-AIX, Solaris. Some parts of AAPP could be tested on HP-UX and/or CygWin, if available.

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 v6 functionality, the testing should verify that there are no significant timing differences between v6 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 v7 or the NPP satellite. 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
8.1 Release note completeness	Inspection	N/A	
8.2 Building and linking	Test	2.1	Coding and compilation testing
8.3 Compile and run on specified platforms	Test	2.5	Portability testing
8.4 Direct readout test cases	Test	2.3.1; 2.3.2; 2.3.3	Integration testing
8.5 MetOp BUFR, MAIA3, IASIPC test cases	Test	2.3.4; 2.3.9	
8.6 MAIA3 with GRIB API	Test	2.3.9	
8.7 ATMS/CrIS SDR ingest	Test	2.3.8	
8.8 ATMS/CrIS BUFR ingest	Test	2.3.5	
8.9 ATMS spatial filtering	Test	2.3.6	
8.10 CrIS spectral filtering and thinning	Test	2.3.7	
8.11 Map ATMS to CrIS	Test	2.3.7	
8.12 ATMS/CrIS 1d BUFR encode	Test	2.3.5	Met Office requirement, optional elsewhere.

8.13 Test cases available	Inspection / test	2.4	Usability of the test cases assessed during validation testing
8.14 MWTS/MWHS ingest and pre-processing	Test	2.3.10	

4. TEST RESULTS

Test results will be included in an appendix to a later release of this document.