NWP SAF

MWIPP User Manual

Version 1.2

February 2023

MWIPP 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 December 2016, 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
0.1	14/06/2018	Nigel Atkinson	Initial draft
0.2	10/10/2018	Nigel Atkinson	Version for issue to beta testers
1.0	14/01/2019	Nigel Atkinson	Initial release
1.1	14/02/2022	Nigel Atkinson	Update to include support for simulated MWI and ICI data
1.2	14/02/2023	Nigel Atkinson	Update the generic BUFR sequence to allow full-orbit MWI/ICI to be encoded. Add discussion of the solar angles for AMSR-2.

Table of Contents

1. INTRODUCTION

This document is the user manual for Version 1 of the Microwave Imager Pre-processor (MWIPP). It incorporates the following elements:

- Installation Guide (section 2)
- Run-time instructions (section 3)
- Software description (section 0)
- Scientific description (section 5)

1.1 Reference documents

[RD-1]   NWPSAF-MO-DS-035, MWIPP Product Specification
[RD-2]   NWPSAF-MO-DS-038, MWIPP Top level design

1.2 Functionality summary

Version 1.0 of the MWIPP has the following functionality:

- Support for SSMIS, AMSR-2, GMI, MWRI instruments
- Support for BUFR and hdf5 formats
- Spatial averaging capability
- Re-mapping capability, specifically re-mapping to a user-defined grid
- The design supports future extension to the microwave imagers on Metop-SG (i.e. MWI and ICI)

The intention is that the NWPSAF SSMIS_PP deliverable will not be developed further, since it is superseded by MWIPP (see section 5.1 for details)

Version 1.1 of the MWIPP has the following additional functionality:

- Support for simulated MWI and ICI data, provided by EUMETSAT[1] in netCDF format
- Re-mapping of feedhorns that are not co-located (as is the case for MWI and ICI)
- Re-mapping of one microwave imager to another (ICI to MWI)

Version 1.2 of the MWIPP has been updated to support the November 2022 EUMETSAT release of MWI/ICI test data. For AMSR-2 the solar angles have been made consistent with those from other instruments.

1.3 System requirements

To install and run MWIPP, you need the following:
- A Linux system, or equivalent
- A Fortran-90 compiler that supports Fortran-2003, e.g. gfortran or ifort
- ecCodes library version 2.7.0 or later (see section 2.1)

---

[1] https://www.eumetsat.int/eps-sg-user-test-data

- If you are building ecCodes, cmake is required
- hdf5 library version 1.8.8 or later, with Fortran-90 enabled (section 2.2). If using hdf5 v1.8.x then Fortran-2003 also needs to be enabled (not necessary with v1.10.x).
- netCDF C and Fortran libraries (required for ICI and MWI, optional for other instruments)
- gmake

If you are not sure what versions of the external libraries are installed on your system, try running commands such as:

```
bufr_dump -V
h5dump –V
cmake --version
```

An optional python plotting routine is included: to use it, you need python2.7 or later, and the Cartopy library, see https://scitools.org.uk/cartopy/docs/latest/. You will also need numpy, h5py and matplotlib.

2. INSTALLATION

This section explains how to install MWIPP and its dependencies (ecCodes, hdf5, netCDF) 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.

It is recommended that you copy the code fragments contained in this section to create your own build scripts. You should first define the following environment variables (directory names should be absolute paths, not relative) and create the relevant directories:

- `TAR` – directory for tar files
- `BUILD` – directory for building ecCodes and/or hdf5
- `ECCODES_INSTALL_DIR` – installation directory for ecCodes. The directory containing lib, bin, include, etc.
- `HDF5_INSTALL_DIR` – installation directory for hdf5. The directory containing lib, bin, include, etc. It can be convenient to install netCDF to the same directory.
- `FC` – Fortran compiler, e.g. gfortran

2.1 Download and install ecCodes (if not already installed)

Please note that the installation of ecCodes[2] uses cmake[3] (see links in footnotes), so cmake should be installed first if not already available on your system.

An example script to download and install ecCodes is given below:

```
#Download ecCodes
  version=2.24.2    #edit this line to select the latest version on the web site
  cd $TAR
  tarfile=eccodes-${version}-Source.tar.gz
  url="https://confluence.ecmwf.int/download/attachments/45757960/"
```

---

[2] https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home
[3] https://cmake.org/

```
  [ -s $tarfile ] || { echo "Downloading ecCodes"; wget ${url}${tarfile}; }
  [ -s $tarfile ] || { echo "Failed to download $tarfile"; exit 1; }

#Unpack
  cd $BUILD
  dir=${tarfile%.tar.gz}
  [ -d $dir ] || { echo "unpacking ecCodes"; tar -xzf $TAR/$tarfile; }

  echo "Building ecCodes ..."

  mkdir -p eccodes_build
  cd eccodes_build

#Configure and make
  cmake \
    -DCMAKE_INSTALL_PREFIX=${ECCODES_INSTALL_DIR} \
    -DCMAKE_Fortran_COMPILER=${FC} \
    $BUILD/$dir
  make || { echo "problem with ecCodes make"; exit 1; }
  make install >install.out || { echo "problem with ecCodes make install"; exit 1; }
  echo "Finished ecCodes build"
```

If you encounter difficulties in installing ecCodes, you are advised to contact ECMWF via software.support@ecmwf.int. (ecCodes is not an NWP SAF deliverable).

2.2 Download and install hdf5 (if not already installed)

See footnote for a link to the hdf5[4] web page. An example installation script for hdf5 is given below:

```
#Download hdf5
  cd $TAR
  tarfile=hdf5-1.10.1.tar.gz
  url="https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.10/hdf5-1.10.1/src/"
  [ -s $tarfile ] || { echo "Downloading HDF5"; wget ${url}${tarfile}; }
  [ -s $tarfile ] || { echo "Failed to download $tarfile"; exit 1; }

  cd $BUILD
  dir=${tarfile%.tar.gz}
  [ -d $dir ] || { echo "unpacking HDF5"; tar -xzf $TAR/$tarfile; }

  cd $dir
  export FC
  mkdir -p $HDF5_INSTALL_DIR

  ./configure --prefix=$HDF5_INSTALL_DIR --enable-fortran \
    || { echo "problem with hdf5 configure"; exit 1; }
  #note that if you are installing hdf5-1.8.x you also need --enable-fortran2003

  make || { echo "problem with hdf5 make"; exit 1; }
  make install || { echo "problem with hdf5 make install"; exit 1; }
  echo "Finished hdf5 build"
```

2.3   Download and install netCDF (if not already installed)

The netCDF library is *optional* for MWIPP v1, but if it is not available then the parts of MWIPP that relate to Metop-SG MWI/ICI processing will not be built.

---

[4] https://support.hdfgroup.org/HDF5/index.html

An example installation script for netCDF is given below. It is convenient to install netCDF to the same installation directory as hdf5.

```
#netCDF C library
  cd $TAR
  tarfile=v4.8.1.tar.gz
  url=https://github.com/Unidata/netcdf-c/archive/refs/tags
  dir=netcdf-c-4.8.1
  [ -f ${dir}.tar.gz ] || wget $url/$tarfile
  [ -f $tarfile ] && [ ! -f ${dir}.tar.gz ] && mv $tarfile ${dir}.tar.gz
  cd $BUILD
  [ -d ${dir} ] || tar -xf $TAR/${dir}.tar.gz
  cd ${dir}
  export LDFLAGS="-L$HDF5_LIB"
  export CPPFLAGS="-I $HDF5_INSTALL_DIR/include"
  ./configure --prefix=$HDF5_INSTALL_DIR --disable-dap && make && make install
  [ $? = 0 ] && echo "netcdf-c built OK" || exit 1

#netCDF Fortran library
  cd $TAR
  tarfile=v4.5.3.tar.gz
  url=https://github.com/Unidata/netcdf-fortran/archive/refs/tags
  dir=netcdf-fortran-4.5.3
  [ -f ${dir}.tar.gz ] || wget $url/$tarfile
  [ -f $tarfile ] && [ ! -f ${dir}.tar.gz ] && mv $tarfile ${dir}.tar.gz
  cd $BUILD
  [ -d ${dir} ] || tar -xf $TAR/${dir}.tar.gz
  cd $dir
  export LD_LIBRARY_PATH=$HDF5_LIB
  export LDFLAGS="-L$HDF5_LIB"
  export CPPFLAGS="-I${ HDF5_INSTALL_DIR }/include"
  ./configure --prefix=$ HDF5_INSTALL_DIR && make && make install
  [ $? = 0 ] && echo "netcdf-f built OK" || exit 1
```

2.4 Download MWIPP

As with other NWPSAF deliverables, the MWIPP package is downloaded via a link on the "Software Downloads" section of the NWP SAF web site. After user registration, click on "Change Software Preferences", select MWIPP and check the box to confirm that you agree to the terms of the license agreement. Download the tar file to a suitable location on your computer (e.g. the $TAR directory that you have already defined). Navigate to the directory where you want to install MWIPP and unpack the tar file using: `tar -xzf ${TAR}/${file}`

2.5 Configure MWIPP

Navigate to the `MWIPP/src` directory, then run the *configure_mwipp.sh* script. By default it uses bash shell, but will work equally well under ksh if you change the first line.

(i)   If you have installed ecCodes, hdf5 and netCDF yourself (as in 2.1, 2.2 and 2.3), and have defined the environment variables `ECCODES_INSTALL_DIR`, `HDF5_INSTALL_DIR`, `FC`, and (optionally) `NETCDF_INSTALL_DIR` then you can run *configure_mwipp.sh* with no arguments, like this

```
./configure_mwipp.sh
```

(ii)   If ecCodes and/or hdf5 and/or netCDF are centrally installed (and therefore `ECCODES_INSTALL_DIR`, `HDF5_INSTALL_DIR`, `NETCDF_INSTALL_DIR` are not set) you can let the script try to determine the library locations: just run

   `./configure_mwipp.sh --fortran-compiler=$FC`

   Be careful in case your system already has environment variables defined for `ECCODES_LIB` and/or `HDF5_LIB` that are unrelated to MWIPP. If this is the case you should either *unset* them or supply a command argument (see below).

(iii)   If necessary, you can specify the various directories explicitly, either via arguments or environment variables. See Table 1.

