

Radiance Simulator v3.2 User Guide

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 December 2016, between EUMETSAT and the Met Office, UK, by one or more partners within the NWP SAF. The partners in the NWP SAF are the Met Office, ECMWF, DWD and Météo France.


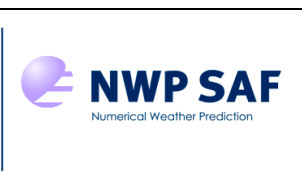
COPYRIGHT 2023, EUMETSAT, ALL RIGHTS RESERVED.

Change record			
Version	Date	Author / changed by	Remarks
0.1	15/01/2020	J. Hocking	First draft
0.2	09/02/2021	J. Hocking	Updates during beta phase
0.3	10/03/2021	J. Hocking	Updates after beta phase
0.4	30/03/2021	J. Hocking	Updates after internal review
0.5	26/04/2021	J. Hocking	Updates after DRR
1.0	18/01/2022	J. Hocking	Updates for v3.1
1.1	20/07/2023	J. Hocking	Updates for v3.2
1.2	26/07/2023	J. Hocking	Updates after internal review

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.....	7
4.	INPUT/OUTPUT FILES	8
4.1	Configuration namelist file	8
4.2	Observation data file	18
4.3	Model data files	20
4.3.1	Met Office UM Fieldsfiles / PP files (<i>file type 0</i>).....	22
4.3.2	GRIB files (<i>file types 1, 6, 7 and 8</i>)	23
4.3.3	NetCDF files (<i>file type 5</i>)	28
4.3.4	NWP-SAF 60L, 91L and 137L profile datasets (<i>file types 2, 3, 4</i>).....	30
4.3.5	Temporal interpolation.....	30
4.4	RTTOV Coefficients.....	31
4.5	Output files	33
4.5.1	Naming convention	33
4.5.2	Contents.....	33
4.5.3	File size	36
5.	PROCESSING DETAILS.....	36
5.1	Use of observation information.....	36
5.1.1	Horizontal interpolation (regular lat-lon grid).....	36
5.1.2	Temporal interpolation.....	37
5.1.3	Footprint simulation.....	37
5.2	Non-standard grids	39
5.2.1	Coordinate rotation	39
5.3	Field conversions.....	40
5.3.1	Potential temperature conversion.....	40
5.3.2	Humidity conversion.....	40
5.3.3	Rain/snow flux conversion	41
5.4	Pressure levels	41
5.5	Surface type.....	42
5.6	Emissivity and BRDF values	43
5.7	Satellite zenith and azimuth angles	43
5.8	Solar radiation	44
5.9	Visible/IR cloud simulations	44
5.10	Visible/IR CAMS aerosol simulations.....	45
5.11	Principal Components simulations	46
5.12	Running RTTOV.....	46
5.13	Quality control	47
5.14	Memory Usage and Batching	48
5.14.1	Memory Requirement for Model Fields	49
5.14.2	Observation Profile Batching	49
5.14.3	Memory Requirement for Output Data	50
5.14.4	Process Batching	50

6.	LIMITATIONS AND KNOWN ISSUES	51
7.	REFERENCES.....	51
8.	ANNEXES.....	52
	ANNEX A – radsim_run.py	52
	ANNEX B – radsim_geo_obs.py	54
	ANNEX C – radsim_plot_example.py	61
	ANNEX D – Example MARS, CDS and ADS requests	63
	ANNEX E – Solar angle calculations	67
	ANNEX F – Cloud liquid water and ice effective diameters	70
	ANNEX G – CADS height assignments	71
	ANNEX H – Footprint calculations.....	72

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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. In order 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.

2. INSTALLATION

2.1 System Requirements

- Unix or Linux operating system
- Fortran-90 compiler with some basic F2003 support
- Python3 (v3.7 or later*) for some ancillary scripts
- 20 MB free disk space for installation (approximate)
- 4GB RAM minimum recommended for running simulations.**

*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.


