

# AAPP v8 Userguide | NWP SAF

ATOVS and AVHRR Pre-processing Package (AAPP)

Document: NWPSAF-MO-UD-036

Version: 8.5

Date: April 2020

*Author: Nigel Atkinson (Met Office)*

This documentation was developed within the context of the EUMETSAT Satellite Application Facility on Numerical Weather Prediction (NWP SAF), under the Cooperation Agreement dated 29 June 2011 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.

**Copyright 2016, EUMETSAT, All Rights Reserved.**

---

## Introduction

The ATOVS and AVHRR Pre-processing Package (AAPP) originated in the late 1990s, in response to the need for a processing package for the direct readout data from NOAA-15 and its successors. Since then, many extensions have been added, e.g. a major enhancement to accommodate Metop in 2006 (AAPP v6) and extension to Suomi-NPP in 2012 (AAPP v7). The package is maintained by the EUMETSAT Satellite Application Facility for Numerical Weather Prediction (NWP SAF). The current major release is v8.

This guide provides information on the capabilities of AAPP, with practical hints on its use that are relevant to both new and experienced users. It also addresses some of the questions that have been posed by users of the over the years. It is intended to be used in conjunction with the other documents which are available on the AAPP web page, including:

- - Installation Guide
  - Scientific Description
  - Software Description
  - AAPP overview
  - Data Formats
  - OPS-LRS User Manual
  - AAPP Top Level Design documents

## Summary of AAPP capabilities

AAPP consists of a large number of tools for different aspects of polar-orbiter satellite data processing. These tools are mainly concerned with level 0 and level 1 processing, i.e. to generate basic quantities such as brightness temperatures, on the original satellite swath. It has a capability to generate cloud mask and sea surface temperature, but does not perform atmospheric temperature/humidity retrievals and is not intended as a visualisation tool: resources for those purposes are available elsewhere (see [Links to external resources](#)).

AAPP tools are available for the following functions:

- Raw data to level 1 processing for HRPT from NOAA POES satellites.
- Level 0 to level 1 processing for AHRPT from Metop.
- Ingest of level 1B files for NOAA POES or Metop satellites and onward processing to level 1c (original instrument grid) or level 1d (one instrument mapped to another).
- Ingest of Scientific Data Record (SDR) files from Suomi NPP, JPSS and FY-3 satellites. The input files are in hdf5 format.
- Spatial and/or spectral filtering, and re-mapping.
- Cloud mask generation (MAIA), for AVHRR and VIIRS.
- BUFR encoding and decoding.
- Conversion of AAPP outputs to hdf5 format (from v7.8)

The instruments currently supported are:

- On NOAA POES: AMSU-A, AMSU-B, MHS, HIRS, AVHRR, MSU
- On Metop: AMSU-A, MHS, HIRS, IASI, AVHRR
- On Suomi-NPP and JPSS: ATMS, CrIS, VIIRS
- On FY-3: MWTS, MWHS, IRAS, HIRAS, MWRI

This Guide describes these capabilities, and also attempts to provide background information on what else you may need, such as raw-data processing tools and ancillary data.

In AAPP, the following conventions are used to describe the different processing levels of data:

- Raw: data as transmitted by the spacecraft
- Level 0: After initial steps such as frame synchronisation have been performed. In the case of Metop, instruments are separated into different files.
- Level 1a: Raw counts, with instruments in separate files.
- Level 1b: Geo-referenced and calibrated data (reversible: calibration coefficients are separated from raw data).
- Level 1c: Geo-referenced and calibrated brightness temperatures and albedo (non-reversible: calibration coefficients are applied to numerical data). In the case of IASI, the spectra are apodized. In the case of microwave sounders, an antenna pattern correction will usually have been applied.
- Level 1d: Mapped and filtered data. Several instruments may be mapped to a common instrument grid. A cloud mask, or other derived products, may be included.

Note that EUMETSAT use a slightly different convention for level 1b: in the EUMETSAT definition the non-reversible application of calibration information has already been done, i.e. the 1b contains brightness temperatures not raw counts.

## What's new in AAPP v8?

The main changes in AAPP version 8, compared with the earlier v7, are as follows:

- Replacement of the old versions of MAIA cloud mask. Previously MAIA2.1 was used for AVHRR on HIRS grid and MAIA3 for AVHRR native grid. Now MAIA4 will be used for both AVHRR and VIIRS. One advantage of that is that GFS forecast fields can be used (readily available on the internet). For details, see [MAIA cloud masks](#).
- Addition of interfaces to ecCodes, ECMWF's new BUFR/GRIB software library. At this stage the interfaces to the old BUFRDC and GRIB\_API packages are being retained, with the new software available as an option. For details, see [BUFR conversion tools](#).
- Support for JPSS-1 (NOAA-20).

Some obsolete features of AAPP (e.g. support for Spot messages) have been removed.

Regarding operating systems and compilers:

- The obsolete g77 compiler is no longer supported. Any g77 users should migrate to gfortran.
- HP-UX and Solaris are no longer supported (there are no known users)
- The NWPSAF no longer has access to an AIX test system, though we will endeavour to assist AIX users.
- Mac OS is now supported.

## Building AAPP and preparing to run

Detailed instructions for installing AAPP are provided in the [AAPP Installation Guide](#). AAPP is usually distributed in the form of source code which must be compiled by the user, though this approach may change in the future as some users have indicated a desire to be able to download binaries. Note that one or more external libraries may be needed to achieve the functionality that you need, for example [BUFRDC](#), [ecCodes](#) or [hdf5](#) libraries. The [GRIB-API](#) library has been superseded by ecCodes.

You can find an example build script *install\_aapp8.sh* in the [downloads directory](#), which you can download to your workstation and customise to suit your requirements. It includes configuration and build steps for the external libraries, for AAPP and for OPS-LRS. Type *./install\_aapp8.sh* to see usage instructions.

An example configuration for AAPP could be:

```
./configure --prefix=/software/nwpsaf/aapp_run_8.1/ --fortran-compiler=gfortran --station=exeter --site-
--external-libs="-L$DIR_BUFRDC/lib -lbufr -L${DIR_HDF5}/lib -lhdf5 -lhdf5_hl -lhdf5_fortran -lhdf5hl_f
--external-includes="-I$DIR_HDF5/include -I$DIR_ECCODES/include"
```

where DIR\_BUFRDC, DIR\_HDF5 and DIR\_ECCODES are the installation directories of the external libraries. The configure step would be followed by `make` and (optionally) `make install`. The AAPP Installation Guide gives more examples.

