NWP SAF

EPS-SG Direct Broadcast Software User Manual

Version: 1.0

20 April 2026

EPS-SG Direct Broadcast Software User Manual

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

Change record			
Version	Date	Author / changed by	Remarks
1.0	20/04/2026	Nigel Atkinson	Initial release

Table of Contents

1. INTRODUCTION

This document is the NWP SAF user manual for the direct broadcast level 0 and level 1 processing software. This version of the manual covers the MWS, METimage (VII) and IASI-NG (IAS) instruments on Metop-SG-A series of satellites. A future version will also include MWI, ICI and SCA.

The software packages have been procured by the EUMETSAT EPS-SG programme. The NWP SAF does not take responsibility for the design of the software or for the scientific content.

1.1 Disclaimers

IASI-NG

In the initial release (April 2026), IAS is not included. Whilst this manual describes the IAS processor, users will have to initially just work with MWS and VII. It is planned that IAS processor, together with a new set of test data, will be provided soon.

Test data

The EUMETSAT Metop-SG A1 satellite is under commissioning and its data are preliminary and undergoing testing. Data may be interrupted without notification. Users receiving these new data through any dissemination means (including, but not limited to EUMETCast) assume all risk related to their use of Metop-SG A1 data and EUMETSAT disclaims any and all warranties, whether expressed or implied, including (without limitation) any implied warranties of merchantability or fitness for a particular purpose.

MWS

Due to a hardware issue, the MWS channels 19 to 23 (183 GHz) are noisy and out of specification. These channels should be used with caution.

1.2 Software delivery

Each package is delivered in the form of a compressed tar file. Modules are supplied for:

- Level 0 processor: processing from raw VCDU or CADU to level 0 (all supported instruments)
- MWS level 1: processing from level 0 to level 1b
- VII level 1: processing from level 0 to level 1b
- IAS level 1: processing from level 0 to level 1c

For each package, pre-built binaries are available, and we recommend that users use the pre-built binaries where possible. For Level 0, MWS and IAS, source code is also available. For VII, due to license restrictions, users requiring source code will need to contact EUMETSAT User Services (ops@eumetsat.int).

You will need two external libraries, which are supplied:

- ESA's "EOCFI" library – available inside the MWS package and the VII package (different versions). This library is also used by the L0 processor.
- L0-reader software – available inside the VII package. Used only by VII.

You will also need some ancillary files that are supplied separately, see section 2.1.

To run the complete IAS processing chain, which includes generation of a cloud mask on the IAS fields of view, you will also need to install the NWC SAF "PPS" package, which is available separately from the NWC SAF (https://www.nwcsaf.org/). It is, however, possible to run the IAS chain without a cloud mask.

1.3 Test data

Test data files are also supplied, containing data from the commissioning phase of Metop-SG-A1 which may be used to test the software. Data are provided in two different raw formats:

- VCDU files: 20th Jan 2026, 08:39:53 to 08:52:02. This is the preferred format.
- CADU files with one file per Virtual Channel (VC), 13th Nov 2025, 08:53:22 to 08:59:27

A data file for fully-multiplexed CADU can also be made available on request, though it is more efficient if the VCs are separated. Discussed further in section 5.

Please note that the above test data are not compatible with version of the IAS processor that is to be released (see section 1.1). New test data will be supplied to support IAS.

For details on the requirements for the user station and raw data formats, see EUMETSAT's Metop-SG Direct Readout Guide[1]. The VCDU (or "transfer frame") is defined in Figure 3 of that document.

1.4 Wrapper scripts

The NWP SAF has created wrapper scripts that are intended to simplify the process of installing and running the software packages.

The wrapper script package also contains the dynamic libraries that are needed to link to the executables.

1.5 System requirements

To install and run the EPS-SG packages, you need a Linux system. To run the pre-built executables, you will need either (i) a Redhat Enterprise Linux 9 (RHEL9) or equivalent operating

[1] https://user.eumetsat.int/resources/user-guides/metop-sg-direct-readout-guide

system (e.g. Rocky Linux 9), or (ii) a portable container, such as *Apptainer*, in which a RHEL9 environment can be simulated.