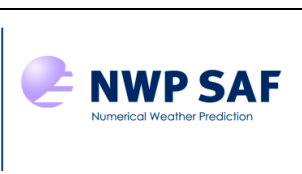
The code has been tested successfully in one or more standard configurations after installation with each of the following Fortran compilers:

- ifort - v15.0.0, v17.0.7
- pgfortran - v18.7-0
- gfortran - v8.1.0, v11.2.0

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 version 1 of RadSim 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.

 <p>The EUMETSAT Network of Satellite Application Facilities</p>	 <p>NWP SAF Numerical Weather Prediction</p>	<p>Radiance Simulator v3.2 User Guide</p>	<p>Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023</p>
--	---	--	--

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 v13.2

Available from <https://nwp-saf.eumetsat.int/site/software/rttov/>

RTTOV coefficient files are available from

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

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 v13 distribution. The ASCII *.dat* files or the HDF5 (*.H5*) files may be used. Note that the use of HDF5 format requires RTTOV to be installed with HDF5 support switched on explicitly. This is recommended as some coefficients are only available in this format.

In addition to the RTTOV optical depth coefficients, input files for visible/IR cloud, MFASIS LUTs, MFASIS-NN, MW scattering (hydrotable), PC-RTTOV or HTFRTC will be needed if using one of those options. The optical depth coefficients are for *clear-sky* simulations using the standard RTTOV model only.

- ecCodes (version 2.0.0 or later, version 2.23.0 or later recommended)

Available from <https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home>

NB For ingest of ICON cloud liquid and ice particle size fields, ecCodes v2.23.0 or later is required. It is generally recommended to use the most recent version of ecCodes.

- netCDF (version 4.0 or later)

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

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 (version 1.8.8 or later, used by netCDF and RTTOV)

Available from <http://www.hdfgroup.org/HDF5/>

- Jasper and libaec (optionally used by ecCodes)

2.3 Compiling the code

All the RadSim code is contained in the distribution file `radsim-3.2.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 in the installation directory

```
tar -xf radsim-3.2.tar.gz
```

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

```
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.

These are defined in the following environment variables:


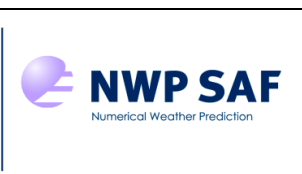
```
FC                # Name of Fortran compiler
ECCODES_DIR      # Path to ecCodes installation directory
NETCDF_DIR       # Path to netCDF-4 installation directory
RTTOV_DIR        # Path to RTTOV installation directory
```

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