The IASI level 1 processor, OPS-LRS, is distributed as a separate sub-package – see the [OPS-LRS User Manual](#). Before building OPS-LRS you should first build AAPP and the external libraries `fftw-3.0.1` and `xerces 7.0`. The external libraries are available via the OPS-LRS auxiliary data link on the [AAPP Downloads](#) page. Note that xerces 7.0 is quite an old library, but later versions are not compatible with

OPS-LRS. OPS-LRS is built using the usual configure > make > make install process. An example configuration for OPS-LRS could be:

```
./configure --prefix=/software/nwpsaf/OPS-LRS-V7.0/ --aapp-prefix=/software/nwpsaf/aapp_run_8.1/ --xrcs-
--fftw-prefix=/software/nwpsaf/ --arch=Linux-gfortran --site-id=UKM --nthreads=4 --optimize=normal
```

You should specify an appropriate number of threads at the configure step, e.g. `--nthreads=4` or `8` depending on your computer resources. If you want to change it after building OPS, simply edit the `OPS_SD.cfg` file in the `OPS-LRS-run/OPS/conf` directory. Note that OPS-LRS runs 6 threads at start-up (WOM, MSGS, MP, JDBS, TES, SD\_FRW) and then  $N$  threads for processing the IASI data. Usually  $N$  is 1 or a multiple of 4 (as IASI has 4 fields of view).

Depending on which components of AAPP you wish to run, you may also need to download some data files from the AAPP ftp site, e.g. data files for MAIA, and auxiliary files for the IASI instrument (the “BRD”, “GRD”, “ODB” files used by OPS-LRS). These are available from the [AAPP Downloads](#) page, together with a script that simplifies the downloading of the IASI files.

It is also recommended to run some of the test cases available on the [AAPP Downloads](#) page.

According to a 2015 survey, the following computer platforms were in use to run AAPP: (i) Linux, (ii) IBM AIX, (iii) Mac OS, (iv) windows running a unix emulator like Cygwin. In the past, HPUX and Solaris were also supported but there are now no known users for these platforms. Commonly used Fortran compilers include (most popular first): (i) gfortran, (ii) g95, (iii) g77 (only suitable for the older parts of AAPP), (iv) ifort, (v) xlf, (vi) pgf.

Before running any of the commands listed in the following sections, you need to set up your environment variables using the `ATOVS_ENV8` file which is located in the top directory:

```
AAPP_PREFIX=name_of_top_directory
. $AAPP_PREFIX/ATOVS_CONF
```

This will automatically add the necessary AAPP directories to your `$PATH`, so that you can run them from any directory that you choose.

You may also need to set `LD_LIBRARY_PATH`, if you have installed external libraries locally (e.g. hdf5 or ecCodes).

## Orbital elements and the satpos file

Orbital elements are required for the navigation process. They are bulletins that provide the satellite position and velocity at a given time, and hence allow the software to predict future positions. Usually, 2-line elements (often abbreviated to TLE) are used. Some users prefer [TBUS](#) bulletins, but these are not available for all satellites and are not described in this section. You don't need to know the TLE format in order to use them, but for reference it is described on the [Celestrak web site](#). There are various sources on the internet for these bulletins, including:

<http://oiswww.eumetsat.org/metopTLEs/html/index.htm> Metop-A/B, NOAA-15/18/19, Suomi-NPP. The Metop bulletins are generated by EUMETSAT directly, the NOAA bulletins are generated from 4-line elements.

<https://is.sci.gsfc.nasa.gov/ancillary/ephemeris/tle/drl.tle> Aqua, Terra, Suomi-NPP, NOAA-20. Generated by NASA.

<https://www.space-track.org/> Requires registration. Generated by NORAD for many satellites using satellite tracking techniques.

<http://celestrak.com/NORAD/elements/> File “weather.txt” contains the satellites of interest. Same data as space-track.

[http://www.shinetek.com.cn/eos\\_data/](http://www.shinetek.com.cn/eos_data/) Also includes the “1line” elements that are needed for FY-3 level 1 processing.

<https://satellite.nsmc.org.cn/portalsite/default.aspx> FY-3 only.

Alternatively, you can extract TLE data for NOAA and Metop satellites from the Metop Multi-Mission Administration Message (MMAM). This can be obtained from the direct broadcast HKTm file, or they can be viewed from the EUMETSAT web site [here](#). Instructions for extracting TLE bulletins from the HKTm level 0 files are given in the [AAPP v7.4 Release Note](#).

For the web-based bulletins, we recommend using the EUMETSAT and NASA TLEs (the first 2 sources in the list), as the bulletins take account of satellite manoeuvres. With bulletins from space-track and celestrak, note that sometimes (rarely) the bulletins are not updated for several days, and this can cause

problems for satellite tracking. There are also occasional issues with inconsistency in the orbit numbers.

For historical bulletins it is best to use the space-track web interface. To process HRPT data for a given day you must have a TLE bulletin that is valid before the start of that day.

There is an AAPP tool available to download current bulletins – called `get_tle`. It uses the environment variables `PAR_NAVIGATION_TLE_URL_DOWNLOAD` and optionally `PAR_NAVIGATION_TLE_USER`, `PAR_NAVIGATION_TLE_PASSWD` that are defined in your `ATOVS_ENV8` file. In the AAPP structure, files are normally placed in monthly subdirectories of `$DIR_NAVIGATION/tle_db` (where `$DIR_NAVIGATION` defaults to `$AAPP_PREFIX/AAPP/orbelems`). You can either use the `get_tle` tool or you can retrieve the files yourself. The files must have names of the form `tle_yyyymmdd_hhmm.txt`, where `yyymmdd_hhmm` is the date/time of your download.