Table 1: User inputs to the script *configure_mwipp.sh*

Script argument	Equivalent environment variable	Purpose
--fortran-compiler=	FC	Compiler: defaults to gfortran
-	ECCODES_INSTALL_DIR	As defined in 2.1. Directory contains "lib" (or "lib64") and "include" subdirectories.
-	HDF5_INSTALL_DIR	As defined in 2.2. Directory contains "lib" (or "lib64") and "include" subdirectories.
-	NETCDF_INSTALL_DIR	As defined in 2.3. Directory contains "lib" (or "lib64") and "include" subdirectories.
--eccodes-lib=	ECCODES_LIB	Location of the ecCodes library files, e.g. *libeccodes_f90.so*
--hdf5-lib=	HDF5_LIB	Location of the hdf5 library files e.g. *libhdf5_fortran.so*
--netcdf-lib=	NETCDF_LIB	Location of the netCDF library files e.g. *libnetcdff.so*
--eccodes-inc=	ECCODES_INC	Location of the ecCodes include files, e.g. *eccodes.mod*
--hdf5-inc=	HDF5_INC	Location of the hdf5 include files, e.g. *hdf5.mod*
--netcdf-inc=	NETCDF_INC	Location of the netcdf include files, e.g. netcdf.*mod*

Note that script arguments always take priority over environment variables, and HDF5_INSTALL_DIR (defined in 2.2) takes priority over HDF5_LIB / HDF5_INC (similarly for ecCodes).

Note also that it is not necessary to specify the location of the "include" directory if it is on the same level as the "lib" directory (e.g. /usr/include … /usr/lib) or if it is a subdirectory of "lib".

You may find that the hdf5 libraries are centrally installed on your system but have not been built with Fortran enabled (*libhdf5_fortran.so* missing*)*, or are a rather old version (v1.8.x) and have not been built with Fortran2003 support. In this case the script will warn you, and you should follow the instructions in 2.2 to create a new library.

The *configure_mwipp.sh* script firstly runs some test programs to verify whether the external libraries are OK. Secondly it creates the necessary directories used in the build process. Thirdly, it creates a file called *Makefile.ARCH* which is used by all the lower-level makefiles. Finally, it

creates a file *mwipp_env.sh* which is placed in the "bin" directory and should be sourced at run time (this file sets up PATH and, if necessary, LD_LIBRARY_PATH).

The MWIPP directory structure is as follows:

> bin (contains executables, and *mwipp_env.sh*)
> build (contains generated modules and libraries)
> src
>> bin (main programs)
>> control (example namelist files, see section 3)
>> libcommon
>> libssmis
>> libamsr2
>> libgmi
>> libmwri
>> libmwi
>> libici
>> … other instrument-specific libraries may be added later

If you prefer, you can create your own version of *Makefile.ARCH*, taking *Makefile.ARCH.in* as a template.

2.6 Make MWIPP

Just type make to compile the software.

There is no specific make install option, but if you want to copy the files in *bin* to somewhere else on your computer, you can do so manually. Use make clean if you want to re-build after changing something.

3. RUNNING THE SOFTWARE

3.1 General comments

A main program is provided for each supported instrument: *amsr2_main.exe*, *gmi_main.exe*, *mwri_main.exe*, *ssmis_main.exe*, *mwi_main.exe*, *ici_main.exe*.

The general philosophy of the main MWIPP executables is:

- Things that are likely to change from run to run (e.g. the names of input and output files) are supplied via command line arguments (e.g. -i infile, -o outfile).

- A top-level namelist (which is read directly by the Fortran executable) defines which processing steps are to be run. The user can create different namelists for different functions. The name of the top-level namelist is entered via a command-line argument (-n).

- Lower-level namelists contain things that are less likely to change (e.g. the WMO identifier of your centre, for BUFR encoding). The names of these namelists are specified in the top-level namelist.

Example namelists are provided in the MWIPP/src/control directory. These can be copied and customised by the user. See also section 3.11.

In addition, some standalone executables are provided which perform specific hdf5 to BUFR conversion facilities: amsr2_hdf2bufr.exe, gmi_hdf2bufr.exe and mwri_hdf2bufr.exe.

For SSMIS, only BUFR format is accepted for input and output, because Unified Pre-Processor (UPP) files are normally distributed in BUFR. For AMSR-2, GMI and MWRI, input can be BUFR or hdf5. For MWI and ICI, only netCDF input is currently supported.

As mentioned in section 2.5, it is recommended to source the MWIPP environment file before running any main program, in order to set up your PATH and/or LD_LIBRARY_PATH:

```
source ${MWIPP_HOME}/mwipp_env.sh
```

or

```
. ${MWIPP_HOME}/mwipp_env.sh
```

where `${MWIPP_HOME}` is the full path of the directory containing the executables (the *bin* directory).

A high-level block diagram of MWIPP is shown in Figure 1.
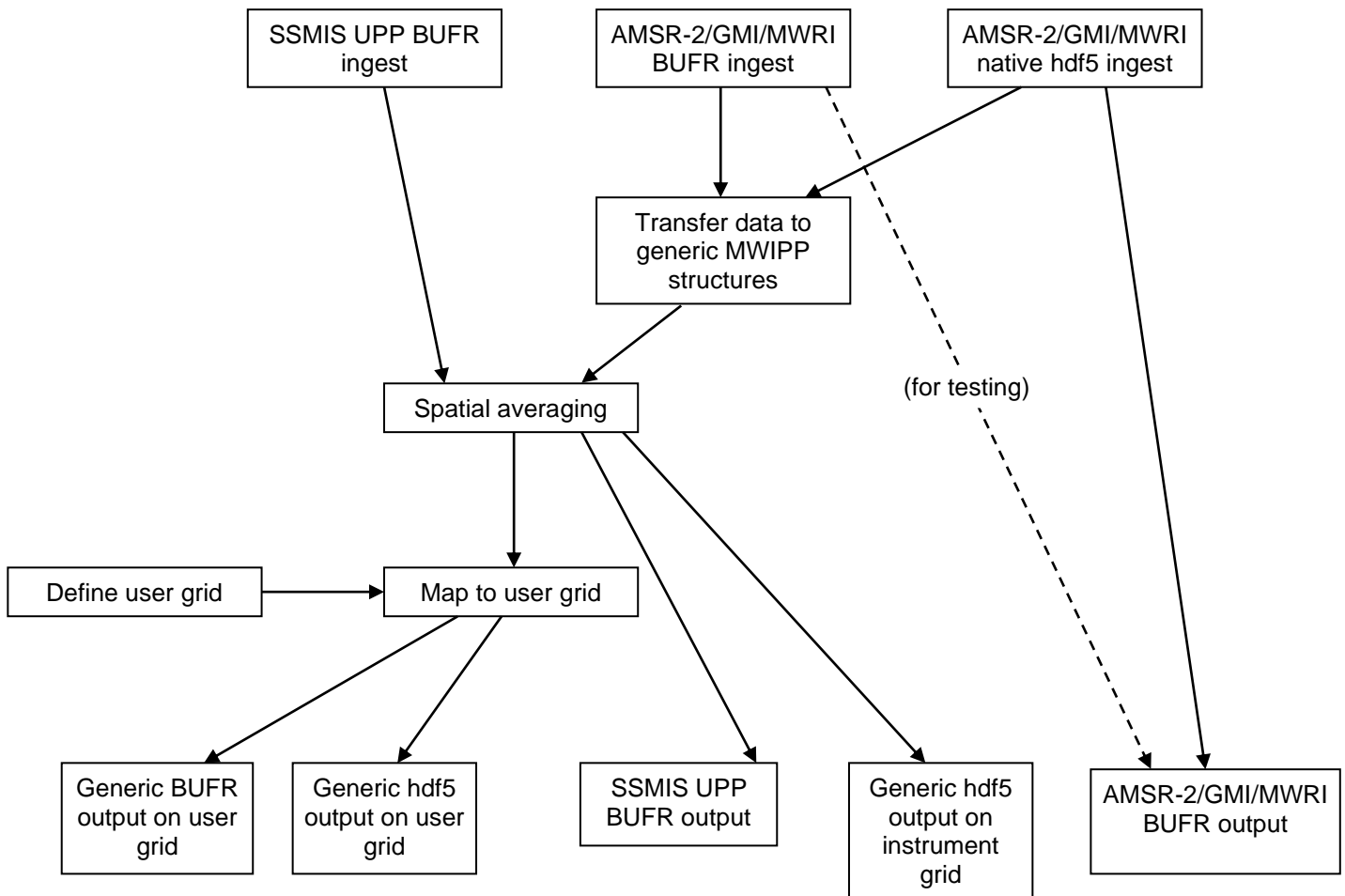
Figure 1: MWIPP block diagram, showing processing flows for SSMIS, AMSR-2, GMI, MWRI

3.2 ssmis_main.exe

The Special Sensor Microwave Imager/Sounder (SSMIS) is a 24-channel microwave imager, with channels ranging in frequency from 19.35 GHz to 190 GHz. For details, see section 5.1. The input data (spatially unaveraged) are in "Unified Pre-processor" (UPP) BUFR format (see Table 19 in the Appendix), in which all channels have been mapped to the grid of the lower atmosphere sounder (LAS) channels (50-57 GHz). In addition, certain instrument artefacts (warm load intrusions and main reflector emission) have been corrected.

ssmis_main.exe is run as follows:

```
ssmis_main.exe -n namelistfile -i infile [-o outfile || -u outfile_usergrid]
```

The supplied namelist file, *ssmis_main.nl*, assumes that you want to replicate the functionality of SSMIS_UPP averaging module, i.e. to read in a BUFR file, perform some averaging and write to an output BUFR file. So you need to specify -i infile -o outfile. The default namelist is as follows:

```
&mwipp_main_namelist

ssmis_read_upp_bufr = .true.
ssmis_average = .true.
ssmis_map_to_usergrid = .false.
ssmis_write_upp_bufr = .true.
ssmis_write_h5 = .false.
ssmis_write_bufr_usergrid = .false.
ssmis_write_h5_usergrid = .false.

ssmis_averaging_namelist = 'ssmis_averaging.nl'
ssmis_usergrid_namelist = 'user_grid.nl'
ssmis_bufr_namelist = 'mwipp_bufr.nl'     !applies only to user grid output

/
```

If you want to activate other functions, simply set the relevant switch to ".true.". The other functions are:

- Write to hdf5 output file instead of, or in addition to, BUFR
- Map to a user-defined grid (see below)
- Write the mapped data to BUFR or hdf5 output file (named via the –u option). If you want both, the program will automatically append suffixes (.h5 and .bufr) to your output file name

The lower-level namelists (e.g. ssmis_averaging.nl) are optional: if a namelist file does not exist then defaults are used.

Mapping to a user grid

The characteristics of the user grid are set up in *user_grid.nl* (this is the default name). If *usergrid_nlat* and *usergrid_nlon* are greater than zero, a regular, global, lat/lon grid is set up. For example,

```
usergrid_nlat = 480
usergrid_nlon = 640
```

defines a grid with 480 latitude points and 640 longitude points, which is often referred to as N320.

Alternatively, you can use a GRIB file to define the user grid. To do this, set *usergrid_nlat* and *usergrid_nlon* to zero, and specify *external_grid_file*, e.g.

```
external_grid_file='t255_n128_lsm.grib'
```

where the above file is an ECMWF land-sea mask GRIB file on a Reduced Gaussian grid. The reading of this GRIB file, and the nearest-neighbour mapping function are handled by ecCodes.