```
HDF_DIR          # Path to HDF5 installation directory
JASPER_DIR       # Path to Jasper installation directory
AEC_DIR          # Path to libaec installation directory
```

Each library path should contain the standard *lib* and *include* subdirectories and in the case of RTTOV a *mod* subdirectory.

This is usually sufficient but in some cases the default compiler options may need to be modified.

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

Notes on compiler options:

Compiler options may be changed by editing the relevant compiler configuration file in the `build` 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 datasets are strictly big-endian byte order with the exception of GRIB and netCDF which are portable formats. It is assumed that the Radiance Simulator 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.


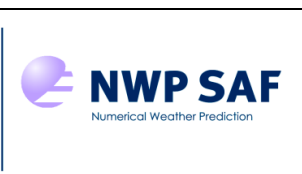
3. RUNNING THE CODE

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 `radsim_cfg.nl` can be specified in environment variable `RADSIM_NL` 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-

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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`)
- Path to directory containing RTTOV coefficient files (variable `rttov_coeffs_dir`) or, for HTFRTC simulations, path to directory containing HTFRTC coefficient files (variable `rttov_htrtc_coeffs_dir`, see sections 4.4 and 5.11)
- Satellite instrument identifiers (variables `platform`, `inst`, `satid`)

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 also 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.

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 namelist consists of the following variables (Fortran intrinsic type and inclusion status or default value are given in parentheses, compulsory variables also highlighted in bold):


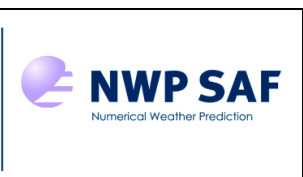
Variable name	Type, Status, Values	Description	Note
Input data			
obs_datafile	<i>character, optional</i>	Path to observation data file. This allows the simulation to be performed at observation locations. The file format is described in section 4.2.	
model_datafile	<i>character, compulsory</i>	Path to model data file. Several formats are supported (see section 4.3).	
model_ancil_datafile	<i>character, optional</i>	Path to DWD ICON “invariant” model data file or ECMWF netCDF file containing surface/single-level fields (see section 4.3).	
model_filetype	<i>integer, compulsory</i>	Model file type identifier. See section 4.3.	
rttov_coefs_dir	<i>character, compulsory (unless running HTFRTC simulations)</i>	Path to RTTOV optical depth coefficients directory. See section 4.4.	
rttov_coefs_type	<i>character, default='.dat'</i>	Coefficient file type, should be '.dat' or '.H5'. Not used if specifying coefficient filenames explicitly.	
rttov_coefs_options	<i>character, optional</i>	Additional string in <i>rtcoef</i> filename such as “_o3co2” or “_ironly” (note this should include any underscore). Not used if specifying coefficient filenames explicitly.	6a
rttov_hydrotable_dir	<i>character, optional</i>	Path to RTTOV-SCATT hydrotable coefficients directory for use with the <i>run_scatt</i> option.	7
rttov_scclld_dir	<i>character, optional</i>	Path to RTTOV visible/IR cloud coefficients directory for use with the <i>ir_addclouds</i> option.	
rttov_mfasis_cld_dir	<i>character, optional</i>	Path to MFASIS LUT directory for use with the MFASIS LUT RTTOV solver.	
rttov_mfasis_nn_dir	<i>character, optional</i>	Path to MFASIS-NN coef file directory for use with the MFASIS-NN RTTOV solver.	6c
rttov_scaer_dir	<i>character, optional</i>	Path to RTTOV visible/IR aerosol coefficients directory for use with the <i>ir_addaerosols_cams</i> option.	6b
rttov_pccoeffs_dir	<i>character, optional</i>	Path to PC-RTTOV coefficients directory for use with the <i>addpc</i> option.	
rttov_hftrtc_coefs_dir	<i>character, optional</i>	Path to HTFRTC coefficients directory for use with the <i>hftrtc</i> option.	
rttov_coeff_file	<i>character, optional</i>	Filename of RTTOV optical depth coefficient file.	