For space-track.org retrieval, you can specify which satellites you are interested in via the environment variable `PAR_NAVIGATION_TLE_CATALOGUE` – a list of NORAD catalogue numbers. It defaults to a list that includes NOAA, Metop, NPP, FY3 and Terra/Aqua satellites (see the `get_tle` script)

When AAPP ingests a TLE file it updates the various “index” files, one for each satellite of interest, e.g. `tle_noaa19.index`. The TLE index file is updated whenever you run the top level script `AAPP_RUN_NOAA` or `AAPP_RUN_METOP` (see section 5). The software only looks for TLE files that are newer than the index file, so to force older files to be ingested you have to use the unix touch command to change the time stamp of the bulletins. If you want to ingest TLE files by hand you can do it like this (after setting up your AAPP environment):

```
tleing -s noaa19
```

The next step (also performed automatically by `AAPP_RUN_NOAA` or `AAPP_RUN_METOP`) is to generate a “satpos” file for a particular time period. This is done by program `satpostle`. You can run it by hand, if needed, like this:

```
satpostle -o -s noaa19 -n 1.2
```

The above example generates a satpos file for today, containing 1.2 days, with position/velocity information every 2 minutes. There are various options if you want to use, for example, a different time period. The result is stored in directory `$DIR_NAVIGATION/satpos`. If you suspect a problem with a satpos file (e.g. you get an error when reading the file), simply delete it and re-generate it.

You should make sure that the lists of satellites (`PAR_NAVIGATION_DEFAULT_LISTESAT`) and bulletins (`PAR_NAVIGATION_DEFAULT_LISTEBUL`) in your `ATOVS_ENV7` file are appropriate.

## Processing direct broadcast data

### NOAA satellites

AAPP assumes that the input HRPT file has been unpacked into 16 bit words (only the least significant 10 bits are used), with 11090 words per “minor frame”, and 6 minor frames per second. For details, see section 4.1 of the [NOAA KLM User Guide](#). AAPP doesn’t care whether the data are in big-endian (most significant byte first) or little-endian (least significant byte first) format, it can handle both.

However, not all reception systems deliver this unpacked format. Some deliver a packed format, which is in effect a continuous bit stream of 10-bit words, packed into 16-bit words. The complication here is that 11090 10-bit words equates to 13862.5 bytes, so fill bits are often inserted. Different stations have different ways of doing this. If you know the format of your reception station then there is an AAPP tool `unpack_noaa_hrpt` that can unpack it. But you may need to modify the source file `unpack_noaa_hrpt.F`, in particular the parameters `bytes_in` and `words_out` (normally set to 27740 and 22180 respectively, to suit Met Office raw files). If you don’t know the format of your raw files then please contact the [NWP SAF Helpdesk](#) and we can help either by providing an analysis tool or by examining a sample of your data.

The main processing is done by script `AAPP_RUN_NOAA`. You need to make sure that the environment variable `$WRK` points to a valid working directory. Then run the command like this:

```
AAPP_RUN_NOAA -i "$instruments" -g "$grids" hrptfile
```

e.g. if `instruments="AMSU-A MHS HIRS AVHRR"` and `grids="HIRS"` then the output will include a HIRS level 1d file (with mapped AMSU, MHS and AVHRR), plus the various level 1b and 1c files. If you don’t need level 1d then just set `grids=" "` (a string containing a blank character). If you need AVHRR 1b but not mapped to HIRS add the “-z” flag.

If you want the MAIA cloud mask on the full-resolution AVHRR grid, you will have to run MAIA separately: see [MAIA cloud masks](#).

See also Figure 1 of the [AAPP version 7 Top Level Design](#).

## Metop satellites

The input to AAPP is “EPS level 0”. Some reception systems deliver this format. But if you only have CADU data (or other raw data) you will have to convert to level 0 yourself. EUMETSAT have made available the *Metopizer* tool which can do this conversion. It is available on their [Software](#) page. The Metopizer tools that you are most likely to need to use are `cadu_to_ccsds` and `ccsds_to_10`. Note that Metopizer version 3.51 has a new version of `ccsds_to_10` that correctly handles the Metop Multi-Mission Administration Message (MMAM) and generates EPS level 0 files that can be input directly to AAPP: see the AAPP test case *metopizer\_and\_AAPP*. With earlier versions of the Metopizer, additional steps were needed (see next paragraph).

The EPS level 0 file is supposed to contain a VIADR containing, among other things, information to convert satellite time to UTC time. This is needed for accurate navigation. Sometimes the VIADR is either missing or contains the wrong information. In this case, AAPP has a tool `patch-level0-from-mmam.exe` to correct it. To run the tool, either (i) add HKTm to the list of instruments when you call `AAPP_RUN_METOP`, or (ii) follow the instructions in section 4 of the [AAPP v7.4 Release Note](#).

If you wish to process IASI, you will need to install the OPS-LRS software (as discussed earlier). Then before running the main AAPP script you will need to set the environment variables `DIR_IASICONFIG` (to point to the auxiliary files) and `PATH_OPS` (to point to the OPS-LRS-run/OPS/perl directory containing the `ops_process` script).

The main processing is done by the script `AAPP_RUN_METOP`. To process a set of EPS Lo files in the current directory, the command is:

```
AAPP_RUN_METOP -i "$instruments" -g "$grids"
```

where *instruments* can be any of AMSU-A, MHS, HIRS, AVHRR, IASI. You can specify different input and output directories if preferred. The Lo files must have file names of the form `*_??_oo_M*Z`, e.g. `AMSA_00_00_M02_20020808181206Z_20020808195406Z_N_O_20020808201206Z`. Remember that M02 is Metop-A, M01 is Metop-B and M03 is Metop-C. There is no HIRS on Metop-C.

## IASI considerations

If IASI is specified in the list of instruments then AVHRR must be included also. First AVHRR is processed to level 1b. Then OPS-LRS is invoked in dump mode using `ops_process`. The output from OPS-LRS is a IASI `l1c` file in EPS format, but the `AAPP_RUN_METOP` script converts this to an AAPP-style “`l1c`” format. If you want to access the EPS format file, it is found in the `$WRK` directory in a hidden directory `M01 (or M02)/.keep/*/to_pgf`.

To run OPS-LRS in granule mode you would have to use lower-level commands – and would need the input Lo files also to be in granule form. The NWPSAF can advise if needed.

