

James Hocking, Met Office, UK

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.

COPYRIGHT 2025, EUMETSAT, ALL RIGHTS RESERVED.

	Change record			
Version	Date	Author / changed by	Remarks	
0.1	06/01/2025	J. Hocking	First draft	
1.0	08/04/2025	J. Hocking	Updates after beta test	
1.0.1	22/04/2025	J. Hocking	Updates after internal review	
1.0.2	11/06/2025	J. Hocking	Updates after DRR	



Table of Contents

1.	INTRODUCTION	4
2.	INSTALLATION	4
2.1	System Requirements	4
2.2	External Libraries	5
2.3	Compiling the code	6
3.	RUNNING THE CODE	8
4.	INPUT/OUTPUT FILES	10
4.1	Configuration namelist file	10
4.2	Observation data file	18
4.3	Model data files	20
4.3.1	Met Office UM Fieldsfiles / PP files (file type 0)	24
4.3.2	GRIB files (file types 1, 6, 7 and 8)	25
4.3.3	NetCDF files (file type 5)	30
4.3.4	NWP-SAF profile datasets (file types 2, 3, 4, 9, 10, 11)	32
4.3.5	Temporal interpolation	33
4.4	RTTOV Coefficients	33
4.5	Output files	35
4.5.1	Naming convention	35
4.5.2	Contents	35
4.5.3	File size	38
5	DDOCESSING DETAILS	38
5.		00
5.1	Use of observation information	38
5.1 5.1.1	Use of observation information Horizontal interpolation (regular lat-lon grid)	38 39
5.1 5.1.1 5.1.2	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation	38 39 40
5.1 5.1.1 5.1.2 5.1.3	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation	38 39 40 40
5.1 5.1.1 5.1.2 5.1.3 5.2	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids	38 39 40 40 41
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation	38 39 40 40 41 42
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions	38 39 40 40 41 42 42
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion	38 39 40 40 41 42 42 43
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1 5.3.2	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion	38 39 40 40 41 42 42 43 43
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1 5.3.2 5.3.3	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion	38 39 40 40 41 42 42 43 43 43
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1 5.3.2 5.3.3 5.3.3 5.4	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels	38 39 40 40 41 42 42 43 43 44 44
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1 5.3.2 5.3.3 5.4 5.5	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type	38 39 40 40 41 42 42 43 43 43 44 45
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values	38 39 40 41 42 43 44 44 45 46
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3.1 5.3.2 5.3.3 5.3.2 5.3.3 5.4 5.5 5.6 5.7	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values Satellite zenith and azimuth angles	38 39 40 41 42 43 43 44 45 46 46
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6 5.7 5.8	Use of observation information. Horizontal interpolation (regular lat-lon grid). Temporal interpolation. Footprint simulation. Non-standard grids. Coordinate rotation . Field conversions. Potential temperature conversion . Humidity conversion . Rain/snow flux conversion . Pressure levels. Surface type. Emissivity and BRDF values . Satellite zenith and azimuth angles . Solar radiation .	38 39 40 41 42 43 44 44 45 46 47
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values Satellite zenith and azimuth angles Solar radiation Hydrometeor simulations	38 39 40 41 42 43 43 44 45 46 46 47 48
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values Satellite zenith and azimuth angles Solar radiation Hydrometeor simulations	38 39 40 41 42 43 44 44 45 46 46 47 48 49
5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3 5.3.1 5.3.2 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 5.11	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values Satellite zenith and azimuth angles Solar radiation Hydrometeor simulations Principal Components simulations	38 39 40 40 41 42 43 43 44 45 46 46 47 48 49 49
5. 5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 5.11 5.12	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values Satellite zenith and azimuth angles Solar radiation Hydrometeor simulations Aerosol simulations Principal Components simulations Running RTTOV	38 39 40 41 42 43 44 45 46 47 48 49 50
5. 5.1 5.1.1 5.1.2 5.2.1 5.2.1 5.3.1 5.3.2 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 5.11 5.12 5.13 5.2 5.3.1 5.3.2 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.2 5.3.1 5.3.2 5.3.2 5.3.2 5.3.1 5.3.2 5.10 5.11 5.12 5.13	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values Satellite zenith and azimuth angles Solar radiation Hydrometeor simulations Principal Components simulations Running RTTOV Quality control	38 39 40 41 42 43 44 45 46 47 49 50 50
5. 5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3.1 5.3.2 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 5.11 5.12 5.13 5.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.4 5.5 5.10 5.10 5.10 5.12 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.3 5.4 5.10 5.11 5.12 5.10 5.12 5.10 5.12 5.10 5.12 5.10 5.12 5.10 5.11 5.12 5.11 5.12 5.11 5.12 5.11 5.12 5.11 5.12 5.13 5.12 5.11 5.12 5.13 5.12 5.13 5.12 5.13 5.12 5.13 5.12 5.13 5.12 5.13 5.12 5.13 5.12 5.13 5.12 5.13 5.14	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values Satellite zenith and azimuth angles Solar radiation Hydrometeor simulations Principal Components simulations Running RTTOV Quality control Memory Usage and Batching	38 39 40 41 42 43 44 45 46 47 49 50 50 50
5. 5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 5.11 5.12 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9 5.11 5.12 5.13 5.14 5.12 5.13 5.14 5.14 5.10 5.14	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values Satellite zenith and azimuth angles Solar radiation Hydrometeor simulations Principal Components simulations Running RTTOV Quality control Memory Usage and Batching Memory Requirement for Model Fields	38 39 40 42 43 44 45 46 47 49 50 52
5. 5.1 5.1.1 5.1.2 5.1.3 5.2 5.2.1 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 5.11 5.12 5.3.3 5.4 5.5 5.6 5.7 5.8 5.9 5.11 5.12 5.12 5.13 5.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.3 5.4 5.5 5.10 5.12 5.3.1 5.3.2 5.3.1 5.3.2 5.3.1 5.3.2 5.3.3 5.4 5.12 5.10 5.12 5.3.1 5.3.2 5.3.1 5.3.2 5.4 5.12 5.10 5.12 5.3.1 5.3.2 5.3.1 5.3.2 5.4 5.12 5.10 5.12 5.10 5.12 5.3.1 5.3.2 5.4 5.5 5.10 5.11 5.12 5.12 5.12 5.12 5.13 5.4 5.12 5.10 5.11 5.12 5.10 5.11 5.12 5.10 5.11 5.12 5.11 5.12 5.11 5.12 5.11 5.12 5.11 5.12 5.11 5.12 5.11 5.12 5.13 5.14 5.12 5.14 5.12 5.14 5.14 5.14 5.14 5.14 5.14 5.14	Use of observation information	3 3 3 9 4 0 4 1 4 2 4 3 3 4 4 4 4 5 4 6 4 7 4 8 9 4 9 5 5 2 2 5 3 5 3 5 5 2 5 3 5 5 5 5 5 5 5
$\begin{array}{c} 5.1\\ 5.1.1\\ 5.1.2\\ 5.1.3\\ 5.2\\ 5.2.1\\ 5.3.1\\ 5.3.2\\ 5.3.3\\ 5.4\\ 5.5\\ 5.6\\ 5.7\\ 5.8\\ 5.9\\ 5.10\\ 5.11\\ 5.12\\ 5.13\\ 5.14\\ 5.14.1\\ 5.14.2\\ 5.14.3\\ \end{array}$	Use of observation information Horizontal interpolation (regular lat-lon grid) Temporal interpolation Footprint simulation Non-standard grids Coordinate rotation Field conversions Potential temperature conversion Humidity conversion Rain/snow flux conversion Pressure levels Surface type Emissivity and BRDF values Satellite zenith and azimuth angles Solar radiation Hydrometeor simulations Principal Components simulations Principal Components simulations Running RTTOV Quality control Memory Usage and Batching Memory Requirement for Model Fields Observation Profile Batching Memory Requirement for Output Data	38 39 40 42 43 44 45 46 47 49 50 52 53



6.	LIMITATIONS AND KNOWN ISSUES	55
7.	GLOSSARY	55
8.	REFERENCES	57
9.	ANNEXES	58
ANNE)	K A – radsim run.py	58
ANNE)	(B – radsim_geo_obs.py	60
ANNE)	(C – radsim_plot_example.py	67
ANNE)	CD – Example MARS, CDS and ADS requests	69
ANNE)	K E – Solar angle calculations	73
ANNE)	K F – Cloud liquid water and ice effective diameters	76
ANNE	K G – CADS height assignments	77
ANNE	K H – Footprint calculations	78
ANNE)	K I – RadSim v3.2 configuration namelist variable equivalents	30



1. INTRODUCTION

This document describes the NWP SAF Radiance Simulator (RadSim) code which is designed to output simulated satellite instrument data from NWP model fields. Details of how to install and run the code, input/output requirements and a description of the processing that is performed are included.

The code is written primarily in Fortran-90 with some additional scripts in Python. To use RadSim, some knowledge of Fortran namelists, the use of netCDF, and the capabilities of the RTTOV radiative transfer model are desirable, but in-depth knowledge is not required.

RadSim v4.0 has the following DOI: <u>https://doi.org/10.15770/EUM_SEC_CLM_1008</u>

When citing RadSim it may be described as:

The Radiance Simulator is an operational product developed and distributed by the EUMETSAT Satellite Application Facility for Numerical Weather Prediction (NWP SAF) and can be downloaded from <u>https://nwp-saf.eumetsat.int/</u>.

2. INSTALLATION

2.1 System Requirements

- Unix or Linux operating system*
- Fortran-90 compiler with some basic F2003 support
- Python3 (v3.10 or later**) for some ancillary scripts
- 20 MB free disk space for installation (approximate)
- 4GB RAM minimum recommended for running simulations***

*RadSim has been tested on RHEL7 and Cray systems. It is expected to be compatible with any Unix or Linux operating system.

**Older versions of Python3 may work but are untested.

***The RAM requirement will vary considerably depending on the application. It may be much lower than this, but if using a standard data file from a recent global NWP model run, the memory requirement will be of this order. A discussion of memory usage can be found in section 5.14.

The code has been tested successfully in multiple configurations after installation with each of the following Fortran compilers:

- ifort v17.0.7, v18.0.5
- pgfortran v18.7-0
- gfortran v8.1.0, v11.2.0
- nagfor v6.1



Support notes:

Newer versions of these Fortran compilers are expected to be compatible. It may also work with some older versions, but these have not been tested and are not supported.

The *xlf* compiler was supported for RadSim v1 but this is no longer the case. A 'legacy' build configuration file is provided, and users may attempt to install and run with *xlf* if they wish. The file can be edited if the specified options are no longer appropriate. However, no additional support can be given for builds with this compiler.

2.2 External Libraries

Fortran versions of the following libraries are required. Each can be downloaded from the given locations if not already available.

• RTTOV v14.0

Available from <u>https://nwp-saf.eumetsat.int/site/software/rttov/</u> RTTOV coefficient files are available from <u>https://nwp-saf.eumetsat.int/site/software/rttov/download/coefficients/coefficient-download/</u>

Coefficient files are also needed. These can be downloaded from the link above but note that many optical depth coefficient files are provided with the RTTOV v14 distribution. The ASCII . dat files or the netCDF (.nc) files may be used. Note that the use of netCDF format requires RTTOV to be installed with netCDF support switched on explicitly. This is recommended as the emissivity/BRDF atlas files, and some coefficient files are only available in this format.

In addition to the RTTOV optical depth coefficients, input files for aerosol and/or hydrometeor optical properties (aertable/hydrotable), MFASIS-NN coefficients, and/or PC-RTTOV coefficients will be needed if using one of those options.

• netCDF (version 4.0 or later)

Available from http://www.unidata.ucar.edu/software/netcdf/

The ecCodes library is an optional dependency but is required when ingesting model data in GRIB format (see section 4.3).

• ecCodes (version 2.23.0 or later)

Available from https://confluence.ecmwf.int/display/ECC

The location of the following libraries may be needed in some cases, but this is system dependent. They are used by one or more of the above libraries so there should be no need to download new versions of these. If installation fails because one of these libraries is missing, then you will need to add it to the user configuration (see below).



• HDF5 (used by netCDF)

Available from https://www.hdfgroup.org/solutions/hdf5/

• JasPer, AEC, and/or OpenJPEG libraries (optionally used by ecCodes)

These libraries are not used by RadSim but may be dependencies of ecCodes depending on how ecCodes was built.

If you compiled RTTOV against an external LAPACK library (this is not the default, so it should be obvious if this is the case – see the RTTOV documentation), then you may need to specify the relevant information for linking RadSim against this same LAPACK library.

2.3 Compiling the code

All the RadSim code is contained in the distribution file radsim-4.0.tar.gz. Access to some external libraries (see above) is also required for compilation.

Installation consists of the following steps:

1. Unpack the distribution file

tar -xf radsim-4.0.tar.gz

This will create a radsim-4.0/ directory containing the following files/directories:

```
build/*
etc/*
src/*
radsim_check_install
radsim_install
readme.txt
user.cfg
```

2. Edit the user.cfg file

You must supply paths to external libraries. Also set the name of the compiler you wish to use; one of ifort, gfortran, pgfortran. If you are compiling on a Cray XC system, then you should specify cray-ifort as the compiler. The compiler should correspond to one of the compiler configuration files in the <code>build/cfg/</code> directory (see below).

These are defined in the following environment variables:

FC# Name of Fortran compilerNETCDF_DIR# Path to netCDF-4 installation directoryRTTOV DIR# Path to RTTOV installation directory



If you are reading model data from GRIB files (section 4.3) then set

ECCODES_DIR # Path to ecCodes installation directory

You may also need to set (this is system dependent)

NWP SAF

HDF_DIR # Path to HDF5 installation directory JASPER_DIR # Path to JasPer installation directory AEC_DIR # Path to AEC installation directory OPENJPEG_DIR # Path to OpenJPEG installation directory Each library path should contain the standard lib or lib64 and include subdirectories and in the case of RTTOV a mod subdirectory. Leave variables undefined (blank/empty) if not required.

As noted in section 2.2, if RTTOV was compiled against an external LAPACK library, then you may need to specify suitable information for linking RadSim against this LAPACK library in the LAPACK_LD environment variable. This is system-dependent, but user.cfg contains an example that should make it clear how to set this based on how the LAPACK library was included in the RTTOV build.

The above steps are usually sufficient but in some cases the default compiler options may also need to be modified.

Notes on compiler options:

Compiler options may be changed by editing the relevant compiler configuration file in the build/cfg/ subdirectory of RadSim. Extra linker options can be included by setting the LDFLAGS variable. These are appended to the default options. It is recommended to use an OpenMP-enabled build of RTTOV and to specify the relevant compiler OpenMP flag in the LDFLAGS variable when compiling RadSim as this will enable multi-threaded simulations. It is also recommended to compile RadSim with OpenMP support by adding the relevant option to the FCFLAGS variable as this is beneficial for the footprint simulations. OpenMP is enabled by default in the compiler configuration files.

All currently supported input profile dataset formats are strictly big-endian byte order apart from GRIB and netCDF which are portable formats. It is assumed that RadSim will be used on a little-endian platform such as a Linux PC and therefore byte-swapping for I/O has been included as a compiler flag in each case. For the *ifort* compiler, the F_UFMTENDIAN environment variable may be used instead to override the I/O byte-order assumption.

3. Run the install script.

./radsim_install

The directory should now contain, in addition, the following files

bin/radsim.exe bin/radsim_run # Wrapper script for radsim.exe



4. Run the install check script (optional)

./radsim_check_install

This runs a simple, quick simulation to check that the RadSim installation was successful.

3. RUNNING THE CODE

RadSim can be run by issuing the following command (in this case from the RadSim installation directory)

bin/radsim_run radsim_cfg.nl

Alternatively, the location of the configuration file <code>radsim_cfg.nl</code> can be specified in environment variable <code>RADSIM_NL</code> and the command run without argument.

radsim_cfg.nl is a Fortran namelist file containing all runtime options, including the location of input and output files and the instrument to be simulated. An example file can be found in the RadSim installation at etc/radsim_cfg_example.nl and a more cut-down version at etc/radsim_cfg_basic.nl. It is recommended to take a copy of one of these files and edit according to the application requirements. Full details of the available options are given in section 4.1 below. At the very least, the following need to be set

- Path to and type of model profile dataset (variables model_filetype and model_datafile, and for DWD ICON GRIB data or ECMWF netCDF data model ancil datafile, see section 4.3)
- Path to directory containing RTTOV coefficient files (variable rttov_coefs_dir, see section 4.4)
- Satellite instrument identifiers (variables platform, inst, satid)

Figure 1 below provides a schematic of the processing flow of RadSim.

Note that for some compilers (notably ifort), certain types of simulation may require the amount of stack memory to be increased. This can be achieved by running the following in the environment in which RadSim is being run:

\$ ulimit -s unlimited
\$ export OMP STACKSIZE=1000M

The second is specifically for OpenMP builds. If you see segmentation faults occurring this is the first thing to try.

Using a job scheduling system:

RadSim allows optional use of the RTTOV parallel interfaces which run multi-threaded simulations using OpenMP. This requires RTTOV and RadSim to be compiled with the



relevant OpenMP compiler flags (see above and the RTTOV documentation). In addition, if there are many instrument channels to simulate (such as with IASI), there may be an advantage in submitting a radiance simulator job to a scheduling system on a shared server in terms of access to system resources.

You may submit the radsim_run script to a job scheduler (some example Loadleveler and PBS directives are included but these should be checked and modified to suit the application) but note that on some systems the shell environment may not be exported. If that is the case, the job may fail when attempting to locate the configuration file. If so, it is recommended that the job is run via your own wrapper script (a modified copy of radsim run for example) that includes the configuration file pathname explicitly.

Alternative calling method

RadSim provides a Python script src/scripts/radsim_run.py which generates a configuration file and optionally calls radsim_run using this configuration file. All RadSim options can be specified via arguments to this script: most arguments are optional, and defaults are used where these are omitted. Annex A provides more information.



Figure 1: schematic of the main input, processing, and output functions of RadSim.



4. INPUT/OUTPUT FILES

4.1 Configuration namelist file

This file contains a single Fortran namelist radsim_nl. You can either edit the example namelist provided or alternatively the radsim_run.py script described above generates a configuration namelist. The table below lists the configuration namelist variables including the corresponding Fortran intrinsic types and inclusion status. Compulsory variables highlighted in bold. Where variables have been renamed since v3.2 the previous variable names are given. See also Annex I which lists RadSim v3.2 variables, gives the v4.0 equivalents, and indicates where default values have changed:

Variable name	Type, Status, Values	Description	Note		
Input data					
obs_datafile	character, optional	Path to observation data file. This allows the simulation to be performed at observation locations. The file format is described in section 4.2.			
model_datafile	character, compulsory	Path to model data file. Several formats are supported (see section 4.3).			
model_ancil_datafile	character, optional	Path to DWD ICON "invariant" model data file or ECMWF netCDF file containing surface/single-level fields (see section 4.3).			
model_filetype	integer, compulsory	Model file type identifier. See section 4.3.			
rttov_coefs_dir (v3.2: rttov_coeffs_dir)	character, compulsory	Path to RTTOV optical depth coefficients directory. See section 4.4.			
<pre>rttov_coefs_type (v3.2: rttov_coeffs_type)</pre>	character, default='.dat'	Coefficient file type, should be '.dat' or '.nc. Not used if specifying coefficient filenames explicitly.			
rttov_coefs_options (v3.2: rttov_coeffs_options)	character, optional	Additional string in <i>rtcoef</i> filename such as "_o3co2" (note this should include any underscore). Not used if specifying coefficient filenames explicitly.			
rttov_aertable_dir (v3.2: rttov_scaer_dir)	character, optional	Path to RTTOV aerosol optical properties directory for use with the <i>aerosols</i> option.			
rttov_hydrotable_dir (v3.2: rttov_hydrotable_dir and rttov_sccld_dir)	character, optional	Path to RTTOV hydrometeor optical properties directory for use with the <i>hydrometeors</i> option.			
rttov_mfasis_nn_dir	character, optional	Path to MFASIS-NN coef file directory for use with the MFASIS-NN RTTOV solver.			
rttov_pccoefs_dir (v3.2: rttov_pccoeffs_dir)	character, optional	Path to PC-RTTOV coefficients directory for use with the <i>enable_pcrttov</i> option.			
<pre>rttov_coef_file (v3.2: rttov_coeff_file)</pre>	character, optional	Filename of RTTOV optical depth coefficient file.			
rttov_aertable_file (v3.2: rttov_scaer_file)	character, optional	Filename of RTTOV aerosol optical properties file for use with the <i>aerosols</i> option.			
rttov_hydrotable_file (v3.2: rttov_hydrotable_file and rttov_sccld_file)	character, optional	Filename of RTTOV hydrometeor optical properties file for use with the <i>hydrometeors</i> option.			
rttov_mfasis_nn_file	character, optional	Filename of MFASIS-NN coef file for use with the RTTOV MFASIS-NN solver.			