rttov_hydratable_file	<i>character, optional</i>	Filename of RTTOV-SCATT hydratable file directory for use with the <i>run_scatt</i> option.	7
rttov_pol_coeff_file	<i>character, optional</i>	Filename of RTTOV-SCATT polarisation input file for use with the <i>run_scatt</i> option and <i>pol_mode=2</i> (only required if the filename has been changed).	5,6c
rttov_scclld_file	<i>character, optional</i>	Filename of RTTOV visible/IR cloud coefficient file for use with the <i>ir_addclouds</i> option.	
rttov_mfasis_cld_file	<i>character, optional</i>	Filename of MFASIS LUT file for use with the RTTOV MFASIS LUT solver.	
rttov_mfasis_nn_file	<i>character, optional</i>	Filename of MFASIS-NN coef file for use with the RTTOV MFASIS-NN solver.	6c
rttov_scaer_file	<i>character, optional</i>	Filename of RTTOV visible/IR aerosol coefficient file for use with the <i>ir_addaerosols_cams</i> option.	6b
rttov_pccoeff_file	<i>character, optional</i>	Filename of PC-RTTOV coefficient file for use with the <i>addpc</i> option.	
rttov_hftrtc_static_file	<i>character, optional</i>	Filename of HFTRTC static coefficient file for use with the <i>hftrtc</i> option.	
rttov_hftrtc_sensor_file	<i>character, optional</i>	Filename of HFTRTC sensor-specific file for use with the <i>hftrtc</i> option.	
emiss_atlas_dir	<i>character, optional</i>	Path to emissivity atlas directory	
brdf_atlas_dir	<i>character, optional</i>	Path to BRDF atlas directory	
read_footprint_file	<i>character, optional</i>	Full path to netCDF file containing footprint data that was created by specifying <i>write_footprint_data</i> in a previous run.	6b
Satellite instrument variables			
platform	<i>character, compulsory</i>	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.	
inst	<i>character, compulsory</i>		
satid	<i>integer, compulsory</i>		
channels	<i>integer array, optional</i>		List of channel numbers to use. If omitted or if all values are 0 then all channels will be simulated.
General control variables			
temporal_data	<i>logical, default=F</i>	If true RadSim will read model data for all forecast times	

		present in the input <i>model_datafile</i> and will carry out temporal interpolation of the model data to observation times (requires <i>obs_datafile</i> 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.	
enable_footprints	<i>logical, default=F</i>	If true RadSim returns the mean radiance over all grid points falling within an ellipse around each observation. Requires <i>obs_datafile</i> to be specified including footprint ellipse major/minor axes (radii) and azimuth angle for each observation.	6a
run_scatt	<i>logical, default=F</i>	Run MW scattering code RTTOV-SCATT. This requires extra fields in the input model data file (see section 4.3) and RTTOV hydrotable files.	
run_mfasis	<i>logical, default=F</i>	Run MFASIS-NN visible fast cloud model. This requires extra fields in the input model data file (see section 4.3), and RTTOV visible/IR cloud and MFASIS-NN coef files. This is a shortcut for setting <i>ir_addclouds</i> and <i>addsolar</i> to true and <i>vis_scatt_model</i> to 4.	
ircloud_use_model_clw_deff	<i>logical, default=F</i>	For cloud simulations using ICON GRIB fields, use the cloud liquid water particle size model field. Requires <i>ir_addclouds</i> to be true and the "CLW Deff" cloud scheme to be selected (<i>ircloud_clw_scheme</i>).	6b
ircloud_use_model_ice_deff	<i>logical, default=F</i>	For cloud simulations using ICON GRIB fields, use the cloud ice water particle size model field. Requires <i>ir_addclouds</i> to be true and the Baum ice cloud scheme to be selected (<i>ircloud_ice_scheme</i>).	6b
output_mode	<i>integer, default=1</i>	Diagnostic output mode. Three levels are defined: 1=Normal, 2=Verbose, 3=Debug.	
seaice_threshold	<i>real, default=0.2</i>	Separation threshold for sea or sea-ice surface type classification. Valid range [0 - 1]	
coast_threshold	<i>real, default=0.2</i>	Difference threshold for coastline. Valid range [0 - 0.5]	1

co2_max_ppmv	<i>real, default=-1.</i>	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. This is mandatory for “PMC-shift” SSU coefficients.	