Alternatively, you may be able to build from source code. See section 2.3.

The memory and CPU requirements vary from instrument to instrument, but as a guide:

- For IAS, at least 64 cores and 128 GB memory are recommended if you wish to process the data in close to real-time. Run time is approximately inversely proportional to the number of cores (at least, when the number of cores is small).
- For VII, at least 12 cores and 128 GB memory are recommended

The number of cores actually used is configurable (via ***VII_1B_AUX_PCNF*** for VII and via the Job Order for IAS).

You will also need access to the EUMETSAT Data Store in order to retrieve ancillary files: see https://user.eumetsat.int/data-access/data-store.

1.6 Registration and acceptance of license agreements

In order to download the EPS-SG software packages, you will need an account on the NWP SAF server. If you do not already have an account, please visit https://nwp-saf.eumetsat.int/site/ and click on the link “Register”.

Having signed in to the site, click on the tab “Software Downloads” and click on “Click here to change Software Preferences”. You will see a list of all the NWP SAF deliverables: tick the boxes for the packages that you wish to download. The packages applicable to EPS-SG are:

- EPS-SG level 0
- MWS level 1
- IASI-NG level 1
- VII level 1

Scroll down to the bottom of the page and accept the license agreement.

Note that requesting any one of the above packages will also give you access to the NWP SAF wrapper scripts and dynamic libraries.

The NWP SAF license agreement can be found at https://nwp-saf.eumetsat.int/site/software/licence-agreement/.

2. INSTALLATION

This section explains how to install the packages assuming you do *not* have administrator privilege – i.e. the software packages are installed in local directories. If you are running as administrator, you should follow your local conventions.

2.1 Download and unpack the tar files

The package files provided with the April 2026 release of the software are shown in Table 1. They will be made available via the "Software Downloads" section of the NWP SAF web site.

Table 1: Package files and their sizes. The IASI-NG package will be added at a future date.

Package file	Size (bytes)
L0_v3.1.0.tar.bz2	8671158
MWS_v2.5.0.tar.bz2	27511348
VII_v4.2.4.tar.bz2	36433130
NWPSAF_EPSSG_wrappers_v1.0.0.tar.bz2	54541294

Download the tar files to a convenient location on your filing system – which we will refer to as the "top directory". Unpack each file in the usual way (e.g. `tar -xf L0_v3.1.0.tar.bz2`).

Depending on your requirements, you may not need to install *all* the instrument packages from Table 1. However, it should be noted that in order to run the L0 processor you will need to download and unpack either the MWS or the VII level 1 package (or both) in order to obtain the bundled EOCFI libraries.

You will also need to download at least some of the ancillary and test files listed in Table 2. These are available at https://nwp-saf.eumetsat.int/downloads/epssg_db_ancillary_files/

Table 2: Ancillary files, test files and documentation package

Package file	Size (bytes)	Comment
LM_SGA1_fullchain-run_tds.tar.bz2	5620050267	VCDU test data and ancillaries
Local_L0_CADU_split_VC.tgz	2329472720	Optional CADU test data
SG_____AUX_RD09___S20070101000000Z_Exxxxxxx xxxxxxxZ_G20210201120000Z_CALV_OPE_OPER.SIP.tgz	2918568630	Used for VII and IAS
SG_____AUX_LSM____S20070101000000Z_Exxxxxxx xxxxxxxZ_G20230508000000Z_CALV_VAL_TEST_T.SIP.tgz	28854654	Used for VII and IAS
'Consolidated documentation package LM V1.zip'	18909537	Optional documentation package from EUMETSAT

Unpack the test data in the top directory.

The AUX_RD09 and AUX_LSM files will be unpacked when you run the test data for the first time (section 3). For now, just download them.

The “consolidated documentation” files are provided by EUMETSAT for reference, according to the list in Table 3.