ritov_mw_pol_coef lile v32:rev.pol_coef lile v34:rev.pol_coef lile rev.pol_coef lile v34:rev.pol_coef lile v35:rev.pol_coef lile v35:rev.pol_coef lile v3			Filename of RTTOV polarisation input file for	
Instruction Display and the methanism is a been changed. rttor_poccoef_file character, optional Filename of PC-RTTOV coefficient file for use optional rttor_az: more_poccoef_file optional Path to emissivity atlas directory. optional optional Path to emissivity atlas directory. optional optional Path to BRDF atlas directory. optional optional Path to BRDF atlas directory. optional optional Path to BRDF atlas directory. optional optional Full path to netCDF file containing footprint data that was created by specifying wither footprint. read_footprint_file character, optional RTTOV platform, instrument and satellite instrument variables platform character, compulsory RTTOV coefficient file names. If coefficient file names are not supplied explicitly, RadStim stid compulsory List of channel numbers to use. If omitted or if all values are or the nall channels will be simulated. integer array, optional If the RadSim will read model data for all forecast times present in the input not period or if all values are or the all channels. forcetast times present in the input observation datastrikes from value aboset of the time the model data for all forecast hours present in the input observation times (fase)	rttov_mw_pol_coef_file	character,	use with the <i>hydrometeors</i> option and	
rttov_pccoef_file (v32: rinxo_pccoef_file) (v32: rinxo_pccoef_file) (v33: rinxo_pccoef_file) (v34: rinxo_pccoef_file) (v35:	(V3.2: httov_poi_coeff_file)	optional	<i>pol_mode=2</i> (only required if the lifename has been changed)	
Index_code_field Character, optional Interaction, with the enable_critev option. emis_atlas_dir Character, optional Path to emissivity atlas directory. fred_atlas_dir Character, optional Path to BRDF atlas directory. read_footprint_file Character, optional Path to BRDF atlas directory. platform Character, optional Full path to netCDF file containing footprint data that was created by specifying with control as in a previous run. SateIllite instrument variables RTTOV platform, instrument and sateIlite inst Character, compulsory integer, optional Integer array, optional RTTOV coefficient file names. If coefficient integer array, optional temporal_data Integer array, optional List of channel numbers to use. If omitted or if all values are of them all channels will be simulated. temporal_data Integer array, optional List of channel numbers to use. If omitted or if all forecast times present in the input model_datafile and will carry out temporal interpolation of the model data for all forecast times in the model_data to observation times (requires obs. datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_data is used. fcst_range(2) integer array, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile	rttov, pagaof, filo	character	Filename of PC-RTTOV coefficient file for use	
Instrument Optimized optimise Instruction optimite (v32: emise, atlas_dir) optional Path to emissivity atlas directory. (v32: emise, atlas_dir) optional Path to BRDF atlas directory. brdf_atlas_dir optional Path to BRDF atlas directory. read_footprint_file character, optional Path to BRDF atlas directory. framate character, optional Full path to netCDF file containing footprint data that was created by specifying write footprint, data in a previous run. statilite instrument variables RTTOV platform, instrument and satellite instrument variables platform character, compulsory RTTOV platform, instrument and satellite integer, optional RTTOV platform, instrument and satellite satid integer array, optional integer array, optional List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. temporal_data logical, default=F If true RadSim will read model data for all interpolation of the model data to observation times (requires obs_datafile to be specified) interpolation of the model data to observation times (requires obs_datafile to be specified) fcst_range(2) integer array, default=F Choose nearest-neighbour (true) or the first foreccast time in the model data to observation time steps in the fil	(v32: rttov_pccoeff_file)	ontional	with the enable pertoy option	
Integration Path to emissivity atlas directory. brd1_atlas_dir optional Path to BRDF atlas directory. brd1_atlas_dir character, optional Path to BRDF atlas directory. read_footprint_file character, optional Full path to netCDF file containing footprint data that was created by specifying write_footprint_data in a previous run. platform character, compulsory FTTOV platform, instrument and satellite inst statid integer, compulsory RTTOV coefficient file names. If coefficient integer, compulsory channels integer array, optional Usit of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. temporal_data logical, default=F If true RadSim will read model data for all forecast times present in the input interpolation of the model data to to bespecified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used. For Met Office fieldsiles only. With the steps on the fieldsfile. If enterporal_data is false, then this can be used to read/simulate a particular time step, for example fact_range=[3.9] will read only fields for times the file. For example, sciller of the interporal_data is false, then the is can be used to read/simulate a particular time step, for example fact_range=[3.9] will read only fields for times thermporal_data is false, then the is can be used to read/simulate a particular time step, for example fact_range=[3.9] will read only fields for times thermpor	emis atlas dir	character		
brdf_atlas_dir character, optional Path to BRDF atlas directory. Image: Computation of the containing footprint data that was created by specifying write, footprint_data in a previous run. Satellite instrument variables Full path to netCDF file containing footprint data that was created by specifying write, footprint_data in a previous run. Satellite instrument variables Platform compulsory Compulsory RTTOV platform, instrument and satellite dentifiers. These are the same as used in the compulsory Satellite instrument variables Compulsory Sate of the mage area suble of the compulsory Will construct them from these variables. Compulsory Its of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. TeadSim control variables For dentifier and will cead model data to all forecast times present in the input model_data to observation forst_range(2) Integer array, default=F Interporal_data true, specify start/end forecast horeast time in the model_data for be specified interporal_data is false, then this can be used to read/simulate a particular time step, fo	(v3.2: emiss_atlas_dir)	optional	Path to emissivity atlas directory.	
read_footprint_file character, optional Full path to netCDF file containing footprint data that was created by specifying write_footprint_data in a previous run. stellite instrument variables RTTOV platform, instrument and satellite identifiers. These are the same as used in the character, compulsory satid Integer, compulsory RTTOV platform, instrument and satellite identifiers. These are the same as used in the RTTOV coefficient file names. If coefficient filenames are not supplied explicitly, RadSim will construct them from these variables. satid Integer, compulsory List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. temporal_data Integer array, optional List of channel numbers to use. If omitted or is all values are 0 then all channels will be simulated. temporal_data Inger array, default=F If true RadSim will read model data for all forecast times present in the input model_datafile to bespecified interpolation of the model_data to observation times of the first forecast time in the model_datafile is used. fcst_range(2) integer array, default=F For Met Office fieldsfiles only. With temporal_data to all solves to the time steps in the file. For example, setting fost_range=[3] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is true. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an dos_datarifile to be spe	brdf_atlas_dir	character, optional	Path to BRDF atlas directory.	
read_lootprint_ine optional data that was created by specting glatform Satellite instrument variables character, compulsory RTTOV platform, instrument and satellite identifiers. These are the same as used in the RTTOV coefficient file names. If coefficient filenames are not supplied explicitly, RadSim will construct them from these variables. satid integer, compulsory RTTOV coefficient file names. If coefficient filenames are not supplied explicitly, RadSim will construct them from these variables. channels integer array, optional List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. temporal_data If true RadSim will read model data for all forecast times present in the input model_datafile and will carry out temporal interpolation of the model data to observation interpolation of the model_datafile to be specified including observation dates/times). See section 4.3.4. If false only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=3,9] will read only the T+3 fields. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).		character.	Full path to netCDF file containing footprint	
Satellite instrument variables platform character, compulsory RTTOV platform, instrument and satellite identifiers. These are the same as used in the RTTOV coefficient file names. If coefficient identifiers. These are the same as used in the RTTOV coefficient file names. If coefficient filenames are not supplied explicitly, RadSim will construct them from these variables. satid integer, compulsory List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. temporal_data RadSim control variables If the RadSim will read model data for all forecast times present in the input model_datafile and will carry out temporal interpolation of the model data to observation times (requires obs, datafile and will carry out temporal interpolation of the model data for the first forecast time in the model_datafile to suse. fest_range(2) integer array, default= [0,0] For Met Office fieldsfiles only. With temporal_data true, specify start/end only fields for times between T+3 and T+9 from the fieldsfile. If targe=[3], will read only fields. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified). 6 multi_surface logical, default=F If true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface toyes (factions are determined by the land fraction and sea-ice fraction data. 6 enable_footprints logical, default	read_tootprint_file	optional	data that was created by specifying	
platform character, compulsory RTTOV platform, instrument and satellite identifiers. These are the same as used in the RTTOV coefficient file names. It coefficient file names are not supplied explicitly, RadSim will construct them from these variables. satid integer, compulsory compulsory channels integer array, optional List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. temporal_data logical, default=F If true RadSim will read model data for all forecast times present in the input model_datafile and will cary out temporal interpolation of the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the <i>model_datafile</i> is used. fcst_range(2) integer array, default = [0.0] For Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, subset of the time steps in the file. For example, subset of the temporal_data is false, then this can be used to read/simulate a particular time step, for example fost_range=[3.3] will read only field stor times for the fost_range=[3.3] will read only field the temporal_data is true. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) temporal interpolation (requires an obs_datafile to be specified). 6 multi_surface logical, default=F If true, makes use of RTTOV's hete		Satellite inst	rument variables	
platform compulsory character, compulsory RTTOV platform, instrument and satellite identifiers. These are the same as used in the RTTOV coefficient file names. If coefficient filenames are not supplied explicitly, RadSim will construct them from these variables. satid integer, compulsory List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. channels integer array, optional List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. temporal_data logical, default=F If true RadSim will read model data for all forecast time in the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used. 6 fcst_range(2) integer array, default = [0,0] For Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,9] will read only the T+3 fields. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified). 6 multi_surface logical, default=F If true, RadSim returns the mean		character		
Inst Identifiers. These are the same as used in the conflicient file names. If coefficient file names. If coefficien	platform	compulsorv	RTTOV platform, instrument and satellite	
inst compulsory If I OV coefficient the names. Ir coefficient satid integer, compulsory integer, optional integer, compulsory channels integer array, optional List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. temporal_data RadSim control variables If true RadSim will read model data for all forecast times present in the input model_data for the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model data for othe first forecast time in the model data for othe first forecast time in the model data for othe first forecast time in the model data for only a subset of the time steps in the file. For example, setting fcst_range(3.9) will read only fields for imps between T+3 and T+9 from the fields for imps fcst_range=(3.3) will read only the T+3 fields. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_ datafile to be specified). 6 multi_surface logical, default=F If true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage (factions are determined by the land fraction and sea-ice fraction data. 6		character.	identifiers. These are the same as used in the	
satid integer, compulsory Integer array, optional Integer array, optional Integer array, optional List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. RadSim control variables Integer array, optional logical, default=F If true RadSim will read model data for all forecast times present in the input model_datafile and will carry out temporal interpolation of the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_data for othe first forecast time in the model_data for othe first forecast time in the model_data for the first forecast time in the model_data for othe section 4.3.4. If false only data for the first forecast time in the model_data for othe section 4.3.4. If false only data for the first forecast time in the model_data for only. With temporal_data is false, then this can be used. 6 fcst_range(2) integer array, default = [0,0] Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified). 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or linear (false) spatial interpolation when temporal_data is true. 6 multi_surface logical, default=F Choose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true. 6 enable_footprints logical, defa	inst	compulsory	RITOV coefficient file names. If coefficient	
Satu compulsory Win consider them not near vanables. channels integer array, optional List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated. RadSim control variables If frue RadSim will read model data for all forecast times present in the input model_datafile and will carry out temporal interpolation of the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used. fcst_range(2) integer array, default = [0,0] For Met Office fieldsfiles only aubset of the time steps in the file. For example, setting fcst_range=[3,3] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) temporal interpolation when temporal_data is true. 6 multi_surface logical, default=F Choose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true. 6 enable_footprints logical, default=F If true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (fand, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data. 6 <td>satid</td> <td>integer,</td> <td>will construct them from these variables</td> <td></td>	satid	integer,	will construct them from these variables	
channelsinteger array, optionalList of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated.RadSim control variablesIf true RadSim will read model data for all forecast times present in the input model_datafile and will carry out temporal interpolation of the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used.fcst_range(2)integer array, default=FFor Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,9] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an default=F6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points failing within an ellipse arruy of all points failing within an ellipse arruy or all grid points failing within an ellipse		compulsory		
channels optional optional all values are 0 then all channels will be simulated. RadSim control variables If true RadSim will read model data for all forecast times present in the input model_datafile and will carry out temporal interpolation of the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used. fcst_range(2) integer array, default=F For Met Office fieldsfiles only. With temporal_data is false, then this can be used to read/simulate a particular time step, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,9] will read only the T+3 fields. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified). 6 interp_nn_temporal logical, default=F Choose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true. 6 multi_surface logical, default=F If true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data. 6 enable_footprints logical, default=F If true RadSim returns the mean radiance ov		integer arrav.	List of channel numbers to use. If omitted or if	
Instructed. RadSim control variables temporal_data If true RadSim will read model data for all forecast times present in the input model_datafile and will carry out temporal interpolation of the model_data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used. fcst_range(2) integer array, default = [0,0] For Met Office fieldSiles only. With temporal_data is false, then this can be used to read/simulate a particular time steps in the file. For example, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldSfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified). 6 multi_surface logical, default=F If true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fraction data. 6 enable_footprints logical, default=F If true RadSim returns the mean radiance over all girld points falling within an ellipse array over all girld points falling within an ellipse 6	channels	optional	all values are 0 then all channels will be	
temporal_data logical, default=F If true RadSim will read model data for all forecast times present in the input model_datafile and will carry out temporal interpolation of the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used. fcst_range(2) integer array, default = [0,0] For Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range=[3.9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3.3] will read only the T+3 fields. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified). 6 multi_surface logical, default=F Choose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true. 6 enable_footprints logical, default=F fit rue, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data. 6		PadSim co	simulated.	
temporal_datalogical, default=Ffrace reactings present in the input model_datafile and will carry out temporal interpolation of the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used.fcst_range(2)integer array, default = [0,0]For Met Office fieldsfiles only. With temporal_data is false, then this can be used to read/simulate a particular time step, in the file. For example, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) temporal interpolation (requires an obs_datafile to be specified).6multi_surfacelogical, default=FIf frue, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6			If true RadSim will read model data for all	
temporal_datalogical, default=Flogical, default=finterpolation of the model data to observation times (requires observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used.fcst_range(2)integer array, default = [0,0]For Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range=[3,3] will read only fields for times to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) spatial interpolation (requires an obs_datafile to be specified).6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6			forecast times present in the input	
temporal_datalogical, default=Finterpolation of the model data to observation times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used.fcst_range(2)integer array, default = [0,0]For Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6		logical, default=F	model datafile and will carry out temporal	
temporal_data default=F times (requires obs_datafile to be specified including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used. fcst_range(2) For Met Office fieldsfiles only. With temporal_data true, specify start/end forecast time in the model_datafile is used. fcst_range(2) integer array, default = [0,0] For Met Office fieldsfiles only a subset of the time steps in the file. For example, setting fost_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fost_range=[3,3] will read only the T+3 fields. 6 interp_nn_spatial logical, default=F Choose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified). 6 interp_nn_temporal logical, default=F Choose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true. 6 multi_surface logical, default=F If true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fraction ata. 6 enable_footprints logical, default=F If true RadSim returns the mean radiance over all grid points falling within an ellipse arround each observation. Returns the mean radiance over all grid points falling within an ellipse			interpolation of the model data to observation	
including observation dates/times). See section 4.3.4. If false only data for the first forecast time in the model_datafile is used.fcst_range(2)integer array, default = [0,0]For Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation Requires6	temporal_data		times (requires obs_datafile to be specified	
section 4.3.4. If false only data for the first forecast time in the model_datafile is used.For Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (false) fraction sare determined by the land fraction and sea-ice fraction data.6			including observation dates/times). See	
forecast time in the model_datafile is used.forecast time in the model_datafile is used.for Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) spatial interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with thre surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Returnses6			section 4.3.4. If false only data for the first	
fcst_range(2)For Met Office fieldsfiles only. With temporal_data true, specify start/end forecast hours to read fields for only a subset of the time steps in the file. For example, setting fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires6			forecast time in the <i>model_datafile</i> is used.	
fcst_range(2)integer array, default = [0,0]integer array, default = [0,0]integer array, default = [0,0]integer array, fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse arction are do beservation. Requires6			For Met Office fieldsfiles only. With	
fcst_range(2)integer array, default = [0,0]integer array, default = [0,0]integer array, fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal_data is true.6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires			temporal_data true, specily stantend forecast	
fcst_range(2)integer array, default = [0,0]fcst_range=[3,9] will read only fields for times between T+3 and T+9 from the fieldsfile. If temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.6interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires6			time steps in the file. For example, setting	
fcst_range(2) Integer analysis Hoger analysis Hoge		integer array.	fcst_range=[3.9] will read only fields for times	_
temporal_data is false, then this can be used to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) spatial interpolation when temporal_data is true.6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires6	fcst_range(2)	default = [0.0]	between T+3 and T+9 from the fieldsfile. If	6
to read/simulate a particular time step, for example fcst_range=[3,3] will read only the T+3 fields.interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation Requires6		[[]]]	temporal_data is false, then this can be used	
example fcst_range=[3,3] will read only the T+3 fields.interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires6			to read/simulate a particular time step, for	
Interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires			example fcst_range=[3,3] will read only the	
interp_nn_spatiallogical, default=FChoose nearest-neighbour (true) or bi-linear (false) spatial interpolation (requires an obs_datafile to be specified).6interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation_Requires			I+3 fields.	
Interp_nn_spatial default=F (raise) spatial interpolation (requires an obs_datafile to be specified). 6 interp_nn_temporal logical, default=F Choose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true. 6 multi_surface logical, default=F If true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data. 6 enable_footprints logical, default=F If true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires 6	intern nn enstiel	logical,	(folge) apatial interpolation (requires or	e
interp_nn_temporallogical, default=FChoose nearest-neighbour (true) or linear (false) temporal interpolation when temporal_data is true.6multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires6	mierp_nn_spatial	default=F	(laise) spatial interpolation (requires an	б
interp_nn_temporallogical, default=Fchroster hearest hea			Choose nearest-neighbour (true) or linear	
default=F (late) temporal_data is true. of multi_surface If true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data. 6 enable_footprints Iogical, default=F If true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires 6	interp nn temporal	logical,	(false) temporal interpolation when	6
multi_surfacelogical, default=FIf true, makes use of RTTOV's heterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires6		detault=F	temporal data is true.	
multi_surfacelogical, default=Fheterogeneous surface capability. Each profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data.6enable_footprintslogical, default=FIf true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires6			If true, makes use of RTTOV's	
multi_surface logical, default=F profile is associated with three surface types (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data. 6 enable_footprints logical, default=F If true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires			heterogeneous surface capability. Each	
default=F (land, sea, sea-ice) and the relevant coverage fractions are determined by the land fraction and sea-ice fraction data. enable_footprints logical, default=F logical, default=F over all grid points falling within an ellipse around each observation. Requires	multi surface	logical,	profile is associated with three surface types	6
enable_footprints logical, default=F If true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires		default=F	(land, sea, sea-ice) and the relevant coverage	
enable_footprints logical, default=F If true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires			tractions are determined by the land fraction	
enable_footprints logical, default=F logical, cover all grid points falling within an ellipse around each observation. Requires			and Sea-ice fraction data.	
default=F around each observation. Requires	enable footprints	logical, default=F	over all grid points falling within an ellinge	
			around each observation. Requires	





		<i>obs_datafile</i> to be specified including footprint ellipse major/minor axes (radii) and azimuth angle for each observation.	
run_mfasis	logical, default=F	Run MFASIS-NN visible fast cloud model. This requires extra fields in the input model data file (see section 4.3), and RTTOV hydrotable and MFASIS-NN coef files. This is a shortcut for setting <i>hydrometeors</i> and <i>solar</i> to true and <i>solar_solver</i> to 2.	
use_model_clw_deff (v3.2: ircloud_use_model_clw_deff)	logical, default=F	For cloud simulations using ICON GRIB fields, use the cloud liquid water particle size model field. Requires <i>hydrometeors</i> to be true and the "CLW Deff" cloud scheme to be selected (<i>visir_clw_properties</i>).	
use_model_ice_deff (v3.2: ircloud_use_model_ice_deff)	logical, default=F	For cloud simulations using ICON GRIB fields, use the cloud ice water particle size model field. Requires <i>hydrometeors</i> to be true and the Baum ice cloud scheme to be selected (<i>visir_ice_properties</i>).	
combine_qcrttov	logical, default=T	The <i>qcrttov</i> output dataset contains RTTOV quality flags which are defined per-channel. If <i>combine_qcrttov</i> is false, these are output without modification. To reduce output file size, when <i>combine_qcrttov</i> is true, the flags for all channels are combined with a logical OR for each profile so that there is just one integer per profile with each flag set if it was set for at least one channel for that profile.	6
output_mode	integer, default=1	RadSim diagnostic output mode. Three levels are defined: 1=Normal, 2=Verbose, 3=Debug. This is separate from the <i>rttov_verbose</i> option which controls RTTOV output.	
seaice_threshold	real, default=0.2	Separation threshold for sea or sea-ice surface type classification. Valid range [0 - 1]. Ignored if <i>multi_surface</i> is true.	
coast_threshold	real, default=0.2	Difference threshold for coastline. Valid range [0 - 0.5]. Ignored if <i>multi</i> surface is true.	1
co2_max_ppmv	real, default=-1.	If a positive value is specified, the RTTOV reference CO ₂ profile is scaled to this maximum value (in ppmv). This requires an RTTOV coefficient file that supports variable CO ₂ . Suggested range: 290-550 ppmv. If negative, the RTTOV background profile is used without modification.	
ssu_co2_cell_pressure	real array, dim(3), default= (-1,-1,-1)	If using SSU PMC shift coefficients, this specifies the cell pressures. Non-positive values will result in the nominal cell pressures being used. If you simulate a subset of the three SSU channels, only specify one or two pressures (for the one or two simulated channels respectively). NB You must also set either <i>co2_data</i> true (with CO ₂ included in the model data) or specify <i>co2_max_ppmv</i> .	
cads_height_assign_threshold	real, default=-1.	If a positive value is specified (usually 0.01), RadSim calculates and outputs channel height assignments consistent with the NWP SAF CADS package (see Annex G).	





calc_geo_sat_angles	logical, default=F	If true, RadSim computes satellite zenith and azimuth angles for a geostationary satellite with sub-satellite point at geo_sat_lat, geo_sat_lon and at altitude geo_sat_height. Applies to both model grids and obs data files. See section 5.7.	
geo_sat_lat real, default=0.0		Latitude of sub-satellite point (degrees, should be zero or very close to zero), used if <i>calc_geo_sat_angles</i> is true	
geo_sat_lon	real, default=0.0	Longitude of sub-satellite point (in interval [0,360) degrees), used if calc_geo_sat_angles is true	
geo_sat_height	real, default=35800	Altitude of satellite (km), used if <i>calc_geo_sat_angles</i> is true	
zen_angle	real, default=0.0	Satellite zenith angle (degrees) to use for model grid runs.	
azi_angle	real, default=0.0	Satellite azimuth angle (degrees) to use for model grid runs (0-360; measured clockwise, east=+90).	
sun_zen_angle	real, default=-1.	Fixed solar zenith angle (degrees) to use for model grid runs: if unspecified (or negative) RadSim computes solar angles for each grid point. Only applies if <i>solar</i> is true.	
sun_azi_angle	real, default=0.0	Fixed solar azimuth angle (degrees) to use for model grid runs: if <i>sun_zen_angle</i> is unspecified RadSim computes solar angles for each grid point (0-360; measured clockwise, east=+90). Only applies if <i>solar</i> is true.	
use_emis_atlas (v3.2: use_emiss_atlas)	logical, default=F	Use an emissivity atlas for land and sea-ice surfaces (for sea-ice this is dependent on the atlas).	2
use_brdf_atlas	logical, default=F	Use the BRDF atlas for land surfaces if the solar option is enabled.	2
emis_atlas_id	integer, default=1	Selects which RTTOV emissivity atlas to use. For IR sensors: 1=UWIRemis, 2=CAMEL single-year, 3=CAMEL climatology. For MW sensors: 1=TELSEM2, 2=CNRM atlas. See RTTOV user guide for details on the atlases.	
emis_atlas_year	integer, default=2015	Select year for emissivity atlas data. Only applicable to MW CNRM atlas (valid values: 2014, 2015) and IR CAMEL v3 single-year atlas (valid values: 2000-2021).	6
camel_atlas_version	integer, default=3	Select version of CAMEL atlas. Valid values: 2, 3.	6
use_all_atlas_months	logical, default=F	When using the emissivity and/or BRDF atlases with the NWP SAF profile datasets, by default the atlas data for the month of the first profile are used for all profiles. If this option is true, then atlas data are loaded for all 12 months and the correct atlas data are used for each profile. Note that this may require a lot of memory.	
default_brdf_land	real, default=-1.	If greater than zero, specifies the BRDF value used for land surfaces in the absence of other data (e.g. BRDF atlas if used) for solar	





		simulations. This replaces the RTTOV default value.	
default_brdf_seaice	real, default=-1.	If greater than zero, specifies the BRDF value used for sea-ice surfaces in the absence of other data (e.g. BRDF atlas if used) for solar simulations. This replaces the RTTOV default value.	
fixed_emis	real, default=-1.	If greater than zero use this fixed emissivity value for all channels and profiles. Not generally recommended, intended for testing purposes.	6
max_array_size	integer, default=500	Maximum size of output array (in MB). Used to define number of output batches.	3
max_profs	integer, default=500000	Maximum number of interpolated profiles. Used to define number of batches for processing obs data.	3
nthreads	integer, default=1	If equal to 2 or more RadSim and RTTOV will make use of multiple threads. This requires RTTOV and RadSim to be compiled with OpenMP flags which is recommended.	
nprofs_per_call	integer, default=5	Number of profiles passed to RTTOV in each call.	
	RTTO	V options	iem)
(these are commuted directly	to RTTOV equival	Set RTTOV verbosity flag. This primarily	ion)
rttov_verbose	logical, default=F	affects printing of warnings when input profiles exceed the min/max training bounds of the gas optical depth regression which are only printed when <i>apply_reg_limits</i> is false. This is independent of <i>output_mode</i> .	6
apply_reg_limits	logical, default=F	Clip input profile values to the limits from the RTTOV optical depth regression where these limits are exceeded.	5
adk_bt	logical, default=T	Jacobians for thermal channels are in terms of brightness temperature (T) or radiance (F).	6
adk_refl	logical, default=T	Jacobians for visible/near-IR channels are in terms of reflectance (T) or radiance (F).	6
opdep13_gas_clip	logical, default=T	Clip negative predicted optical depths to zero for individual gases for v13 predictor coefficients.	5
interp_mode	integer, default=4	Select the RTTOV interpolation mode. Valid modes: 1-5.	5
o3_data (v3.2: ozone_data)	logical, default=F	Use input ozone profiles (must be present in model data).	
co2_data	logical, default=F	Use input CO_2 profiles (must be present in model data, currently only in CAMS 137L and MACC 60L profile datasets).	6
n2o_data	logical, default=F	Use input N ₂ O profiles (must be present in model data, currently only in CAMS 137L profile dataset).	6
co_data	logical, default=F	Use input CO profiles (must be present in model data, currently only in CAMS 137L and 60L profile datasets).	6
ch4_data	logical, default=F	Use input CH ₄ profiles (must be present in model data, currently only in CAMS 137L and MACC 60L profile datasets).	6





so2_data	logical, default=F	Use input SO ₂ profiles (must be present in model data, currently only in CAMS 137L and 60L profile datasets).	6
solar (v3.2: addsolar)	logical, default=F	Enable solar radiation for affected channels.	
rayleigh_single_scatt	logical, default=T	Enable/disable Rayleigh single-scattering calculation for visible channels.	5
rayleigh_max_wavelength	real, default=2.0	Maximum channel central wavelength (microns) above which Rayleigh scattering calculations are omitted.	5
rayleigh_min_pressure	real, default=0.0	Minimum pressure (hPa) below which Rayleigh scattering calculations are omitted.	5
nlte_correction (v3.2: do_nlte_correction)	logical, default=F	Apply the non-local thermodynamic equilibrium bias correction (requires an RTTOV coefficient file which supports this).	
refraction (v3.2: addrefrac)	logical, default=T	Include effects of atmospheric refraction.	5
plane_parallel	logical, default=F	Enable strict plane-parallel geometry (NB this is automatically enforced if the DOM solver is selected for scattering simulations).	5
rad_down_lin_tau	logical, default=T	Use linear-in-tau for downwelling radiance calculation for clear-sky and Chou-scaling solvers.	5,6
use_t2m	logical, default=T	Use 2m temperature field in simulations.	6
use_q2m	logical, default=T	Use 2m water vapour field in simulations.	6
mw_sea_emis_model (v3.2: fastem_version)	integer, default=3	Select sea surface emissivity model for MW sensors. Valid values: 1=FASTEM-5, 2=FASTEM-6, 3=SURFEM-Ocean.	
ir_sea_emis_model	integer, default=2	Select sea surface emissivity model for IR sensors. Valid values: 1=ISEM, 2=IREMIS.	5
solar_sea_refl_model (v3.2: solar_sea_brdf_model)	integer, default=1	Select solar sea reflectance model. Valid values: 1=Elfouhaily <i>et al.</i>	5
lambertian (v3.2: do_lambertian)	logical, default=F	Treat surface as Lambertian reflector.	
lambertian_fixed_angle	logical, default=F	Use fixed angle for downwelling Lambertian radiances (T) or use parameterisation for angle (F) (only applies if <i>lambertian</i> option is true)	5
clw_data	logical, default=F	Use input cloud liquid water profile (applies to non-scattering MW simulations only).	
permittivity_param (v3.2: mw_clw_scheme)	integer, default=2	Select the cloud liquid water permittivity parameterisation to use with <i>clw_data</i> =T. Valid values: 1=Liebe, 2=Rosenkranz, 3=TKC.	5
clw_cloud_top (v3.2: mw_clw_cloud_top)	real, default=322hPa	Pressure level above which CLW is ignored in input cloud profiles (only applies if <i>clw_data</i> is true).	5
aerosols (v3.2: ir_addaerosols_cams)	logical, default=F	Include aerosol scattering in simulations using CAMS or ICON-ART aerosol fields. This option requires additional aerosol optical property files to be specified (see <i>rttov_aertable_dir/file</i> above).	
hydrometeors (v3.2: run_scatt and ir_addclouds)	logical, default=F	Include hydrometeor scattering in simulations. This option requires additional hydrometeor	