ssu_co2_cell_pressure	<i>real array, dim(3), default=(-1,-1,-1)</i>	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 <i>co2_max_ppmv</i> .	6a
cads_height_assign_threshold	<i>real, default=-1.</i>	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).	6a
calc_geo_sat_angles	<i>logical, default=F</i>	If true, RadSim computes satellite zenith and azimuth angles for a geostationary satellite with sub-satellite point at <i>geo_sat_lat</i> , <i>geo_sat_lon</i> and at altitude <i>geo_sat_height</i> . Applies to both model grids and obs data files. See section 5.7.	6b
geo_sat_lat	<i>real, default=0.0</i>	Latitude of sub-satellite point (degrees, typically zero), used if <i>calc_geo_sat_angles</i> is true	6b
geo_sat_lon	<i>real, default=0.0</i>	Longitude of sub-satellite point (in interval [0,360) degrees), used if <i>calc_geo_sat_angles</i> is true	6b
geo_sat_height	<i>real, default=35800</i>	Altitude of satellite (km), used if <i>calc_geo_sat_angles</i> is true	6b
zen_angle	<i>real, default=0.0</i>	Satellite zenith angle (degrees) to use for model grid runs.	
azi_angle	<i>real, default=0.0</i>	Satellite azimuth angle (degrees) to use for model grid runs (0-360; measured clockwise, east=+90).	
sun_zen_angle	<i>real, default=-1.</i>	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>addsolar</i> is true.	
sun_azi_angle	<i>real, default=0.0</i>	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>addsolar</i> is true.	
use_emiss_atlas	<i>logical, default=F</i>	Use an emissivity atlas for land surfaces.	2
use_brdf_atlas	<i>logical, default=F</i>	Use the BRDF atlas for land surfaces if the <i>addsolar</i> option is enabled.	2
emis_atlas_id	<i>integer, default=1</i>	Selects which RTTOV emissivity atlas to use. For IR sensors: 1=UWIRemis, 2=CAMEL 2007, 3=CAMEL climatology. For MW sensors: 1=TELSEM2, 2=CNRM atlas.	
use_all_atlas_months	<i>logical, default=F</i>	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.	6a
default_brdf_land	<i>real, default=-1.</i>	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.	6a
default_brdf_seaice	<i>real, default=-1.</i>	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.	6a
max_array_size	<i>integer, default=500</i>	Maximum size of output array (in MB). Used to define number of output batches.	3
max_profs	<i>integer, default=500000</i>	Maximum number of interpolated profiles. Used to define number of batches for processing obs data.	3
nthreads	<i>integer, default=1</i>	If equal to 2 or more RadSim and RTTOV will make use of multiple threads. This requires	

		<h2>Radiance Simulator v3.2 User Guide</h2>	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

		RTTOV and RadSim to be compiled with OpenMP flags which is recommended.	
nprofs_per_call	<i>integer, default=5</i>	Number of profiles passed to RTTOV in each call.	
RTTOV-specific options (these are commuted directly to RTTOV equivalents. See RTTOV Users Guide for more information)			
addsolar	<i>logical, default=F</i>	Enable solar radiation for affected channels.	
rayleigh_depol	<i>logical, default=T</i>	Enable depolarisation in Rayleigh phase function.	5,6c
rayleigh_single_scatt	<i>logical, default=T</i>	Enable/disable Rayleigh single-scattering calculation for visible channels.	5,6a
rayleigh_max_wavelength	<i>real, default=2.0</i>	Maximum channel central wavelength (microns) above which Rayleigh scattering calculations are omitted.	5,6a
rayleigh_min_pressure	<i>real, default=0.0</i>	Minimum pressure (hPa) below which Rayleigh scattering calculations are omitted.	5,6a
ozone_data	<i>logical, default=F</i>	Use input ozone profile	
clw_data	<i>logical, default=T</i>	Use input cloud liquid water profile (applies to non-scattering MW simulation only)	
mw_clw_scheme	<i>integer, default=2</i>	Select the cloud liquid water permittivity parameterisation to use with <i>clw_data=T</i> . Valid values: 1=Liebe, 2=Rosenkranz, 3=TKC.	5
mw_clw_cloud_top	<i>real, default=322hPa</i>	Pressure level above which CLW is ignored in input cloud profiles (only applies if <i>clw_data</i> is true).	5
cc_threshold	<i>real, default=0.001</i>	RTTOV-SCATT <i>cc_threshold</i> parameter which excludes cloudy columns with a cloud fraction below this value.	5