After completion of OPS-LRS, if “`$grids`” includes IASI then `atovpp` will be run. A key data files is `IASI.fdf`, which contains the channel selection to be used. Also, if Principal Components are requested in the `IASI.fdf` then you will need the appropriate eigenvector files; these are specified in file `iasi_eigenvectors_spec.dat` which by default is in `$DIR_PREPROC` (i.e. `$AAPP_PREFIX/AAPP/data/preproc`), but can be elsewhere if you define an environment variable `DIR_IASI_PREPROC`.

See also Figure 3 of the [AAPP version 7 Top Level Design](#).

## Suomi-NPP and JPSS satellites

To process raw S-NPP/JPSS direct readout data you will need to run external programs, as follows:

- ○ ■ [RT-STPS](#) (from NASA) to convert from raw data to Raw Data Record (RDR, with instruments in separate files).
- [CSPP](#) (from University of Wisconsin) or [IPOP](#) (NASA) to perform the processing to Sensor Data Record (SDR).

You may also need to run [nagg](#) (part of the `hdf5` tools) to manipulate granules. ATMS and CrIS granules are normally 32 seconds in duration, so a typical pass generates many granules.

The AAPP tools `atms_sdr` and `cris_sdr` can ingest the SDRs for ATMS and CrIS, generating AAPP-style “`l1c`” files. You can then run:

- `atovpp` – to map ATMS to CrIS, i.e. create CrIS level 1d; or

- `atms_beamwidth` if you just want to do beamwidth manipulation on ATMS without the re-mapping
- `cris_channels` if you want to do a channel selection for CrIS level 1c without re-mapping ATMS.

Key files that you may need to change are `atms_beamwidth.dat` and `CRIS.fdf`, by default located in `$DIR_PREPROC`.

Note that ATMS level 1d by default comprises antenna-pattern-corrected brightness temperatures. This is discussed further in [Antenna pattern correction](#).

For more information, please see the document [Appendix to scientific description – Pre-processing of ATMS and CrIS](#).

## FY-3 satellites

Level 0 and Level 1 processors are provided by CMA, see <http://satellite.cma.gov.cn/portalsite/default.aspx> -> Tools -> Softwares. Currently these tools support MWTS, MWHS, IRAS, HIRAS, MWRI, VIRR and MERSI.

After running the CMA software packages `fy3l0db` and `fy3l1db` (or `fy3c0db`, `fy3c1db` in the case of FY-3C), the AAPP tools `iras_sdr`, `mwhs2_sdr`, `mwhs_sdr`, `mwri_sdr`, `mwts2_sdr`, `mwts_sdr`, `hiras_sdr` can be used to convert the hdf5 files into AAPP-style binary files.

You can then run tools such as `mwhs2_beamwidth`, `mwhs_beamwidth`, `mwts2_beamwidth` if you wish to do spatial filtering. Also, there are re-mapping tools `mwhs2_to_iras`, `mwhs_to_mwts`, `mwts2_to_iras`, `mwts2_to_mwts2`. Key files that you may need to change are `mwhs_beamwidth.dat`, `mwhs2_beamwidth.dat`, `mwts_beamwidth.dat`, `mwts2_beamwidth.dat`. Note that the MWTS and MWHS “.11c” files have slots for mapped instruments, so there is no “.11d” format needed.

## Processing level 1 data from satellite agencies

### NOAA 1b

AAPP is commonly used to process level 1b sounder data from NOAA CLASS archive, or level 1b data disseminated by NOAA in near real time. See Figure 2 of the [AAPP version 7 Top Level Design](#). Since NOAA-15, the NOAA 1b format and AAPP 1b formats for sounder instruments have been identical. You can run `atovin` and `atovpp` individually. Input files can be specified by name or you can use the defaults which are `aman.11b`, `ambn.11b`, `hrsn.11b`.

In the case of AVHRR, and the older NOAA satellites, NOAA level 1b is different from AAPP level 1b. In AAPP v7.6, the tool `noaa_class_to_aapp` was introduced to convert from NOAA format to AAPP format, and the tool `avhrr_aapp_to_class` to convert from AAPP format to NOAA format. See the [AAPP v7.6 Release Note](#). In fact, `noaa_class_to_aapp` calls one of several different executables depending on the instrument that it finds in the 1b header record.

Also, bear in mind that for AVHRR the NOAA 1b files can have 8-bit, 10-bit or 16-bit words for the instrument counts. In the AAPP 1b, these are always 16-bit words. For post-NOAA-15 satellites the differences are described further in the AAPP Data Formats document.

An earlier AAPP tool, `hrpt1b_noaa`, was used to do a partial conversion of NOAA 1b to AAPP 1b for AVHRR, but this program is superseded by `noaa_class_to_aapp`. If you find a case that is not correctly handled, please notify the [NWPSAF Helpdesk](#).

See also the NOAA “POD User Guide” and “KLM User’s Guide”, available at <https://www1.ncdc.noaa.gov/pub/data/satellite/publications/podguides/>

### EPS level 1b/1c

The EUMETSAT archive can be used to extract AMSU, HIRS, MHS, IASI and AVHRR files in native EPS format. However whilst AAPP can handle IASI and AVHRR, there are no AAPP tools to handle AMSU, HIRS or MHS in EPS format. The user is recommended to specify BUFR output for these instruments (see [BUFR conversion tools](#)).

To convert IASI 11c from EPS format to AAPP format, the tool is `convert_iasilc` (part of the “`iasi-tools`”). For AVHRR, the command is `convert_avh1b` (in “`metop-tools`”). The AVHRR tool does not perform a full conversion back to raw counts; it generates scaled radiances instead.

### Suomi-NPP, JPSS and FY-3

For these missions the format used by the agency (NOAA or CMA) for SDR files is hdf5, and they can be converted to AAPP-style level 1b files using the previously-mentioned tools `atms_sdr`, `cris_sdr`, etc. Data

are also commonly distributed in BUFR, see [BUFR conversion tools](#).

## Antenna pattern correction for microwave sounders

Microwave sounder brightness temperatures are normally corrected for the fact that antenna sidelobes are sensitive to radiation originating from outside the earth's disk (i.e. from cold space or the satellite itself). The uncorrected temperatures are known as “antenna temperatures”.