Table 3: Documents provided by EUMETSAT for reference. The directories are relative to “Consolidated documentation package LM V1”

Directory	Document file
common	EPS-SG Generic Auxiliary Data Specification (GADS) 718291 v5B [Approved].pdf
	EPS-SG Generic Product Format Specification (GPFS) (702108 V5).pdf
IASI-NG	IASING-ICD-4200-104-TS (IASING L1CLOP Interface Control Document) - v2.3.pdf
	IASING-SP-4200-0272-CNES_0900_EPS-SG IASI-NG Level 1C Product Format Specification.pdf
	IASING-SP-4200-0273-CNES_0900_EPS-SG IASI-NG Level 1C Auxiliary Data Specification.pdf
	IASING-SUM-4200-126-TS (IASING L1CLOP Software User Manual) - v2.5.pdf
L0	EPS-SG Level 0 Product Format Specification (703928 v4B).pdf
	EPS-SG Local Mission L0 Processor - Interface Control Document_TO_BE_UPDATED.docx
	EPS-SG Local Mission L0 Processor - User Manual.docx
MWS	EPS-SG MWS Level 1B Auxiliary Data Specification (ADS) (777588 V5).pdf
	EPS-SG MWS Level 1B Product Format Specification (PFS) (777550 V5).pdf
	ICD-MWS EPS-SG Local Processor Interface Control Document v4.1_signed.pdf
	UM-MWS EPS-SG Local Processor User Manual v3.1_signed.pdf
VII	EPS-SG Local Processors Re-engineering Processor Prototypes VII EPS-SG LP Interface Control Document.docx
	EPS-SG VII Level 1B Auxiliary Data Specification 777147 v5A [Approved] (2).pdf
	EPS-SG VII Level 1B Product Format Specification (777138 v5A) [Approved].pdf
	UM-VII EPS-SG Local Processor User Manual v2.1.pdf

2.2 Installation of binaries using the wrapper script

To install the binaries, it is recommended to go to the *scripts* directory (having unpacked the NWPSAF_EPSSG_wrappers file) and use the NWP SAF script ***install_epssg_packages.sh.***

```
usage(){
  echo 'usage: install_epssg_packages.sh -t topdir -d installdir [PACKAGES]'
  echo ' where PACKAGES is from the list: EOCFI L0reader L0 MWS IAS VII'
  echo ' and if PACKAGES is omitted then all are installed'
}
```

The “topdir” provides the location of the packages (i.e. the directory where you unpacked the tar files). Each package is provided in an rpm file, except for IASI-NG which uses a gzipped tar file. Locations relative to “topdir” are set in the script according to Table 4.

Table 4: Locations of the RPM files or gzipped tar file containing the binaries

Directory	RPM or .tgz file
MWS/COTS	EOCFI-CPP-4.23-4.23-1.el9.x86_64.rpm
VII/COTS	EOCFI-CPP-4.25-4.25-1.el9.x86_64.rpm
VII/COTS	L0-READER-1.2-1.2-1.el9.x86_64.rpm
L0/Software	EPS-SG_Local_Mission_L0_Processor-1426d97-1.el8.x86_64.rpm
MWS/Software	eps-sg-mws-l1-lp-2.4.0-2.4.0-1.el9.x86_64.rpm
VII/Software	epssg-VII-L1B-LP-3.0.4-1.el9.x86_64.rpm
IASING/Software	L1CLOP-bin_03.01.02.tgz

The script runs *rpm2archive* (system utility) for each rpm file, then unpacks the resulting gzipped tar file in the specified installation location (which we refer to as ${EPS_SG_INSTALL}. This needs no root privilege. Alternatively, if the user has root privilege, then they can use standard system tools such as *yum*.

The packages are installed into the following locations. The principal executables are indicated in ***bold***:

```
${EPS_SG_INSTALL}/
├── opt/
│   ├── EOCFI-CPP-4.xx/
│   │   └── [subdirectories]/
│   ├── L0-READER-1.x/
│   │   └── [subdirectories]/
│   ├── facilities/
│   │   └── epssg/
│   │       └── VII-L1B-LP/
│   │           ├── bin/          # Contains VII_L1B_LP
│   │           └── lib/
│   └── mws/
│       └── eps-sg-mws-l1-lp-2.4.0/
│           ├── bin/              # Contains mws_l1_lp.bin
│           ├── lib/
│           └── schemas/
├── usr/
│   ├── bin/                      # Contains execute_L0processor.py
│   ├── lib/
│   ├── lib64/
│   └── share/
└── IAS/                          # Contains L1clopLocalLauncher.sh
    ├── conf/
    ├── java/
    ├── lib/
    └── licenses/
```

2.3 Building from source code

2.3.1 MWS

Instructions for building the MWS package from source are provided in the *UM-MWS EPS-SG Local Processor User Manual* in the documentation package (Table 3).