2	NWP SAF
	Numerical Weather Prediction

		optical property files to be specified (see	
		RTTOV scattering solver for thermal	
thermal solver	integer.	radiation: 1=DOM, 2=Chou-scaling, 3=delta-	
(v3.2: ir_scatt_model)	default=3	Eddington (applies only if hydrometeors or	
		aerosols true).	
	1.1	RTTOV scattering solver for solar radiation:	
solar_solver	integer,	1=DOM, 2=MFASIS-NN (applies only if solar	
(V3.2: VIs_scatt_model)	default=1	true and hydrometeors or aerosols true).	
	le vie el	If true enables Rayleigh multiple scattering	
rayleigh_multi_scatt	logical,	calculations with the solar DOM solver. Note	
(v3.2: dom_rayleign)		that this is very computationally expensive.	
dom natroomo	integer,	Number of streams for DOM calculations	
dom_nstreams	default=8	(only applies if DOM solver active).	
dom occuracy	roal default_0.0	DOM accuracy parameter (only applies if	
dom_accuracy		DOM solver active).	
dom opdop throshold	roal default_0.0	DOM absorption optical depth threshold (only	
		applies if DOM solver active).	
	logical	If true, enables Tang et al modification to	
chou_tang_mod	dofault–E	Chou-scaling solver (only applies if Chou-	6
	ueraun_r	scaling solver active).	
		Empirical factor used in Tang et al	
chou tang factor	read,	modification to Chou-scaling (only applies if	6
	default=0.1	Chou-scaling solver active and	0
		chou_tang_mod true).	
		Select treatment of polarisation for MW	
mw_pol_mode	integer,	scattering simulations: 0=no pol, 1=empirical,	
(v3.2: pol_mode)	default=1	2=ARO scaling (requires additional pol coef	
		file).	
ice polarisation	real. default=1.4	Factor used in empirical <i>mw_pol_mode</i> (see	5
		the RITOV user guide).	-
		Select Baran scheme to use for VIS/IR	
baran ice version	integer,	nydrometeor simulations with Baran ice	5.6
	default=2	properties. Valid Values: 1=Baran2014,	,
		Z=Baran2018.	
	interest	Ice particle size parameterisation to use for	
icede_param	Integer,	VIS/IR hydrometeor scattering with the Baum	5
(v3.2: ircioud_icede_param)	delauit=2	ice properties. Valid values. 1-4 (see RTTOV	
		VIS/IP aloud liquid water antical properties to	
visir_clw_properties	integer,	use with hydrometeers-T. Valid values:	
(v3.2: ircloud_clw_scheme)	default=1	1= OPAC $2="Deff" properties$	
		VIS/IR ice cloud optical properties to use with	
visir_ice_properties	integer,	hydrometeors=T. Valid values: 1=Baum	
(v3.2: ircloud_ice_scheme)	default=2	2=Baran	
		Select cloud overlap parameterisation Valid	
		options: 0=auto-select (IR=>1. MW=>3).	
overlap param	integer,	1=max/random, 2=two-column max frac.	5.6
··	aetault=0	3=two-column hydro-weighted. See RTTOV	, -
		user guide for details on these schemes.	
		Specify the RTTOV col_threshold variable for	
col_threshold	roal dataut 0	hydrometeor simulations which treats cloud	
(v3.2: clacol_threshold and cc_threshold)	real, default=0.	columns with weights below this value as	
		clear.	
two col may frac may n	real,	Parameter for the two-column max frac cloud	6
	default=750.	overlap parameterisation: maximum hydro	0



	Radiance S
NWP SAF	
Numerical Weather Prediction	

		frac is computed from TOA down to this	
		pressure (hPa).	
enable_pcrttov	logical,	Enable PC-RTTOV simulations (requires PC-	
(V3.2: addpc)		RTIOV coefficient file).	
npscores	Integer,	Number of PC scores to compute for PC-	
·	default=300	RTTOV simulations.	
pc_band (v3.2: ipcbnd)	lnteger, default=1	Spectral band for PC-RTTOV simulations.	5
pc_reg_set	integer,	Regression channel set for PC-RTTOV	
(v3.2: ipcreg)	default=1	simulations.	
	Output co	ntrol variables	1
output_dir	character, optional	the current directory (see section 4.5 for more details).	
output_file	character, optional	Output file name. If unspecified, RadSim constructs a filename based on the sensor and data validity time (see section 4.5).	
write_footprint_file	character, optional	Filename of netCDF file to which footprint information can be written for re-use in a subsequent run. The file can be found in <i>output_dir</i> .	
write_radiances	logical, default=F	Write radiances to the output file. The default is to write brightness temperatures (and reflectances when <i>solar</i> is true).	
write_clearsky	logical, default=F	Write clear-sky radiances/brightness temperatures/reflectances in addition to total values (for use with <i>hydrometeors</i> option).	6
write_emis (v3.2: write_emiss)	logical, default=F	Write emissivity values to the output file.	
write_brdf	logical, default=F	Write BRDF values to the output file for use with <i>solar</i> option	
write_profiles	logical, default=F	Write model profile data to the output file, including all fields that were used in the simulation (fields will be interpolated to obs positions if that option has been used).	
write_lation	logical, default=F	Write latitude/longitude data only to the output file (can be useful for model grid runs if full profile output is not required).	
write_geom_height	logical, default=F	Write geometric heights of pressure half- levels and full-levels computed by RTTOV.	
write_trans	logical, default=F	Write layer transmittances to the output file.	4
write_tjac	logical, default=F	Write temperature Jacobians to the output file.	4
write_qjac	logical, default=F	Write humidity Jacobians to the output file.	4
write_o3jac	logical, default=F	Write ozone Jacobians to the output file.	4
write_tskinjac	logical, default=F	Write skin temperature Jacobians to the output file.	
write_wind10mjac	logical, default=F	Write 10m wind u/v component Jacobians to the output file.	
write_emisjac (v3.2: write_emissjac)	logical, default=F	Write surface emissivity Jacobians to the output file (see the RTTOV user guide for more information about emissivity Jacobians).	



Table Notes

- 1. This is the difference between the assumed land fraction for simulation (i.e. 0 or 1) and the interpolated land fraction. If the difference is greater than this, the point is flagged as *coast*.
- 2. Use of an emissivity atlas (and BRDF atlas for visible channels) is recommended but these must be downloaded separately, hence the default is not to use them.
- 3. If there are difficulties running the code due to lack of memory, these parameters may be adjusted downwards to help reduce the load. Note that the maximum number of interpolated profiles does not apply if using the model grid.
- 4. Use of these options significantly increases the size of the output file and, in the case of Jacobians, the run time.
- 5. These RTTOV options may be modified, but the default values are recommended and usually there is no need to change the values unless you have a specific reason for doing so.
- 6. These options are new in RadSim v4.0.

The following configuration namelist variables were removed in RadSim v4.0 (compared to RadSim v3.2) as they are no longer supported by RTTOV or are deprecated: *rayleigh_depol, rttov_mfasis_cld_dir, rttov_mfasis_cld_file, rttov_htfrtc_coefs_dir, rttov_htfrtc_static_file, rttov_htfrtc_sensor_file, htfrtc*

4.2 Observation data file

This optional file contains observation metadata. The purpose is to allow simulations to be performed at specific locations with realistic satellite viewing geometry (see section 5.1 for more details). If not specified, the simulation will be performed at model profile locations assuming a nadir zenith angle (this can be changed in the configuration namelist with the *zen_angle* variable).

Two versions are available for observation data files in ASCII format with slightly different formats. Both versions are text files with data for one observation point per line. Version 1 files are the same as used in version 1 of RadSim. It is also possible to provide observation data files in netCDF format which is recommended when simulating a very large number of observations. The formats are described below.

Version 1 (ASCII)

For each observation the following need to be provided, in this order, as a space-separated list of values (all are *real* values):

- Latitude (degrees, range -90.0 -> 90.0)
- Longitude (degrees, range -180.0 -> 180.0, or 0.0 -> 360.0)
- Surface elevation (m)
- Land-sea mask (0.0 -> 1.0) this is effectively a land fraction so 0.0=sea, 1.0=land
- Satellite zenith angle (degrees)



Observation data are preceded by a short header section. An example file (obsdata_example_v1.txt) can be found in the etc subdirectory of the installation directory.

Version 2 (ASCII)

This is similar to version 1 except that a variable number of columns can be supplied, and the order is flexible. The header section defines the order of the columns. The following fields may be supplied:

Field	Name	Index	Туре	Units/comments
Longitude	lon	1	real	degrees, range -180 - 180 or 0 – 360,
				mandatory
Latitude	lat	2	real	degrees, range -90 - 90, mandatory
Surface elevation	zsurf	3	real	m
Land-sea mask	lsm	4	real	land fraction, 0-1
Satellite zenith angle	satzen	5	real	degrees, should not exceed RTTOV
				maximum (~84°)
View identifier	viewid	6	integer	not used by RadSim
Scan line	scanline	7	integer	not used by RadSim
Scan position	scanpos	8	integer	not used by RadSim
Satellite azimuth angle	satazim	9	real	degrees, 0 – 360
Year	year	10	integer	
Month	month	11	integer	1 - 12
Day	day	12	integer	day of month, 1 - 31
Hour	hour	13	integer	0 - 23
Minute	minute	14	integer	0 - 59
Solar zenith angle	solzen	15	real	degrees
Solar azimuth angle	solazim	16	real	degrees, 0 – 360
Footprint semi-major axis	rmajor	17	real	km
Footprint semi-minor axis	rminor	18	real	km
RTTOV surface type	rtsurf	19	integer	0=land, 1=sea, 2=sea-ice
Magnetic field strength	be	20	real	Gauss
Cosine of angle between	cosbk	21	real	range -1 - 1.
magnetic field and radiation path				

Notes:

- View identifier, scan line, and scan position are identifiers/index values, there is no formal definition, and they are not used by RadSim.
- The satellite azimuth angle is relevant for MW sea surface emissivity and solar BRDF calculations, and footprint simulations (see below).
- The solar zenith and azimuth angles are relevant for solar-enabled simulations (*solar*=T): note both solar angles must be specified or neither in this case. The solar zenith angle is required (alone) for NLTE simulations (*nlte_correction*=T).
- Azimuth angles are measured clockwise, with due north being zero and due east being +90°.
- The observation date/time is mandatory if the *temporal_data* option is set to true in the configuration namelist: all fields (year, month, day, hour, minute) must be specified. Model fields are then interpolated in both space and time to each observation. For solar simulations, if the solar angles are not specified then the



date/time fields must be present and RadSim will compute solar zenith and azimuth angles for each observation from the latitude, longitude, date and time.

- If the *enable_footprints* option is set to true in the configuration namelist for footprint simulations, the footprint radii must both be specified along with the satellite azimuth angle (see section 5.1.3 and Annex H).
- For greater control over the surface type associated with each observation you can specify this explicitly, but this is ignored if *multi_surface*=T (section 5.5) or *enable_footprints*=T (section 5.1.3).
- If running simulations with a Zeeman-enabled RTTOV optical depth coefficient file (section 4.4), the magnetic field variables must be specified: these are input directly to the RTTOV *Be* and *cosbk* profile variables.

An example ASCII file (obsdata_example.txt) can be found in the etc subdirectory of the installation directory.

netCDF observation data files

A valid netCDF observation data file contains one-dimensional datasets with the names and types given in the table above. Only mandatory fields must be supplied, and all others are optional as in the version 2 ASCII observation data files. All datasets have the same dimension/size which is the number of observations to be simulated. The name of this dimension is arbitrary.

A python script (radsim_geo_obs.py) is provided in the src/scripts subdirectory which can be used to generate ASCII or netCDF obs data files for simulating a full disc scan (or subset thereof) for many geostationary imagers. This is described in Annex B.

4.3 Model data files

Atmospheric profile data can be obtained from a number of sources. The following are currently supported (the file type code used in the configuration namelist is given in parentheses):

- Met Office UM fieldsfiles/PP files (<u>file type 0</u>)
- GRIB files:
 - data from ECMWF including CAMS (*file type 1*)
 - data from the DWD ICON model (*file type 6*)
 - data from the HARMONIE-AROME model (*file type 7*)
 - data from JMA (<u>file type 8</u>)
- NetCDF files:
 - data from ECMWF (excluding CAMS) (<u>file type 5</u>)
- NWP SAF 60L profile dataset from ECMWF analyses (*file type 2*)
- NWP SAF 91L profile dataset from ECMWF short-range forecasts (*file type 3*)
- NWP SAF 137L profile dataset from ECMWF short-range forecasts (*file type 4*)
- NWP SAF 60L profile dataset from MACC short-range forecasts (*file type 9*)
- NWP SAF 60L profile dataset from CAMS short-range forecasts (*file type 10*)
- NWP SAF 137L profile dataset from CAMS short-range forecasts (*file type 11*)



Required Fields

Regardless of the data source, the following fields must be available in the file, either directly or in an equivalent form that can be converted easily. Available conversions are described in section 5.3. Further details on file contents can be found in the following subsections. If optional features of RTTOV are enabled (see Table 4.1 and example configuration files radsim_cfg_*.nl) then additional fields are required. These are covered in the section Optional Fields below.

Pressure levels

Surface pressure is always required. Generally, it is possible to specify either the pressure half-levels or pressure full-levels. In some cases, neither are necessary if coefficients are available to compute them. See section 5.4 for full details about pressure levels for the various model inputs.

Required fields on full-levels (the same levels must be used in each case)

- Temperature (K)
- Specific humidity (kg/kg)

Required surface fields

- Skin temperature
- Surface (10m) wind speed (as separate latitudinal and longitudinal or u and v components)
- Land-sea mask
- Orographic height
- Sea-ice fraction (see note below)

Note: sea-ice fraction is not compulsory (the simulations will still run) because it may not be a relevant parameter for certain profile data sets e.g. for a limited area model that does not cover high latitudes, but it should always be provided if possible.

Optional Fields

The following fields are required for the indicated options (see Table 4.1) *in addition* to the required fields listed above.

Option use_t2m=T (enabling use of near-surface temperature)

• Temperature (at 2m, K)

Option use_q2m=T (enabling use of near-surface water vapour)

• Specific humidity (at 2m, kg/kg)



Option o3_data=T (enabling variable ozone in simulations)

Required on full-levels:

• Specific ozone concentration (kg/kg or mg/kg)

If ozone is an optional variable gas for the RTTOV optical depth coefficient file being used and no ozone field is provided, RTTOV uses its own background profile for ozone in the simulations. Information on the RTTOV background profiles is available in the RTTOV documentation and on the RTTOV web site:

https://nwp-saf.eumetsat.int/site/software/rttov/download/coefficients

Option co2_data=T (enabling variable CO2 in simulations)

Required on full-levels:

• Specific CO₂ concentration (kg/kg)

See ozone above. Note that if a *co2_max_ppmv* value is specified it will take precedence over the use of the CO₂ fields in the model data file.

Option $n2o_data=T$ (enabling variable N_2O in simulations)

Required on full-levels:

• Specific N₂O concentration (kg/kg)

See ozone above.

Option co_data=T (enabling variable CO in simulations)

Required on full-levels:

• Specific CO concentration (kg/kg)

See ozone above.

Option $ch4_data=T$ (enabling variable CH_4 in simulations)

Required on full-levels:

• Specific *CH*⁴ concentration (kg/kg)

See ozone above.



Option so2_data=T (enabling variable SO₂ in simulations)

Required on full-levels:

• Specific SO₂ concentration (kg/kg)

See ozone above.

Option clw_data=T (enabling of cloud liquid water emission in non-scattering microwave simulations)

Required on full-levels:

• Specific liquid water content (cloud-liquid-water, kg/kg)

Option hydrometeors=T (hydrometeor scattering)

Required on full-levels (cloud/hydrometeor concentrations are grid-box average values):

- Specific liquid water content (cloud-liquid-water, kg/kg)
- Specific ice content (cloud-ice-water, kg/kg)
- Rain content/rate (optional, microwave sensors only, kg/kg or kg/m²/s)
- Snow content/rate (optional, microwave sensors only, kg/kg or kg/m²/s)
- Area cloud fraction (0-1)

Optional for ICON GRIB files:

- Cloud liquid water radius (m)
- Cloud ice water radius (m)
- Density (kg/m³)

Option aerosols=T (aerosol scattering for CAMS or ICON-ART species)

Optional on full-levels (values are grid-box average specific concentrations):

- Sea salt size bin 1 (CAMS / ICON-ART, kg/kg)
- Sea salt size bin 2 (CAMS / ICON-ART, kg/kg)
- Sea salt size bin 3 (CAMS / ICON-ART, kg/kg)
- Dust size bin 1 (CAMS / ICON-ART, kg/kg)
- Dust size bin 2 (CAMS / ICON-ART, kg/kg)
- Dust size bin 3 (CAMS / ICON-ART, kg/kg)
- Hydrophilic organic matter (CAMS only, kg/kg)
- Hydrophobic black carbon / Soot (CAMS / ICON-ART, kg/kg)
- Sulphate (CAMS only, kg/kg)

4.3.1 Met Office UM Fieldsfiles / PP files (file type 0)

These must contain *unpacked* data. RadSim will detect if this is not the case and exit without further processing. If access to UM utility scripts is available, then the *ieee* tool can be used to unpack the data prior to running the simulation, otherwise you will need to request an unpacked file from your data provider (warning note: unpacked files may be up to an order of magnitude larger than packed files, it's best to filter out unwanted fields when unpacking the file – use T+0 fields and the STASH codes defined below).

PP files must have the '.pp' suffix in the file name. Note that PP files should contain 32-bit values. 64-bit PP files are non-standard and not supported.

The file should contain the STASH items listed in the following table. They should be valid for the same validity time. If STASH items are present for more than one validity time, then the earliest validity time will be selected unless the *temporal_data* option in the configuration namelist is set to true.

STASH code	Description
4 (or 16004)	Theta (or temperature)
10	Specific humidity
407	Pressure half-levels (rho levels), optional but strongly recommended (see note 1)
408	Pressure full-levels (theta levels), optional (see note 1)
409	Surface pressure
24	Surface skin temperature
3209	10m wind (u-component)
3210	10m wind (v-component)
31	Sea-ice fraction
33	Orography

The following STASH items may also be required depending on what options are selected.

STASH code	Description	Usage notes
3236	Temperature at 1.5m	Required for use_t2m
3245 or 3237	Relative or specific humidity at 1.5m	Required for use_q2m
60	Ozone (mixing ratio)	Required for o3_data
254	Specific liquid water content (qcl)	Required for clw_data, hydrometeors
12	Specific ice content (qcf)	Required for hydrometeors (see note 2)
271	Specific ice content (qcf2) - RAL3	Optional for hydrometeors (see note 2)
186	Rain rate	Optional for hydrometeors (MW only, see note 3)
187	Snow rate	Optional for hydrometeors (MW only, see note 2)
265	Area cloud fraction	Optional for hydrometeors but strongly
		recommended (see note 4).
266	Bulk cloud fraction	Optional for <i>hydrometeors</i> only if area cloud fraction unavailable (see note 4).

Notes:

1. It is strongly recommended that the model data file includes stash 407 (pressure on rho levels). In that case stash 408 (pressure on theta levels) is optional as these are computed by RadSim if not found. If stash 407 is not present, then stash 408 must



be included in the file, and the pressures on rho levels will be computed, but this is an approximation and is not strictly consistent with the true UM rho level pressures.

- 2. UM diagnostic output usually includes all ice, including precipitating ice, in the *qcf* field (STASH code 12). This is used with the ice cloud particle properties in the default RTTOV v14 hydrotables if running for a MW sensor with the *hydrometeors* option. For RAL3 the model cloud ice and precipitating ice have been separated into the *qcf* and *qcf2* fields: it is recommended that both are present (in which case they are summed in cloud ice), otherwise at least one of them should be present. The capability to use snow rate has been included but is not currently applicable.
- 3. Rain is not compulsory the simulation can still be run without a rain-rate field if it's not available.
- 4. Either area cloud fraction or bulk cloud fraction must be present for cloud simulations. Area cloud fraction is strongly recommended in preference.

4.3.2 GRIB files (file types 1, 6, 7 and 8)

GRIB files originating from ECMWF, and JMA, and GRIB files containing data from the HARMONIE-AROME and DWD ICON (and ICON-ART) models are supported. Support for additional sources may be added in later releases. You must specify the correct value in the *model_filetype* configuration namelist variable.

ECMWF GRIB data (*file type 1*)

For ECMWF data, model fields must be specified on a regular latitude/longitude grid. Example MARS/CDS/ADS requests are given in Annex D. The file should contain the following fields as defined by the ecCodes *ParamId* key value.

ParamId	Description		
3 (or 130)	Theta (or temperature)		
133 (or 157)	Specific (or relative) humidity		
54	Pressure levels (optional, see note 1 below)		
134 (or 152)	Surface pressure		
235	Surface skin temperature		
165	10m wind (u-component)		
166	10m wind (v-component)		
31	Sea-ice fraction		
129	Geopotential (for orography)		
172	Land/sea mask (land fraction)		

The following fields may also be required depending on which options are selected.

ParamId	Description	Usage notes
167	Temperature at 2m	Required for use_t2m
168	Dewpoint temperature at 2m	Required for use_q2m
203	Ozone (mixing ratio)	Required for o3_data
246	Specific liquid water content (qcl)	Required for <i>clw_data, hydrometeors</i> (see
		section 5.9)
247	Specific ice content (qcf)	Required for hydrometeors (see section 5.9)
75	Rain (mixing ratio)	Optional for hydrometeors (MW only)
76	Snow (mixing ratio)	Optional for hydrometeors (MW only)
248	Area cloud fraction	Required for hydrometeors

210001	CAMS sea salt size bin 1 (mixing ratio)	Optional for <i>aerosols</i> (see section 5.10)
210002	CAMS sea salt size bin 2 (mixing ratio)	Optional for <i>aerosols</i>
210003	CAMS sea salt size bin 3 (mixing ratio)	Optional for <i>aerosols</i>
210004	CAMS dust size bin 1 (mixing ratio)	Optional for aerosols
210005	CAMS dust size bin 2 (mixing ratio)	Optional for aerosols
210006	CAMS dust size bin 3 (mixing ratio)	Optional for <i>aerosols</i>
210007	CAMS hydrophilic organic matter (mixing ratio)	Optional for aerosols
210010	CAMS hydrophobic black carbon (mixing ratio)	Optional for <i>aerosols</i>
210011	CAMS sulphate (mixing ratio)	Optional for <i>aerosols</i>
164	Total cloud cover	Not used in simulations (see note 2)
186	Low cloud cover	Not used in simulations (see note 2)
187	Medium cloud cover	Not used in simulations (see note 2)
188	High cloud cover	Not used in simulations (see note 2)

Notes:

- Pressure full-levels are usually not included explicitly and are instead reconstructed from the surface pressure field and a set of coefficients. Note that ingest of pressure half-levels is not currently supported so these must be computed (either from pressure full-levels and the surface pressure or using the surface pressure and the relevant coefficients).
- 2. If present, these additional fields will be read in, interpolated onto the observation times/locations, and written out to the output file along with the other profile data. They are not used in the simulations.