For AMSU and MHS, this is handled in AAPP by the “efficiency factors” provided in data file `fdf.dat`. For each channel, and each field of view, three numbers are provided: the fraction of the signal that originates from earth, the fraction from the spacecraft and the fraction from space. To simplify the calculation, it is usually assumed that the spacecraft is at the same temperature as the earth.

The efficiency factors are computed from pre-launch measurements of antenna pattern, by the agency responsible for supplying the instrument. However, in some cases there are different versions of the correction available in AAPP, e.g. a version 1 that was available at the time of launch or a version 2 that was issued later. The version number is stored in the level 1c header (e.g. `ama1c_h_vnantennacorr`).

The antenna correction is applied by AAPP routine `atovin`, called automatically by `AAPP_RUN_NOAA` or `AAPP_RUN_METOP`.

The default is to use the first set of antenna corrections for each satellite. However, it is possible for the user to specify that a particular version of the antenna correction be used for a particular satellite, via environment variables, e.g.

```
export FDFSAT="METO-02 NOAA-18"
export FDFVN="2 1"
```

will use the second set for Metop-A and the first set for NOAA-18. In fact, this configuration is recommended for DBNet/RARS applications – to maintain consistency with global data generated by EUMETSAT. Be careful with the satellite names, they must be in the above format (7 characters).

If you are processing historic data, please note that there are some errors in the AMSU-B/MHS sections of the `fdf.dat` file. For NOAA-15, -16, -17 and -18, channel 19 incorrectly contains the channel 16 corrections and channel 20 incorrectly contains the channel 17 correction. This is a bug dating right back to AAPP v1.0 in 1998. Thanks to Hans et al. (Remote Sens. 2019, 11, 548) for pointing this out. Note also that NOAA-18 MHS contains the NOAA-15 AMSU-B correction, for all channels. These instruments are no longer operating.

AAPP also provides a tool to allow you to apply or undo an antenna correction:

```
atovin_antcorr [-f "files"] [-z] [instrument1] [instrument2]
```

where the `-z` flag is used to undo a correction. The `instrument1/2` can be any of AMSU-A, AMSU-B, MHS. The header variable `ama1c_h_vnantennacorr` or `amb1c_h_vnantennacorr` is used to determine which version of the correction to undo. If the version number is set to zero then it is assumed that no correction is applied.

In the case of ATMS, the AAPP level 1c format includes both antenna temperatures and brightness temperatures. (The former are taken from the Temperature Data Record, TDR, while the latter are from the Sensor Data Record, SDR). ATMS level 1d only has one set of temperatures, and these are by default *brightness temperatures*. Different centres may have different conventions: if you need your local data to use antenna temperatures (which was the practice at the Met Office prior to 2019), you will need to modify the Fortran program `atms_beamwidth.F`: change `atms1c_btemps` to `atms1c_anttemps` the first time it appears (not the second time). Then recompile that program.

## BUFR conversion tools

The **B**inary **U**niversal **F**orm for the **R**epresentation of meteorological data (**BUFR**) is a binary data format maintained by the World Meteorological Organization (WMO). It is a table driven code. The latest version of the tables can be found [here](#), in particular note the “BUFR Table B” (descriptors), “BUFR Table D” (sequences), “Code and Flag Tables”, and the “Common Code Tables”.

AAPP has interfaces with the ECMWF [BUFRDC](#) software package. AAPP v8 also has interfaces to [ecCodes](#) and it is anticipated that in the future there will be a migration from BUFRDC to ecCodes.

Supported instruments and AAPP formats are shown in the following table.

Instrument	AAPP format	Table D sequence	Data on GTS?	Notes
AMSU-A	1c	310009	Yes	Used for global and regional (DBNet) exchange.
AMSU-B/MHS	1c	310010	Yes	Used for global and regional (DBNet) exchange.
HIRS	1c	310008	Yes	Used for global and regional (DBNet) exchange.
HIRS	1d	–	No	Met Office sequence; uses some local descriptors. Includes mapped AMSU/MHS.
IASI	1c	340001	No	In use prior to 2010. Fixed set of 8700 channels.
IASI	1c	340007	Yes	Supersedes 340001. Includes additional quality information. See note below on the ENHANCED_IASI environment variable.
PCIASI	1c	340008	Yes	Used by DBNet. Includes channel subset and/or PC scores. Number of channels/PCs set by delayed replication.
IASI	1d	–	No	Met Office sequence. Includes mapped AMSU/MHS.
ATMS	1c	310061	Yes	
ATMS	1d	–	No	Met Office sequence.
CRIS	1c	310060	Yes	Number of channels set by delayed replication. Now deprecated: superseded by the CRISFSR sequence
CRISFSR	1c	–	Yes	As for CrIS 1c, but with provision for extra guard channels and a VIIRS cluster analysis. AAPP support using ecCodes only, not BUFRDC.
CRIS	1d	–	No	Met Office sequence. Includes mapped ATMS.
MWTS/MWTS2 /MWHS/MWHS2 /IRAS/MWRI	1c	–	No	Legacy sequence originally devised by ECMWF in conjunction with CMA; superseded by the sequence below for all except MWRI.
MWTS2/MWHS2/IRAS	1c	310070/310071 /310072	Yes	New sequences devised by CMA in 2017. To create data using these sequences using AAPP, set the environment variable USE_VASS_SEQUENCE=Y before running <i>eccodes_encodebuf_1c</i> . Not available with BUFRDC.
MWTS2/MWHS2/IRAS	1d	–	No	Met Office sequence. Flexible with up to 2 mapped instruments.

BUFR Section 3 (the data description section) contains either the Table D sequence given in the table, or where there is no sequence number it contains the list of Table B descriptors.

### BUFR encode/decode using BUFRDC

To run the AAPP BUFR tools, using the BUFRDC library, you need to do the following:

- ○ Ensure that AAPP has been built correctly, linked to BUFRDC.
- ○ Define and export the environment variable BUFR\_TABLES – pointing to the directory



containing the BUFR tables (with trailing /). Usually it is recommended to use the tables provided with BUFRDC.