At the Met Office we were able to build the MWS package without requiring administrator privilege as follows:

- Use conda to assemble the required dynamic libraries. The yaml file used to create the conda environment *epssg* was as follows:

```
name: epssg

channels:
  - conda-forge

dependencies:
  - netcdf-cxx4=4.3.1
  - eigen=3.4.0
  - xerces-c=3.2
  - spdlog=1.10.0
  - fmt=8.1.1
  - gtest=1.11.0
  - libxml2=2.9.13
  - mkl
```

- Use ***install_epssg_packages.sh*** (see section 2.2) to unpack the ESA ***EOCFI*** library and to install it into $EPSSG_INSALL. Alternatively, it can be downloaded from the ESA web site (after registration). Source code for this ESA library is not publicly available.
- Run the supplied script *build_mws_from_source.sh*

2.3.2 VII

To obtain the VII source code, please contact EUMETSAT User Services, as this requires a special license.

Then follow the instructions in the *UM-VII EPS-SG Local Processor User Manual* in the documentation package (Table 3).

2.3.3 IASI-NG

The *IASING L1CLOP Software User Manual* (in the documentation package), does not provide any instructions for building the code from source. However, if you unpack the file *L1CLOP-source_03.02.01.tgz* you will find a ***readme.md*** file that gives instructions. The build procedure uses *Maven*, see https://maven.apache.org/

The NWP SAF has not built IASI-NG from source. If you do so, please provide feedback (section 9). We have noted that the Java executables run without modification on both RHEL7 and RHEL9 systems. Therefore, portability appears to be less of a problem that it would be for C++ or Fortran source files.

2.3.4 Level 0

The *EPS-SG Local Mission L0 Processor - User Manual* (in the documentation package) does not provide instructions for building the code from source. However, the L0/Software/epssg-l0-lm-v3.1.0 directory contains a ***readme.md*** file that gives instructions. The build procedure uses Meson, see https://mesonbuild.com/.

The NWP SAF has not built L0 from source. If you do so, please provide feedback (section 9).

2.4 Data directories

The data directories, to be used at run-time, are independent of the installation directory. The EPS-SG wrapper scripts use the following structure for the data directories. Directories in grey are not used, but the software may create them. The directory structure is created when you run the VCDU test case for the first time (section 3.1).

```
${EPS_SG_DATA}/
├── AUX/
│   ├── common/
│   ├── IAS/
│   ├── MWS/
│   └── VII/
├── ccsds_out/
├── conf/                    # Contains l0procconf.xml
├── ias_l1/                  # IAS level 1 output directory
├── l0products/              # Outputs from L0 processor; inputs to L1 processors
├── mmam_out/
├── mws_cof_l1/
├── mws_l1/                  # MWS level 1 output directory
├── vcdu/                    # Input directory for raw VCDU files
├── vii_l1/                  # VII level 1 output directory
└── nwcsaf_l2/               # NWCSAF level 2 cloud mask
```

2.5 Access to EUMETSAT data store

For processing your own data, the EUMETSAT tool ***eumdac*** is used to download current Metop-SG orbit parameters from the EUMETSAT Data Store. You will need to set up an account for the EUMETSAT User Portal and to install the ***eumdac*** software.

For details, see https://user.eumetsat.int/data-access/data-store

You do not need to access the Data Store when running with the supplied VCDU test data.

3. RUNNING THE TEST CASES

3.1 Basic running of L0 and L1 from VCDU

EUMETSAT have provided a set of raw data, together with the required ancillary files, for a Metop-SG-A1 overpass, in VCDU format (i.e. Transfer Frames). The file name is ***LM_SGA1_fullchain-run_tds.tar.bz2***. Download this file and unpack it (see section 2.1).