DWD ICON and ICON-ART GRIB data (file type 6)

For ICON and ICON-ART data, the model data are specified on an unstructured triangular grid. Most fields are provided in the file specified in the *model_datafile* configuration namelist variable. However, a few model fields are fixed and must be provided to RadSim in a separate "invariant" input file specified in the *model_ancil_datafile* variable.

The files should contain the following fields as defined by the ecCodes *Paramld* key values. Multiple parameter IDs are available for some variables: depending on environment variable settings for ecCodes definitions, different parameter IDs can be returned by the ecCodes routines and both alternatives are listed below. The corresponding variable *shortName* key values and ICON variable names are also given:

ParamId(s)	ShortName(s)	ICON var name	Description
130 / 500014	t	temp	Temperature on full levels
133 / 500035	q / qv	qv	Specific humidity on full levels
54 / 500001	pres / p	pres	Pressure full-levels (optional, see note 2)
54 / 500001	pres / p	pres_ifc	Pressure half-levels (optional, see note 2)
134 / 500000	sp / ps	pres_sfc	Surface pressure
500010	t_g	t_g	Surface skin temperature
165 / 500027	10u / u_10m	u_10m	10m wind (u-component)
166 / 500029	10v / v_10m	v_10m	10m wind (v-component)
31 / 500069	ci / fr_ice	fr_seaice	Sea-ice fraction
172 / 500054	lsm / fr_land	fr_land	Land/sea mask (see note 1)
500007	hsurf	hsurf	Land surface height (see note 1)
502309 / 250003	clat	clat	Latitude (see note 1)
502310 / 250004	clon	clon	Longitude (see note 1)

WP SAF

The following fields may also be required depending on which options are selected.

ParamId(s)	ShortName(s)	ICON var name	Description	Usage notes
167 / 500011	2t / t_2m	t_2m	Temperature at 2m	Required for use_t2m
168 / 500017	2d / td_2m	td_2m	Dewpoint temperature at 2m	Required for use_q2m
203 / 500242	o3	o3	Ozone (mixing ratio)	Required for o3_data
503056	qc_dia	tot_qc_dia	Total specific liquid water content	Required for <i>clw_data,</i> <i>hydrometeors</i> (see section 5.9)
503057	qi_dia	tot_qi_dia	Total specific ice content	Required for <i>hydrometeors</i> (see section 5.9)
260020 / 500102	rwmr / qr	qr	Rain (mixing ratio)	Optional for <i>hydrometeors</i> (MW only)
260021 / 500103	snmr / qs	qs	Snow (mixing ratio)	Optional for <i>hydrometeors</i> (MW only)
260257 / 500098	ccl / clc	clc	Cloud cover	Required for hydrometeors
500779 / 503566	recloud	reff_qc	Cloud liquid water effective radius	Optional for <i>hydrometeors</i> with <i>visir_clw_properties</i> = 2 (VIS/IR only, see note 3)
500781 / 503568	reice	reff_qi	Cloud ice water effective radius	Optional for <i>hydrometeors</i> with <i>visir_ice_properties</i> = 1 (VIS/IR only, see note 3)
3089 / 500545	den	rho	Density	Optional for <i>hydrometeors</i> with <i>visir_clw_properties</i> = 2 (VIS/IR only, see note 4)
503699	bcarb	soot	Hydrophobic black carbon / soot	Optional for <i>aerosols</i>
503241	dusta	dust_insol_acc	Dust size bin A	Optional for <i>aerosols</i>
503242	dustb	dust_insol_coa	Dust size bin B	Optional for <i>aerosols</i>
503243	dustc	dust_giant	Dust size bin C	Optional for <i>aerosols</i>
503247	seasa	nacl_sol_acc	Sea salt size bin A	Optional for <i>aerosols</i>
503248	seasb	nacl_sol_coa	Sea salt size bin B	Optional for <i>aerosols</i>
503249	seasc	nacl_giant	Sea salt size bin C	Optional for <i>aerosols</i>

Notes:

- 1. Land/sea mask (172 or 500054), land surface height (500007), latitude (502309 or 250003), longitude (502310 or 250004) are invariant fields and must be specified in the invariant *model_ancil_datafile* as noted above. The latitude and longitude correspond to the centre of each grid triangle ("clat" and "clon").
- 2. The pressure half- and full-levels are required inputs to RTTOV simulations when using ICON data. The two fields use the same parameter IDs but are distinguished by the *typeOfLevel* attribute which is "*generalVerticalLayer*" for full-levels and "*generalVertical*" for half-levels. Note either the half-levels or the full-levels must be present in the model data file (or both). If either is not present, RadSim computes them from the other on the assumption that full-level pressures are the mean of the log of the adjacent half-level pressures (this gives half-level pressures have a relative difference up to ~10^-5 to those from the ICON model, i.e., within 0.01 hPa at 1000 hPa).
- 3. For VIS/IR cloud scattering simulations (*hydrometeors* true) using cloud liquid or ice optical properties that explicitly depend on particle size (*visir_clw_properties=2, visir_ice_properties=1*), the ICON cloud liquid or ice effective radius fields may be



present in the file. To enable use of them, the *use_model_clw_deff* and/or *use_model_ice_deff* configuration namelist options must be set to true. In this case the explicit fields will be used instead of the respective RTTOV particle size parameterisations. See Annex F for more information about how these fields are used by RadSim. Note that the ICON fields are effective radii in metres. When profile data are written to the output netCDF file, these fields are converted to effective diameters in microns.

4. The density is used only for the parameterisation of cloud liquid water particle effective diameter which is required with the RTTOV CLW Deff liquid water optical properties (*visir_clw_properties* = 2). If the density field is not present, the parameterisation is computed by RTTOV using its own calculation of the air density. The parameterisation is given in Annex F.

HARMONIE-AROME GRIB data (*file type 7*)

For HARMONIE-AROME data, model fields must be specified on a regular latitude/longitude grid. Pressure levels are always computed, so the GRIB file must contain the coefficients for this calculation unless the data are on 65 levels in which case the coefficients are stored in the RadSim code. HARMONIE-AROME fields are identified using the ecCodes *indicatorOfParameter* and *indicatorOfTypeOfLevel* key values. In addition, the *level* key is required to identify 2D fields. The following fields are mandatory.

indicatorOf	indicatorOf	level	Description
Parameter	TypeOfLevel		
11	109	-	Temperature
51	109	-	Specific humidity
1	105	0	Surface pressure
11	105	0	Surface skin temperature
33	105	10	10m wind (u-component)
34	105	10	10m wind (v-component)
91	105	0	Sea-ice fraction
6	105	0	Geopotential (for orography)
81	105	0	Land/sea mask

The following fields may also be required depending on which options are selected.

indicatorOf	indicatorOf	level	Description	Usage notes
Parameter	TypeOfLevel			
11	105	2	Temperature at 2m	Required for use_t2m
17	105	2	Dewpoint temperature at	Required for use_q2m
			2m	
66	105	0	Show depth	Optional, only used with IR land
00	105	0		surface emissivity atlases, see note 1
76	109	_	Specific liquid water content	Required for clw_data, hydrometeors
70	109	-	(qcl)	(see section 5.9)
58	100	_	Specific ice content (acf)	Required for hydrometeors (see
50	109	-	Specific ice content (qci)	section 5.9)
181	109	-	Rain (mixing ratio)	Optional for hydrometeors (MW only)
184	109	-	Snow (mixing ratio)	Optional for hydrometeors (MW only)
71	109	-	Area cloud fraction	Required for hydrometeors
71	105	0	Total cloud cover	Not used in simulations (see note 2)
73	105	0	Low cloud cover	Not used in simulations (see note 2)



74	105	0	Medium cloud cover	Not used in simulations (see note 2)
75	105	0	High cloud cover	Not used in simulations (see note 2)

Notes:

- Snow depth (in metres) is optional. If present the RTTOV snow_fraction profile variable is set to snow_depth / 0.1, this value being clipped to lie in the interval [0,1]. This simple linear relationship where 10cm of snow implies full (100%) snow cover is broadly similar to the relationships used in the HARMONIE-AROME model (Sander Tijm, KNMI, personal communication). In RTTOV, the snow_fraction is only used by the IR land surface emissivity atlases.
- 2. If present, these additional fields will be read in, interpolated onto the observation times/locations, and written out to the output file along with the other profile data. They are not used in the simulations.

JMA GRIB data (file type 8)

GRIB data from JMA including JRA-55 reanalysis datasets and fields from the operational NWP model are supported. Currently only clear-sky simulations are supported as cloud fields were not present in the test datasets: hydrometeor simulations may be implemented in a future release. The GRIB files should include the coefficients for calculating the pressure levels. If these coefficients are not present, then RadSim uses the *NV* variable to determine the number of levels. In this case the fields must be on either 60 or 100 vertical levels: RadSim has the relevant coefficients for these. If *NV* is set to zero, RadSim assumes fields are on 60 levels.

Model fields must be supplied on a regular latitude/longitude grid. JMA fields are identified using the ecCodes *ParamId* and *typeOfLevel* key values. In some cases, multiple *ParamIDs* are supported. The following fields are supported and are mandatory except for ozone.

ParamId	typeOfLevel	Description	
130	hybrid	Temperature	
133	hybrid	Specific humidity	
54 / 134	surface	Surface pressure	
194	surface	Surface skin temperature	
131 / 165	heightAboveGround	10m wind (u-component)	
132 / 166	heightAboveGround	10m wind (v-component)	
130 / 167	heightAboveGround	Temperature at 2m (only required for use_t2m)	
133 / 174096	heightAboveGround	Specific humidity at 2m (only required for use_q2m)	
31 / 3091	surface	Sea-ice fraction	
129	surface	Geopotential (for orography)	
172	surface	Land/sea mask (land fraction)	
203	hybrid	Ozone (mixing ratio) (see note 1)	
260624	hybrid	Ozone (mixing ratio) (see note 1)	

Notes:

1. Ozone is optional. The two ozone *ParamIDs* correspond to fields with different units (kg/kg and mg/kg respectively). Only one of the two fields should be supplied, and one must be present if the *o3_data* namelist configuration variable is set to true.



4.3.3 NetCDF files (*file type 5*)

RadSim can currently ingest netCDF files obtained by running the *grib_to_netcdf* tool (from the ecCodes library) on ECMWF files containing GRIB data. Other sources of data may work so long as they conform to the standards/format of the files produced by *grib_to_netcdf* as described below. Additional data sources may be explicitly added in later releases.

ECMWF netCDF data

Model fields must be specified on a regular latitude/longitude grid. See section 4.3.2 on ECMWF GRIB data. GRIB data obtained via MARS or CDS can be converted to netCDF format using the *grib_to_netcdf* tool for use with RadSim but note that the *-T* option must *not* be used.

NetCDF data are supplied to RadSim in two files, one containing atmospheric/multi-level data in the *model_datafile* namelist configuration variable, and the second containing surface/single-level data in the *model_ancil_datafile* variable. The two files must be consistent in terms of the spatial grid and time steps.

The files must contain the following dimensions: *latitude, longitude, time* or *valid_time*. In addition, the atmospheric data file must contain the *level* or *model_level* dimension. All atmospheric variables must have dimensions (*time/valid_time, level/model_level, latitude, longitude*) and all surface (single-level) variables must have dimensions (*time/model_time, latitude, longitude*). Note that even if the files contain data for only a single validity/data time, the *time/valid_time* dimension must be present and, in that case, it is of length one. By default, *grib_to_netcdf* writes variables as SHORT integer values. RadSim will ingest these, but the variables can alternatively be of any numerical data type.

All variables containing model fields should have a numerical *missing_value* or *GRIB_missingValue* attribute specifying the missing value indicator. If variables containing model fields have either *scale_factor* or *add_offset* attributes (or both), the corresponding linear transformation (scale and offset, in that order) is applied to the data.

The files must contain *latitude* and *longitude* variables (floats or doubles, each sized according to the corresponding dimension) containing the latitudes and longitudes which define the grid.

If the integer *time* variable is present, containing the values of the *time* dimension, it must specify hours since some particular date/time (by default the date/time is midnight on 01/01/1900). RadSim obtains the reference date from the *units* attribute of the of the *time* variable. This is a character string with the following format:

"hours since YYYY-MM-DD HH:MM:SS.0"

If the integer *valid_time* variable is present, containing the values of the *valid_time* dimension, it must specify seconds since some particular date/time (by default the date/time is midnight on 01/01/1970). RadSim obtains the reference date from the *units* attribute of the of the *valid_time* variable. This is a character string with the following format:



"seconds since YYYY-MM-DD"

RadSim relies on the attribute having one of these formats to determine the validity and data times for the model fields.

The files (atmospheric/multi-level and surface/single-level) should contain the following named variables. The corresponding ECMWF ecCodes *Paramld* key value is given in brackets.

Variable name	Description
(GRIB ParamID)	(atm = atmospheric file, sfc = surface file)
t (130)	Temperature, atm
q (133)	Specific humidity, atm
p (54)	Pressure full-levels, atm (optional, see note 1 below)
sp (134)	Surface pressure, <i>sfc</i>
skt (235)	Surface skin temperature, sfc
u10 (165)	10m wind (u-component), sfc
v10 (166)	10m wind (v-component), sfc
siconc (31)	Sea-ice fraction, sfc
z (129)	Geopotential (for orography), sfc
lsm (172)	Land/sea mask (land fraction), sfc

The following fields may also be required depending on which options are selected.

Variable name	Description	Usage notes
(GRIB ParamID)	(atm = atmospheric file, sfc = surface file)	
t2m (167)	Temperature at 2m, sfc	Required for use_t2m
d2m (168)	Dewpoint temperature at 2m, sfc	Required for use_q2m
o3 (203)	Ozone (mixing ratio), atm	Required for o3_data
clwc (246)	Specific liquid water content (qcl), atm	Required for clw_data, hydrometeors
		(see section 5.9)
ciwc (247)	Specific ico content (acf) atm	Required for hydrometeors (see section
		5.9)
crwc (75)	Rain (mixing ratio), atm	Optional for hydrometeors (MW only)
cswc (76)	Snow (mixing ratio), atm	Optional for hydrometeors (MW only)
cc (248)	Area cloud fraction, atm	Required for hydrometeors
tcc (164)	Total cloud cover, sfc	Not used in simulations (see note 2)
lcc (186)	Low cloud cover, sfc	Not used in simulations (see note 2)
mcc (187)	Medium cloud cover, sfc	Not used in simulations (see note 2)
hcc (188)	High cloud cover, sfc	Not used in simulations (see note 2)

Notes:

- 1. Pressure levels are usually not included explicitly and need to be reconstructed from the surface pressure field and a set of coefficients and ingest of pressure half-levels is not currently supported. Note that the netCDF files do not contain these coefficients: RadSim has coefficients for the ECMWF 60, 91 and 137 levels so netCDF files must contain fields on one of these level sets or they must contain the pressure full-levels.
- 2. If present, these additional fields will be read in, interpolated onto the observation times/locations, and written out to the output file along with the other profile data. They are not used in the simulations.

4.3.4 NWP-SAF profile datasets (*file types 2, 3, 4, 9, 10, 11*)

RadSim supports reading the NWP SAF profile datasets from ECMWF analyses/short-range forecasts on 60L, 91L, and 137L (file types 2, 3, and 4 respectively), the 60L profile dataset from MACC short-range forecasts (file type 9), the 60L profile dataset from CAMS short-range forecasts (file type 10), and the 137L profile dataset from CAMS short-range forecasts (file type 11).

These datasets contain the fields required to run the optional hydrometeor (scattering) simulations if desired, or CLW absorption for MW non-scattering simulations. Note that the 60L ECMWF dataset (file type 2) does not contain rain or snow data. All datasets contain ozone data. The 60L MACC dataset includes CO₂ and CH₄ data and can also be used for aerosol simulations with a CAMS aerosol optical property file. The 60L CAMS dataset includes CO and SO₂ data. The 137L CAMS dataset includes all optional variable trace gases supported by RTTOV (O₃, CO₂, N₂O, CO, CH₄, SO₂) and can be used for aerosol simulations, a suitable RTTOV optical depth coefficient file must be used that supports the relevant variable gas(es). To include the gas fields in the simulation set the corresponding flag to true (e.g., *co2_data* for CO₂).

The 91L and 137L ECMWF datasets, and the 60L and 137L CAMS datasets include a date for each profile, but no time information. Therefore, if you run solar simulations for these datasets without specifying fixed solar zenith and azimuth angles in the configuration namelist, all profiles are simulated using solar angles calculated assuming a validity time of 00UTC (see section 5.8). The 60L ECMWF and MACC datasets include the hour (0, 6, 12, 18) associated with each profile along with the date, and these are used to calculate solar angles if necessary.

These profile datasets are neither arranged on a regular lat-lon grid nor representative of a single data-time so they cannot be interpolated to observation positions or times. Therefore, simulations are performed at profile locations. If an observation data file is specified in the configuration file, then it is simply ignored in these cases.

If an emissivity and/or BRDF atlas is used with a profile dataset, then by default atlas data for the month of the *first* profile are used for all profiles in the dataset. If the *use_all_atlas_months* configuration namelist variable is set to true, then RadSim will load the atlas data for all months and use the correct atlas data for each profile. Be aware that this option may require a large amount of memory as atlas data for all 12 months are loaded before the simulations commence.

To specify the NWP SAF 91L or 137L datasets the *model_datafile* variable should point to the relevant *.atm file. RadSim will then look for the associated *.sfc file in the same directory. For the other datasets, simply set the *model_datafile* variable to the required profile file. For the profile datasets in ASCII format with multiple files differentiated by variable, you can concatenate the files (belonging to the same dataset). For example, this allows you to simulate all profiles from one dataset together.

The documentation accompanying the datasets gives further details about the contents.



4.3.5 Temporal interpolation

Temporal interpolation is supported for ECMWF GRIB and netCDF data, DWD ICON GRIB data, HARMONIE-AROME GRIB data, and Met Office UM fieldsfiles (but not UM PP files or the NWP SAF profile datasets). The model data file(s) should contain the same fields on the same grid for two or more forecast times (the exception to this is the ICON invariant data file which is read in once and the same fields are applied to all model time steps).

RadSim *always* reads model data for all time steps in the input file: this can therefore potentially require a large amount of memory. The only exception to this is for Met Office fieldsfiles where you can specify a range of times to read in via the *fcst_range* namelist variable. For example, if you set *fcst_range* = 3, 9 then only timesteps between T+3 and T+9 (inclusive) will be read from the model file. In general, it is recommended to restrict the model data to only those fields and time steps required. Model data files must not contain fields for more than one thousand time steps (it is highly unlikely that this limit will be exceeded in practice).

Temporal interpolation is activated by setting *temporal_data* to true in the configuration namelist. In this case an *obs_datafile* must be specified and it must contain *year, month, day, hour* and *minute* columns specifying the date/time of each observation. Any observations that fall outside the range of times represented in the model data will be flagged (see section 5.13) and will be excluded from the simulations.

All model fields are linearly interpolated in time by default. If the *interp_nn_temporal* configuration namelist option is set to true, then fields from the nearest time step are used instead. If an emissivity and/or BRDF atlas is used, the atlas data for the month of the first model file validity time are used for all simulations. (NB the *use_all_atlas_months* option has no effect with temporal interpolation).

4.4 RTTOV Coefficients

Some details about RTTOV coefficients including a summary of the versions available can be found in the RTTOV Users Guide under the section "Summary of RTTOV concepts". Further information on the files can be found in the RTTOV section of the NWP SAF website:

https://nwpsaf.eu/site/software/rttov/download/coefficients/

Many of the smaller optical depth coefficient files are provided in the RTTOV distribution (subdirectory rtcoef rttov14). Look there first before downloading any files. RTTOV optical depth coefficient files are organised according to the predictor version and the number of coefficient levels and can be found in subdirectories named rtcoef rttov14/rttovXpredYL/ where X is the (13 predictor version is recommended) and Y is the number of levels (usually 54 for multispectral sensors and 101 for hyperspectral sounders). RTTOV v14 currently supports all ASCII optical depth coefficient files compatible with RTTOV v13 but see the RTTOV coefficient web page above for more information about recommended and deprecated coefficient files, and also to download larger coefficient files (e.g., for hyperspectral sounders) in netCDF format.



When enabling trace gases in the simulation (O₃, CO₂, N₂O, CO, CH₄, SO₂) you must ensure the coefficient file being used supports the variable gas(es) in question.

When assigning the path to the coefficients directory in the control namelist file, assign to the appropriate subdirectory e.g.,

rttov_coefs_dir=/path_to_rttov14/rtcoef_rttov14/rttov13pred54L

You can specify the explicit filename for the coefficient file in the *rttov_coef_file* configuration namelist variable: RadSim will look for this file in the *rttov_coefs_dir* directory. If you do not specify the filename, the name of the file is constructed in RadSim from the (*platform, satid, instrument*) triplet specified in the configuration namelist file. For the optical depth coefficient files, the filename is constructed as *rtcoef_platform_satid_instrument.dat/nc* (see below for more information about the file extension). Many optical depth coefficient files now include additional information such as the variable gases supported. If you are not specifying the filename explicitly, you can rename coefficient files (where required) or create symbolic links to match the above convention. Alternatively, a configuration variable *rttov_coefs_options* is available which can be used to specify any additional text that appears in the filename between "instrument" and the file extension (".dat"/".nc") such as "_o3co2" or "_7gas".

Depending on the type of simulation, additional coefficient files may be required: hydrometeor and/or aerosol optical property files, MFASIS-NN coef files, and PC-RTTOV coefficient files. In each case, there is a namelist variable for the directory containing the file(s) which must be specified, and then you can either specify the filename explicitly (recommended) or let RadSim construct the filename, noting that some filenames contain extra identifying information that RadSim will not include and you may need to rename files or create symbolic links unless you specify the filenames explicitly.

The *rttov_coefs_type* variable specifies the file extension used when RadSim constructs the filenames. It can be set to ".nc" to read netCDF input files instead of ASCII files (default: ".dat"). This applies to all input files except the ARO scaling look-up table file (see below).

The *rttov_coefs_dir* variable is mandatory, and an RTTOV optical depth coefficient file must be supplied for all simulation types.

For aerosol scattering simulations using CAMS or ICON-ART fields, an RTTOV CAMS or ICON-ART (respectively) aerosol optical property file (prefix "rttov_aertable_") must also be supplied: it must contain optical properties for the same set of channels as the optical depth coefficient file.

For hydrometeor scattering simulations, a hydrometeor optical property file (prefix "rttov_hydrotable_") must also be supplied: it must contain optical properties for the same set of channels as the optical depth coefficient file. For MFASIS-NN simulations, an MFASIS-NN coefficient file is required in addition to the hydrometeor optical property file.

For hydrometeor simulations with microwave sensors it is important that either the default NWP SAF hydrotables are used, or otherwise if a custom hydrotable is generated that particles 1, 2, 4 and 5 in the hydrotable file correspond respectively to rain, snow, cloud



liquid water and cloud ice water (this requires that a custom file contains at least 5 hydrometeor types, even though the third one - graupel in the NWP SAF hydrotables - is not currently used). RTTOV provides a new treatment of polarisation, specified by setting the new *mw_pol_mode* option to 2 in the config namelist. In this case, RTTOV requires the "ScalingFactorForBulkProperties.rssp" file. This should be in the same directory as the hydrotable. If the name of the file is changed then it must be specified in the *rttov_pol_coef_file* config namelist option, otherwise the default filename will be assumed.

For PC-RTTOV simulations, a PC coefficient file (prefix "pccoef_") must be supplied. This must be compatible with the supplied optical depth coefficient file: see the RTTOV user guide for more information.