- For encoding, you may also need to define some other environment variables, such as

ORIGINATING_CENTRE	e.g. 74=Met Office; defaults to 255=missing
SUB_CENTRE	defaults to 0. Important for DBNet (RARS) stations.
CENTRE_ID	centre code for use in Section 4, where applicable; defaults to 255.
MASTER_TABLE	The version of the BUFR tables to use.
LOCAL_TABLE	Defaults to 0
BUFR_EDITION	Defaults to 4
MESSAGE_SUBTYPE	Local subtype, can be anything you like but if not specified some defaults are provided
ENHANCED_IASI	Should be set to "Y" if you want to use 340007 sequence.
USE_OBS_TIME	Set to Y if you want to put the observation time in Section 1 (recommended). Otherwise
ATMS_THIN	e.g. set to 3 if you want to only encode 1 spot in 3 in the 1d sequence.
MWTS2_THIN	As above
MWHS2_THIN	As above
IRAS_THIN	As above

The encoder is invoked like this:

```
aapp_encodebufr_1c -i "${infile}" "${instruments}"
```

where “\${instruments}” is a list of instruments (as in the above table), one for each input file, e.g. “AMSU-A MHS HIRS”. The output is one or more files with suffix “.bufr”.

The decoder is invoked like this:

```
aapp_decodebufr_1c -i "${bufrfiles}" [${instruments}]
```

In most cases the list of instruments is optional, but is required if the instrument cannot be identified from the BUFR sequence number, e.g. for MWTS, MWTS2, MWHS, MWHS2, IRAS. The output is one or more files with suffix “.11c”. Decoders have not been implemented for level 1d.

It is worth noting that NCEP use some different sequences to those listed in the table. For CrIS there is a local sequence number of 310193 but the actual sequence is the same as 310060. For IASI they use a sequence 361207 (“NC21241”) which is similar to 340001 but using delayed replication to specify the channel subset; this is not currently supported by AAPP.

## BUFR encode/decode using ecCodes

If you are using ecCodes, there are a few differences:

- The AAPP encode/decode programs are named *eccodes\_encodebufr\_1c* and *eccodes\_decodebufr\_1c*
- Although LOCAL\_TABLE defaults to zero on encode, ecCodes is more tolerant if LOCAL\_TABLE is non-zero. If a local table is not found, it will attempt to use the standard tables.
- There is no BUFR\_TABLES environment variable. BUFR tables are automatically located in the ecCodes directory structure, and this is sufficient for datasets that use the standard tables (i.e. all the level 1c datasets).
- Some of the level 1d datasets require local tables, which are supplied with AAPP and can be accessed using an environment variable ECCODES\_DEFINITION\_PATH. In order to work with these level 1d datasets, you can define (e.g. in ATOVS\_ENV8):

```
export ECCODES_DEFINITION_PATH=$AAPP_PREFIX/AAPP/data/tools_eccodes/definitions:$DIR_ECCODES/share/
```

where the first part points to the local definitions and the second part points to the definitions that are supplied with ecCodes. For example, AAPP subdirectory definitions/bufr/tables/0/local/1/74/0 would contain tables for centre 74 (Met Office) local table 1. It is only necessary to include non-standard definitions in these local tables.

- A new BUFR sequence for CrIS Full-Spectral-Resolution data (devised by NOAA) is implemented in the ecCodes interface.
- Some of the level 1d encoders are not yet implemented in the ecCodes interface.

Note that AAPP can be built with both BUFRDC and ecCodes linked in – allowing either interface to be used. However, AAPP should *not* be built with both GRIB\_API and ecCodes.

For CrIS and IASI, the default is to encode 1 message per scan. In AAPP v7, this was set to 6 messages per scan in order to reduce message lengths, but the use of 1 message per scan has several advantages: (i) runs faster, (ii) produces a better-compressed output, (iii) the number of bulletins per pass (for direct broadcast) is less than 100, as recommended in the GTS manual. The default can be over-ridden via environment variables MESSAGES\_PER\_SCAN\_IASI, MESSAGES\_PER\_SCAN\_PCIAI, MESSAGES\_PER\_SCAN\_CRIS, MESSAGES\_PER\_SCAN\_CRISFSR. For other instruments, one or more scans are encoded into a message; the settings are hard-coded (e.g. 5 scans for AMSU-A, MHS and HIRS; 3 scans for ATMS).

## MAIA cloud masks

In AAPP v7, three versions of MAIA (Mask AVHRR for Inversion ATOVS) were included in AAPP:

- MAIA v2.1, AVHRR cloud mask on the HIRS grid
- MAIA v3, AVHRR cloud mask on the full-resolution AVHRR grid
- MAIA4, VIIRS cloud mask

In AAPP v8 they are merged into an updated MAIA4.

An NWP model background field is needed to run MAIA4. In the past, users have had to rely on ECMWF or Météo-France model fields but as of AAPP v7.9 (Feb 2015) it has been possible to use GFS model fields for MAIA4 – these are freely available on the Internet and are downloaded automatically when needed.

In order to use GFS forecast files, you will have to have an environment variable:

```
MAIA4_USE_GFS="yes"
```

It is possible to define also:

```
MAIA4_REMOTE_GFS_DIR=url
```

By default the url is set to [http://jpssdb.ssec.wisc.edu/cspp\\_v\\_2\\_o/ancillary/](http://jpssdb.ssec.wisc.edu/cspp_v_2_o/ancillary/) but can be defined to <http://jpssdb.ssec.wisc.edu/ancillary/> to process old data.

You can specify the directory in which forecast files are stored via the environment variable:

```
DIR_FORECAST
```

The method of generating each cloud mask is shown in the following table:

	<b>HIRS grid</b>	<b>AVHRR grid</b>	<b>VIIRS</b>
<b>Main script</b>	AAPP_RUN_NOAA / AAPP_RUN_METOP	MAIA4_RUN_AVHRR	MAIA4_RUN
<b>Sub-script</b>	avh2hirs	maia4_Avhrr	maia4
<b>Input</b>	HIRS 11d + AVHRR 11b	AVHRR11b or PFS11b or Lo	VIIRS SDR files and GMODO/GMTCO files
<b>Output</b>	HIRS 11d	avhCT file (hdf5)	viiCT file (hdf5)
<b>Specific environment variables</b>	RUN_MAIA (yes/no)	–	–

Note that for the AVHRR to HIRS mapping, running the full MAIA4 is significantly slower than the old MAIA2.1. Therefore an option (RUN\_MAIA=no) has been provided to just map the AVHRR radiances to HIRS grid – i.e. the user gets the mean and standard deviation of the spots that fall within the HIRS footprint, for each AVHRR channel. This is the default.