You will need to define and export the following environment variables:

- EPSSG_INSTALL – the location where you installed the software (section 2.2)
- EPSSG_DATA – the location where you wish to install the data (section 2.4)

Then go to the *scripts* directory and run the following:

```
./run_EPSSG_test_data.sh [options] instruments
```

where the options are as follows:

```
-n 0: do not run level 0 processing
-n 1: do not run level 1 processing
-n 01: do not run either level 0 or level 1 processing
-c: CADU option, see section 3.2
-h: display help and exit
```

and *instruments* are taken from: MWS, VII, IAS. If no instruments are specified, then the AUX files from the L0 and L1 packages are copied to $EPSSG_DATA but no processing is run.

If the run is successful then the output files will be created in the following subdirectories of $EPSSG_DATA: **mws_l1**, **vii_l1**, **ias_l1**.

Note:

- If you are processing IAS then it is recommended to include VII also. The script will run them in the correct order (VII first).
- By default, the VII level 2 cloud mask is not generated and will not be included in the IAS level 1 output.

3.2 Running L0 and L1 from CADU

EUMETSAT have provided 6 minutes of raw data in CADU format. This file does not include all the ancillary data, therefore you will need (i) to first run the VCDU test case (if you haven't done so already), and (ii) access to the EUMETSAT Data Store (section 2.5).

The file name is **Local_L0_CADU_split_VC.tgz**. Download this file to a suitable location on your file system, create a directory *Local_L0_CADU_split_VC* and unpack the tar file.

Then run *run_EPSSG_test_data.sh* as in section 3.1 but adding a flag **-c**

You will find that L0 processing for VII takes significantly longer with CADU than with VCDU. Therefore, we recommend that you use VCDU input where possible.

The *run_EPSSG_test_data.sh* script also contains a -C option for processing fully-multiplexed CADU data (single file containing all instruments). However, we have found that this takes a very long time to run, and so we have not provided fully-multiplexed test data on the NWP SAF web site.

3.3 Inclusion of VII L2 cloud mask in IAS output

To include the VII L2 cloud mask in the IAS L1 output, the sequence of steps is as follows:

1. Process VII and IAS to level 0
2. Process VII to level 1b
3. Run the NWCSAF cloud mask to process VII to level 2
4. Process IAS to level 1c, making use of the cloud mask

Step 3 requires the use of NWP forecast files. GFS forecast files are convenient, and freely available, but only remain on-line for one month. Therefore, we recommend that you omit the cloud mask step when processing the fixed test data, but you can introduce it when running in near-real-time mode.

More details are provided in the next section.

4. THE NWCSAF PPS SOFTWARE

4.1 Installing the NWCSAF PPS software

To receive the PPS software, you will need to register with the Nowcasting SAF at https://www.nwcsaf.org/.

Navigate to Software Download -> NWCSAF PPS. The **PPS vEPSSG-SAFbeta6** is currently available as a beta release (see link to beta releases at the bottom of the page), though you should look out for new releases.

For full details on installation, see the Installation Notes document on the web page for the NWCSAF/PPS Beta releases.

We recommend the Conda installation method, creating a Conda environment called **ppsEPSSG**. If we download files into a directory called $PPS_DATA, then the steps are as follows:

1. Download the PPS vEPSSG-SAFbeta6 conda package
2. Download and uncompress the “Installation script – 3rd party s.w. only” (ppsvEPSSG_SAFbeta6_conda_environment.yml)
3. Download ancillary data (available from the NWCSAF web pages). Some of these are quite large (the last one is 7GB).

```
acpg_EmissivityData_20201024.tar.gz
acpg_PhysiographyData_20201217.tar.gz
acpg_v2021_staticdata.tar.gz
acpg_vEPSSG_SAFbeta6.tar.gz
CPP_anc_v2021_patch2.tar.gz
gap_filled_snowfree_c6_albedo.v20201009.tar.gz
```