4.5 Output files

4.5.1 Naming convention

Output data are written to netCDF-4 classic format using internal compression for array variables. You can optionally specify the output file name in the *output_file* configuration namelist variable. Otherwise, RadSim creates a file name with the following naming convention:

radsim-<platform>_<satid>_<instrument>-<datatime>.nc e.g. radsim-metop_2_amsua-201401010012.nc

The data-time suffix (of the form YYYYMMDDHHMM) is optional and will be omitted, along with the last '-' separator, if the profile dataset is not related to a single validity time (i.e. the NWP SAF profile datasets). The data-time corresponds to the validity time of the first set of fields in the file. By default, the output file is written to the current directory, but you can specify an alternative directory in the *output_dir* configuration namelist variable.

4.5.2 Contents

The following *attributes* are written to the file

- *platform, satid, instrument* character strings identifying the simulated instrument.
- channels integer array of the channels used.
- *validity_time* 5 element integer array containing year, month, day, hour, minute. This is not written if the profile data have multiple validity times.
- data_time 5 element integer array containing year, month, day, hour, minute. This is
 not written if the profile data have multiple validity times. This may differ to the validity
 time, for example, if the data correspond to a forecast.
- *wavenumbers* array of wavenumber values corresponding to the channels used, taken from the RTTOV coefficient file.
- *RMDI* fill value used when floating point data are missing
- *IMDI* fill value used when integer data are missing

Most output fields are floating point arrays with the first dimension the number of simulated observations (labelled *obs* in the file). Other dimensions may be *levels* (the number of



pressure half-levels, see section 4.3), *layers* (the number of pressure full-levels), *surfaces* (the number of surfaces associated with each simulated profile, typically either 3 or 1 depending on whether the *multi_surface* configuration namelist option is true or false respectively, see section 5.5), and *channels*.

For the purpose of the simulation, fields are divided into observation and model fields. The former consist of any information in the observation data file (if applicable), plus any simulated data. The latter consist of model profile data. By default, **all observation fields are written to the file**. Observation fields are listed below:

Fields that are always present:

- *lat, lon* observation coordinates
- satzen satellite zenith angles
- *rtsurf* RTTOV surface type (land, sea or sea-ice)
- *qcflags* quality control flags (see section 5.13 for more details)
- qcrttov RTTOV quality flags (see section 5.13 for more details)
- *qcinfo* additional quality control information (see section 5.13 for more details)
- *bt* OR *radiance* simulated top-of-atmosphere brightness temperatures or radiances; only one of these is written

Fields that may be present depending on the options chosen:

- *refl* simulated top-of-atmosphere reflectances (if solar radiation enabled, but not written if *write_radiances* is true)
- *bt_clear, radiance_clear* simulated top-of-atmosphere clear-sky brightness temperatures or radiances; only one of these is written (output if *write_clearsky* is true).
- *refl_clear* simulated top-of-atmosphere clear-sky reflectances (if solar radiation enabled and *write_clearsky* is true, but not written if *write_radiances* is true)
- *emis* surface emissivities
- *brdf* surface BRDFs (if solar radiation enabled)
- *tjac, qjac, o3jac* temperature, humidity and ozone Jacobians
- *tskinjac, wind10mujac, wind10mvjac, emisjac* skin temperature 10m wind u/v, and surface emissivity Jacobians
- trans level-to-space transmittances (only for IR/MW channels; these are zero for visible/near-IR channels)
- geometric_height_half altitudes of pressure half-levels computed by RTTOV (m)
- *geometric_height* altitudes of pressure full-levels computed by RTTOV (m)
- cads_height_assignment height assignments (interpolated level numbers) consistent with the NWP SAF CADS package (see Annex G)
- *viewed, scanline, scanpos* identifiers/index values that may be provided in the input data file
- *satazim* satellite azimuth angles
- solzen, solazim solar zenith and azimuth angles
- year, month, day, hour, minute observation dates/times
- *footprint_rmajor* observation footprint semi-major axis (km)
- footprint_rminor observation footprint semi-minor axis (km)
EUMETSAT Network of opplication

• *be* – magnetic field strength (Gauss)

NWP SAF

- *cosbk* cosine of angle between magnetic field and radiation path
- nprof_per_footprint the number of profiles averaged in each footprint
- viewed, scanline, scanpos ancillary identifiers from obs data file

Model fields are not written out by default, but they can be included by use of the *write_profiles* option in the namelist file. All fields are those used in the simulation (before conversion to RTTOV units), therefore if simulating at observation positions, they are interpolated values not input values, and there will be as many data points in each field (per level) as there are observations. Also, any input fields that are converted to a required field are discarded after conversion and therefore not available (e.g., dewpoint temperature is converted to specific humidity and then discarded). The full list of possible fields is as follows:

Fields that are always present if using the *write_profiles* option:

- *lat* latitude (degrees, -90 90)
- Ion longitude (degrees, 0-360)
- *lsm* land-sea mask (0-1)
- *zsurf* surface height (m)
- seaice sea-ice fraction (0-1)
- *pstar* surface pressure (Pa)
- tskin surface skin temperature (K)
- *u10* 10m wind u-component (m/s)
- v10-10m wind v-component (m/s)
- ph pressure half-levels (Pa)
- *p* pressure full-levels (Pa)
- *t* temperature profiles (K)
- *q* specific humidity profiles (kg/kg)
- *rh* relative humidity profiles (%)

Fields that may be present depending on the options chosen and the contents of the data file:

- *t2* temperature at 2m (K)
- q2 specific humidity at 2m (kg/kg)
- z height at pressure full-levels (m)
- *clw* cloud liquid water profiles (kg/kg)
- *ciw* cloud ice water profiles (kg/kg)
- *rain* rain profiles (either kg/m²/s or kg/kg, depending on input profile)
- *snow* snow profiles (either kg/m²/s or kg/kg, depending on input profile)
- *cfrac* cloud fraction profiles (0-1)
- *clw_deff* cloud liquid water effective diameter (microns)
- *ciw_deff* cloud ice water effective diameter (microns)
- sea_salt1 CAMS / ICON-ART sea salt size bin 1 (kg/kg)
- sea_salt2 CAMS / ICON-ART sea salt size bin 2 (kg/kg)
- sea_salt3 CAMS / ICON-ART sea salt size bin 3 (kg/kg)



- *dust1* CAMS / ICON-ART dust size bin 1 (kg/kg)
- dust2 CAMS / ICON-ART dust size bin 2 (kg/kg)
- dust3 CAMS / ICON-ART dust size bin 3 (kg/kg)
- *hphil_omat* CAMS hydrophilic organic matter (kg/kg)
- *hphob_bcar* CAMS hydrophobic black carbon / ICON-ART soot (kg/kg)
- sulphate CAMS sulphate (kg/kg)
- o3 ozone profiles (kg/kg)
- co2 CO₂ profiles (kg/kg)
- *n*2*o* N₂O profiles (kg/kg)
- co CO profiles (kg/kg)
- ch4 CH4 profiles (kg/kg)
- so2 SO2 profiles (kg/kg)
- *density* air density (kg/m³)
- *snow_depth* snow depth (m)
- total_cc/low_cc/medium_cc/high_cc unused in simulations but interpolated and written out with other model data if total/low/medium/high cloud cover fields present in ECMWF/HARMONIE-AROME input files.
- year, month, day, hour, minute output per profile for NWP SAF profile datasets only.

When running on a model grid it can be useful to output just the latitude/longitude without the rest of the profile data: this can be achieved by setting *write_latlon* to true in the configuration namelist. The result is a significantly smaller output file that contains enough data to make geographical plots of simulation results.

4.5.3 File size

The size of the output files depends on the configuration, notably the number of instrument channels, the number of profiles and whether some of the more expensive options are chosen such as the production of Jacobians or transmittances (these are 3-dimensional arrays over obs, channels and levels). An AMSU-A simulation in a simple configuration may only require around 5MB, whereas a similar configuration for IASI using all 8461 channels may require considerably more than 2GB. Output files are netCDF-4 classic format using internal compression for array variables.

5. PROCESSING DETAILS

5.1 Use of observation information

Observation meta-data are provided via the observation data file which is described in section 4.2. The purpose is to allow more realistic simulations of a satellite instrument inflight by specifying observation positions and the viewing angle, rather than doing the calculations at model grid points and assuming a fixed view angle.

For Met Office UM fields, HARMONIE-AROME GRIB data, JMA GRIB data, and ECMWF IFS (GRIB or netCDF) data, the model fields must be arranged on a rectangular lat-lon grid



(this includes regular grids with a rotated pole – see section 5.2). DWD ICON data are specified on an unstructured triangular grid.

Model fields are interpolated horizontally to observation positions, and, if temporal interpolation is enabled, an additional linear interpolation to the date/time of the observation is performed. In the case of surface type and surface height, if there are values in the observation data file then they are used in preference to the interpolated values.

For data on a regular lat-lon grid, bilinear interpolation is used by default for the horizontal interpolation. For data on an unstructured grid, the data from the nearest grid point are used. Note that the bi-linear spatial and linear temporal interpolations may lead to physically inconsistent profiles. In addition, it is not clear that interpolation of cloud fields (for example) is generally a good idea. For this reason, two options are provided, *interp_nn_spatial* and *interp_nn_temporal*, both false by default. If set to true, then nearest neighbour interpolation is used spatially and/or temporally (respectively). In other words, the data from the nearest grid point and/or nearest timestep are used.

5.1.1 Horizontal interpolation (regular lat-lon grid)

A standard bi-linear interpolation technique is used. The interpolated value is a weighted combination of the values at the 4 nearest neighbouring points that form an enclosing rectangular *grid cell*.

A regular lat-lon grid is defined with a datum coordinate and values for the lat and lon intervals between adjacent points. For a point P, the grid cell index in each axis direction is then the difference between the datum coordinate and the position of P, divided by the interval.

Model fields are stored as 1D-arrays so the single combined grid cell index is dependent on the ordering of grid points in the field. A regular grid is assumed by default to be arranged row-wise, i.e., the data for the first row is followed by the data for the second row and so-on, however column-wise ordering is also allowed.

For a grid cell with vertices [(x,y), (x+dx,y), (x,y+dy), (x+dx, y+dy)] and corresponding field values $[a_1, a_2, a_3, a_4]$, the interpolated field value *b* is given by:

$$b = (1 - w_x)(1 - w_y)a_1 + w_x(1 - w_y)a_2 + (1 - w_x)w_ya_3 + w_xw_ya_4$$

where the weights w_x and w_y are:

 $w_x = (x_p - x)/dx$ $w_y = (y_p - y)/dy$

If *interp_nn_spatial* is true, the computed weights are reset to the nearest integer (0 or 1).

IETSAT fork of cation cilities	NUMPERCARE	Radiance Simulator v4.0 User Guide	Doc ID Version Date
---	---	---------------------------------------	---------------------------

5.1.2 Temporal interpolation

If temporal interpolation is enabled a simple linear interpolation in time is applied after the horizontal interpolation is applied to the model fields at the time steps immediately before and after the observation time.

For horizontally interpolated data $[b_1, b_2]$ at forecast times $[t_1, t_2]$ respectively, the temporally interpolated field value *c* for observation time t_p ($t_1 \le t_p \le t_2$) is given by:

 $c = (1 - w_t)b_1 + w_t b_2$

Where the weight w_t is: $w_t = (t_p - t_1)/(t_2 - t_1)$ If *interp_nn_temporal* is true, the computed weight is reset to the nearest integer (0 or 1).

5.1.3 Footprint simulation

This option is intended for use with convective-scale model data where the model grid boxes are significantly smaller than the observation footprints and it allows sub-footprint variability in surface properties and cloud fields to be captured in the simulated radiances. It cannot be used with the NWP SAF profile datasets.

RadSim models footprints as ellipses with the semi-major axis aligned along the view path from the observation location to the satellite. Footprint simulations are enabled by setting the *enable_footprints* configuration option to true. In this case, the obs data file must include the satellite azimuth angles (which specifies the orientation of the footprint ellipse), and the semi-major/minor axes of the footprint ellipses in km. For each observation, RadSim simulates radiances for all grid points that lie within the specified ellipse around the observation and calculates the mean radiance. If *write_radiance* is false, brightness temperatures (for channels at wavelengths above 3µm) are calculated from the mean radiances. The brightness temperature calculations are consistent with those done in RTTOV. When solar radiation is enabled and *write_radiance* is false, the mean reflectance is calculated over the grid points within the footprint for channels at wavelengths below 5µm (reflectance is a linear function of radiance).

The footprint calculation itself is described in Annex H. Note that for local area model fields, footprints for observations near the edge of the model grid will only include grid points up to the edge of the grid. RadSim does not flag these "partial" footprints. You must ensure your model data fully cover the observations you are simulating.

If solar radiation is enabled and the observation data file does not contain solar zenith and azimuth angles, RadSim calculates the solar zenith and azimuth angles for the given observation location and applies them to all grid points within the footprint.

It is strongly recommended *not* to specify the land sea mask (land fraction) and orography data in the obs data file for footprint simulations: if these are specified then the given values (surface type and surface height) will be applied to all grid points in the footprint. If these



data are omitted from the obs data file then the model values are used at each grid point: this allows different surface types to be simulated within the footprint, for example.

The footprint simulation can be used with temporal interpolation: the model profiles at grid points within the footprint are interpolated in time and then averaged over the footprint.

If no model grid points are found within a footprint, this is noted in the *qcinfo* quality flag (see section 5.13), and the model grid is interpolated to the observation location as described above. If this happens it suggests that the model resolution is too coarse compared to the footprint size and this option may not be appropriate. If any grid point within the footprint is classified as coast, then the "coast" *qcinfo* flag is set. Quality flags for an observation are set in *qcflags* if any profile/channel within a footprint triggers one: this applies to RTTOV errors, and missing emissivity/BRDF values from the atlases (see section 5.13). Similarly, RTTOV quality flags are set in *qcrttov* if any profile in the footprint triggers one (see section 5.13).

Note that the footprint simulations are not compatible with some other options: in particular, any options for per-profile or per-channel outputs (such as emissivities, BRDFs, transmittances, Jacobians, CADS height assignments, geometric heights) are not available.

The code which identifies the grid points falling within each footprint is quite expensive to run: this routine can take advantage of multiple threads via OpenMP which can significantly reduce the run-time. RadSim is compiled with OpenMP support enabled by default (see section 2.3) and you can specify the number of threads to use via the *nthreads* configuration namelist variable.

The footprint metadata can be written out and then read in for subsequent runs which can result in faster run-times if the metadata do not change. This is applicable in cases where the same model grid is used, and the observation locations do not change between runs (though observation *times* may differ). In particular, this may apply to observations for geostationary sensors. In the first run you can specify an output filename using the *write_footprint_file* configuration namelist variable. The footprint data are written to the specified netCDF file (NB this file is overwritten if it already exists) in the specified *output_dir*. On subsequent runs where the footprint data file in the *read_footprint_file* configuration namelist variable. Even when using this feature, it is recommended to compile with OpenMP flags and make use of multiple threads for the RadSim runs.

5.2 Non-standard grids

As noted, for DWD ICON fields a simple nearest-neighbour approach is always used for interpolation. For all other model data, the model data points must have a well-defined spacing, i.e., the grid should be rectangular. Rotated lat-lon grids, commonly used in limited area models, can be used if they are rectangular in their original positions, but other types of grid cannot. If the grid is not recognised to be of a type that can be interpolated, then the simulation is done at model data point positions. The code will issue a warning message if interpolation has been requested but is not available for that grid.



5.2.1 Coordinate rotation

If a grid has a rotated pole, observation data coordinates (which are assumed to have standard lat-lon coordinates) are transformed to the rotated grid before interpolation. Model data, interpolated or not as appropriate, are then transformed to standard coordinates. The rotation transformation is described below.

Using a left-handed coordinate system with the polar axis as the z-axis, the forward transformation (standard->rotated coordinates) is effectively a sequence of two axis rotations, a z-axis rotation or longitude transformation, such that the rotated pole then lies on the 180.0 degree meridian and then a y-axis rotation or latitude transformation. To convert from rotated to standard coordinates, the transformation, including the rotation sequence, is reversed.

If the rotated pole is at position (λ_p , ϕ_p) then the rotation angles are:

$$\begin{aligned} \theta_z &= \pm (\lambda_p + \pi) \\ \theta_y &= \pm (\pi/2 - \varphi_p) \end{aligned}$$

+ve for the forward transformation, -ve for the reverse.

The z-axis rotation is then:

$$\lambda_z = \lambda - \theta_z$$

The y-axis rotation:

 $\varphi_{y} = \sin^{-1} \left(\sin \varphi \cos \theta_{y} - \cos \lambda \cos \varphi \sin \theta_{y} \right)$ $\lambda_{y} = d \cos^{-1} \left(\cos \lambda \cos \varphi \cos \theta_{y} + \sin \varphi \sin \theta_{y} \right)$

Note: The input coordinates (λ , ϕ) to each rotation change meaning depending on the order of the rotation sequence, e.g., if the z-axis rotation is done first then λ is the original longitude coordinate for the z-axis rotation and then $\lambda = \lambda_z$ for the y-axis rotation.

5.3 Field conversions

Some fields that are required to run RTTOV may not be available directly but can be derived from other fields. RadSim will convert, where necessary: potential temperature to temperature, relative to specific humidity and dewpoint temperature to specific humidity, rain/snow flux to mass mixing ratio, and ozone mixing ratio in mg/kg to kg/kg.

RTTOV requires profiles to be ordered top-down. This is contrary to the convention for some model data profiles which are ordered bottom-up. Both vertical axis orientations have been allowed for; a check is performed on the first profile to determine the orientation on the assumption that pressure is monotonically decreasing in height. The profile is then flipped over if necessary. These per-profile conversions are discarded with each iteration, the data available at the end of processing, and which may be written to the output file, is in the



original orientation. Temperature and humidity fields remain in the original model units, but rain, snow and ozone will be in the converted units when conversions take place.

5.3.1 Potential temperature conversion

Temperature T is obtained from potential temperature θ using the standard formula:

$$T = \theta \left(\frac{p}{p_0}\right)^{\kappa}$$

Where:

p = pressure $p_0 = \text{reference pressure } (=1000 \text{ hPa})$ $\kappa = R/c_p$ (R is the specific gas constant for dry air, c_p is the specific heat capacity)

5.3.2 Humidity conversion

Surface (2m) humidity information may be stored as dewpoint temperature T_d or relative humidity. In both cases, in order to obtain specific humidity q it is necessary to calculate specific humidity at saturation.

Specific humidity q (units kg/kg) is defined in terms of pressure as:

 $q = \frac{\varepsilon e}{(p - (1 - \varepsilon)e)}$

Where: $\varepsilon = Rd/Rv$ (ratio specific gas constant of dry air and water vapour) e = vapour pressure p = atmospheric pressure

The saturation vapour pressure *e*_s is calculated using the commonly used Magnus Formula:

$$e_s = a \exp\left(\frac{bT}{c+T}\right)$$

Where *T* is temperature and *a*, *b*, *c* are empirically derived constants with different values for liquid water and ice. It is assumed that if T < 273.15 K then ice coefficients should be used.

For the dewpoint conversion we can substitute $T=T_d$ and $e=e_s$ in the above equations and calculate q.

Relative humidity *r* is defined as e/e_s but to a very good approximation $r = q/q_s$, therefore *q* is simply:

 $q = q_s r$

Substituting $e=e_s$ in the first equation allows q_s to be calculated.

The EUMEISAT Network of Satellite Application Facilities	Radiance Simulator v4.0 User Guide	Doc ID Version Date	: NWPSAF-MO-UD-061 : 1.0.2 : 11/06/2025
---	---------------------------------------	---------------------------	---

5.3.3 Rain/snow flux conversion

RTTOV v14 supports input of rain/snow fluxes instead of mixing ratios, but the feature is deprecated, so RadSim implements the conversion for the NWP SAF profile datasets (except for the 137L CAMS dataset) and UM fields. RadSim only passes rain/snow as mixing ratios to RTTOV. This means that output rain/snow fields are *always* mixing ratios for all input datasets. The conversion follows that in RTTOV (Geer *et al.*, 2007) and gives identical results:

$$m = \left(\frac{af}{\rho}\right)^b \left(\frac{Rt}{M_{\rm air}p}\right)$$

where *f* is the rain/snow flux in kg/m²/s, *m* is the corresponding mixing ratio in kg/kg, *t* is the temperature (K), *p* is the pressure (Pa), *R* is the universal gas constant (8.314462618 J/mol/K), M_{air} is the molar mass of dry air (28.9644 g/mol), ρ is the density of rain or snow (assumed 1.0g/cm³ for rain, 0.1g/cm³ for snow), and *a* and *b* are constants stored as ratios in the RTTOV source code:

	а	b
Rain	3600.0 / 20.89	1.0 / 1.15
Snow	3600.0 / 29.51	1.0 / 1.10

5.4 Pressure levels

RTTOV v14 requires pressure "half-levels" to be provided. The top half-level is typically at or near 0 hPa, and the bottom half-level is always at the surface. These half-levels form the upper and lower bounds of layers. Each layer is associated with a pressure "full-level", and all input fields (temperature, gas concentrations, aerosols, hydrometeors) are provided on pressure full-levels. The pressure full-levels are an optional input to RTTOV, and if they are not provided, RTTOV computes them as the mean of adjacent half-level pressures. See the RTTOV documentation for more information on the pressure levels.

Models including the IFS, CAMS, ICON, HARMONIE-AROME, JMA use the same half-level and full-level terminology. For these models, the "number of levels" typically refers to the number of full-levels. However, for RTTOV and in the output RadSim netCDF file, be aware that the "number of levels" refers to the number of half-levels. For the Met Office Unified Model, the half-levels correspond to UM rho levels, and the full-levels correspond to UM theta levels.

For the IFS, CAMS, HARMONIE-AROME, and JMA models, the pressure half-levels are computed from the surface pressure using a set of coefficients. RadSim has coefficients in the code for some standard model configurations: IFS/CAMS on 60L, 91L, 137L; HARMONIE-AROME on 65L; JMA on 60L, 100L. For other vertical grids, the input GRIB file must contain the coefficients. The pressure full-levels are computed from the pressure half-levels in the same way that RTTOV does, so these are not input to RTTOV in the simulations. Therefore, typically, for these models only the surface pressure is mandatory, and the model GRIB file does not need to contain the vertical pressure half- or full-levels. For ECMWF fields in netCDF format, pressure half-level ingest is not supported and nor is



reading of coefficients from the input file, so the fields must be on a supported standard set of levels (60L, 91L, 137L) for which the coefficients are stored in the RadSim code, or the netCDF file must include the pressure full-levels field. In the latter case, the half-level pressures are computed from the full-level pressures. The surface pressure must always be included in the model file.

For the ICON model, the surface pressure must be present in the model file. The file must also contain either the pressure half-levels or pressure full-levels (or both). If either is missing, RadSim computes it from the other. Both half- and full-levels are passed into RTTOV.

For the Met Office Unified Model, it is strongly recommended that the model data file contains the pressure half-levels (rho levels). The pressure full levels (theta levels) are computed through interpolation of the rho levels in exner pressure if the theta levels are not in the input file. This calculation is consistent with the UM. If the rho levels are not present, then the theta level pressures must be in the model file. In this case the rho level pressures are approximated as the mean of adjacent theta level pressures, with the top rho level set to 0 hPa and the bottom level set to the surface pressure. This is not consistent with the UM and as such is not generally recommended. Both half- and full-levels are passed into RTTOV.

5.5 Surface type

By default, RTTOV requires the underlying surface to be classified as either *land*, *sea* or *ice* (sea-ice). All input profile datasets should contain a land-sea mask (LSM) either explicitly or by proxy. If there is no LSM but there is an orography field then the surface is classified as *sea* if the surface height is 0.

Some datasets may use a floating-point definition of the LSM to accommodate mixed surface types in a grid-box near coastlines. This is, in effect, a land fraction such that 0=sea only and 1=land only. In RadSim it is assumed that such points are sea if the LSM value < 0.5, otherwise land.

In the case of the model LSM being interpolated to observation positions, the observation value, if it is available, is used in preference to the model derived value as interpolation of surface type is unreliable. The interpolated value is a land fraction. In cases where the land fraction is some value other than 0 or 1, i.e., between those values, a check is made against the assumed surface type (always 0 or 1 before sea-ice is considered) and flagged as coastline if the difference is greater than a defined threshold. This threshold can be changed in the configuration namelist file (section 4.1).

Ideally, the model dataset should also include a sea-ice fraction field which can be used to distinguish between *sea* and *ice*, however this is not compulsory as it may not always be available. If sea-ice fraction is present, then the surface is reclassified from *sea* to *ice* if the fraction of ice is greater than a defined threshold. This value can be changed in the configuration namelist file (section 4.1).



The observation data file may contain the *rtsurf* field which can be used to explicitly set the surface type to land (0), sea (1), or sea-ice (2) for each observation. This is ignored for footprint simulations and for the *multi_surface* option described next.