For further information on MAIA, please see the AAPP [Software description](#) and the [MAIA version 4 Scientific User Manual](#).

## Conversion of AAPP binary formats to HDF5

The tool `convert_to_hdf5` was introduced in AAPP v7.8 (as a result of an action from the ITSC-19 conference). It converts AAPP level 1c/1d files to hdf5 format, and can also be used for AVHRR level 1b. The hdf5 output has a number of top-level attributes, which in general correspond to the items in the 1b/1c/1d header record. In some cases the names or units are changed but these are intended to be self-explanatory. There are also groups like “Data”, “Geolocation”, containing the main datasets, with attributes to describe them. You can use a tool such as “h5dump” to examine the output file.

You will find that there are strong similarities between the resulting hdf5 products, for different instruments. This should make it easier to ingest AAPP outputs into other applications, without having to know the details of the AAPP 11c format.

The hdf5 format is more flexible than the traditional AAPP 11c format. For example, an option has been introduced in `convert_to_hdf5` to output just a limited geographical subset in the AVHRR 11c.h5 product: see the [AAPP v7.13 Release Note](#).

## Special considerations for DBNet contributing stations

The Direct Broadcast Network ([DBNet](#)) is the programme run by WMO aimed at distributing direct broadcast sounder data to NWP centres worldwide, with timeliness suitable for regional NWP models. The ATOVS component is known as RARS (Regional ATOVS Retransmission Service). In recent years the concept has been extended to cover IASI, ATMS, CrIS and the FY-3 sounder system known as VASS (=MWTS+MWHS+IRAS). AAPP is used to a greater or lesser degree in all the processing chains, but not all stations will be contributing to all services.

The table below shows which parts of AAPP and linked libraries are needed for each service. It also shows the requirements for external software. Please note that in most cases “BUFR” can be either BUFRDC or ecCodes, but we recommend that centres start to use ecCodes. (Note that ecCodes is mandatory if you are processing NOAA-20 CrIS).

Satellite/instrument	AAPP build requirements	AAPP configuration	External requirements
NOAA ATOVS	BUFR	–	–
Metop ATOVS	BUFR	–	Raw to EPS level 0 (e.g. Metopizer)
Metop IASI	OPS-LRS + BUFR	500 IASI channels + 300 PCs (config file IASI.fdf)	Raw to EPS level 0 (e.g. Metopizer)
ATMS/CrIS	hdf5 (C interface) + BUFR	As of April 2020: 431 channels for both SNPP and NOAA-20 (CRIS_FSR.fdf)	RT-STPS + CSPP SDR software (or IPOPP)
FY-3 sounders	hdf5 (C interface) + BUFR	–	fy3clodb and fy3cl1db (RT-STPS could be an alternative to fy3clodb)

The processing steps are summarised below. We give an outline of the processing; the station operator will need to fill in details.

## Steps common to all data

Set up environment variables BUFR\_TABLES (directory – BUFRDC only), ORIGINATING\_CENTRE (Common Code Table C1), SUB\_CENTRE (indicates the station). For MASTER\_TABLE, AAPP will provide defaults, but you can set it if required. It is recommended to set LOCAL\_TABLE=0. Also, you should set USE\_OB\_TIME=Y (BUFRDC interface only). Remember to export all the environment variables.

A script is needed to detect incoming files, run the various stages of processing and disseminate products. The implementation approach will vary from station to station.

In late 2017, DBNet centres were asked to add certain flags to the calls to `aapp_encodebufr_1c`: “-N” for ATOVS (to encode NEDT) and “-v none” for other instruments (to suppress encoding of software version numbers). For AAPP v8, the defaults in this script have been changed to reflect current operational practice, and the flags are not needed, though it does no harm to include them.

The AAPP script `prepare_dbnet_bufr_for_gts` may be used to prepare data for transmission on the GTS. It automatically generates a suitable output file name, that is consistent with the contents (e.g. centre, sub-centre) and adds bulletin headers.

## NOAA ATOVS

```
AAPP_RUN_NOAA -i "AMSU-A MHS HIRS" -g " " hrptfile
eccodes_encodebufr_1c -i "amsual1c*.11c mhs11c*.11c hirs11c*.11c" AMSU-A MHS HIRS
```

(or you can call the legacy `aapp_encodebufr_1c` if you are not yet ready to use ecCodes). Rename the \*.bufr according to DBNet convention. Add GTS headers using the script `gtsheaders_bufr.pl` (now included in AAPP). Disseminate. Here is an example showing the use of `gtsheaders_bufr.pl`, for AMSU-A data from centre KWBC (Washington):

```
gtsheaders_bufr.pl $amsua_in $amsua_out INAX KWBC
```

The different instruments have different letter codes (AMSU-A=A, MHS=M, HIRS=H, IASI=Q, ATMS=S, CrIS=C, MWHS-2=K, MWTS-2=T, IRAS=I).

## Metop ATOVS

Generate the LO files and put them in a working directory.

If necessary, modify the OBT-UTC parameters as discussed in [Processing direct broadcast data](#).

Set environment variables for Metop-A: export FDFSAT=METO-O2; export FDFVN=2

```
AAPP_RUN_METOP -i "AMSU-A MHS HIRS" -g " "
```

```
eccodes_encodebufr_1c -i "amsu11c*.11c mhs11c*.11c hirs11c*.11c" AMSU-A MHS HIRS
```

(or you can call the legacy `aapp_encodebufr_1c` if you are not yet ready to use `ecCodes`). Rename the `*.bufr` according to DBNet convention. Add GTS headers (as described above). Disseminate.

## Metop IASI

Generate the IASI and AVHRR LO files and put them in a working directory.

If necessary, modify the OBT-UTC parameters as discussed in [Processing direct broadcast data](#).

```
AAPP_RUN_METOP -i "AVHRR IASI" -g " "
```

```
iasi_1c_to_pc iasifile.11c iasifile.lpc
```

```
eccodes_encodebufr_1c -i iasifile.lpc PCIASI
```

Rename the `iasifile.bufr` according to DBNet convention. Add GTS headers. Disseminate.