4. Run the following commands (which can take an hour or more):

```
cd $PPS_DATA
conda env create -f ppsvEPSSG_SAFbeta6_conda_environment.yml
conda activate ppsEPSSG
conda install pip
tar xzf pps_EPSSG_SAFbeta6_conda_packages.tgz
pip install level1c4pps
conda install -c file://${PPS_DATA}/pps_EPSSG_SAFbeta6_conda_packages -c conda-forge \
   nwcsaf_cpp nwcsaf_acpg nwcsaf_hrw level1c4pps pyhdf
cd acpg_vEPSSG
python pps_make_install_static_data.py --repo $PPSDATA --dest $PPSDATA/data/pps
```

The resulting PPS directory structure has the following form (with key input directories highlighted in bold):

```
$PPS_DATA/data/pps/
├── export/
├── import/
│   ├── ICEMAP_data/
│   ├── IMAGER_data/
│   └── NWP_data/
├── intermediate/
│   ├── ANC_remapped/
│   ├── AUX_on_region/
│   ├── AUX_remapped/
│   ├── EMISS_remapped/
│   ├── HRW_remapped/
│   ├── ICEMAP_remapped/
│   ├── IMAGER_remapped/
│   └── NWP_remapped/
└── static/
    ├── AUX_data/
    ├── CPP_data/
    └── EMISS_data/
```

4.2 Running the NWCSAF PPS software

See the NWP SAF script ***run_nwcsaf_pps.sh***. It assumes:

- Environment variables EPSSG_DATA and PPS_DATA have been set up and point to the correct directory structures.
- Conda environment ppsEPSSG has been set up. The script activates this environment automatically.
- Internet access is available, to download GFS forecast files.
- The required GFS forecast files are available from https://bin.ssec.wisc.edu/pub/eosdb/ancillary/. These only remain on-line for one month.

The script has three parts:

1. Converts the format of the VII L1b file to PPS format (using ***metimage2pps.py***), taking input data from $EPSSG_DATA/vii_l1, and writing the output to $PPS_DATA/data/pps/import/IMAGER_data.
2. Downloads a GFS forecast file into $PPS_DATA/data/pps/import/NWP_data. Note that this step was implemented outside of PPS due to difficulties in getting the supplied code to work through a firewall. Other forecast files (e.g. ECMWF) may be used instead, see the PPS documentation. If you are not behind a firewall, you can try the automated method of downloading GFS files (again, see PPS documentation).
3. Runs PPS for each VII granule (using ***ppsRunAll.py***) and moves products to $EPSSG_DATA/vii_l2_cma

Please note that the script does not implement any housekeeping. You should delete old forecast files or cloud mask files before running the script, i.e. delete unwanted files from the following directories:

- $EPSSG_DATA/vii_l1
- $PPS_DATA/data/pps/import/IMAGER_data
- $PPS_DATA/data/pps/import/NWP_data
- $EPSSG_DATA/vii_l2_cma

5. PROCESSING YOUR OWN DIRECT BROADCAST DATA

5.1 The *run_EPSSG.sh* script

The main wrapper program which can be used for all instruments is called *run_EPSSG.sh*. Its usage is as follows (use the “-h” option to show usage):

```
run_EPSSG.sh [-d YYYYmmdd] [-c] [-p] instruments
```

where

- instruments are taken from: L0_NAV, L0_MWS, L0_VII, L0_IAS, MWS, VII, IAS.
- you need to supply the date when running the L0 processor

- the -c option is for CADU input
- the -p option parallelises the L0 processor

The script requires environment variable EPSSG_DATA to be defined. You will also need EPSSG_INSTALL to be defined, unless the packages have been installed in the root directory.

Note that all instruments require “NAVATT” information, therefore you should include L0_NAV in the instrument list. The date of the acquisition must be supplied, since there is no established file naming convention for VCDU files.

As a simple example, to process MWS from VCDU (i.e. all data in $EPSSG_DATA/vcdu) to level 1b, you can use:

```
run_EPSSG.sh -d YYYYmmdd -p L0_NAV L0_MWS MWS
```

See section 5.5 for an example of the usage of this script in the IAS processing chain.

5.2 Processing VCDU or EVCDU data

You can process your own VCDU/EVCDU data in a similar way to the test data of section 3. You will need to be able to download ancillary files from the EUMETSAT Data Store: see section 5.4.