3.3 amsr2_main.exe

AMSR-2 is a 14-channel microwave imager, with frequencies from 6.9 GHz to 89 GHz. For details, see section 5.2.

amsr2-main.exe is run in the same way as described in 3.2 for SSMIS:

```
amsr2_main.exe -n namelistfile -i infile [-o outfile || -u outfile_usergrid]
```

The main difference is that the input file can be either BUFR or hdf5. The hdf5 input is "native format", i.e. as you would obtain from JAXA (e.g. GW1AM2_201305080433_226D_L1SGBTBR_1110110.h5).

The BUFR definition (for input files, or for output files on the instrument grid) uses a EUMETSAT local sequence 3-50-254, see Table 17 in the Appendix. This sequence is included in ecCodes version 2.7.0 (or later). The use of a local sequence means that any BUFR output files that you produce should have the centre identifier set to 254 (i.e. EUMETSAT), with master table version 19 and local table version 1. These values are therefore hard-coded, they are not set in a namelist.

The other options are invoked in the same way as described in 3.2.

3.4 gmi_main.exe

GMI is a 13-channel microwave imager, with frequencies from 10.65 GHz to 190 GHz. For details, see section 5.3. The channels are divided into two functional groups: 9 lower frequencies, referred to as group "S1" and 4 higher frequencies, group "S2". The two groups have slightly different swath widths, and for BUFR products they are separated into different files. The usage for gmi-main.exe is as before:

```
gmi_main.exe -n namelistfile -i infile [-o "outfiles" || -u outfile_usergrid]
```

But note that two "outfiles" need to be specified under −o. For native hdf5 input, a single input file is used, but if BUFR is used for input then two files must be specified. If you decide to map to a user grid, then the two groups of channels are merged.

A typical native-format input file is 1C.GPM.GMI.XCAL2016-C.20180419-S182139-E182637.V05A.RT-H5.

The BUFR sequence used for input files, or for output files on the instrument grid, is 3-40-012 (see Table 18 in the Appendix). You can specify your own WMO centre identifier via *mwipp_bufr.nl.*

3.5 mwri_main.exe

MWRI (as flown on FY-3A to FY-3D) is a 10-channel microwave imager with channels from 10.65 GHz to 89 GHz. For details, see section 5.4. It is understood that future MWRI instruments may have additional channels.

The usage for mwri-main.exe follows that previously described in 3.2:

```
mwri_main.exe -n namelistfile -i infile [-o outfile || -u outfile_usergrid]
```

A typical native-format (hdf5) input file is FY3C_MWRIA_GBAL_L1_20180613_0715_010KM_MS.HDF.

The BUFR sequence currently used in MWIPP (see Table 16 in the Appendix) follows that defined in AAPP (it does not have a WMO sequence number), but an official sequence may be defined in the future (as has been done for other FY-3 instruments, such as MWHS-2). As with GMI, you can specify your own WMO centre identifier via *mwipp_bufr.nl*.

The software currently supports FY-3C and FY-3D. Note that according to test data from June 2018 the MWRI on FY-3D has more samples per scan (266) than that on FY-3C (254). The existing BUFR format only supports a maximum of 254 samples, therefore the default is that any additional samples are discarded when creating BUFR files.

The hdf5 files for FY-3A and FY-3B follow a different layout; these earlier satellites are not currently supported in MWIPP but could be added if users have a need.

3.6 mwi_main.exe

This executable can process both MWI and ICI. MWI is a 26-channel microwave imager with channels from 18.7 GHz to 183 GHz, while ICI is a 13-channel microwave imager with channels from 183 GHz to 664 GHz. When processed together, they form a 39-channel super-instrument.

The usage for mwi-main.exe is as follows:

```
mwi_main.exe -n namelistfile -i infile_mwi [-j infile_ici] [-o outfile_bufr] [-h
outfile_h5]
```

The program runs the following steps:

1. Reads the input MWI file, converting radiance to BT and interpolating the geolocation tie points
2. Averages MWI spatially (if requested)
3. Reads the input ICI file (if supplied), converting radiance to BT and interpolating the geolocation tie points
4. Averages ICI spatially (if requested)
5. Re-maps all channels from both instruments to a specified MWI feedhorn (the feedhorn for 118 GHz is chosen, see Figure 3). This step incorporates thinning in the along-scan direction, by default to 9.6 km (1 spot in 6)
6. Additional thinning may be applied (along-track and along-scan)
7. Write out output data in BUFR and/or hdf5.

3.7 ici_main.exe

This executable is similar to mwi_main.exe, but it allows processing of ICI on its own. The ICI channels are all mapped to the 183 GHz locations (see Figure 4).

The usage for ici-main.exe is as follows:

```
ici_main.exe -n namelistfile -i infile_ici [-o outfile_bufr] [-h outfile_h5]
```

3.8 Running the standalone hdf5 to BUFR converters

These converters are provided in order to simplify the conversion process for the users – there is no need for top-level namelists. The lower-level routines used are the same ones as are used in sections 3.3, 3.4 and 3.5.

```
amsr2_hdf2bufr.exe -i infile -o outfile
gmi_hdf2bufr.exe -i infile -o "outfile_s1 outfile_s2"
mwri_hdf2bufr.exe -i infile -o outfile
```

WMO centre and sub-centre identifiers can be specified via *mwipp_bufr.nl*, except in the case of AMSR-2 where they are hard-coded (see 3.3).

3.9 Generating quick-look images

A python tool *mwipp_quicklook.py* is provided to generate quick-look imagery for a specified channel of the generic MWIPP hdf5 products, either on the original grid or the user grid. Plate Carrée projection is used. The images are displayed on the screen and/or saved to file. The usage is as follows:

```
usage: mwipp_quicklook.py [-h] -i INPUTFILE [-c CHANNEL] [-r LATLONRANGE]
                          [-s SYMSIZE] [-nd] [-ns]

arguments:
  -h, --help             show this help message and exit
  -i INPUTFILE, --in INPUTFILE
                         input h5 file
  -c CHANNEL, --channel CHANNEL
                         Channel to display, or 'surface' or 'rain'
  -r LATLONRANGE, --range LATLONRANGE
                         geographical limit: 'x0 x1 y0 y1', default '-180 180
                         -90 90'
  -s SYMSIZE, --symsize SYMSIZE
                         symbol size, default 4
  -nd, --no_display      don't display image on screen
  -ns, --no_save         don't save image
```

3.10   Test cases

A range of test cases is included as part of MWIPP, which can be downloaded from the NWP SAF web site. Detailed instructions are provided in README files. For more information, see https://nwp-saf.eumetsat.int/site/software/mwipp/.

3.11   Namelist variables

The list of possible namelist variables is given in Table 2 to Table 7. Note that namelist files can contain comments, but you should leave blank the first character in a comment line.

Table 2: Definition of top-level namelist: *mwipp_main_namelist* in file *[INSTR]_main.nl.* Replace [INSTR] by one of the following: ssmis, amsr2, gmi, mwri. Note that mwi and ici are covered in Table 6 and Table 7 respectively.

Name	Values	Purpose
[INSTR]_readhdf	.true. / .false.	Read an hdf5 input file?
[INSTR]_readbufr	.true. / .false.	Read a BUFR input file?
[INSTR]_write_h5	.true. / .false.	Create hdf5 output file on the native grid?
[INSTR]_writebufr	.true. / .false.	Create BUFR output file on the native grid?
[INSTR]_average	.true. / .false.	Perform spatial averaging?
[INSTR]_map_to_usergrid	.true. / .false.	Map the averaged data to a user-defined grid?
[INSTR]_write_bufr_usergrid	.true. / .false.	Create BUFR output file on the user-defined grid?
[INSTR]_write_h5_usergrid	.true. / .false.	Create hdf5 output file on the user-defined grid?
[INSTR]_averaging_namelist	String	File name of the lower-level averaging namelist (see Table 3)
[INSTR]_usergrid_namelist	String	File name of the lower-level namelist that defines the user grid (see Table 4)
[INSTR]_bufr_namelist	String	File name of the lower-level namelist that gives locally-defined parameters related to BUFR (see Table 5)

Table 3: Definition of the averaging namelist: *mwipp_averaging*

Name	Values	Purpose
sigma	Default 50.0	If *use_boxcar* is false, sigma is the distance, in km, at which the applied Gaussian weight has fallen to exp(-0.5) = 0.61. Set to 50.0 to replicate the functionality of SSMIS_PP. If *use_boxcar* is true, sigma is a simple distance threshold (see 4.2.2 for further explanation).
min_weight	Default 0.01	Allows selection of a suitable averaging domain.
nweights	Default 100	Number of surrounding points to use in the averaging
skipscans	Default 50	Allows you to discard the leading and trailing scans. Set to 50 for SSMIS in order to replicate the functionality of SSMIS_PP. Note that this figure can be reduced if desired (e.g. if you are not dealing with global files with significant overlap between orbits).
rain_threshold	Default 0.1	A spot is flagged as rain if more than this fraction of neighbouring unaveraged spots are flagged as rain. Relies on the rain flagging in the input data; no new rain tests are applied in MWIPP.
rain_averaging_threshold	Default 0.5	Unaveraged BTs are used if more than this fraction of neighbouring unaveraged spots are flagged as rain.
always_average	.true. / .false.	False (default): rain-flagged spots are excluded from the averaging; true: they are included.
use_boxcar	.true. / .false.	False (default): gaussian averaging; true: boxcar with *sigma* as the distance threshold.
ChannelsToBeAveraged	List of channels (default all)	e.g. 1,2,3,4,5. A value of 0 will disable averaging. A value of -1 will select all channels.

Table 4: Definition of the namelist defining the user grid: *usergrid_namelist*

Name	Values	Purpose
usergrid_nlat	integer	Number of latitude points in a regular grid
usergrid_nlon	integer	Number of longitude points in a regular grid
external_grid_file	String	If usergrid_nlat and usergrid_nlon are both zero, this is the name of a GRIB file that defines the user grid.