RTTOV v14 provides a capability to simulate heterogeneous surfaces. This can be enabled in RadSim by setting the *multi_surface* option to true. In this case each profile is associated with three surface types: land, sea, and sea-ice. The land area coverage fraction is given by the LSM. The sea area coverage fraction is either (1-LSM) if the sea-ice fraction field is not present, or otherwise (1-LSM)*(1-sea_ice_fraction). RTTOV automatically computes the sea-ice area fraction to ensure a total coverage of 1. The three surface types are then treated independently in terms of emissivity and reflectance as described in the next section.

5.6 Emissivity and BRDF values

For sea surfaces, emissivities (for MW and IR channels) and BRDFs (for solar-affected channels if solar radiation is enabled) are calculated inside RTTOV. For land/sea-ice, fixed defaults apply for MW emissivity (0.95/0.92 respectively), and values are provided inside RTTOV for IR emissivity and for solar BRDF. The default BRDF values in RTTOV are fixed constants, and you can specify alternative values via the *default_brdf_land* and *default_brdf_seaice* configuration namelist options.

Alternatively, there are emissivity and BRDF atlases available (see the RTTOV user guide for details) for both MW and visible/IR over land. Some emissivity atlases may provide values for sea-ice surfaces as well (see the RTTOV user guide). The atlases are not used by default, but are highly recommended, particularly for microwave emissivities and for BRDF values, as the fixed values are unlikely to be very accurate given there is significant variation at those frequencies. If *multi_surface=T* (see section 5.5) the atlases will be used to obtain emissivities for the land and sea-ice surfaces, and BRDFs for land surfaces only.

5.7 Satellite zenith and azimuth angles

RadSim requires the satellite zenith angle to be specified, and under certain circumstances (when surface emissivity and reflectance models are used, and for footprint simulations) the satellite azimuth angle is also required. When running RadSim using an observation data file (see sections 4.2 and 5.1), the satellite zenith and azimuth angles may be specified explicitly for each observation. If the satellite angles are unspecified in the observation data file, or for runs on model grids (no observation data file), fixed satellite zenith and azimuth angles may be specified via the *zen_angle* and *azi_angle* configuration namelist variables.

When simulating geostationary sensors RadSim can optionally calculate the satellite zenith and azimuth angles. This applies both to runs on model grids and runs using observation data files. This is activated by setting the *calc_geo_sat_angles* configuration namelist variable to true. RadSim then computes the zenith and azimuth angles for a geostationary sensor with a sub-satellite point located at the latitude and longitude given by the *geo_sat_lat* and *geo_sat_lon* configuration namelist variables, and at altitude *geo_sat_height* (in km). The latitude and longitude both default to 0° and the latitude should generally be very close to zero but may vary to account for satellite movement about the nominal sub-satellite point. The satellite altitude defaults to 35800 km and if specified should



typically be close to this value. The calculation of the zenith and azimuth angles is the same as that described in Annex B for the *radsim_geo_obs.py* script that generates observation data files for geostationary sensors.

If *calc_geo_sat_angles* is true, this overrides any fixed *zen_angle* and/or *azi_angle* specified in the configuration namelist. Where an observation data file contains the satellite zenith and/or azimuth angle, these are *not* overwritten by the RadSim calculations. If the observation data file contains only one or other of the angles, RadSim computes the other. RTTOV imposes a maximum for the satellite zenith angle that depends on the *rtcoef* coefficient file being used (see the RTTOV documentation). For coefficients based on v13 predictors or the old v9 predictors, the maximum zenith angle is 85.3° while for the deprecated v7 and v8 predictor files the maximum angle is 75°. Where the computed zenith angles exceed the relevant limit, RadSim sets the relevant bit in the *qcflags* quality control output (see section 5.13), the observation or grid point is not simulated, and corresponding outputs for that observation are zero. In this case the azimuth angle is not computed and remains set to zero. Where the computed zenith angle to 90°. Note that RadSim does not check satellite zenith angles from observation data files, so if you input zenith angles explicitly in the observations data file, they must all be within the RTTOV limits.

5.8 Solar radiation

Solar radiation is enabled by setting the *solar* configuration namelist variable to true. A solarenabled RTTOV optical depth coefficient file is required: in practice this means a file based on the v13 predictors or the older v9 predictors contained in a <code>rttov13pred*L/</code> or <code>rttov9pred*L/</code> subdirectory of the <code>rtcoef rttov14/</code> directory.

If solar radiation is enabled then brightness temperatures are output for channels with wavelengths above 3 microns, and reflectances are output for channels below 5 microns unless the *write_radiances* namelist variable is true in which case only radiances are output for all channels.

Solar simulations require solar zenith and azimuth angles. If you are running simulations on a model grid, you can specify fixed angles using the *sun_zen_angle* and *sun_azi_angle* configuration namelist variables. These angles are then used for all simulations. If you do not specify a *sun_zen_angle* greater than or equal to zero, then RadSim computes the solar zenith and azimuth angles for each grid point based on the validity time.

If you supply an obs data file for solar-enabled simulations, then you can again supply fixed angles to use for all simulations using the *sun_zen_angle* and *sun_azi_angle* configuration namelist variables. If you do not specify these, then you must either include both the solar zenith and azimuth angle fields, or the full date/time fields in the obs data file. In the former case, the input angles are used directly in the simulations while in the latter case, RadSim computes the solar angles from the date/time and lat/lon location of each observation.

The calculations for solar zenith and azimuth angles are given in Annex E.



5.9 Hydrometeor simulations

Hydrometeor simulations are enabled by setting the *hydrometeors* option to true. They require at least cloud liquid water, cloud ice water, and cloud fraction fields. For microwave sensors, rain and snow are optional inputs in addition. All hydrometeor concentrations are grid-box averages. The corresponding hydrometeor optical property file must be specified in the configuration namelist (sections 4.1 and 4.4). The RTTOV user guide provides information on the optical properties contained in the NWP SAF hydrotable files.

For microwave sensors, the rain, snow, liquid cloud, and ice cloud optical properties in the NWP SAF hydrotables are used for those respective hydrometeor fields. If you want to use your own custom hydrotable, see the information about this in section 4.4. The approximate treatments of polarisation implemented in RTTOV can be selected via the *mw_pol_mode* option (see the RTTOV user guide). The ARO-scaling polarisation treatment requires an additional look-up table to be read in with the coefficients: see section 4.4.

For visible/IR sensors, the optical properties for liquid and ice cloud are selected via the visir_clw_properties and visir_ice_properties configuration namelist variables. All options are available with all model data inputs. When the OPAC cloud liquid water properties are selected, RadSim currently uses only the maritime and continental stratiform particle types, selected according to the surface type of the profile (see section 5.5, if *multi_surface* is true then it is land if the LSM is greater than 0.5): continental over land, maritime otherwise. The Deff cloud liquid water and Baum ice properties require cloud liquid and ice water effective diameters respectively. In most cases these are computed within RTTOV using a parameterisation based on Martin et al. (1994) for cloud liquid water or otherwise one of four parameterisations selected by the *icede_param* option. The exception is for DWD ICON data. Optionally, the ICON GRIB file may include fields of cloud liquid and/or ice particle radius. In this case, the use_model_clw_deff and/or use_model_ice_deff options may be set to true to use these fields instead of the RTTOV parameterisations. Note that modifications to the data from the fields are made in the case of very small diameters: this treatment is described in Annex F. If the ICON density field is present and the cloud liquid water radius field is not present, the cloud liquid water size parameterisation is computed by RadSim using model densities instead of those computed internally by RTTOV (see Annex F).

The RTTOV thermal radiance solver is selected via the *thermal_solver* option. It defaults to delta-Eddington (3) which is the fast solver that works in the IR and MW. For IR scattering, the Chou-scaling solver (2) is also available, optionally with the Tang modification (*chou_tang_mod, chou_tang_factor,* see the RTTOV user guide). The DOM solver (1) accounts for multiple scattering but is slow. The RTTOV options related to the DOM solver are also available (*dom_nstreams, dom_accuracy, dom_opdep_threshold*, see the RTTOV user guide).

Solar scattering simulations are enabled by setting both *hydrometeors* and *solar* to true. The default solver for solar radiation is the RTTOV DOM implementation (for details see the RTTOV user guide). This is quite computationally expensive to run and note that it enforces a strict plane-parallel atmosphere for *all* simulated channels (so atmospheric curvature and refraction are not taken into account). You can enable Rayleigh multiple scattering with the DOM solver using the *rayleigh_multi_scatt* option. The solver for solar radiation can be



specified in the *solar_solver* configuration namelist variable. MFASIS-NN is selected by setting it to 2. A shortcut is provided for MFASIS-NN: setting the *run_mfasis* variable to true automatically sets *hydrometeors* and *solar* to true and selects MFASIS-NN as the solver. MFASIS-NN is trained using the Baum ice optical properties and so the *visir_ice_properties* should be set to 1 for MFASIS-NN simulations (RadSim does this automatically).

MFASIS-NN simulations require an MFASIS-NN coefficient file in addition to the hydrometeor optical property file. MFASIS-NN currently supports only a subset of visible/near-IR channels, and any visible/near-IR channels not supported (and with no significant thermal radiation) will have zero output radiances.

RTTOV cloud overlap options are available. The most important ones are *overlap_param* to select the cloud overlap parameterisation, and *col_threshold* to set the minimum weight for cloud columns to be considered. In most cases it is not recommended to change the *overlap_param* from the default which is to use maximum/random overlap for VIS/IR sensors, and the RTTOV hydrometeor-weighted two column scheme for MW sensors.

5.10 Aerosol simulations

RadSim can ingest CAMS fields optionally including nine of the CAMS aerosol species for which RTTOV provides optical properties (see ECMWF GRIB data in section 4.3.2). CAMS aerosol fields are also present in the 137L CAMS and 60L MACC profile datasets (section 4.3.4). RadSim can also ingest ICON-ART fields optionally including five of the ICON-ART aerosol species for which RTTOV provides optical properties (see ICON GRIB data in section 4.3.2). Aerosol simulations are not currently supported for any other model input.

Aerosol simulations are enabled by setting *aerosols* to true. Any or all of the supported CAMS or ICON-ART species may be present in the input file: all aerosol fields present will be used in the simulations. The corresponding CAMS or ICON-ART aerosol optical property file must be specified in the configuration namelist (sections 4.1 and 4.4).

The *thermal_solver* and *solar_solver* options discussed in section 5.9 for hydrometeor scattering simulations apply also to aerosol scattering simulations, along with the options related to particular solvers (DOM, Chou-scaling), and the Rayleigh multiple scattering option. However, the MFASIS-NN solver *cannot* be used for solar radiation as it does not yet support aerosol simulations. If solar radiation is enabled, then the default (and only) RTTOV solar solver for aerosols is DOM which is computationally expensive.

Aerosols can be enabled alongside hydrometeor simulations if all relevant hydrometeor fields are present in the input file.

5.11 Principal Components simulations

PC-RTTOV is described in the RTTOV user guide. It provides a more efficient way of simulating many or all channels for hyperspectral IR sounders. Within RadSim, PC-RTTOV can be used for non-solar clear-sky simulations optionally including variable O_3 , CO_2 , N_2O , CO, and CH₄. In addition, either the NLTE option (*nlte_correction=T*) or hydrometeor scattering (section 5.9) can be enabled, but not both together.



Radiances are reconstructed for the *channels* specified in the namelist configuration file: if *channels* is omitted or all values are zero, then all sensor channels will be reconstructed. It is not currently possible to output PC scores. The IR emissivity atlases can be used (section 5.6), and the CAMEL v3 climatology atlas is recommended.

To enable PC-RTTOV set the *enable_pcrttov* configuration namelist variable to true. In this case a PC-RTTOV coefficient file is required, and the corresponding optical depth coefficient file must also be supplied (section 4.4). The number of PC scores to simulate is specified in the *npcscores* configuration namelist variable. This can range from one up to a maximum of 400. PC-RTTOV provides a choice among different predictor sets. This is specified in the *pc_reg_set* variable and is a value in the range 1-4: the higher the number the more predictors are used which improves the accuracy but increases the run-time. See the RTTOV user guide for full details on PC-RTTOV, the associated options, and recommended values.

5.12 Running RTTOV

By default. simulations use the *rttov_direct* or *rttov_parallel_direct* subroutines (the latter if *nthreads* is greater than 1), but if any Jacobian output has been requested then *rttov_k* or *rttov_parallel_k* is run instead, and this increases the run time.

See section 4.4 for information about the coefficient files required for the different simulation types.

The number of profiles passed to RTTOV per call may be specified in the *nprofs_per_call* variable in the configuration namelist. This defaults to 5. When running very memory intensive simulations (e.g. hydrometeor simulations for all channels of a hyperspectral sounder) it may be useful or necessary to reduce this. Conversely, for less memory-intensive scenarios performance gains may be possible by increasing this value, especially when running multi-threaded simulations.

5.13 Quality control

Quality control flags are set in the *qcflags* variable which is written to the output file. If *qcflags* has a value of zero then there were no errors, otherwise one or more of the following flags will have been set (flag bit numbers here correspond to those defined in the Fortran90 standard for the intrinsic bit manipulation procedures):

Flag/Bit number	Description
0	Observation is outside the model domain
1	Error reported by RTTOV
2	Bad emissivity value from atlas
3	Observation date/time lies outside the model data file temporal range
4	Bad BRDF value from atlas
5	Computed satellite zenith angle exceeds RTTOV maximum



Flag 0: If the observation location is outside the model domain, then no simulation will be performed. Output values for radiances or brightness temperatures/reflectances will be zero in this case (this is the value used to initialise these variables in RTTOV).

Flag 1: If there is an error reported by RTTOV, it means a fatal problem occurred and the simulation was aborted. Radiances will usually not have been calculated for all channels/profiles passed in the same call to RTTOV and the corresponding outputs will be zero.

Flag 2: If there is a bad emissivity value (outside the range 0-1) where an emissivity atlas is being used then the simulations for all channels for that profile are carried out with the fixed emissivity values used when no atlas is used. This applies to land and sea-ice surfaces.

Flag 3: This applies only if *temporal_data* is true (i.e. temporal interpolation is enabled and the model data includes fields at multiple forecast times). If the observation date/time lies outside the temporal range of the data, no simulation is performed, and output radiances or brightness temperatures/reflectances are zero.

Flag 4: If there is a bad BRDF value (less than or equal to zero) where the BRDF atlas is being used then the simulation for the *affected channel for that profile* is carried out with the fixed BRDF value used when no atlas is used. If there are valid BRDFs for other channels for that profile, then they *are* used (compare with Flag 2 above for emissivity). This flag is not set if the solar zenith angle for the profile exceeds the maximum angle RTTOV allows for solar simulations (~85°).

Flag 5: When *calc_geo_sat_angles* is true and the geostationary satellite zenith angle is computed (see section 5.7), this flag is set for any grid points or observations where the computed angle exceeds the maximum allowed by RTTOV. The observation is not simulated, and the corresponding outputs are all zero.

The *qcinfo* variable contains other quality information. Two flags have been defined:

Flag/Bit number	Description
0	Coastline point
1	Empty footprint

Flag 0: The condition for triggering the coastline flag is described in section 5.5. Use of data points with this flag set should be done with caution – there may be surface skin temperature and emissivity errors that affect the accuracy of the data.

Flag 1: If footprint simulations are enabled (see section 5.1.3), this flag is set in the case where no grid points fall within an observation footprint and regular interpolation was used instead. This indicates that the model grid resolution may be too coarse for footprint simulations.

The *qcrttov* variable contains the RTTOV quality flags. These are described in the RTTOV user guide. The RTTOV quality flags are output for each channel for each profile. This output can be quite large if simulating many channels, and some flags are the same for all channels



for each profile, so by default RadSim combines the flags for all channels for each profile using a logical OR. In this case, *qcrttov* is a one-dimensional array, with one value per simulated profile, and each flag is set in *qcrttov* if it was set for at least one channel for the corresponding profile. To obtain per-channel *qcrttov* output, set the *combine_qcrttov* configuration namelist option to false: then it will be a two-dimensional array with one value per channel and per profile, containing a copy of the RTTOV quality outputs. For PC-RTTOV simulations, the *combine_qcrttov* option is ignored and the RTTOV quality outputs are always combined for all PC-RTTOV predictor channels (these are different to the channels for which reconstructed radiances are computed – see the RTTOV user guide for more information).

When doing footprint simulations, the RTTOV quality flags are combined with a logical OR for all profiles falling within each footprint so that a flag is set in *qcrttov* if it was set for at least one profile in the footprint. The treatment of per-channel flags and the *combine_qcrttov* option are the same as described above.

5.14 Memory Usage and Batching

As stated in the system requirements section in section 2.1, the recommended amount of RAM that should be available to run RadSim is 4GB. This is a very rough guide, some applications require more, some much less. In standard configurations (simulating for 1-4 orbits or at model grid points for a small number (<100) of channels), the main determining factor is the size of the model grid, even if simulating at observation locations. All model fields are stored in memory at the same time in the early stages of the run, irrespective of the setup.

5.14.1 Memory Requirement for Model Fields

Global models

As an example, the Met Office global model grid from July 2014 onwards has 1152 rows and 1536 columns for a total of 1,769,472 grid points. Each of these has 71 levels. The minimum amount of memory required to store the necessary multi-level fields (4 of them - pressure, temperature, humidity, cloud water), with 8-byte values is then:

4 x 71 x 1152 x 1536 x 8 bytes = 3.7GB

Other contributions (surface fields etc) take the requirement to over 4GB. Different grid sizes scale accordingly. If the option *write_profiles* is switched on, additional fields such as relative humidity increase this further.

It is desirable that the available RAM is sufficient to cover this requirement. If less is available, then the code will run slowly. In some circumstances the requirement may exceed the amount of *virtual memory* available to the system, in which case it is impossible for the simulation to run.



SAF Datasets

With just a few thousand profiles represented, these are very small by comparison to a global grid and therefore have a very low memory requirement.

5.14.2 Observation Profile Batching

If simulating at observation locations, then for small numbers of profiles, the input model fields are discarded after interpolation, reducing the memory burden for subsequent processing. For exceptionally large numbers of profiles, the interpolated dataset may be greater in size than the original, carrying with it a risk that the memory requirement exceeds available resources. This may also occur with footprint simulations which involve storing data for multiple profiles per observation. To mitigate this, RadSim can process the observation profiles in batches.

In most cases batching is restricted to those cases where the number of profiles is greater than the number of model grid points. The exception to this is for footprint simulations and it is quite common that you will need to use the *max_profs* configuration namelist variable (see below) to avoid running out of memory for footprint simulations. In this case, *max_profs* is always respected. For footprint simulations you may need to reduce this significantly e.g. to a value of a few thousand or even a few hundred.

If more than one batch is used, *the input model data are only discarded at the end of the run*. The effect this has on the overall memory requirement is to increase the amount used to store the model data by an amount proportional to the batch size. In the example calculation above, the batch size can be added to the number of data points, e.g., if the batch size is 500,000 then the above calculation becomes:

4 x 71 x (1152 x 1536 + 500000) x 8 bytes = approx 4.8GB

This is the minimum requirement (before taking into account other contributions) regardless of how many profiles there are in total.

Profile batching is automatic. It is triggered if there are more points defined in the observation data file than both the number of model grid points and also the batch size (with the noted exception of footprint simulations for which *max_profs* is always respected). Some examples are given in the table below.

Number of obs	Number of model grid points	Batch size	Number of batches
	500.000	100,000	10
1 000 000	500,000	750,000	2
1,000,000	2,000,000	100,000	1
	(enable_footprints = .false.)	750,000	1

Table: Batching examples

The batch size can be changed in the configuration namelist file to suit the application (see section 4.1). Note that the *max_profs* value is the number of *observations* for batching. With



footprint simulations, you will expect multiple profiles per observation, so as noted above you may want to reduce *max_profs* significantly to account for this. This is also important because the process batching (see below) is not supported for footprint simulations.

5.14.3 Memory Requirement for Output Data

The main contributor to the output data volume is the direct result of the simulations, i.e., the brightness temperatures/reflectances that are calculated in the radiative transfer model. There is one value per channel per profile. Output data are all stored as 4-byte values, there is no need for the extra precision of 8-byte values after the simulations have been performed.

For example, if the simulations are for AMSU-A (15 channels) for a modest number of profiles (100,000) then the memory requirement is simply

15 x 100000 x 4 bytes = 5.7MB

This is insignificant when compared to the model data requirement. If simulating for many channels (e.g., 8461 channels with IASI), the output data does become a significant factor, e.g.,

8461 x 100000 x 4 bytes = 3.2GB

5.14.4 Process Batching

The memory load imposed by the size of the output data can be reduced by running the simulations in batches and writing a subset of the results to the output file for each batch. This is a further subdivision of the profile batching. Each of the input profile batches is processed in 1 or more output batches.

The size of these batches is controlled not by the number of observations (profiles) but by defining a maximum array size for the output data via the *max_array_size* configuration namelist variable. Thus, if the maximum array size is set at 500MB (the default), then the above example for IASI requires 7 batches to run.

The picture is complicated slightly if any of the extra output options are switched on. Some of these require much greater amounts of memory to store all the data but the same array size restriction applies. It should be possible to process any number of channels through the batching control variables.

Note that process batching is not supported for footprint simulations so you should control the memory requirements via the observation batching (*max_profs* variable).



6. LIMITATIONS AND KNOWN ISSUES

These are documented in the accompanying Release Note. Please report any additional problems via the NWP SAF helpdesk at <u>https://nwp-saf.eumetsat.int/site/help-desk/</u>.

7. GLOSSARY

ABI	Advanced Baseline Imager
ADS	Atmosphere Data Store
AEC	Adaptive Entropy Coding
Aertable	Aerosol optical property file for use with RTTOV
AGRI	Advanced Geostationary Radiation Imager
AHI	Advanced HIMWARI imager
AMI	Advanced Meteorological Imager
AMSU-A	Advanced Microwave Sounding Unit - A
ARO	Azimuthally Randomly Oriented
ASTER	Advanced Spaceborne Thermal Emission and Reflection
	radiometer
BRDF	Bi-directional Reflectance Distribution Function
CADS	Cloud and Aerosol Detection Software
CAMEL	Combined ASTER and MODIS Emissivity database over Lan
CAMS	Copernicus Atmosphere Monitoring Service
CDS	Climate Data Store
CLW	Cloud Liquid Water
CNRM	Centre National de Recherches Météorologiques
Deff	Effective diameter
DOM	Discrete Ordinates Method
DWD	Deutscher Wetterdienst
ecCodes	ECMWF's software package for decoding GRIB files
ECMWF	European Centre for Medium Range Weather Forecasting
FASTEM	MW surface emissivity model
GB	Gigabytes
GRIB	GRIdded Binary or General Regularly-distributed Information in
	Binary form
HARMONIE-AROME	Convection-permitting, high-resolution limited-area weather
	forecasting model used in Europe and the Mediterranean
HDF5	Hierarchical Data Format version 5
HIMAWARI	Japanese geostationary satellite platform
Hydrotable	Hydrometeor optical property file for use with RTTOV
IÁSI	Infrared Atmospheric Sounding Interferometer
ICON	Icosahedral Nonhydrostatic model (DWD's NWP model)
ICON-ART	Couples ICON to Aerosol and Reactive Trace gases module
IR	Infrared
IREMIS	Physically-based IR sea surface emissivity model in RTTOV
ISEM	Older IR sea surface emissivity model in RTTOV
JasPer	Library for image manipulation and coding
JMA	Japan Meteorological Agency





JPEG	Joint Photographic Experts Group
JRA	Japanese ReAnalysis
KNMI	Royal Netherlands Meteorological Institute
LAPACK	Linear Algebra PACKage
LSM	Land/Sea Mask
MACC	Monitoring Atmospheric Composition and Climate
MARS	Meteorological Archival and Retrieval System
MB	Megabytes
MFASIS-NN	Method for FAst Satellite Image Synthesis – Neural Network
MODIS	Moderate-resolution Imaging Spectro-radiometer
MW	Microwave
netCDF	network Common Data Form
NIR	Near-Infrared (see VIS)
NITE	Non Local Thermodynamic Equilibrium
NOAA	National Oceanic and Atmospheric Administration
NWP	Numerical Weather Prediction
OPAC	Optical Properties of Aerosols and Clouds
Onen IPEG	Open source JPEG 2000 codec
OpenMP	Application Programming Interface supporting multi-platform
Openini	shared-memory parallel
PRS	Portable Batch System
PC	Principal Components
	Prossure Modulated Coll
	Mot Office bipary file format for model fields
	Rendem Access Memory
	Rational Atmosphere and Land 2 (limited area configuration of
RALJ	Mot Office LIM
	Red Het Enterprise Linux
	Reunal Enterprise Linux Redictive Transfer for TOVS
	Radialive Transler for TOVS
	Salellile Applications Facility
SEVIRI	Spinning Enhanced Visible and Initiated Imager
SSU STACL and a	Stratospheric Sounding Unit
	Identifier for Met Office UM fields
SURFEM-Ocean	SURface Fast Emissivity Model for Ocean
TELSEM2	I ool to Estimate Land Surface Emissivities from Microwaves
TIDOO	
TIROS	Television Infrared Observation Satellite
TOA	Top Of Atmosphere
IOVS	TIROS Operational Vertical Sounder
UM	Unified Model
UIC	Coordinated Universal Time
UV	Ultraviolet (see VIS)
VIS	Visible (here used synonymously with UV/VIS/NIR)



8. REFERENCES

Geer, A. J., P Bauer, P. Lopez, 2007: Lessons learnt from the 1D+4D-Var assimilation of rain and cloud affected SSM/I observations at ECMWF, ECMWF Tech. Memo. 535

Martin, GM, DW Johnson, and A Spice, 1994: The Measurement and Parameterization of Effective Radius of Droplets in Warm Stratocumulus Clouds. *J. Atmos. Sci.*, **51**, 13, 1823-1842, <u>https://doi.org/10.1175/1520-0469(1994)051<1823:TMAPOE>2.0.CO;2</u>

Saunders, R., J. Hocking, E. Turner, S. Havemann, A. Geer, C. Lupu, J. Vidot, P. Chambon, C. Köpken-Watts, L. Scheck, O. Stiller, C. Stumpf, E. Borbas, 2020: RTTOV v13 Science and Validation Report. Available at:

https://nwp-saf.eumetsat.int/site/download/documentation/rtm/docs_rttov13/rttov13_svr.pdf (last accessed 27 Nov 2020).