Before processing your own data, if you haven’t already done so you should first run ***run_EPSSG_test_data.sh*** (section 3) in order to set up the correct directory structure in $EPSSG_DATA and to install ancillary files. You can run this without specifying any instruments if you want to save time.

There are some values in ***L0_config.sh*** that you may wish to customise:

- timeslot_duration=3600. This controls the size of the output granules. A value 3600 usually produces only one granule for a DB pass (or two if the data straddles an hour boundary). See comment in section 7.
- evcdu_offset=20. EVCDU may contain some extra bytes compared to VCDU. Depends on your front end.
- scname=SGA1. Currently only the SGA1 satellite is supported.
- descrambling=${descrambling:-False}. Applicable to CADU input data.
- The input directory, normally $EPSSG_DATA/vcdu

5.3 Processing CADU data

EPS-SG produces a very large volume of data: an overpass of 12 minutes is ~8GB of raw data, if you include all Sat-A local mission instruments. Just to parse a file with all the data multiplexed consumes a significant time. Therefore, the traditional approach of having the front-end generate a single file, containing the full overpass, and processing it in one go, is not optimal. A second

drawback is that the L0 processor is going to loop over the full stream one time per Virtual Channel (VC), thus multiplying the run time. A work-around (though not ideal) is to launch parallel jobs, one per VC. This way the work is externally parallelized.

Instead, it is much better to ask the front-end to produce different files, one for each VC. This yields smaller inputs to the L0 processor, and on top of that, different L0 instances, per VC, can be launched in parallel. With this strategy, a reasonable end to end timeliness can be achieved. This is the recommended approach.

A gzipped file of test data, ***Local_L0_CADU_split_VC.tgz***, is available. It contains one CADU file per VC. To process it, unpack the file and run ***run_EPSSG_test_data.sh*** with the "-c" flag set. However, we have noticed that the run-time for VII is larger than it is with VCDU. We recommend that you use VCDU input, rather than CADU, whenever possible.

5.4 Updating auxiliary files

The auxiliary files to be used initially are specified in the following scripts:

- L0_config.sh
- mws_config.sh
- ias_config.sh
- vii_config.sh

Some of these will need to be either regularly or occasionally updated. The software handles two of these automatically:

1. Predicted orbit files – AUX_POFD – are published daily
2. IERS Bulletin A files – AUX_IBA – are published weekly with a validity period of 12 months.

They can both be retrieved from EUMETSAT Data Centre using ***eumdac*** (section 2.5). This is managed by the scripts *get_current_pofd.sh* and *get_current_iba.sh*, which are called from *L0_generate_joborder.sh*, *MWS_generate_joborder.sh*, *IAS_generate_joborder.sh* and *VII_generate_joborder.sh*.

Alternatively, the IERS Bulletin A could be downloaded from the IERS web site[2].

There are many other auxiliary files that are currently hard coded into the **config.sh* files. Some of these may need periodic updating: please look out for announcements at https://nwp-saf.eumetsat.int/site/metop-sg-direct-broadcast-software/.

[2] https://datacenter.iers.org/data/latestVersion/bulletinA.txt

5.5 IAS processing chain

The complete IAS processing chain (including cloud mask) would be run like this (after setting up environment variables such as EPSSG_DATA, PPS_DATA and EPSSG_INSTALL):

```
run_EPSSG.sh -d $YYYYmmdd -p L0_IAS L0_VII  #L0 for IAS and VII
run_EPSSG.sh VII                            #L1 processing for VII
run_nwcsaf_pps.sh                           #generate cloud mask
run_EPSSG.sh IAS                            #run L1 processing for IAS
```

This command sequence will attempt to process all files that the user provides in $EPSSG_DATA/vcdu.

5.6 Housekeeping

The *run_EPSSG.sh* script does not perform any housekeeping, other than (for clarity) removing empty *l0products* subdirectories after running the L0 processing.

We recommend that you empty the following directories before processing a pass:

- $EPSSG_DATA/vcdu
- $EPSSG_DATA/l0products

You should also periodically clean the output directories (e.g. $EPSSG_DATA/vii_l1) in order to remove old, bulky output files.