Table 5: Definition of the namelist for BUFR encoding: *mwipp_bufr*

Name	Values	Purpose
originating_centre	integer	Your centre, as defined in Common Code Table C-1
sub_centre	integer	Sub-centre, see Common Code Table C-12
master_table	integer	BUFR tables version
local_table	integer	Local tables version (normally 0)
local_subtype	integer	The user-defined subtype in Section 1 (any value)
max_subsets	integer	Maximum number of subsets to be encoded in one BUFR message.
use_mwipp_v1_0_sequence	logical	Applies to the generic BUFR sequence only. True: use the old sequence provided in MWIPP v1.0 (default for heritage instruments) False: use the new sequence provided in MWIPP v1.1 (default for MWI/ICI)

Table 6: Definition of the namelist for *mwi_main.exe*

Name	Values	Purpose
mwi_average	.true. / .false.	Perform spatial averaging for MWI?
mwi_averaging_namelist	String	File name of the lower-level averaging namelist (see Table 3)
ici_average	.true. / .false.	Perform spatial averaging for ICI?
ici_averaging_namelist	String	File name of the lower-level averaging namelist (see Table 3)
spot_thinning_remap	integer	Spot thinning at the remap stage. Default 6 (i.e. thin to 9.6 km)
spot_thinning	integer	Spot thinning in the output. Default 1
scan_thinning	integer	Scan thinning in the output. Default 1
bufr_namelist	String	File name of the lower-level namelist that gives locally-defined parameters related to BUFR (see Table 5)

Table 7: Definition of the namelist for ici_main.exe

Name	Values	Purpose
ici_average	.true. / .false.	Perform spatial averaging for MWI?
ici_averaging_namelist	String	File name of the lower-level averaging namelist (see Table 3)
spot_thinning_remap	integer	Spot thinning at the remap stage. Default 4 (i.e. thin to 10.8 km)
spot_thinning	integer	Spot thinning in the output. Default 1
scan_thinning	integer	Scan thinning in the output. Default 1

bufr_namelist	String	File name of the lower-level namelist that gives locally-defined parameters related to BUFR (see Table 5)

4. SOFTWARE DESCRIPTION

4.1 General comments

This section provides a "bottom-up" description of the software components.

All the software components are written in Fortran90, with the exception of the hdf5-handling routine for AMSR-2, which uses some Fortran2003 constructs to handle variable-length hdf5 strings.

Subroutines are either instrument-specific (in which case they are placed in the libxxxx directories, where xxxx is the instrument) or they are generic (in which case they are placed in libcommon).

Subroutines and data are contained within "Modules" under the following circumstances:
- To facilitate sharing of variables and arrays (e.g. the instrument data arrays)
- Where a subroutine takes a deferred-length variable as an argument (avoids the necessity to use interface blocks)
- To group together subroutines that have a related function (e.g. those related to the user-defined grid)

The sections below provide a brief functional description of each subroutine. For more details, please see the comments the source code.

4.2 Generic subroutines and modules (in *src/libcommon*)

4.2.1 *hpsort.f90*

A public-domain Heapsort subroutine to sort the elements of an array into ascending order, and modified to return an index to the original sample points, as well as the sorted points.

Reference: http://jean-pierre.moreau.pagesperso-orange.fr/Fortran/sort3_f90.txt

4.2.2 *mwipp_averaging_mod.f90*

A module containing routines to compute weights for averaging and then to smooth a BT field. The method is the same as was used in SSMIS_PP.

Contains the following subroutines:

SUBROUTINE read_averaging_namelist(namelist_file)
    Reads the namelist that controls averaging parameters (e.g. the effective distance). See section 3.11 for more details.

SUBROUTINE compute_distsq(xref,yref,zref,xtest,ytest,ztest,distsq)
    Compute the square of the distances between 2D swath coordinates (xref, yref, zref) and a single test point (xtest, ytest, ztest)

SUBROUTINE compute_weights(x,y,z,sigma,thresh,nweights,use_boxcar,dspot,dscan,&
weights_out)

> Compute the weights to be used for smoothing. If *use_boxcar* is false, the weights, w_i , at a distance r_i from a reference spot, are defined as

$$w_i = \exp\left(-\frac{r_i^2}{2\sigma^2}\right)$$

> where σ is the averaging distance (input variable *sigma*). If *use_boxcar* is true, the weights are 1.0 if r_i is less than or equal to w_i and 0.0 otherwise. The subroutine examines a region in the centre of the 2D input data field (coordinates *x, y, z*). First it looks at the mid-swath points and finds how many scans are needed before the computed Gaussian weight has dropped below a threshold. Then it examines all the swath points in that range of scan lines, computes the weights and sorts them into descending order. The number of weights retained is *nweights*. Finally, the spot and scan offsets are returned (*dspot, dscan*), together with the weights.

SUBROUTINE modify_btemps(bt,surface,rain,nweights,dspot,dscan,weights,&
nchans_total,channels,rain_threshold,rain_averaging_threshold,always_average,&
bt_out,surface_out,rain_out)

> Smooths a BT field using the pre-computed weights. A channel list (*channels*) defines which channels are to be averaged. For each reference spot, its neighbours are examined (a total of *nweights* spots) and a weighted average BT is computed. If *always_average* is true then rain-contaminated spots are included, otherwise they are excluded. Spots where the brightness temperature is negative (i.e. missing) are excluded. If more than a defined fraction of spots are flagged as rain (configurable, via *rain_ threshold*, typically 0.1), the output flag is set as rain. If *always_average* is false and too many neighbouring spots are rain-contaminated (configurable, via *rain_averaging_threshold*, typically 0.5) then the input BT is passed unchanged to the output. If too many spots are missing (>25%) then the output BT is set to missing. If any of the spots used in the averaging have a different surface type from the reference spot, the output surface type is set to 6, meaning "coast". Note that the rain and/or surface flags may be missing for some instruments.

4.2.3 *mwipp_data_mod.f90*

A module containing generic data arrays used in MWIPP (*mwipp_bt, mwipp_lat*, etc.). Contains

SUBROUTINE allocate_mwipp_data(nchan,nspot,nscan,allocate_angles)

> Note that satellite and solar zenith and azimuth angles are not available for SSMIS, so do not need to be allocated.

SUBROUTINE deallocate_mwipp_data

4.2.4 *mwipp_latlon_mod.f90*

Conversion between latitude/longitude and Cartesian coordinates, assuming a spherical earth. Contains

SUBROUTINE latlon_to_xyz(lat,lon,x,y,z)

SUBROUTINE latlon_to_xyz_1d(lat,lon,x,y,z)

4.2.5 *mwipp_map_to_usergrid_mod.f90*

SUBROUTINE get_usergrid(namelist_file,lat_user,lon_user,igrib,npoints)
> Read the namelist that specifies the type of user-defined grid to be used: either defined in a GRIB file or defined from the number of latitude/longitude points

SUBROUTINE project_onto_usergrid(lat_in,lon_in,igrib,npoints,spots,scans)
> Project swath data onto a regular, global lat/lon grid. If two or more points in the swath project onto the same grid point, use the one that is closest to the grid point. The output *spots* and *scans* arrays give the swath coordinates for each point in the user grid.

SUBROUTINE project_onto_latlon_grid(lat_in,lon_in,nlat,nlon,spots,scans)
> Project swath data onto a grid defined in GRIB, using the ecCodes routine *codes_grib_find_nearest*. If two or more points in the swath project onto the same grid point, use the one that is closest to the grid point. The output *spots* and *scans* arrays give the swath coordinates for each point in the user grid.

SUBROUTINE project_simple_thin(lat_in,lon_in,spot_thinning,scan_thinning,npoints,spots,scans)
> Convert 2D arrays into a generic 1D array structure that is a thinned version of the original grid.

SUBROUTINE generate_latlon_grid(nlat,nlon,lat_out,lon_out)
> Called from *get_usergrid*: sets up a regular lat/lon grid. The output arrays are 1-D.

SUBROUTINE externally_defined_grid(filename,lat_out,lon_out,igrib,npoints)
> Called from *get_usergrid*: opens and reads the GRIB file, using calls to ecCodes.

4.2.6 *mwipp_write_bufr_usergrid_mod.f90*

SUBROUTINE write_bufr_usergrid(outfile_name,bufr_namelist,&
  spots_user,scans_user,lat_user,lon_user,bt_user,satzen_user,sataz_user,&
  startchan,endchan)
> Writes a BUFR output file using the generic BUFR sequence shown in Table 14 or Table 15 in the Appendix. By default, MWI and ICI are encoded with the second sequence and the other instruments with first (for continuity), but this can be changed using the *use_mwipp_v1_0_sequence* variable in the BUFR namelist (set *.false.* to use the newer sequence for heritage instruments). Output latitude and longitude are those of the chosen samples. Provision is made for different groups of channels to have different sets of zenith and azimuth angles (needed for GMI, not used for other instruments).

4.2.7 *mwipp_write_hdf.f90*

SUBROUTINE mwipp_write_hdf(outfile)
> Saves the generic MWIPP data (defined in *mwipp_data_mod*) to an output hdf5 file. The file format is self-describing.

4.2.8 mwipp_write_hdf_usergrid_mod.f90

SUBROUTINE write_hdf_usergrid(outfile,spots_user,scans_user,lat_user,lon_user,bt_user,&
satzen_user,sataz_user,startchan,endchan)
> Saves the re-mapped MWIPP data to an output hdf5 file. The file format is self-describing. Output latitude and longitude are those of the chosen samples. Provision is made for different groups of channels to have different sets of zenith and azimuth angles (needed for GMI).

4.3 SSMIS subroutines and modules (in *src/libssmis*)

4.3.1 ssmis_read_bufr.f90

SUBROUTINE ssmis_read_bufr(infile_name)
> Reads in a UPP BUFR file and stores selected information in the MWIPP data structure.

4.3.2 ssmis_readwrite_bufr.f90

SUBROUTINE ssmis_readwrite_bufr(infile_name,outfile_name)
> Reads in a UPP BUFR file, replaces the brightness temperatures, rain flags and surface flags with the values already in memory (i.e. after averaging) then writes to an output UPP BUFR file.

4.4 AMSR-2 subroutines and modules (in *src/libamsr2*)

4.4.1 amsr2_data_mod.f90

A module defining AMSR-2 data arrays. These are closely related to the hdf5 native format datasets. Contains

SUBROUTINE allocate_amsr2_data
SUBROUTINE deallocate_amsr2_data

4.4.2 amsr2_read_bufr.f90

SUBROUTINE amsr2_read_bufr(infile_name)
> Reads BUFR data with sequence 3-50-254. This is a EUMETSAT local sequence, based on master table 19 and local table 1.

4.4.3 amsr2_write_bufr.f90

SUBROUTINE amsr2_write_bufr(outfile)
> Writes BUFR data with sequence 3-50-254. This is a EUMETSAT local sequence, based on master table 19 and local table 1. The 'bufrHeaderCentre' is hard-coded as 254 (EUMETSAT).

4.4.4 amsr2_read_hdf.f90

SUBROUTINE amsr2_read_hdf(infile)
>   Read AMSR-2 hdf5 data into memory.

4.4.5 amsr2_data_to_mwipp.f90

SUBROUTINE amsr2_data_to_mwipp
>   Transfer AMSR-2 data held in memory to the generic MWIPP structure. Note that AMSR-2 native format has higher resolution at 89 GHz; this routine uses the resolution of the lower-frequency channels, and only uses 1 spot in 4 at 89 GHz.

4.5 GMI subroutines and modules (in *src/libgmi*)

4.5.1 gmi_data_mod.f90

A module defining GMI data arrays. These are closely related to the hdf5 native format datasets. Contains

SUBROUTINE allocate_gmi_data
SUBROUTINE deallocate_gmi_data

4.5.2 gmi_read_bufr.f90

SUBROUTINE gmi_read_bufr(infiles)
>   Reads BUFR data with sequence 3-40-012. Two input files need to be specified in the "infiles" string, one for the lower frequencies (S1 channels) and one for the high frequencies (S2 channels).