9. ANNEXES

ANNEX A – radsim_run.py

The script src/scripts/radsim_run.py can be used to generate RadSim configuration files and optionally run RadSim in one step from the command line. All RadSim namelist configuration options can be specified via arguments to this script. Most arguments have the same name as the namelist variables and as such are not explained further here: see section 4.1. Usage information can be printed as follows:

```
$ src/scripts/radsim run.py -h
usage: radsim run.py [-h] --config file FILE [--write config only T|F]
                      [--radsim bin dir DIR] --model datafile FILE
                      [--model ancil datafile FILE]
                     --model filetype 0|1|2|3|4|5|6|7|8|9|10|11
                      [--rttov coefs type .dat|.nc]
                      [--rttov coefs options STRING] [--rttov coefs dir DIR]
                      [--rttov aertable dir DIR] [--rttov hydrotable dir DIR]
                      [--rttov mfasis nn dir DIR] [--rttov pccoefs dir DIR]
                      [--rttov coef file FILE] [--rttov aertable file FILE]
                      [--rttov hydrotable file FILE]
                      [--rttov mfasis nn file FILE]
                      [--rttov mw pol coef file FILE]
                      [--rttov pccoef file FILE] [--obs datafile FILE]
                      [--read footprint file FILE] [--write footprint file FILE]
                      [--emis atlas dir DIR] [--brdf atlas dir DIR]
                     --platform STRING --satid INTEGER --inst STRING
                      [--channels INTEGER [INTEGER ...]] [--temporal data T|F]
                      [--fcst range INT or INT, INT] [--interp_nn_spatial T|F]
                      [--interp nn temporal T|F] [--enable footprints T|F]
                      [--multi_surface T|F] [--use_emis_atlas T|F]
                      [--emis atlas id 1|2|3] [--emis_atlas_year EMIS_ATLAS_YEAR]
                      [--camel atlas version 2|3] [--use brdf atlas T|F]
                      [--use all atlas months T|F] [--run mfasis T|F]
                      [--use model clw deff T|F] [--use model ice deff T|F]
                      [--calc geo sat angles T|F] [--geo sat lat FLOAT]
                      [--geo sat lon FLOAT] [--geo sat height FLOAT]
                      [--zen_angle FLOAT] [--azi angle FLOAT]
                      [--sun zen angle FLOAT] [--sun azi angle FLOAT]
                      [--co2 max ppmv FLOAT]
                      [--ssu co2 cell_pressure FLOAT FLOAT FLOAT]
                      [--cads height assign threshold FLOAT]
                      [--default_brdf_land FLOAT] [--default_brdf_seaice FLOAT]
                      [--fixed emis FLOAT] [--seaice threshold FLOAT]
                      [--coast threshold FLOAT] [--combine qcrttov T|F]
                      [--output_mode 1|2|3] [--max_array_size INTEGER]
[--max_profs INTEGER] [--nthreads INTEGER]
                      [--nprofs per call INTEGER]
                      [--rttov verbose T|F] [--apply reg limits T|F]
                      [--adk_bt T|F] [--adk_refl T|F] [--opdep13_gas_clip T|F]
                      [--interp_mode 1|2|3|4|5] [--o3_data T|F] [--co2_data T|F]
                      [--n2o_data T|F] [--co_data T|F] [--ch4_data T|F]
                      [--so2_data T|F] [--solar T|F]
                      [--rayleigh_single_scatt T|F]
                      [--rayleigh_max_wavelength FLOAT]
                      [--rayleigh min pressure FLOAT] [--nlte correction T|F]
```



NWP SAF

[--refraction T|F] [--plane parallel T|F] [--rad down lin tau T|F] [--use t2m T|F] [--use q2m T|F] [--mw sea emis model 1|2|3] [--ir sea emis model 1|2] [--solar sea brdf model 1] [--lambertian T|F] [--lambertian fixed angle T|F] [--clw data T|F] [--permittivity param 1|2|3] [--clw cloud top FLOAT] [--aerosols T|F] [--hydrometeors T|F] [--thermal solver 1|2|3] [--solar solver 1|2] [--rayleigh_multi_scatt T|F] [--dom_nstreams INTEGER] [--dom_accuracy FLOAT] [--dom_opdep_threshold FLOAT] [--chou_tang_mod T|F] [--chou_tang_factor FLOAT] [--visir_clw_properties 1|2] [--visir_ice_properties 1|2|3] [--icede param 1|2|3|4] [--baran ice version 1|2] [--mw pol mode 0|1|2] [--ice polarisation FLOAT] [--overlap param 0|1|2|3|4] [--col threshold FLOAT] [--two col max frac max p FLOAT] [--enable pcrttov T|F] [--npcscores INTEGER] [--pc reg set INTEGER] [--pc band INTEGER] [--output dir DIR] [--output file FILE] [--write_profiles T|F] [--write_latlon T|F] [--write geom height T|F] [--write trans T|F] [--write tjac T|F] [--write qjac T|F] [--write o3jac T|F] [--write tskinjac T|F] [--write wind10mjac T|F] [--write emisjac T|F] [--write radiances T|F] [--write clearsky T|F] [--write emis T|F] [--write brdf T|F]

The --config_file, --model_datafile, --model_filetype, --platform, --satid and --inst arguments are mandatory, all others are optional. Optional arguments take their RadSim defaults if omitted. The following table lists the arguments which are specific to the script (i.e. which do not correspond to configuration namelist variables).

Argument	Value	Description
-h		Print out usage information.
config_file	character	Name of output configuration namelist file.
write_config_only	T F, default F,	If T write out namelist file and stop. If F write out file
	optional	and run RadSim.
radsim_bin_dir	character,	Location of RadSim <i>bin</i> /directory. It is not
	default "bin/",	necessary to specify this if calling radsim_run.py
	optional	from the top-level RadSim directory.



ANNEX B – radsim_geo_obs.py

The script src/scripts/radsim_geo_obs.py can be used to generate observation data files representing full disc scans for various geostationary sensors such as SEVIRI, ABI, AHI, AMI and AGRI.

The primary requirements are the latitudes/longitudes of the pixels. These are provided in a netCDF file which may optionally contain additional data such as pixel scan times, land fractions and orography. Examples of such files for several sensors are available on the RadSim web site (<u>https://nwp-saf.eumetsat.int/site/software/radiance-simulator/</u>), and you can create your own.

Arguments to the script include the relevant netCDF input file, various information related to the satellite (orbital height, sub-satellite latitude and longitude, footprint radius at nadir) and to the scan (scan start date/time, length of scan in minutes). You can also opt to restrict the output observations by specifying a maximum zenith angle, and/or minimum/maximum latitude and longitude bounds.

The output is a RadSim obs data file containing one observation for each pixel. To write a netCDF format obs data file, specify a ".nc" file suffix. The script calculates the satellite zenith and azimuth angles and writes these out together with the pixel latitudes/longitudes. Optionally, the obs data file may also contain the date/time of each pixel (observation), land fraction and/or orography data, the footprint semi-major/minor axes (radii) for each pixel scaled from the specified radius at nadir, and *scanline* and *scanpos* fields.

See below for more information on the various aspects of this tool. The arguments to the script are as follows:

```
$ python radsim_geo_obs.py -h
usage: radsim_geo_obs.py [-h] --output_file FILE --data_file FILE
[--sat_lat FLOAT] --sat_lon FLOAT
[--sat_height FLOAT]
[--nadir_footprint_radius FLOAT] [--scan_time FLOAT]
[--date_time INT INT INT INT]
[--fixed_date_time T|F] [--max_sat_zen FLOAT]
[--min_lat FLOAT] [--max_lat FLOAT]
[--min_lon FLOAT] [--max_lon FLOAT]
[--write_lsm T|F] [--write_orog T|F]
[--write sat angles T|F] [--write scan data T|F]
```

Argument	Value	Description
-h		Print out usage information.
output_file	character	Name of output obs data file. Use a ".nc" file extension to output a netCDF obs data file.
data_file	character	Name of input netCDF file containing pixel lat/lon and optionally other data for a specific geostationary sensor.
sat_lat	float, default 0., optional	Latitude of sub-satellite point in the interval [-90, 90] (degrees).





Radiance Simulator v4.0 User Guide

sat_lon	float	Longitude of sub-satellite point. The value must lie within the range of longitudes specified in the input netCDF file (degrees).
sat_height	float, default 35800 km, optional	Satellite height (km).
nadir_footprint_radius	float, optional	Footprint radius (km) of sub-satellite pixel. If specified, the footprint radii columns are written to the output obs data file. Only required if you want to use the footprint simulations in RadSim.
scan_time	float, optional	Length of time to scan all pixels defined in the netCDF file (minutes). Only required if you want to enable temporal interpolation and/or solar radiation in RadSim and the netCDF file does not contain a <i>scan_time</i> dataset.
date_time	five integers: year, month, day, hour, minute, optional	Date/time at which the scan begins. Only required if you want to enable temporal interpolation and/or solar radiation in RadSim.
fixed_date_time	T/F, default F, optional	If T, then all pixels are assigned the date/time specified indate_time.
max_sat_zen	float, default 85 degrees, optional	Maximum zenith angle (degrees) to include in output obs data file. It is not recommended to increase beyond the default.
min_lat	float, optional	Minimum latitude (degrees) to include in output obs data file, valid range [-90,90].
max_lat	float, optional	Maximum latitude (degrees) to include in output obs data file, valid range [-90,90].
min_lon	float, optional	Minimum longitude (degrees) to include output obs data file, must lie within range of longitudes in netCDF file.
max_lon	float, optional	Maximum longitude (degrees) to include output obs data file, must lie within range of longitudes in netCDF file.
write_lsm	T/F, default F, optional	If true, includes the land/sea mask (land fraction) in the output obs data file. Requires corresponding pixel data in the netCDF file.
write_orog	T/F, default F, optional	If true, includes the orography in the output obs data file. Requires corresponding pixel data in the netCDF file.
write_sat_angles	T/F, default T, optional	If true, write satellite zenith and azimuth angles to output file. Recommended true (the default).
write_scan_data	T/F, default F, optional	If true, write out <i>scanline</i> and <i>scanpos</i> fields to obs data file. This can help to identify obs but is for information only and has no impact on the simulations. Requires two-dimensional lat/lon arrays.

Format of input netCDF files

Example netCDF files can be found on the RadSim website and can be used as templates for your own if you wish to create new ones. Each file is specific to a geostationary sensor,

to a specific resolution of that sensor, and (usually) to a specific scan area/pattern of that sensor.

The data arrays within the files can be either one or two dimensional. All arrays must be the same size. It is also highly recommended to specify the pixels in the order in which they are scanned to ensure observation times are specified correctly (see below).

The input netCDF file **must** contain the following datasets:

- "*lat*", float array, pixel latitudes (degrees)
- "*Ion*", float array, pixel longitudes (degrees), dimensions must be same as "lat"

Latitude values outside the range [-90, +90] are used to determine space-view and any other "invalid" pixels which are to be excluded from any further calculations. Any pixel in the image which does not correspond to an Earth-view **must** have an invalid latitude (i.e. less than -90 or greater than +90).

Longitudes must lie either in the range [-180, +180] or [0, 360]. The longitude range should be chosen so that longitudes are monotonic east-west across the image (e.g. the dataset must not jump between 0 and 360 or between -180 and +180 within the image). The values of the satellite longitude and any minimum/maximum longitude bounds must lie within the range of longitudes in this dataset.

For sensors that operate at different resolutions for different channels (for example having more pixels for visible channels than IR channels, such as AHI), separate netCDF files are required for each resolution, and simulations must be performed separately for channels operating at each resolution.

The netCDF file may also contain further optional datasets:

- "*scan_time*", int array, time in minutes from the beginning of the scan at which each pixel is scanned, dimensions must be same as "lat"
- "*Ism*", float array, pixel land fraction, dimensions must be same as "lat"
- "orog", float array, pixel orography (m), dimensions must be same as "lat"

The dates/times in the observation data file are required if temporal interpolation and/or solar radiation are to be enabled in the RadSim simulations. If these are not required, the dates/times are not required in the output obs data file, and the *--date_time* argument may be omitted. However, if either of these capabilities are to be used then there are three options for specifying the pixel (observation) dates/times:

 For the most accurate results, provide the scan_time dataset in the input netCDF file. This specifies the number of minutes into the scan (rounded to the nearest integer) that each pixel is scanned according to the sensor scan schedule. This allows for sensors which scan multiple pixels (of fields of view) simultaneously. The pixel date/times are simply set to the specified scan start --date_time plus the offset given in the scan_time dataset. Note that if the sensor scan schedule changes, then the netCDF file would require an updated scan_time dataset.



- 2. Specify the total length of the scan from first pixel to last in the --scan_time argument. Pixel scan times are calculated by assuming the first pixel is scanned at the specified scan --date_time and the last pixel is scanned at this time plus the provided -- scan_time. The times for pixels are linearly interpolated between these two times assuming each pixel is scanned in the order provided in the datasets. Times are rounded to the nearest minute. Note that this calculation works well for SEVIRI which scans each pixel in order (so long as the pixels are specified in scan order in the netCDF file), but is less accurate for AHI and ABI, for example, because multiple pixels are detected simultaneously by these sensors.
- 3. Specify *--fixed_date_time* T so that all pixels are assigned the date/time provided in the *--date_time* argument (which in this case could be the time of the midpoint of the scan, for example).

Note that RadSim only accounts for observation times at a resolution of integer minutes for the purposes of temporal interpolation and solar angle calculations. Therefore, the second option above may be sufficient for many applications, even though it does not strictly represent the true order of pixel acquisition. Also note that for the second option above, if you specify a "cut-out" by maximum zenith angle and/or latitude/longitude bounds, it will not impact the observation (pixel) date times because they are calculated based on the whole pixel datasets in the netCDF file. If you want to create observation data files for a satellite which scans a limited area (such as the MSG rapid scan service), you should create a new netCDF file containing only the latitudes/longitudes for the limited region being scanned. Alternatively, include the pixel scan times in the netCDF file, as per the first option.

The "*Ism*" (land/sea mask) dataset is specified as a land fraction and values must lie within the range [0-1] for all pixels with valid latitudes/longitudes. If the dataset is present, then specifying the *--write_Ism T* argument will result in these values being written to the observation data file and this will be used in preference to any interpolated land fraction from the model fields.

The "*orog*" (orography) dataset is specified in metres. If the dataset is present, then specifying the *--write_orog T* argument will result in these values being written to the observation data file and this will be used in preference to any interpolated orography from the model fields.

Calculation of satellite zenith and azimuth angles

The derivation of the angles assumes a spherical Earth of radius R_e and a satellite at height R_h above the Earth's surface (Fig B1). The satellite line of sight has a view angle β which makes a satellite zenith angle of θ_{sat} at the surface, and the view path has length R_p . Finally, the central angle between the sub-satellite point and the intersection of the line of sight with the Earth's surface is α .



Figure B1: schematic diagram of a satellite illustrating zenith angle calculation.

From Fig B1, by considering the two right-angled triangles formed by adding the dashed line we can see that:

$$R_{eh} = R_e + R_h = R_e \cos \alpha + R_p \cos \beta$$

From the Sine Rule:

$$\frac{\sin\beta}{R_e} = \frac{\sin\alpha}{R_p}$$

By combining the previous two equations we may obtain:

$$\tan\beta = \frac{\sin\alpha}{\frac{R_{eh}}{R_e} - \cos\alpha}$$

Noting from Fig B1 that:

$$\theta_{sat} = \alpha + \beta$$

we can expand:

$$\frac{\sin \theta_{sat} \cos \alpha - \cos \theta_{sat} \sin \alpha}{\cos \theta_{sat} \cos \alpha + \sin \theta_{sat} \sin \alpha} = \frac{\sin \alpha}{\frac{R_{eh}}{R_e} - \cos \alpha}$$

Rearranging, we have:

$$\frac{R_{eh}}{R_e}(\sin\theta_{sat}\cos\alpha - \cos\theta_{sat}\sin\alpha) = \sin\theta_{sat}$$

from which we obtain:

$$\theta_{sat} = \tan^{-1} \frac{\sin \alpha}{\cos \alpha - \frac{R_e}{R_{eh}}}$$

For a satellite over latitude λ_s and longitude Λ_s and an observation latitude, longitude of λ_o , Λ_o , the central angle α between these two points (as in Fig B1) satisfies:

$$\cos \alpha = \sin \lambda_o \sin \lambda_s + \cos \lambda_o \cos \lambda_s \cos (\Lambda_o - \Lambda_s)$$



$$\sin \alpha = \sqrt{1 - \cos^2 \alpha}$$

Note that observations where:

$$\cos\alpha > \frac{R_e}{R_{eh}}$$

lie below the horizon (i.e. the zenith angle exceeds 90°).

The zenith angle calculation is generic. However, for the sake of simplicity the azimuth angle calculation assumes an ideal geostationary satellite above the equator ($\lambda_s = 0^\circ$) as shown in Figure B2.



Figure B2: schematic diagram of a geostationary satellite illustrating azimuth angle calculation.

Here point O represents an observation at latitude λ_o and longitude Λ_o , and point S is the sub-satellite point at latitude $\lambda_s = 0^\circ$ and longitude Λ_s . Point A has latitude 0° and longitude Λ_o . Then the arc OA lying along the meridian through Λ_o is subtended by angle λ_o and arc AS lying along the equator is subtended by angle $(\Lambda_s - \Lambda_o)$. We wish to determine the azimuth angle denoted ϕ_{sat} in Fig B2.

Noting that OAS is a right spherical triangle, from Napier's rules we directly obtain that:

$$\sin \lambda_s \tan \phi_{sat} = \tan(\Lambda_s - \Lambda_o)$$

From this we obtain ϕ_{sat} taking care to specify the correct quadrant so that the computed azimuth angle lies in the range [0,360] as required. Note also that RTTOV defines azimuth angles clockwise with respect to North, so, finally, the computed values are converted to this convention.



Calculation of pixel footprint radii

RadSim models footprints as ellipses with the semi-major axis aligned along the view path from the observation location to the satellite (see section 5.1.3 and Annex H). The increase in the footprint semi-major/minor axes (radii) with view angle compared to that specified at nadir is computed by considering the increase in path length from the satellite to the surface (which increases both radii equally), and the angle that the satellite line of sight makes with the surface (which increases the footprint semi-major axis only).

To calculate the path length R_p (see Fig B1) we recall again the Sine Rule:

$$\frac{\sin\beta}{R_e} = \frac{\sin\alpha}{R_p} = \frac{\sin\theta_{sat}}{R_{eh}}$$

together with:

$$\theta_{sat} = \alpha + \beta$$

to obtain (after some rearrangement):

$$R_p = R_{eh} \cos \beta - R_e \cos \theta_{sat}$$

The second factor, due to the angle at which the line of sight meets the surface, is given simply by $\sec \theta_{sat}$.

Therefore, to obtain the footprint semi-minor axis, the nadir footprint radius is multiplied by:

$$\frac{R_p}{R_h}$$

And to obtain the footprint semi-major axis, the nadir radius is multiplied by:

$$\frac{R_p}{R_h}\sec\theta_{sat}$$



ANNEX C – radsim_plot_example.py

The script src/scripts/radsim_plot_example.py can be used to plot simulated brightness temperatures, reflectances and radiances from RadSim. It can also plot surface emissivities/BRDFs and profile datasets from the output file. If the simulations were run on a regular lat/lon grid, then data can be plotted using the "pcolormesh" matplotlib method. Otherwise, a scatter plot is produced. This script is intended to be modified for your own purposes.

If you plot a per-channel dataset such as bt, refl, emis, or brdf, you must specify the *--channel* argument, which lies between 1 and the number of simulated channels. If you want to plot a model field such as the land/sea mask or sea-ice field (for example), then simply omit the *--channel* argument. If you want to plot a single level of a multi-level model field such as temperature, then you can use the *--channel* argument to specify the level to plot.

The script can also optionally plot differences in a given dataset between two files: both files must contain the specified dataset, and the datasets must have the same size.

NB The output NetCDF file must contain lat/lon fields: this can be achieved by setting either the *write_latlon* or *write_profiles* configuration namelist variables to true. The former option yields much smaller netCDF files if the full profile data are not required.

Usage information can be printed as follows:

Argument	Value	Description
-h		Print out usage information.
filename	character	Name of output NetCDF file.
var_name	character, default	Name of variable to plot: defaults to brightness
	'bt', optional	temperatures ('bt').
channel	integer, optional	Channel number to plot. This is mandatory for data stored per-channel (such as bt, refl, radiance, emis or brdf). The channels are numbered from 1 to n, where n is the number of channels simulated (i.e. this does not necessarily relate to the original instrument channel numbering). For multi- level profile data (such as temperature, "t"), you can use thechannel argument to specify the level to plot. For single-level or surface profile datasets (e.g. sea-ice) you must omit this argument.
scatter	T/F, default F, optional	Produce a scatter plot instead of a regular gridded plot. This happens automatically if the script detects that the data are not on a regular lat/lon grid.
size	float, default 1.5, optional	Size of blobs in scatter plots.

The EUMETSAT Network of Satellite Application Facilities	
---	--

vmin	float, default dataset minimum, optional	Minimum value for colour scale.
vmax	float, default dataset maximum, optional	Maximum value for colour scale.
cmap	character, default 'jet'/'seismic', optional	Colour map to use for the plot (see matplotlib documentation for options). Default is 'seismic' for difference plots, or 'jet' otherwise.
title	character, optional	Title for the plot. If this is not supplied, the script generates a title based on the input arguments.
diff	character, optional	Second output netCDF file: if specified, the script plots the differences in specified dataset between the file specified infilename and the one specified indiff. The two files must be consistent: in particular, the datasets must be stored on the same lat/lon locations in both files.



ANNEX D – Example MARS, CDS and ADS requests

There is good documentation for MARS on the web: <u>https://software.ecmwf.int/wiki/display/UDOC/MARS+user+documentation</u>

This Annex is intended to give advice and examples for constructing MARS requests to obtain ECMWF GRIB files that may be used with RadSim. For full details of the MARS keywords see the MARS documentation. ERA-5 data can now be obtained via the CDS. CAMS data are available via the ADS.

General notes:

- For MARS requests where you want to run RadSim on the GRIB file, specify surface data (*levtype=sfc*) followed by vertical profile data (*levtype=ml*) in a single request.
- If using the Python API, then you may need to download atmospheric (multi-level) and surface (single-level) fields into separate files and then concatenate them using ecCodes.
- Always use *levtype=ml* for vertical profile data.
- Always retrieve data for *all* available model levels, not a subset of levels.
- Use *grid=X/Y* to request data on a regular lat/lon grid at the specified longitude (*X*) and latitude (*Y*) resolution (*X*, *Y* in degrees).
- Only include the fields you require (e.g. don't include cloud-related fields if you only need to perform clear-sky simulations). See section 4.3.2 for information about the required and optional fields.
- For temporal interpolation use *step=0/1/...* to retrieve fields at multiple forecast times. Do not do this if you do not require temporal interpolation (or specify only a single *step* value).
- The target keyword is used to specify the path to the output file on your system.