pol_mode	<i>integer, default=1</i>	RTTOV-SCATT <i>pol_mode</i> option to select treatment of polarisation: 0=no pol, 1=empirical, 2=ARO scaling (requires additional pol coeff file).	6c
ice_polarisation	<i>real, default=1.4</i>	RTTOV-SCATT <i>ice_polarisation</i> parameter which controls the polarisation approximation (see the RTTOV user guide).	5,6a
ir_addaerosols_cams	<i>logical, default=F</i>	Run visible/IR aerosol scattering simulations using CAMS aerosol fields.	6b
ir_addclouds	<i>logical, default=F</i>	Run visible/IR cloud scattering code. Note that this option also requires additional RTTOV cloud coefficient files to be	

		specified (see <i>rttov_scclد_file</i> above).	
<code>ir_scatt_model</code>	<i>integer, default=2</i>	RTTOV scattering solver for thermal radiation: 1=DOM, 2=Chou-scaling (applies only if <i>ir_addclouds</i> is true).	5
<code>vis_scatt_model</code>	<i>integer, default=1</i>	RTTOV scattering solver for solar radiation: 1=DOM, 2=single-scattering, 3=MFASIS LUT, 4=MFASIS-NN (the last two apply only if <i>ir_addclouds</i> and <i>addsolar</i> are true).	
<code>dom_nstreams</code>	<i>integer, default=8</i>	Number of streams for DOM calculations (only applies if DOM solver selected).	
<code>dom_accuracy</code>	<i>real, default=0.0</i>	DOM accuracy parameter (only applies if DOM solver selected).	
<code>dom_opdep_threshold</code>	<i>real, default=0.0</i>	DOM absorption optical depth threshold (only applies if DOM solver selected).	
<code>dom_rayleigh</code>	<i>logical, default=F</i>	If true enables Rayleigh multiple scattering calculations with the solar DOM solver. Note that this is very computationally expensive.	6a
<code>clدcol_threshold</code>	<i>real, default=1.E-5</i>	Specify the RTTOV <i>clدcol_threshold</i> variable for visible/IR cloud simulations which excludes cloud columns with weights below this value.	5,8
<code>ircloud_clw_scheme</code>	<i>integer, default=1</i>	Cloud liquid water optical properties to use with <i>ir_addclouds=T</i> . Valid values: 1=OPAC, 2="Deff" properties.	
<code>ircloud_ice_scheme</code>	<i>integer, default=3</i>	Ice cloud optical properties to use with <i>ir_addclouds=T</i> . Valid values: 1=Baum, 2=Baran2014, 3=Baran2018.	
<code>ircloud_icede_param</code>	<i>integer, default=2</i>	Ice particle size parameterisation to use with <i>ir_addclouds=T</i> and <i>ircloud_ice_scheme=1</i> . Valid values: 1-4.	5,9
<code>fastem_version</code>	<i>integer, default=7</i>	Version of the FASTEM microwave emissivity model to use for MW simulations. Valid values:0-7 (0 selects the TESSEM2 model, 7 selects SURFEM-Ocean).	
<code>ir_sea_emis_model</code>	<i>integer, default=2</i>	Select IR sea surface emissivity model. Valid values: 1-2.	5
<code>solar_sea_brdf_model</code>	<i>integer, default=2</i>	Select solar sea BRDF model. Valid values: 1-2.	5
<code>do_lambertian</code>	<i>logical, default=F</i>	Treat surface as Lambertian reflector.	

lambertian_fixed_angle	<i>logical, default=F</i>	Use fixed angle for downwelling Lambertian radiances (T) or use parameterisation for angle (F) (only applies if <i>do_lambertian</i> option is true)	5
do_nlte_correction	<i>logical, default=F</i>	Apply the non-local thermodynamic equilibrium bias correction (requires an RTTOV coefficient file which supports this).	
apply_reg_limits	<i>logical, default=T</i>	Clip input profile values to the limits from the RTTOV optical depth regression where these limits are exceeded.	5
opdep13_gas_clip	<i>logical, default=T</i>	Reset individual gas optical depths to zero when negative values are predicted.	5,6c
interp_mode	<i>integer, default=-1</i>	Select the RTTOV interpolation mode. Valid modes: 1-5. If a value less than 1 or greater than 5 is specified (the default), then RadSim selects the recommended mode as described in the RTTOV user guide (mode 4 if the number of model levels exceeds the number of RTTOV coefficient levels, otherwise mode 1).	5
addrefrac	<i>logical, default=T</i>	Include effects of atmospheric refraction.	5