4.5.3 gmi_write_bufr.f90

SUBROUTINE amsr2_write_bufr(outfiles,bufr_namelist)
>   Writes BUFR data with sequence 3-40-012. Two output files need to be specified in the "outfiles" string, one for the lower frequencies (S1 channels) and one for the high frequencies (S2 channels).

4.5.4 gmi_read_hdf.f90

SUBROUTINE amsr2_read_hdf(infile)
>   Read GMI hdf5 data into memory.

4.5.5 gmi_data_to_mwipp.f90

SUBROUTINE gmi_data_to_mwipp(channelgroup)
>   Transfer AMSR-2 data held in memory to the generic MWIPP structure. "channelgroup" is an integer, 1 or 2.

4.6 MWRI subroutines and modules (in *src/libmwri*)

4.6.1 *mwri_data_mod.f90*

A module defining MWRI data arrays. These are closely related to the hdf5 native format datasets. Contains

  SUBROUTINE allocate_mwri_data
  SUBROUTINE deallocate_ mwri_data

4.6.2 *mwri_read_bufr.f90*

  SUBROUTINE mwri_read_bufr(infile_name)
      Reads BUFR data. It assumes use of the AAPP BUFR sequence shown in Table 16 (originally devised by CMA). Note that in 2018 a new sequence was published for the FY-3 sounder instruments; the routine will be updated if a similar sequence is defined for MWRI.

4.6.3 *mwri_write_bufr.f90*

  SUBROUTINE mwri_write_bufr(outfile,bufr_namelist)
      Writes BUFR data, using the AAPP BUFR sequence shown in Table 16 (originally devised by CMA). Note that in 2018 a new sequence was published for the FY-3 sounder instruments; the routine will be updated if a similar sequence is defined for MWRI.

4.6.4 *mwri_read_hdf.f90*

  SUBROUTINE mwri_read_hdf(infile)
      Read MWRI hdf5 data into memory.

4.6.5 *mwri_data_to_mwipp.f90*

  SUBROUTINE mwri_data_to_mwipp
      Transfer MWRI data held in memory to the generic MWIPP structure.

4.7   MWI subroutines and modules (in *src/libmwi*)

4.7.1 *mwi_data_mod.f90*

A module defining MWI data arrays. These are closely related to the netCDF native format datasets. Contains

  SUBROUTINE allocate_mwi_data
  SUBROUTINE deallocate_mwi_data
  SUBROUTINE allocate_mwi_bt
  SUBROUTINE deallocate_mwi_bt

4.7.2 *mwi_read_netcdf.f90*

SUBROUTINE mwi_read_netcdf(infile_mwi)
    Read MWI netCDF data into memory.

4.7.3 *mwi_read_hdf5.f90*

SUBROUTINE mwi_read_hdf5(infile_mwi)
    Used for variable length strings that can't be handled with NetCDF4 API. Requires
    Fortran2003 support in compiler and hdf5 library.

4.7.4 *mwi_bt_convert.f90*

SUBROUTINE mwi_bt_convert
    Convert MWI radiance to brightness temperature.

4.7.5 *mwi_interpolate_geo.f90*

SUBROUTINE mwi_interpolate_geo
    Interpolate MWI geolocation tie points supplied in the L1B input.

4.7.6 *mwi_data_to_mwipp.f90*

SUBROUTINE mwi_data_to_mwipp
    Transfer MWI data held in memory to the generic MWIPP structure

4.7.7 *remap_mwi_and_ici.f90*

SUBROUTINE remap_mwi_and_ici (mwi_reference_band, ici_present, spotthin)
    Re-map MWI and ICI (if present) to a common MWI grid, in the MWIPP data area.

4.8 ICI subroutines and modules (in *src/libici*)

4.8.1 *ici_data_mod.f90*

A module defining ICI data arrays. These are closely related to the netCDF native format datasets.
Contains

SUBROUTINE allocate_ici_data
SUBROUTINE deallocate_ici_data
SUBROUTINE allocate_ici_bt
SUBROUTINE deallocate_ici_bt

4.8.2 *ici_read_netcdf.f90*

SUBROUTINE ici_read_netcdf(infile_ici)
    Read ICI netCDF data into memory.

4.8.3   ici_read_hdf5.f90

SUBROUTINE ici_read_hdf5(infile_ici)
> Used for variable length strings that can't be handled with NetCDF4 API. Requires Fortran2003 support in compiler and hdf5 library.

4.8.4   ici_bt_convert.f90

SUBROUTINE ici_bt_convert
> Convert MWI radiance to brightness temperature.

4.8.5   ici_interpolate_geo.f90

SUBROUTINE mwi_interpolate_geo
> Interpolate MWI geolocation tie points supplied in the L1B input.

4.8.6   ici_data_to_mwipp.f90

SUBROUTINE mwi_data_to_mwipp
> Transfer MWI data held in memory to the generic MWIPP structure

4.8.7   remap_ici.f90

SUBROUTINE remap_ici (ici_reference_band, spotthin)
> Re-map ICI channels to a common ICI grid, in the MWIPP data area.

4.9 Main programs (in *src/bin*)

4.9.1   ssmis_main.f90

This program includes the functionality of the NWP SAF "SSMIS UPP Averaging Module". It comprises the following steps. All except the first are optional:

1. Read command line options and namelists (see section 3.2 for usage instructions)
2. Read the input "UPP" BUFR file into memory
3. Convert latitudes and longitudes into Cartesian coordinates
4. Compute the averaging weights
5. Carry out averaging, modifying the BTs, the rain flags and the surface flags.
6. Write the data to an hdf5 file on the original grid
7. Define the user grid (if applicable)
8. Project the BTs onto the user-defined grid
9. Write BUFR and/or hdf5 on the user grid
10. Re-read the "UPP" BUFR file and create an output file with modified BTs.

4.9.2   amsr2_main.f90 and mwri_main.f90

All except the first step are optional:

1. Read command line options and namelists (see section 3.3 and 3.5 for usage instructions)
2. Read the input file into memory – either a BUFR file or a native-format hdf5 file
3. Write the unaveraged data to BUFR
4. Transfer the instrument-specific data structures to the MWIPP generic structure
5. Convert latitudes and longitudes into Cartesian coordinates
6. Compute the averaging weights
7. Carry out averaging, modifying the BTs
8. Write the data to an hdf5 file on the original grid
9. Define the user grid (if applicable)
10. Project the BTs onto the user-defined grid
11. Write BUFR and/or hdf5 on the user grid

4.9.3 gmi_main.f90

All except the first step are optional:

1. Read command line options and namelists (see section 3.4 for usage instructions)
2. Read the input file into memory – either a BUFR file or a native-format hdf5 file
3. Write the unaveraged data to BUFR
4. Define the user grid (if applicable)
   For each group of channels (S1 or S2):
       a. Transfer the instrument-specific data structures to the MWIPP generic structure
       b. Convert latitudes and longitudes into Cartesian coordinates
       c. Compute the averaging weights
       d. Carry out averaging, modifying the BTs
       e. Write the data to an hdf5 file on the original grid
       f. Project the BTs onto the user-defined grid. Temporarily store in "bt_user" array
5. Write BUFR and/or hdf5 on the user grid

4.9.4 HDF5 to BUFR converters: amsr2_hdf2bufr.f90, gmi_hdf2bufr.f90, mwri_hdf2bufr.f90

1. Read command line options and namelists (see section 3.8 for usage instructions)
2. Read the input hdf5 file into memory
3. Where applicable (i.e. GMI and MWRI), read the *mwipp_bufr.nl*, which specifies the originating centre, etc.
4. Write to the output BUFR file or files.

4.9.5 mwi_main.f90

1. Read command line options and namelists (see section 3.6 for usage instructions)
2. Read the MWI netCDF file into memory
3. Convert MWI radiance to brightness temperature
4. Interpolate the MWI geolocation tie points
5. Convert latitudes and longitudes into Cartesian coordinates
6. Compute the averaging weights
7. Carry out averaging, modifying the BTs, the rain flags and the surface flags.

8. Repeat steps 2 to 7 for ICI, if present
9. Map data for both instruments to a common grid (based on the MWI 118 GHz channels)
10. Transfer to the generic MWIPP data structures. In this case the "user grid" is simply a thinned version of the MWI grid.
11. Write BUFR if requested, using the BUFR sequence of Table 15
12. Write hdf5 if requested.

4.9.6 **ici_main.f90**

1. Read command line options and namelists (see section 3.6 for usage instructions)
2. Read the ICI netCDF file into memory
3. Convert ICI radiance to brightness temperature
4. Interpolate the ICI geolocation tie points
5. Convert latitudes and longitudes into Cartesian coordinates
6. Compute the averaging weights
7. Carry out averaging, modifying the BTs, the rain flags and the surface flags.
8. Map data for the different feedhorns to a common grid (based on the ICI 183 GHz channels)
9. Transfer to the generic MWIPP data structures. In this case the "user grid" is simply a thinned version of the ICI grid.
10. Write BUFR if requested, using the BUFR sequence of Table 15
11. Write hdf5 if requested.

5. SCIENTIFIC DESCRIPTION

5.1 SSMIS

SSMIS comprises four different groups of channels, each with different native spatial resolution and sampling: see Table 8. These are referred to as Imager (91 GHz and higher), Environmental (19.35 to 37 GHz), Lower Atmosphere Sounding (LAS, 50.3 to 59.4 GHz and the lowest peaking 60.8 GHz channel) and Upper Atmosphere Sounding (UAS, 60.8 to 63.3 GHz).