Example 1: MARS request for all RadSim fields, operational analysis on 91L, 0.5 degree lat/lon resolution, GRIB format.

```
retrieve,
  class=od,
  date=2012-09-22,
  expver=1,
  levtype=sfc,
  param=31.128/129.128/134.128/165.128/166.128/167.128/168.128/172.128/235.128,
  stream=oper,
  time=00:00:00,
  type=an,
  grid=0.5/0.5,
  target="ec_2012092200_oper_an.grib"
retrieve,
  levelist=1/to/91,
  levtype=m1,
  param=75/76/130/133/203/246/247/248
```

Example 2: MARS request for clear-sky fields including CLW for MW simulations, operational deterministic forecast on 137L, T+0/1/3/6 forecast (multiple time steps), 2.0 degree lat/lon resolution, GRIB format.

```
retrieve,
 class=od,
 date=2018-04-30,
 expver=1,
 levtype=sfc,
 param=31.128/129.128/134.128/165.128/166.128/167.128/168.128/172.128/235.128,
 stream=oper,
 time=12:00:00,
 step=0/1/3/6,
 type=fc,
 grid=2.0/2.0,
  target="ec 2018043012 oper fcast T0136.grib"
retrieve,
 levelist=1/to/137,
  levtype=ml,
  param=130/133/246
```



Example 3: CDS requests for atmospheric and surface clear-sky fields including ozone for visible/IR simulations, ERA-5 on 137L, T+0 forecast (single time step), 1 degree lat/lon resolution, netCDF format.

```
import cdsapi
c = cdsapi.Client()
c.retrieve('reanalysis-era5-complete', {
    'class' : 'ea',
    'expver' : 'l',
    'stream' : 'oper',
    'type' : 'fc',
    'param' : '130/133/203',
    'levtype' : 'ml',
    'levelist': 'l/to/137',
    'date' : '2017-07-01',
    'time' : '06',
    'grid' : 'l.0/1.0',
    'format' : 'netcdf',
}, 'ec_2017070106_era5_T0_atm.nc')
```

```
import cdsapi
c = cdsapi.Client()
c.retrieve('reanalysis-era5-complete', {
    'class' : 'ea',
    'expver' : '1',
    'stream' : 'oper',
    'type' : 'fc',
    'param' : '31.128/129.128/134.128/165.128/166.128/167.128/168.128/172.128/235.128',
    'levtype' : 'sfc',
    'date' : '2017-07-01',
    'time' : '06',
    'grid' : '1.0/1.0',
    'format' : 'netcdf',
}, 'ec_2017070106_era5_T0_sfc.nc')
```



Example 4: ADS request for CAMS data including supported aerosol fields and cloud fields for VIS/IR cloud simulations.

```
import cdsapi
credentials = dict()
credentials['url'] = "https://ads.atmosphere.copernicus.eu/api/v2"
credentials['key'] = "xxx"
c = cdsapi.Client(url=credentials['url'], key=credentials['key'])
c.retrieve(
      'cams-global-reanalysis-eac4',
      {
             'date': '2021-06-30/2021-06-30',
             'format': 'grib',
             'variable': [
                   '10m_u_component_of_wind', '10m_v_component_of_wind', '2m_dewpoint_temperature',
                  '2m_temperature', 'dust_aerosol_0.03-0.55um_mixing_ratio',
'dust_aerosol_0.55-0.9um_mixing_ratio',
'dust_aerosol_0.9-20um_mixing_ratio', 'fraction_of_cloud_cover',
                  'hydrophilic_organic_matter_aerosol_mixing_ratio', ______'
'hydrophobic_black_carbon_aerosol_mixing_ratio', 'land_sea_mask', 'sea_ice_cover',
                   'sea_salt_aerosol_0.03-0.5um_mixing_ratio', 'sea_salt_aerosol_0.5-5um_mixing_ratio',
                   'sea salt aerosol 5-20um mixing ratio',
                   'skin_temperature', 'specific_cloud_ice_water_content',
                   'specific_cloud_liquid_water_content',
                   'specific humidity', 'sulphate_aerosol_mixing_ratio', 'surface_geopotential',
'surface_pressure', 'temperature',
            ],
            ],

'model_level': [

'1', '2', '3',

'4', '5', '6',

'7', '8', '9',

'10', '11', '12',

'13', '14', '15',

'16', '17', '18',

'19', '20', '21',

'22', '23', '24',

'25', '26', '27',

'28', '29', '30',
                   '28', '29', '30',
'31', '32', '33',
                   '34', '35', '36'
'37', '38', '39'
                  '40', '38', '39',
'40', '41', '42',
'43', '44', '45',
'46', '47', '48',
'49', '50', '51',
'52', '53', '54',
'55', '56', '57',
                   '58', '59', '60',
            ],
'time': '00:00',
      },
       'ec 2021063000 cams all.grib')
```


ANNEX E – Solar angle calculations

The following calculations are ultimately derived from JavaScript taken from the NOAA website, converted to Fortran 90. The code is based on the book, Astronomical Algorithms by Jean Meeus. The same calculations are used for operational processing of satellite data at the Met Office.

In the following: $deg2rad = \pi/180$ and $rad2deg = 180/\pi$ (factors for converting between degrees and radians), and λ/Λ are the latitude and longitude in degrees respectively, where the longitude lies in the range 0-360°.

Solar zenith angle (θ)

This first requires calculation of the equation of time (*EoT*) which is defined as the difference between local apparent time (t_{sol} , or "sundial time") and local mean time (which is UTC clock time plus a correction for longitude), and the solar declination (δ).

The calculation for *EoT* is as follows:

Calculate Julian day (jd) from date (year, month, day) including fraction of current day computed from UTC time (hour, minute, second)

$$\begin{array}{l} \mbox{if (month \leq 2) then} \\ \mbox{year := year - 1} \\ \mbox{month := month + 12} \\ \mbox{end if} \\ \mbox{a = } \frac{\mbox{year}}{100} \\ \mbox{b = 2 - a + } \frac{a}{4} \\ \mbox{jd = int(365.25 \times (year + 4716)) + int(30.6001 \times (month + 1)) + day + b - 1524.5} \\ \mbox{+ } \frac{\mbox{hour \times 3600 + minute \times 60 + second}}{86400} \end{array}$$

Calculate Julian century (jc) relative to the year 2000 (36525 days per century)

jc = (jd - 2451545) / 36525

Calculate the mean obliquity of the ecliptic (ε_m) in degrees

$$nseconds = 21.448 - jc \times (46.8150 + jc \times (0.00059 - jc \times 0.001813))$$

$$\varepsilon_m = 23 + (26 + \frac{nseconds}{60})/60$$

Calculate the corrected obliquity of the ecliptic (ε_c) in radians



Calculate eccentricity of Earth's orbit (e)

 $e = 0.016708634 - jc \times (0.000042037 + 0.0000001267 \times jc)$

Calculate the geometric mean longitude of the sun ($\overline{\Lambda_{sol}}$, 0-360°)

$$\overline{\Lambda_{sol}} = modulo(280.46646 + jc \times (36000.76983 + 0.0003032 \times jc), 360)$$

Calculate geometric mean anomaly of the sun (M) in radians

 $M = (357.52911 + jc \times (35999.05029 - 0.0001537 \times jc)) \times deg2rad$

Calculate the Equation of Time (EoT) in minutes

$$z = (\tan(\varepsilon_c/2))^2$$

$$c_1 = \sin(2 \times \overline{\Lambda_{sol}} \times deg2rad)$$

$$c_2 = \cos(2 \times \overline{\Lambda_{sol}} \times deg2rad)$$

$$EoT = \begin{bmatrix} (z \times c_1) - (2 \times e \times \sin M) + (4 \times e \times z \times c_2 \times \sin M) - \\ (z^2 \times c_1 \times c_2) - (\frac{5}{4} \times e^2 \times \sin 2M) \end{bmatrix} \times 4 \times rad2deg$$

The solar declination is calculated as follows:

Calculate equation of centre (EoC) in degrees

$$EoC = \sin M \times (1.914602 - jc \times (0.004817 + 0.000014 \times jc)) + \sin 2M \times (0.019993 - 0.000101 \times jc) + \sin 3M \times 0.000289$$

Calculate true longitude (Λ_{sol}^t , in degrees) and apparent longitude (Λ_{sol}^a , in radians) of sun

$$\begin{aligned} \Lambda^{t}_{sol} &= \overline{\Lambda_{sol}} + EoC \\ \Lambda^{a}_{sol} &= (\Lambda^{t}_{sol} - 0.00569 - 0.00478 \times \sin \omega) \times deg2rad \end{aligned}$$

Calculate solar declination (δ)

 $\sin \delta = \sin(\varepsilon_c) \times \sin(\Lambda^a_{sol})$

The solar zenith angle is calculated as follows:

Calculate time offset (toffset) in minutes including longitude adjustment

 $t_{offset} = EoT + 4 \times lon$

Calculate the true solar time (t_{sol}) in minutes for a given time (hour, minute, second UTC)

 $t_{sol} = hour \times 60 + minute + second/60 + t_{offset}$

Calculate the hour angle of the sun (h) in radians

$$h = \left(\frac{t_{sol}}{4} - 180\right) \times deg2rad$$

Calculate solar zenith angle (θ_{sol}), recall latitude is λ

 $\cos\theta_{sol} = \sin\lambda\,\sin\delta + \cos\lambda\cos\delta\cos h$

Solar azimuth angle (ϕ_{sol})

 $\tan \phi_{sol} = \frac{\cos \delta \sin h}{\sin \lambda \cos \delta \cos h - \cos \lambda \sin \delta}$



ANNEX F – Cloud liquid water and ice effective diameters

For visible/infrared cloud simulations where the RTTOV cloud liquid water "Deff" (effective diameter) optical properties or Baum ice properties are selected, the cloud liquid and ice effective diameters are required (respectively). RTTOV provides one parameterisation for liquid cloud diameter based on Martin *et al.* (1994) which is described in the RTTOV v13 Science and Validation Report (Saunders *et al.*, 2020), and four parameterisations for ice cloud diameter (see the RTTOV user guide) that may be selected via the *icede_param* option. In most cases, the RTTOV parameterisations must be used.

However, when ingesting DWD ICON model data, the GRIB file may optionally contain fields for liquid and/or ice particle radius. RadSim will use these fields to specify cloud particle diameters in RTTOV explicitly if the *use_model_clw_deff* and/or *use_model_ice_deff* options are true. For cloud liquid water, in cases where the specified model D_{eff} is less than 5 μ m (and non-zero) it is clipped to 5 μ m. For cloud ice, in cases where the specified model D_{eff} is less than 10 μ m (and non-zero), the cloud ice concentration is scaled by a factor $(10/D_{eff})$ to ensure an approximately constant extinction coefficient, and then the diameter is clipped to 10 μ m. This is consistent with the treatment of cloud particle size at DWD. If the model liquid/ice D_{eff} is zero, then the corresponding cloud concentration is set to zero, also consistent with the treatment at DWD.

Where the ICON cloud liquid water D_{eff} field is not used, the ICON GRIB file may contain the model density field. In this case, instead of using the RTTOV CLW D_{eff} parameterisation, the equivalent calculation is done by RadSim as follows:

$$D_{\rm eff} = 2 \times 10^6 \left(\frac{3 \rm LWC}{4 \pi k n_{tot} \rho_w}\right)^{\frac{1}{3}}$$

where $D_{\rm eff}$ is the particle effective diameter (μ m), LWC is the cloud liquid water concentration (kg/m³), *k* is 0.67 for continental clouds and 0.80 for maritime clouds (no units), n_{tot} is the cloud particle number density (cm⁻³) and ρ_w is the density of water (taken as 1000 kg/m³). The continental and maritime values of n_{tot} are parameterised in terms of aerosol concentration in the layer immediately below the cloud. For simplicity, the values used in RadSim and RTTOV are 100 and 300 cm⁻³ for maritime and continental clouds respectively which correspond approximately to the mid-points of the two ranges in Figure 8c of Martin *et al.* As in RTTOV, computed $D_{\rm eff}$ smaller than 10 μ m are reset to 10 μ m.

The NWP model cloud concentration fields are in units of kg/kg and these are converted to LWC (kg/m³) by multiplying by the density field from the model data. The above calculation of $D_{\rm eff}$ is consistent with that in RTTOV, except that the RTTOV conversion of the cloud units from kg/kg to kg/m³ is slightly different (the density being calculated within RTTOV), and as such the results are not identical for simulations based on DWD ICON fields with/without the density field.

It should be noted that the Martin *et al.* parameterisation was explicitly determined for warm stratocumulus clouds and as such may be limited in its applicability in convective regimes, for example. It is recommended to use the ICON cloud particle size fields if available. Ingest of cloud particle sizes is not available for other model types.



ANNEX G – CADS height assignments

If the *cads_height_assign_threshold* configuration namelist variable is set to a positive number, it activates the output of channel height assignments consistent with the NWP SAF Cloud and Aerosol Detection Software (CADS) package.

The height for each profile and channel is calculated as the first level *j* starting from the surface and working upwards for which the following relation is satisfied:

|clear - overcast(j)| / clear = threshold

where *clear* is the RTTOV simulated clear-sky radiance for the channel, *overcast(j)* is the RTTOV simulated overcast radiance for the channel assuming an opaque, black cloud at level *j* and *threshold* is the value specified in *cads_height_assign_threshold* (this is usually set to 0.01 i.e. 1%). The value returned is a real number, representing the first interpolated "level number" counting upwards from the surface at which the above expression is satisfied.

If the left-hand side of the above expression exceeds the threshold for j=nlevels (i.e. for the bottom level), then the returned level number is *nlevels*. If the left-hand side of the above expression is less than the threshold for all levels (very unlikely), then the returned level number is 2. The returned value lies in the range [2,*nlevels*].

Technical note: RTTOV calculates overcast radiances assuming an opaque cloud at the *bottom* of each *layer*. This means that in RTTOV, the overcast array is indexed from 1 to *nlayers* (rather than *nlevels*) and in practice *overcast(j)* in RTTOV refers to the overcast radiance with an opaque cloud at level j+1. Regardless of this technical detail, the value returned by RadSim for the height assignment corresponds to the actual (interpolated) level number at which the overcast layer would lie such that the above expression is satisfied.



ANNEX H – Footprint calculations

As noted in section 5.1.3, RadSim models footprints as ellipses with the semi-major axis aligned along the view path from the observation location to the satellite. The satellite azimuth angle for the obs location therefore gives the azimuth angle (as measured clockwise from north) of the semi-major axis of the footprint ellipse.

Let ϕ_o be the satellite azimuth angle of the observation, and let R_y and R_x be the semi-major and semi-minor axes (radii) of the footprint ellipse in km. These are specified in the obs data file for each observation.

The footprint calculations assume a spherical Earth of radius R_e =6371km. Let *d* be the great circle distance between an observation at latitude λ_o and longitude Λ_o and a model grid point at λ_m , Λ_m , and let α be the central angle between them in radians. Then we have:

 $d = R_e \alpha$ $\cos \alpha = \sin \lambda_o \sin \lambda_m + \cos \lambda_o \cos \lambda_m \cos (\Lambda_o - \Lambda_m)$

When identifying grid points that lie within the footprint, we can immediately exclude all grid points that are a distance larger than R_y from the observation. We therefore exclude grid points for which:

$$\cos\alpha < \cos\left(R_y/R_e\right)$$

For each remaining grid point falling within the circle of radius R_y of the observation, we apply the following procedure: we first compute the azimuth angle of the grid point relative to the observation location and subtract from this the observation satellite azimuth angle to obtain the azimuth angle of the grid point measured clockwise from the semi-major axis of the ellipse. We then consider Cartesian axes with the y axis parallel to the semi-major axis and centred at the observation location. We may calculate the x and y distance of the grid point from the observation with respect to these axes and finally deduce whether the point lies within the footprint using the standard equation of an ellipse. This process is elucidated below.

Figure H1 below schematically illustrates an observation O and a model grid point M. In Fig H1 b) (middle) we see the observation with associated azimuth angle (ϕ_o) and semi-major and semi-minor axes of the footprint (R_y and R_x respectively). Also marked is the azimuth angle of the grid point with respect to the observation ϕ_m .

In Fig H1 a) (left) we see the same observation and grid point on the Earth sphere, with the north pole N and centre of the Earth C also indicated. With reference to quantities defined above, we can observe the following angles (all angles are expressed in radians):

$$\widehat{OCM} = \alpha \qquad \widehat{OCN} = \frac{\pi}{2} - \lambda_o$$
$$\widehat{MCN} = \frac{\pi}{2} - \lambda_m \qquad \widehat{NOM} = \phi_m$$

We note the distances $CO = CN = CM = R_e$ and distance $OM = d = R_e \alpha$.



Figure H1: schematic of a satellite observation at location O and a model grid point M.

- a) On the spherical Earth with the north pole (N) and Earth centre (C) marked.
- b) The observation footprint modelled as an ellipse on the Earth's surface.
- c) A detail of b) indicating the distances d_x and d_y that are calculated to determine whether M lies within the footprint ellipse around O.

We calculate ϕ_m using the cosine rule for spherical geometry:

$$\cos\left(\frac{\pi}{2} - \lambda_m\right) = \cos\left(\frac{\pi}{2} - \lambda_o\right)\cos\alpha + \sin\left(\frac{\pi}{2} - \lambda_o\right)\sin\alpha\cos\phi_m$$

which yields:

$$\cos\phi_m = \frac{\sin\lambda_m - \sin\lambda_o \cos\alpha}{\cos\lambda_o \sin\alpha} \qquad (H1)$$

We are then able to calculate distances d_x and d_y indicated in Fig H1 c) (right):

$$d_x = R_e \alpha \sin(\phi_m - \phi_o)$$

$$d_y = R_e \alpha \cos(\phi_m - \phi_o)$$

Finally, from the standard equation of an ellipse we determine that M lies within the footprint if the following is satisfied:

$$\frac{d_x^2}{R_x^2} + \frac{d_y^2}{R_y^2} \le 1$$

There are two special cases where the denominator in equation H1 above is zero. The first is when O and M are coincident ($\alpha = 0$) in which case the grid point clearly lies within the footprint. The second is when $\cos \lambda_o = 0$ which occurs when O lies on the north or south pole. In this case we take ϕ_m to be the difference in longitude between the M and that given for O. The longitude of O should be specified such that it is consistent with the given observation azimuth angle if O were moved infinitesimally off the pole. When O lies on the north pole, ϕ_m is set to π minus the difference in longitude (in radians).

ANNEX I – RadSim v3.2 configuration namelist variable equivalents

The following table lists the RadSim v3.2 configuration namelist variables and gives the RadSim v4.0 equivalents where they exist. See section 4.1.

Variables that have changed name are highlighted in red. The table also indicates where default values have changed.

RadSim v3.2 variable name	RadSim v4.0 variable name	
	Input data	
obs_datafile	obs_datafile	
model_datafile	model_datafile	
model_ancil_datafile	model_ancil_datafile	
model_filetype	model_filetype	
rttov_coeffs_dir	rttov_coefs_dir	
rttov_coeffs_type	rttov_coefs_type	
rttov_coeffs_options	rttov_coefs_options	
rttov_hydrotable_dir	rttov_hydrotable_dir	
rttov_sccld_dir	rttov_hydrotable_dir	
rttov_mfasis_cld_dir	N/A (MFASIS-LUT no longer supported).	
rttov_mfasis_nn_dir	rttov_mfasis_nn_dir	
rttov_scaer_dir	rttov_aertable_dir	
rttov_pccoeffs_dir	rttov_pccoefs_dir	
rttov_htfrtc_coeffs_dir	N/A (HTFRTC no longer supported).	
rttov_coeff_file	rttov_coef_file	
rttov_hydrotable_file	rttov_hydrotable_file	
rttov_pol_coeff_file	rttov_mw_pol_coef_file	
rttov_sccld_file	rttov_hydrotable_file	
rttov_mfasis_cld_file	N/A (MFASIS-LUT no longer supported).	
rttov_mfasis_nn_file	rttov_mfasis_nn_file	
rttov_scaer_file	rttov_aertable_file	
rttov_pccoeff_file	rttov_pccoef_file	
rttov_htfrtc_static_file	N/A (HTFRTC no longer supported).	
rttov_htfrtc_sensor_file	N/A (HTFRTC no longer supported).	
emiss_atlas_dir	emis_atlas_dir	
brdf_atlas_dir	brdf_atlas_dir	
read_footprint_file	read_footprint_file	
Satelli	te instrument variables	
platform	platform	
inst	inst	
satid	satid	
channels	channels	
RadSim control variables		
temporal_data	temporal_data	
enable_footprints	enable_footprints	
run_scatt	hydrometeors (same RTTOV option used for VIS, IR, MW)	
run_mfasis	run_mfasis	
ircloud_use_model_clw_deff	use_model_clw_deff	
ircloud_use_model_ice_deff	use_model_ice_deff	
output_mode	output_mode	
seaice_threshold	seaice_threshold	
coast_threshold	coast_threshold	
co2_max_ppmv	co2_max_ppmv	
ssu_co2_cell_pressure	ssu_co2_cell_pressure	



NWP SAF

cads height assign threshold	cads height assign threshold
calc geo sat angles	calc geo sat angles
geo sat lat	geo sat lat
geo sat lon	geo sat lon
geo sat height	geo sat height
zen angle	zen angle
azi angle	azi angle
sun zen andle	sun zen angle
sun azi angle	sun azi andle
use emiss atlas	use emis atlas
use brdf atlas	use brdf atlas
emis atlas id	emis atlas id
use all atlas months	use all atlas months
default brdf land	default brdf land
default brdf seaice	default brdf seaice
max array size	max array size
max_profs	max profs
nthreads	nthreads
nprofs per call	nprofs per call
	RTTOV options
(these are commuted directly to RTTOV equivalents, see RTTOV Users' Guide for more information)	
apply reg limits (default true)	apply reg limits (default is now false)
opdep13 gas clip	opdep13 gas clip
interp mode (default 4 if nlev >	
ncoef lev 1 otherwise)	interp_mode (default is now 4)
ozone data	o3 data
addsolar	solar
ravleigh depol	N/A (no longer an RTTOV option)
rayleigh_deper	ravleigh single scatt
rayleigh max wavelength	rayleigh max wavelength
rayleigh min pressure	rayleigh_max_wavelength
do nite correction	nlte correction
addrefrac	refraction
plane parallel	plane parallel
fastem version (default 7=SURFEM-	
Ocean)	mw_sea_emis_model (default is now 3=SURFEM-Ocean)
ir sea emis model	ir sea emis model
solar sea brdf model (default	
2=Elfouhaily et al)	solar_sea_refl_model (default 1=Elfouhaily et al)
do lambertian	lambertian
lambertian fixed angle	lambertian fixed angle
clw data (default true)	clw data (default is now false)
mw clw scheme	permittivity param
mw_clw_cloud_top	clw cloud top
ir addaerosols cams	aerosols
ir addclouds	hydrometeors (same RTTOV option used for VIS_IR_MW)
ir scatt model (default 2=Chou-scaling)	thermal solver (default is now 3=delta-Eddington)
vis scatt model	solar solver
dom rayleigh	ravleigh multi scatt
dom nstreams	dom nstreams
dom accuracy	dom_accuracy
dom onden threshold	dom opden threshold
nol mode	mw_pol_mode
ice polarisation	ice_polarisation
ireland dw. schomo	



ircloud_ice_scheme	visir_ice_properties
ircloud_icede_param	icede_param
cc_threshold (default 1.E-5)	col_threshold (default is now 0.)
cldcol_threshold (default 1.E-3)	col_threshold (default is now 0.)
addpc	enable_pcrttov
htfrtc	N/A (no longer an RTTOV option)
npscores	npscores
ipcbnd	pc_band
ipcreg	pc_reg
Output control variables	
output_dir	output_dir
output_file	output_file
write_footprint_file	write_footprint_file
write_radiances	write_radiances
write_emiss	write_emis
write_brdf	write_brdf
write_profiles	write_profiles
write_latlon	write_lation
write_geom_height	write_geom_height
write_trans	write_trans
write_tjac	write_tjac
write_qjac	write_qjac
write_o3jac	write_o3jac
write_tskinjac	write_tskinjac
write_wind10mjac	write_wind10mjac
write_emissjac	write_emisjac