The eigenvector file supplied with AAPP (the so-called “version 104”) can be used for Metop-A, B or C. If in the future a different eigenvector file is needed for different satellites then you can create a dedicated directory for each satellite, and then use environment variable `$DIR_IASI_PREPROC` to point to the appropriate directory. But that is not necessary at present.

## ATMS/CrIS

Run RT-STPS (script `[RT-STPS-home]/rt-stps/bin/batch.sh`) to generate RDR files for ATMS and CrIS.

Run CSPP scripts `atms_sdr.sh` and `cris_sdr.sh` – preferably in parallel in order to optimise the timeliness. These generate SDR granules. DBNet convention is to generate full-spectral-resolution CrIS SDR files for S-NPP and NOAA-20 (use the “-f” flag). Prior to April 2020 S-NPP used normal-spectral-resolution.

For ATMS, loop over granules, (with time stamp `${datestr}`), and run AAPP tools:

```
sdr=$(ls SATMS_npp_${datestr}*.h5)
```

```
tdr=$(ls TATMS_npp_${datestr}*.h5)
```

```
atms_sdr $sdr $tdr
```

After the granule loop:

```
combine_1c SATMS*.11c          #generates atms.11c
```

```
eccodes_encodebufr_1c -i atms.11c ATMS      #generates atms.bufr (or use aapp_encodebufr_1c)
```

Similarly, for CrIS, run:

```
sdr=$(ls SCRIF_npp_${datestr}*.h5)
```

This assumes you have configured CSPP to use Full Spectral Resolution mode – which is standard practice for both S-NPP and NOAA-20 from April 2020.

After the granule loop:

```
combine_1c SCRIF*.11c          #generates cris.11c
```

```
cris_channels cris.11c cris_selection.11c  #the channel selection is specified in CRIS_FSR.fdf
```

```
eccodes_encodebufr_1c -i cris_selection.11c CRISFSR      #generates cris_selection.bufr
```

Finally, re-name `atms.bufr` and `cris_selection.bufr` according to DBNet convention. Add GTS headers. Disseminate.

## FY-3C sounders

Due to problems with the FY-3C MWTS-2 scanner, only MWHS-2, IRAS and MWRI are described in this

section.

Run `fy3clodb script Fy3VassL0db.csh`. Note that input files must be placed in the correct directory (`fy3clodb/data/org` – see `fy3clodb` documentation).

Move the MWHS Lo file (`*MWHSX.DAT`) from the output directory of `fy3clodb` (i.e. from `fy3clodb/data/vass_10`) to the input directory of `fy3cl1db` (i.e. to `fy3cl1db/data/mwhs_10`). Similarly for IRAS and MWRI, if required.

Run the `fy3cl1db` scripts `Fy3MwhsL1db.csh`, `Fy3IrasL1db.csh` and `Fy3MwriL1db.csh`. (Note that GPSXX files are not needed for MWHS2 or IRAS, though these are needed for MWRI and MWTS-2).

Run the AAPP ingest and BUFR encoding for MWHS-2:

```
mwhs2_sdr -o mwhs.11c $mwhs_file
export USE_VASS_SEQUENCE=Y
eccodes_encodebufr_1c -i mwhs.11c MWHS2
```

Re-name `mwhs.bufr` according to DBNet convention. Add GTS headers. Disseminate. If IRAS and/or MWRI is required, the approach is the same as for MWHS-2. The environment variable `USE_VASS_SEQUENCE` ensures that WMO-approved BUFR sequences are used for MWHS-2 and IRAS (not yet available for MWRI). These sequences were introduced in mid-2019, replacing the ad-hoc sequences that were in use before then.

## FY-3D sounders

The sounders on FY-3D are MWHS-2, MWTS-2 and HIRAS. The microwave imager is MWRI. HIRAS is a hyperspectral sounder that is similar to CrIS. Level 0 and level 1 processors are invoked in a similar way to FY-3C except that

- For level 0 processing, the scripts are `Fy3VassLodb.csh` and `Fy3HirasLodb.csh`
- There is a new level 1 script `Fy3HirasL1db.csh`

AAPP can ingest and process MWHS-2, MWTS-2 and MWRI data as described in the previous section.

In the case of HIRAS, CMA have not yet defined an official channel selection or BUFR sequence. As an interim measure, AAPP has been set up so that you can use the same channels as CrIS, encoded with the CrIS BUFR sequence that was introduced with NOAA-20:

```
hiras_sdr -o hiras.11c $hiras_file
hiras_channels hiras.11c hiras_channels.11c
eccodes_encodebufr_1c -i hiras_channels.11c HIRAS
```

Our understanding is that the HIRAS instrument can in principle be processed in FSR (full spectral resolution) or non-FSR mode, but the current version of the CMA processing package uses non-FSR mode.

HIRAS is not yet implemented as a DBNet service.

## What to do if you encounter a problem

There are a number of resources that can help if you encounter a problem:

1. The [AAPP Bugs Page](#) – a web page giving a list of current and past issues.
2. The [NWP SAF Helpdesk](#) – your questions will be directed to an AAPP developer.
3. The [NWP SAF Forums](#) – with AAPP Announcements and general discussion sections. If you wish, you can get automatic email notification of new posts.

There is also a Products and Software Working Group that meets at [International TOVS Study Conferences](#), and AAPP issues are often discussed there.

## Links to external resources

### Change record

Version	Date	Author / changed by	Remarks

1.0	Jan 2016	Nigel Atkinson	
1.1	April 2016	Nigel Atkinson	Update direct broadcast and BUFR sections
1.2	July 2016	Nigel Atkinson	Update for AAPP v7.13
1.3	Sept 2016	Nigel Atkinson	Update links. Add information about OPS-LRS and DBNet.
8.0	Dec 2017	Nigel Atkinson	For release with AAPP v8.
8.1	Oct 2018	Nigel Atkinson	Update the BUFR section and the TLE sources
8.2	Dec 2018	Nigel Atkinson	Update the DBNet section
8.3	Apr 2019	Nigel Atkinson	Update the section that describes FY-3 processing
8.4	Feb 2020	Nigel Atkinson	General update, including links
8.5	April 2020	Nigel Atkinson	After DBNet migration to CrIS FSR