Table 8: SSMIS channels

Channel number	Frequency (GHz)	RF Band-width (MHz)	Polaris-ation	3dB Footprint (km)	Sample Spacing (km)	NEΔT at 305K (K)	NEΔT Spec. (K)
12	19.35	355.0	H	46.5x 73.6	25	0.35	0.7
13	19.35	356.7	V	46.5x 73.6	25	0.34	0.7
14	22.235	407.5	V	46.5x 73.6	25	0.45	0.7
15	37.0	1615.	H	31.2 x 45.0	25	0.26	0.5
16	37.0	1545.	V	31.2 x 45.0	25	0.22	0.5
1	50.3	380.0	H	37.7 x 38.8	37.5	0.21	0.4
2	52.8	388.8	H	37.7 x 38.8	37.5	0.20	0.4
3	53.596	380.0	H	37.7 x 38.8	37.5	0.21	0.4
4	54.40	382.5	H	37.7 x 38.8	37.5	0.20	0.4
5	55.50	391.3	H	37.7 x 38.8	37.5	0.22	0.4
6	57.29	330.0	RC	37.7 x 38.8	37.5	0.26	0.5
7	59.4	238.8	RC	37.7 x 38.8	37.5	0.25	0.6
24	$60.792668 \pm 0.357892 \pm 0.050$	106.0	RC	37.7 x 38.8	37.5	0.38	0.7
23	$60.792668 \pm 0.357892 \pm 0.016$	29.28	RC	75.2 x 75.0	75	0.37	0.6
22	$60.792668 \pm 0.357892 \pm 0.0055$	10.48	RC	75.2 x 75.0	75	0.58	1.0
21	$60.792668 \pm 0.357892 \pm 0.002$	5.16	RC	75.2 x 75.0	75	0.86	1.8
19	63.283248 ± 0.285271	2.72	RC	75.2 x 75.0	75	1.23	2.4
20	60.792668 ± 0.357892	2.70	RC	75.2 x 75.0	75	1.18	2.4
17	91.655 ± 0.9	2836.	V	13.2 x 15.5	12.5	0.19	0.9
18	91.655 ± 0.9	2822.	H	13.2 x 15.5	12.5	0.19	0.9
8	150.0 ± 1.25	3284.	H	13.2 x 15.5	12.5	0.53	0.88
9	183.31 ± 6.6	1025.	H	13.2 x 15.5	12.5	0.56	1.2
10	183.31 ± 3.0	2038.	H	13.2 x 15.5	12.5	0.39	1.0
11	183.31 ± 1.0	3052.	H	13.2 x 15.5	12.5	0.38	1.25

Several different file formats are in use for SSMIS level 1 data, including:
- Temperature data records (TDR) in binary format – raw antenna temperatures plus housekeeping
- Sensor data records file (SDR) in binary format – processed brightness temperatures, with spillover corrections etc., but no correction for artefacts like solar intrusions of reflector emission.

- BUFR versions of the TDR product, separate files for each of the four channel groups. Sequence 3-10-025 is used for each channel group. One spot per subset.

- A BUFR version of the SDR format: the data are distributed by EUMETSAT. Separate files for each of the four channel groups. Uses a BUFR sequence in which a complete swath is contained within one subset, with the different fields of view being specified by replication. This is unusual.

- Unified Pre-processor (UPP) BUFR files: all channels are mapped to the LAS grid. Sequence 3-10-025. One spot per subset.

For NWP purposes, the most useful of these level 1 products is the last one (UPP), for two reasons: (i) the re-mapping to the LAS grid is convenient for assimilation because all channels are co-located; (ii) instrument artefacts, namely warm load intrusions and reflector emission, have been corrected. See Bell et al., 2008 and Swadley et al., 2008. The correction of these instrumental artefacts, and the remapping to LAS grid, are normally performed by the UPP software run by the Naval Research Laboratory (NRL). The MWIPP package does *not* attempt to replicate these instrument-specific functions.

The original NWP SAF SSMIS_PP deliverable (Bell, 2006) ingested the TDR BUFR files. First, the four channel groups were mapped to the LAS grid. Next, the data were averaged. Finally, the solar intrusions were flagged and the reflector emission is corrected. The SSMIS_PP only supported DMSP F16, and is now largely obsolete.

In 2010 the "SSMIS UPP Averaging Module" was released as a standalone package, but formally part of SSMIS_PP. Its primary function is to average the data in order to reduce instrument noise, which is critical for sounding channels. The averaging function is identical to that used in SSMIS_PP. However, the input is a single UPP BUFR file (on the LAS grid), i.e. no re-mapping is performed in the package. This software is widely used in NWP systems.

For the MWIPP package, the software has been completely re-written. The scientific basis of the averaging is the same as before: for each position across the swath, a set of weights is used to compute weighted average BT from up to 200 neighbouring samples. A Gaussian weighting is normally used, as described in section 4.2.2.

The input SSMIS data includes a rain flag which has been generated by the UPP. Two modes are available in MWIPP for treating this rain flag:

- The first mode is based on the behaviour of the original SSMIS_PP package. Rain-contaminated spots are *not* used in the averaging of neighbouring spots. If a high percentage (nominally >10%) of a spot's neighbours are flagged as rain then the output rain flag is set for that spot. If this percentage is greater than a second threshold (nominally 50%) then unaveraged BTs are used for that spot. Note that this behaviour differs slightly from that in the original SSMIS_PP: in that software, unaveraged BTs were only used if *all* the 200 neighbouring spots were rainy; this gives a misleading indication of the BT fields in precipitating areas.

- In the second mode, averaging is always performed, regardless of the rain flag. The output rain flag is set as before, so the user still has the ability to reject rain-contaminated spots. But valid BT fields are generated for channels that are not sensitive to rain (e.g. stratospheric or mesospheric channels).

The surface type (defined in the UPP) is also examined and is set to "mixed" in the output file if any of the spots used in the averaging have a different surface type from the reference spot. Finally, looking at each channel, if more than 25% of the neighbouring spots have "missing" BT then the

BT for the spot itself is set to missing (this test was originally introduced because of missing BTs at the edge of the swath for some MWRI channels).

The ability to map to a user-defined grid has been added, see details in section 3. Only a nearest-neighbour mapping has been implemented. In principle, more sophisticated techniques could be considered, e.g. Backus-Gilbert. But this would increase the run-time (probably significantly), and no NWP-related requirement for a more complex mapping has been identified (see the review of requirements in RD-1).

Other changes in MWIPP compared with the SSMIS_UPP are:
- The weights are computed during execution of the program, using the central portion of the supplied input data file, rather than being obtained via a data file. This allows the user to alter the averaging distance, if required.
- The *ecCodes* BUFR/GRIB package replaces the Met Office's "MetDB BUFR" software. As far as BUFR is concerned, this is a technical change rather than a scientific one, but the additional GRIB functionality facilitates mapping to a user-defined grid.

Note that the incoming SSMIS data does not include solar angles. Satellite zenith and azimuth angles are included, but not for each spot: they are on a coarse grid via the "base point" positions and angles. These base point angles are not used in the MWIPP processing, but are copied to the output UPP BUFR file.

5.2 AMSR-2

The AMSR-2 channel channel characteristics are shown in Table 9. See also Shimoda (2011).

Table 9: AMSR-2 channels (from WMO Oscar[5] database)

Central frequency (GHz)	Bandwidth (MHz)	Polarisations	NEΔT	IFOV	Pixel
6.925	350	V, H	0.3 K	35x62 km	10x10 km
7.3	350	V, H	0.3 K	35x62 km	10x10 km
10.65	100	V, H	0.6 K	24x42 km	10x10 km
18.7	200	V, H	0.6 K	14x22 km	10x10 km
23.8	400	V, H	0.6 K	11x19 km	10x10 km
36.5	1000	V, H	0.6 K	7x12 km	10x10 km
89.0	3000	V, H	1.1 K	3x5 km	5x5 km

AMSR-2 data, in EUMETSAT-defined BUFR format, are used (or are planned to be used) at several NWP centres. The hdf5 to BUFR conversion capability implemented in MWIPP is intended to assist those centres that do not have access to EUMETCast, and therefore needed to process the original hdf5 product.

Note the wide range of IFOV sizes in Table 9. Unlike SSMIS, these are all window channels, therefore the benefits of averaging for noise reduction are less clear. However, if the averaging function of MWIPP is used (e.g. 50km Gaussian averaging), this will tend to achieve a more uniform field of view. Note, however, that MWIPP is not designed to achieve strict equality in beam widths: to do that would require different treatment for the different channels, and is beyond the current scope of the package.

---

[5] https://www.wmo-sat.info/oscar/instruments

Note that by convention the BUFR product is unaveraged. To output averaged data on the original grid, you can configure MWIPP to output in hdf5. Alternatively, averaged data can be mapped to a user-defined grid (as described for SSMIS).

The 89 GHz channel is sampled at four times the rate of the other channels (there are two feedhorns, each sampled every 5km along-scan, compared with 10km for the other channels). The BUFR product retains this increased sampling, but for the other parts of MWIPP the 89 GHz samples are thinned to the same sampling rate as the other channels.

The input h5 files include satellite and solar zenith and azimuth angles. The solar angles in the input files are defined by JAXA to be *relative to the satellite angles*, according to Fig 4.2-1 of the AMSR2 Level 1A product format specification[6]. In MWIPP v1.0 and v1.1 these were incorrectly represented in the output files, but this error was subsequently corrected and the solar angles that appear in BUFR products from MWIPP v1.2 are defined to be the zenith angle and azimuth angle of the sun as seen from the ground, relative to local vertical and due north respectively.

There is no rain flag or surface type categorisation (unlike SSMIS), therefore the MWIPP averaging uses all available spots. If a requirement is identified for better rain and surface discrimination, consideration could be given for a future enhancement of MWIPP to add a surface atlas and rain detection.

5.3 GMI

The GMI channel characteristics are shown in Table 10. See also Draper et al., 2015.

Table 10: GMI channels (from WMO Oscar database)

Central frequency (GHz)	Bandwidth (MHz)	Polarisations	NEΔT	IFOV	Pixel
10.65	100	V, H	0.96 K	19x32 km	12x13.4 km
18.7	200	V, H	0.84 K	11x18 km	6.0x13.4 km
23.8	400	V	1.05 K	9.2x15 km	6.0x13.4 km
36.5	1000	V, H	0.65 K	8.6x14 km	6.0x13.4 km
89.0	6000	V, H	0.57 K	4.4x7.2 km	3.0x13.4 km
166.0	4000	V, H	1.5 K	4.4x7.2 km	3.0x13.4 km
183.31 ± 7	2000	V	1.5 K	4.4x7.2 km	3.0x13.4 km
183.31 ± 3	2000	V	1.5 K	4.4x7.2 km	3.0x13.4 km

The four highest frequency channels (166 GHz and above) are treated as a separate sub-instrument and have their own geolocation information in the "1C" product. The swath is wider for the low-frequency group, see Figure 2.