6. TYPICAL RUN TIMES

Typical run times are given in Table 5, for the VCDU test case of section 3.1, with 12 minutes (720 seconds) of data. The server node has the following characteristics:

- AMD EPYC 7763 64-Core Processor
- Up to 120 threads available to user applications
- RHEL9.7 operating system

In the tables below, the L0 processors are run in parallel; therefore, the times in square brackets do not contribute to the end-to-end timeliness figure. The input data comprises 12 minutes of data in VCDU format.

Table 5: MWS processing chain with 4 threads and 16 GB memory

Process	Run time (s)
L0 NAV	[8]
L0 MWS	8
L1 MWS	7

Total	**15 (=0.25 minutes)**

Table 6: VII processing chain with 16 threads and 120 GB memory

Process	Run time (s)
L0 NAV	[8]
L0 VII	282
L1 VII	210
Total	**492 (=8.2 minutes)**

Table 7: IAS and VII processing chain configuration 1, with 60 threads and 120 GB memory

Process	Run time (s)
L0 NAV	[8]
L0 VII	285
L0 IAS	[56]
L1 VII	246
L1 IAS (no cloud mask)	1104
Total	**1635 (=27.3 minutes)**

Table 8: IAS and VII processing chain, configuration 2, with 120 threads and 240 GB memory

Process	Run time (s)
L0 NAV	[8]
L0 VII	261
L0 IAS	[56]
L1 VII	221
L1 IAS (no cloud mask)	926
Total	**1408 (=23.5 minutes)**

Note that the IAS L1 processor can give an exit code of 1, due to quality flags, even though valid products are generated.

PPS cloud mask was not included in the above due to unavailability of GFS forecast files. However, another case (not included in the distribution) indicated that running the PPS would add an additional 8.5 minutes of processing time.

If your system front-end can deliver raw VCDU files *while the acquisition is still in progress*, it should be possible to reduce the latency of the final product (though this has not been verified in

the current release). For example, if VII raw data are received as 1-minute granules during acquisition, then the VII L1 files and PPS cloud mask could be generated while the acquisition is in progress, ready for use in the IAS processing. It is a recommendation from EUMETSAT that the IAS processor should be launched after the acquisition is complete.

7. OTHER POINTS

MWS level 1 processing needs to be explicitly run for each L0 granule, whereas for VII and IAS it is possible to combine multiple L0 granules into a single L1 product. If you require a single L1 product for MWS, this may be done by setting the parameter *timeslot_duration* in ***L0_config.sh*** to a large number (e.g. 3600 seconds or larger) so that there is only a single L0 file for each instrument. This is the default setting.

8. KNOWN ISSUES AND LIMITATIONS

This delivery of the EPS-SG Local Mission SW for satellite SG-A is intended to facilitate the deployment of the Local Mission processing chain at the local stations, and to be reasonably representative in terms of interfaces, usage and run time performance. However, it is not intended to meet operational quality in the generated products, including the 10% consistency with the global mission. This is justified by the fact of being released in parallel to the Cal-Val activities for all the instrument chains involved, which are at different stages of progress. It is expected that during and after Cal-Val, processors and auxiliary data will be evolved to meet the mission quality requirements.

The VII L1B processor is affected by an anomaly that makes the processor crash occasionally because of a problem in the landmark correction function. This is visible in the delivered test dataset, where one of the granules of VII L1B (the one starting at 08:39:59) is missing. Landmark correction is an optional part of the processing and can be deactivated by configuration. In order to do this, to avoid the processor crash, the following parameter needs to be configured

```
<landmark_enable>false</landmark_enable>
```

in the auxiliary file ***PCNF.xml*** under the auxiliary data SIP folder:
SGA1_VII_1B_AUX_PCNF_G_S20251031090000Z_Exxxxxxxxxxxxxx_G20260203083000Z_CALV_VAL_TEST_T.SIP

9. USER SUPPORT

If you have any questions about the software or problems running it, please contact the NWP SAF Helpdesk at https://nwp-saf.eumetsat.int/site/help-desk/. Depending on the nature of the issue, we will either answer your question or will refer it to EUMETSAT for further investigation.