---

[6] https://gportal.jaxa.jp/gpr/assets/mng_upload/GCOM-W/AMSR2_Level1_Product_Format_EN.pdf

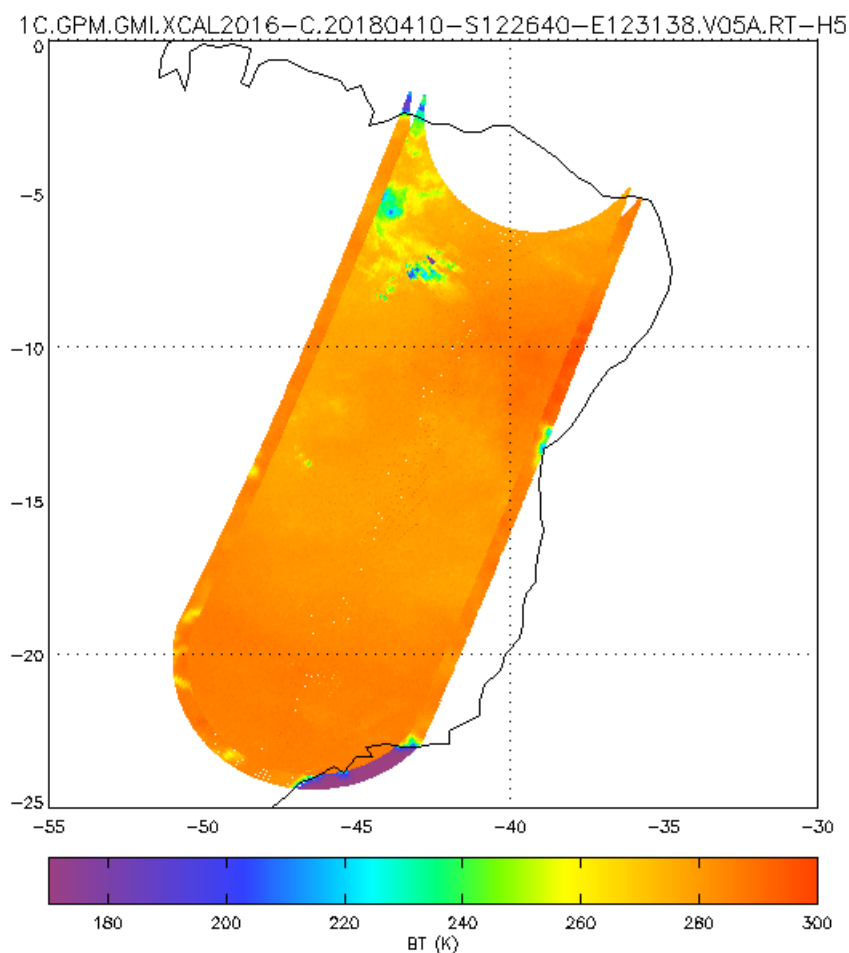1C.GPM.GMI.XCAL2016-C.20180410-S122640-E123138.V05A.RT-H5

Figure 2: GMI images at 10.65 GHz and 166 GHz shown super-imposed. The swath is wider at 10.65 GHz.

JAXA also produce a "1C-R" product in which the high frequencies have been mapped to the spot positions of the low frequencies. This product can be used as input to MWIPP if preferred.

The input h5 files include satellite zenith angle, but not satellite azimuth or solar angles. As with AMSR-2, there is no rain flag or surface type categorisation, therefore the MWIPP averaging uses all available spots.

5.4 MWRI

The MWRI channel characteristics are shown in Table 11. See also Yang et al, 2011.

Table 11: MWRI channels (from WMO Oscar database)

Central frequency (GHz)	Bandwidth (MHz)	Polarisations	NEΔT	IFOV	Pixel
10.65	180	V, H	0.5 K	51 x 85 km	40 x11.2 km
18.7	200	V, H	0.5 K	30 x 50 km	40 x11.2 km
23.8	400	V, H	0.5 K	27 x45 km	20 x11.2 km

36.5	400	V, H	0.5 K	18 x30 km	20 x11.2 km
89	3000	V, H	0.8 K	9 x15 km	10 x11.2 km

The channels are co-located, which simplifies the interpretation of the data. For some channels, the outermost one or two spots can have a missing BT; the MWIPP software takes this into account.

The MWRI data files include a surface category but not a rain flag. The MWIPP averaging uses all available spots, flagging spots as mixed surface type if appropriate.

5.5 MWI and ICI

MWI and ICI are, respectively, the MicroWave Imager and Ice Cloud Imager on the Metop-SG-B satellite series (see Bell et al., 2017). Channel characteristics are given in Table 12 and Table 13.

Table 12: MWI channels (from Accadia et al., 2020)

Central frequency (GHz)	Bandwidth (MHz)	Polarisations	NEΔT (K)	IFOV (km)
18.7	200	V, H	0.8	50
23.8	400	V, H	0.7	50
31.4	200	V, H	0.9	30
50.3	400	V, H	1.1	30
52.61	400	V, H	1.1	30
53.24	400	V, H	1.1	30
53.75	400	V, H	1.1	30
89.0	4000	V, H	1.1	10
118.7503±3.2	2x500	V	1.3	10
118.7503 ± 2.1	2x400	V	1.3	10
118.7503 ± 1.4	2x400	V	1.3	10
118.7503 ± 1.2	2x400	V	1.3	10
165.5 ± 0.75	2x1350	V	1.2	10
183.31 ± 7.0	2x2000	V	1.2	10
183.31 ± 6.1	2x1500	V	1.2	10
183.31 ± 4.9	2x1500	V	1.2	10
183.31 ± 3.4	2x1500	V	1.2	10
183.31 ± 2.0	2x1500	V	1.3	10

Table 13: ICI channels (from Accadia et al., 2020)

Central frequency (GHz)	Bandwidth (MHz)	Polarisations	NEΔT (K)	IFOV (km)
183.31 ± 7.0	2x2000	V	0.8	16
183.31 ± 3.4	2x1500	V	0.8	16
183.31 ± 2.0	2x1500	V	0.8	16
243.2 ± 2.5	2x3000	V, H	0.7	16
325.15 ± 9.5	2x3000	V	1.2	16
325.15 ± 3.5	2x4000	V	1.3	16
325.15 ± 1.5	2x1600	V	1.5	16

448 ± 7.2	2x3000	V	1.4	16
448 ± 3.0	2x2000	V	1.6	16
448 ± 1.4	2x1200	V	2.0	16
664 ± 4.2	2x5000	V, H	1.6	16

The two instruments will have the same viewing geometry (incidence angle close to 53°) and the same scan azimuth range (±65°), hence the same swath width. They will be counter-rotating with respect to each other.

We note that:
- MWI has channels in both the oxygen sounding bands at 50-54 GHz and 118 GHz. Although the main purpose is to provide information on cloud and precipitation, rather than temperature profiling, it is likely that some SSMIS-like spatial averaging may be required on these channels, primarily to reduce noise.
- Both instruments carry channels at 183 GHz. This will facilitate accurate intercomparison.
- ICI senses in three water vapour bands (183, 325 and 448 GHz), hence the possibility of retrieving information on the vertical distribution of ice.

Products to be disseminated in near-real-time by EUMETSAT Central Facilities will include level 1B radiances (the primary focus of MWIPP) and level 2 products including Liquid Water Path and Ice Water Path. The radiances are expected to be disseminated in both BUFR and netCDF.

Each instrument has multiple feedhorns, as shown in Figure 3 and Figure 4. At level 1B, the data from the different feedhorns are not co-aligned, therefore it is for the MWIPP software to map all feedhorns to a common reference. We have chosen to use the 118 GHz feedhorn of MWI, since that is in the centre of the MWI cluster.
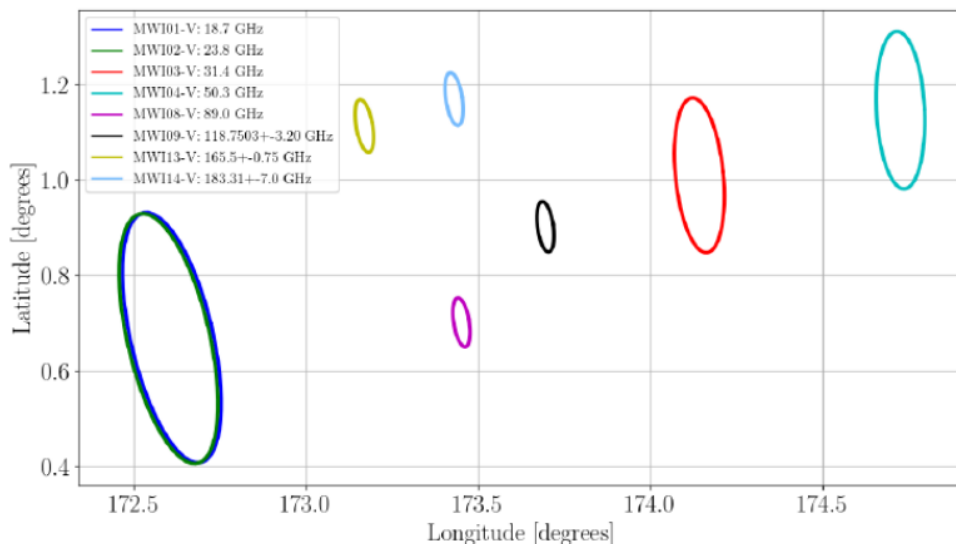

Figure 3: Fields of view for the MWI feedhorns (from Accadia et al., 2020)
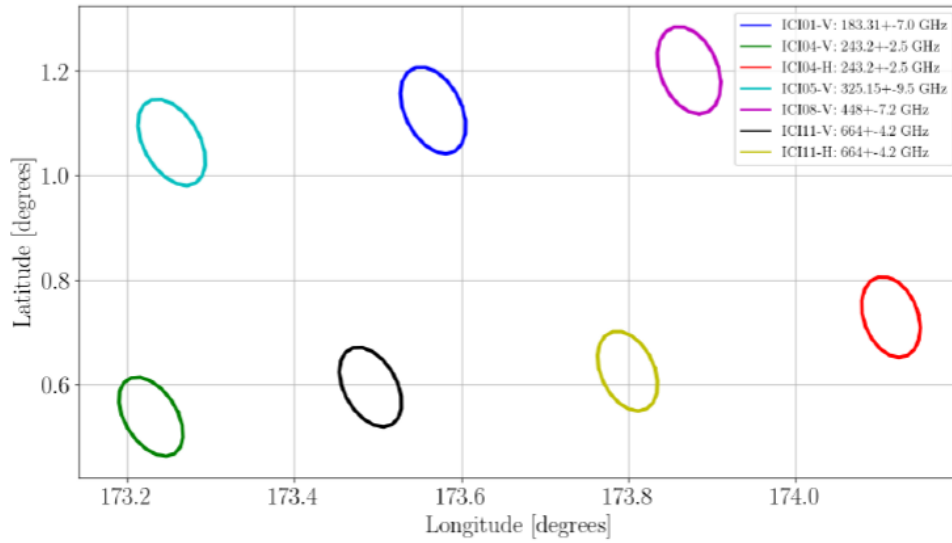
Figure 4 Fields of view for the ICI feedhorns (from Accadia et al., 2020)

We should also note that at level 1B all channels are highly over-sampled in the scan direction (1.6km spacing for MWI, 2.7km for ICI). This is primarily to facilitate re-mapping, but it means that the level 1B product is not expected to be optimal for direct assimilation into NWP.

The MWI/ICI processing chain in MWIPP therefore requires the following steps:

1. Ingest MWI and ICI data files
2. Perform spatial smoothing for each instrument, as implemented for SSMIS (sections 4.2.2 and 5.1)
3. Re-map all bands to a common feedhorn, thinning the result in the scan direction
4. Optionally, perform further thinning of the BT fields (along scan and along-track)
5. Write out the result in BUFR (see Table 15) and/or hdf5

The re-mapping process is illustrated in Figure 5. It works as follows:

- For each MWI scan, find the ICI scan that immediately precedes it, based on time
- Taking a typical scan line, find, for each required spot in the reference band, the coordinates of 4 spots in the non-reference band that map onto that reference spot: 2 spots in one scan and 2 spots in the following scan. Compute weights for interpolation.
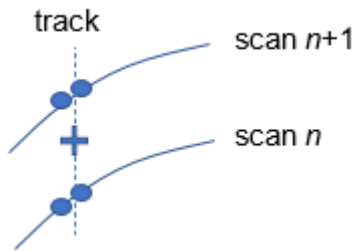- Perform bilinear interpolation of the brightness temperature field, for each channel.

Figure 5: Illustration of the brightness temperature re-mapping. BTs from the circles are interpolated to the cross position.

As noted previously, it is expected that global MWI and ICI level 1 products will be disseminated by EUMETSAT in both BUFR and netCDF formats. However, at the time of release of MWIPP v1.1 only netCDF test data are available. Support for BUFR ingest will be added in a later release.

6. REFERENCES

Accadia, C., V. Mattioli, F. De Angelis, J. Ackermann, S. Di Michele, I. Hans, P. Schlüssel, P. Colucci and A. Canestri, 2020: "Passive Microwave and sub-mm wave imaging missions at EUMETSAT", presentation from *4th workshop on assimilating satellite cloud and precipitation observations for NWP*, ECMWF, 3-6 Feb 2020, https://events.ecmwf.int/event/146/timetable/

Bell, W., 2006, "A Preprocessor for SSMIS Radiances Scientific Description", document NWPSAF-MO-UD-014

Bell, W., English, S.J., Candy, B., Atkinson, N., Hilton, F., Baker, N., Swadley, S., Campbell, W., Bormann, N., Kelly, G. and Kazumori, M., 2008: The Assimilation of SSMIS Radiances in Numerical Weather Prediction Models. IEEE Transactions on Geoscience and Remote Sensing, Vol 46 No 4, April 2008, pp 884-900.

Bell, W., S. Bühler, S. Crewell, P. Eriksson, A. Geer, J-F. Mahfouf, F. S. Marzano, C. Prigent, R. Tonboe, 2017: "MWI-ICI Science Plan, a report from the MWI-ICI Science Advisory Group", available from https://www.eumetsat.int/science-plans-future-missions

D. W. Draper, D. A. Newell, F. J. Wentz, S. Krimchansky and G. M. Skofronick-Jackson, "The Global Precipitation Measurement (GPM) Microwave Imager (GMI): Instrument Overview and Early On-Orbit Performance," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 7, pp. 3452-3462, July 2015. doi: 10.1109/JSTARS.2015.2403303

H. Shimoda, "Overview of GCOM," 2011 IEEE International Geoscience and Remote Sensing Symposium, Vancouver, BC, Canada, 2011, pp. 4134-4137. doi: 10.1109/IGARSS.2011.6050143

Swadley, S., Poe, G., Baker, N., Kunkee, D., Boucher, D., Bell, W., Brown, S, Long, E., Prata, A., 2008, "Observed F-16 and F-17 Anomalies Detailed Analysis of the Root Causes, and the Path Forward", proc. ITSC-16, http://cimss.ssec.wisc.edu/itwg/itsc/itsc16/program.html

Yang, H., Weng, F., Lv L., Lu N., Liu G., Bai M., Qian Q., He J., and Xu H., 2011: The FengYun-3 Microwave Radiation Imager On-Orbit Verification, IEEE Transactions on Geoscience and Remote Sensing, Vol 49 No 11, 4552-4560.

APPENDIX: BUFR SEQUENCES

Table 14: Generic BUFR sequence used by MWIPP for microwave imager data mapped to a user grid (first version)

Descriptor	Meaning
001007	satellite identifier
301011	year, month, day
301013	hour, minute, second
202126, 007001, 202000	satellite height (100m precision)
201132, 005041, 201000	scan line number (increased to 12 bits – max 4094)
005043	fov number (max 254)
301023	lat,lon (0.01 deg precision)
013040	surface flag (includes 0=land, 5=ocean, 6=coast)
020029	rain flag (1=rain, 0=no rain)
007025	solar zenith angle
005022	solar azimuth
104000, 031001	Replicate 4 descriptors this number of times (angles)
025140	start channel
025141	end channel
007026	satellite zenith angle (0.0001 deg precision)
005021	satellite azimuth angle
102000, 031001	Replicate 2 descriptors this number of times (channels)
005042	channel number
012163	brightness temperature (0.01K precision)

Table 15: Generic BUFR sequence used by MWIPP for microwave imager data mapped to a user grid (second version). This sequence is used by default for MWI/ICI. Differences from the first version are shown in blue. The lines in orange are omitted if there is only one set of angles (which is the case for MWI/ICI).

Descriptor	Meaning
001007	satellite identifier
001033	centre
001034	subcentre
002019	instrument
301011	year, month, day
301012	hour, minute
207003, 004006, 207000	second
202126, 007001, 202000	satellite height (100m precision)
025085	orbit angle
201133, 005041, 201000	scan line number (increased to 13 bits – max 8190)
201131, 005043, 201000	fov number (max 2046)
301021	lat,lon (0.00001 deg precision)
013040	surface flag (includes 0=land, 5=ocean, 6=coast)
020029	rain flag (1=rain, 0=no rain)
021166	land fraction
010001	surface height
007025	solar zenith angle
005022	solar azimuth
104000, 031001	Replicate 4 descriptors this number of times (angles)

025140	start channel
025141	end channel
007026	satellite zenith angle (0.0001 deg precision)
005021	satellite azimuth angle
104000, 031001	Replicate 4 descriptors this number of times (channels)
005042	channel number
002153	frequency in Hz
002104	polarisation (0=H, 1=V)
012163	brightness temperature (0.01K precision)

Table 16: BUFR sequence used for MWRI on original grid (defined unofficially by CMA in 2014 and implemented in AAPP). May be superseded in due course.

Descriptor	Meaning
001033	centre
001034	subcentre
001007	satellite identifier
002019	satellite instruments (938 = MWRI)
005040	orbit
201136, 005041, 201000	scan line
005043	fov number
301011	year, month, day
301013	hour, minute, second
301021	latitude, longitude (0.0001 deg)
007002	height (surface)
008012	land/sea qualifier (0=land, 1=sea, 2=coast)
007024	satellite zenith angle (0.01 deg precision)
005021	satellite azimuth angle
007025	solar zenith angle
005022	solar azimuth
108000, 031001	Replicate 8 descriptors this number of times (channels)
005042	channel number
002153	centre frequency
002154	bandwidth
201132, 202129, 012063, 202000, 201000	brightness temperature (0.01K precision)

Table 17: BUFR sequence 3-50-254 for AMSR-2 radiances (EUMETSAT local definition)

Descriptor	Meaning
001007	satellite identifier
002019	satellite instruments (478 = AMSR-2)
005040	orbit
004001	year
004002	month
004003	day
004004	hour
004005	minute
004006	second
202126, 007001, 202000	satellite height (100m precision)
005001	latitude (0.0001 deg precision)

006001	longitude (0.0001 deg precision)
201133, 005041, 201000	scan line number (increased to 13 bits – max 8190)
025070	major frame count (missing)
007025	solar zenith angle
005022	solar azimuth
102009	repeat 2 descriptors 9 times …
002151	radiometer identifier (missing)
012064	Instrument temperature (missing)
007024	satellite zenith angle (0.01 deg precision)
005021	satellite azimuth
005043	fov number
105012	Repeat 5 descriptors 12 times (channels 1 to 12) …
005042	channel number
002153	centre frequency
033032	quality flag for ATOVS (missing)
002104	polarisation (0=H, 1=V)
012163	brightness temperature
109004	Repeat 9 descriptors 4 times (89 GHz odd spots, AH, AV, BH, BV) …
005001	latitude (0.0001 deg precision)
006001	longitude (0.0001 deg precision)
005043	fov number
005043	fov number (the two fov numbers are to be added together)
005042	channel number
002153	centre frequency
033032	quality flag for ATOVS (missing)
002104	polarisation (0=H, 1=V)
012163	brightness temperature
109004	Repeat 9 descriptors 4 times (89 GHz even spots, AH, AV, BH, BV) …
005001	latitude (0.0001 deg precision)
006001	longitude (0.0001 deg precision)
005043	fov number
005043	fov number
005042	channel number
002153	centre frequency
033032	quality flag for ATOVS (missing)
002104	polarisation (0=H, 1=V)
012163	brightness temperature (0.01K precision)

Table 18: BUFR sequence 3-40-012 for GMI radiances. Channels 1-9, 10-13 are encoded separately

Descriptor	Meaning
001007	satellite identifier
002019	satellite instruments (519 = GMI)
008091	coordinates significance (0 = satellite)
005001	latitude of satellite
006001	longitude of satellite
007002	height of satellite
005063	roll
005064	pitch

005066	yaw
005041	scan line
005067	number of scan lines (used to store fov number)
301011	year, month, day
301012	hour, minute
004007	seconds within a minute (microsecond accuracy)
008091	coordinates significance (1 = observation)
005001	latitude
006001	longitude
107000, 031001	Replicate 7 descriptors this number of times (channels)
005042	channel number
002153	centre frequency
002104	polarisation (0=H, 1=V)
040028	GMI quality flag
007024	satellite zenith angle (0.01 deg precision)
040027	sun glint angle
012063	brightness temperature (0.1K precision)

Table 19: BUFR sequence 3-10-025 for SSMIS UPP (24 channels on the grid of the Lower Atmosphere Sounding channels, 60 spots per scan)

Descriptor	Meaning
001007	satellite identifier
008021	time significance (28 = start of scan)
004001	year
004002	month
004003	day
004004	hour
004005	minute
201138, 202131, 004006, 202000, 201000	second (precision 0.001 sec)
201132, 005041, 201000	scan line number (increased to 12 bits – max 4094)
201129, 005043, 201000	fov number (max 510)
005002	latitude (0.01 deg)
006002	longitude (0.01 deg)
013040	surface flag (0=land, 2=near coast, 3=ice, 4=possible ice, 5=ocean, 6=coast)
020029	rain flag (0=no, 1=rain)
104024	repeat 4 descriptors 24 times (channels)
005042	channel number
012163	brightness temperature (0.01K precision)
021083	warm target counts
021084	cold target counts
115003	repeat 15 descriptors 3 times (ephemeris data)
004001	year
004002	month
004003	day
201142, 202131, 004026, 202000, 201000	time period
005001	latitude

006001	longitude
201138, 202129, 007001, 202000, 201000	height of station (0.1km) – there's a mistake here, the units are supposed to be m but are interpreted as km.
008021	time significance (19 = orbit start)
004001	year
004002	month
004003	day
004004	hour
004005	minute
005040	orbit number
101003	Repeat 1 descriptor 3 times
012070	warm load temperature
025054	SSMIS sub-frame ID
101004	Repeat 1 descriptor 4 times
025055	multiplexer housekeeping
008007	dimensional significance (1=line)
104028	Repeat 4 descriptor 28 times
005002	base point latitude (0.01 deg)
006002	base point longitude (0.01 deg)
002111	incidence angle
005021	azimuth