plane_parallel	<i>logical, default=F</i>	Enable strict plane-parallel geometry (NB this is enforced if DOM solver is selected for visible/IR cloud simulations).	5
addpc	<i>logical, default=F</i>	Enable PC-RTTOV simulations (requires PC-RTTOV coefficient file).	
htfrtc	<i>logical, default=F</i>	Enable HTFRTC simulations (requires HTFRTC static and sensor-specific coefficient files).	
npcscores	<i>integer, default=300</i>	Number of PC scores to compute for PC-RTTOV and HTFRTC simulations.	
ipcbnd	<i>integer, default=1</i>	Spectral band for PC-RTTOV simulations.	5
ipcreg	<i>integer, default=1</i>	Regression channel set for PC-RTTOV simulations.	
Output control variables			
output_dir	<i>character, optional</i>	Path to output directory. The default is to use the current directory (see section 4.5 for more details).	
output_file	<i>character, optional</i>	Output file name. If unspecified, RadSim	6a

		constructs a filename based on the sensor and data validity time (see section 4.5).	
write_footprint_file	<i>character, optional</i>	Filename of 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> .	6b
write_radiances	<i>logical, default=F</i>	Write radiances to the output file. The default is to write brightness temperatures.	
write_emiss	<i>logical, default=F</i>	Write emissivity values to the output file.	
write_brdf	<i>logical, default=F</i>	Write BRDF values to the output file for use with <i>addsolar</i> option	
write_profiles	<i>logical, default=F</i>	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_latlon	<i>logical, default=F</i>	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	<i>logical, default=F</i>	Write geometric heights of pressure levels computed by RTTOV.	6a
write_trans	<i>logical, default=F</i>	Write layer transmittances to the output file.	4
write_tjac	<i>logical, default=F</i>	Write temperature Jacobians to the output file.	4
write_qjac	<i>logical, default=F</i>	Write humidity Jacobians to the output file.	4
write_o3jac	<i>logical, default=F</i>	Write ozone Jacobians to the output file.	4
write_tskinjac	<i>logical, default=F</i>	Write skin temperature Jacobians to the output file.	6a
write_wind10mjac	<i>logical, default=F</i>	Write 10m wind u/v component Jacobians to the output file.	6a
write_emissjac	<i>logical, default=F</i>	Write surface emissivity Jacobians to the output file (see the RTTOV user guide for more information about emissivity Jacobians).	6a

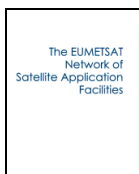
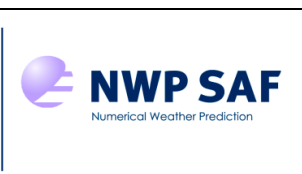
		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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 v3.0 (a), RadSim v3.1 (b), or in RadSim v3.2 (c).
7. These options are equivalent to the “mietables” options in the previous release for specifying the RTTOV-SCATT optical property files for MW scattering simulations.
8. Equivalent to the *cldstr_threshold* variable in RadSim v2.
9. Equivalent to the *ircloud_idg* variable in RadSim v2.

The following configuration namelist variables were removed in RadSim v3.0 (compared to RadSim v2.2) as they are no longer supported by RTTOV or are deprecated:
fix_hgpl, *lradiance*, *use_totalice*, *mw_clw_calc_on_coef_lev*, *apply_band_correction*, *reg_limit_extrap*.

4.2 Observation data file


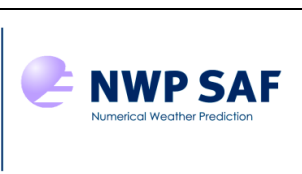
This optional file contains observation metadata. The purpose is to allow simulations to be performed at specific locations with realistic satellite viewing geometry (see section 5.1 for more details). If not specified, the simulation will be performed at model profile locations assuming a nadir zenith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

Two versions are available 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. The formats are described below.

Version 1

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)

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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



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. Additional fields that may be supplied are

- *View identifier* (any integer, not used by RadSim)
- *Scan line* (any integer, not used by RadSim)
- *Scan position* (any integer, not used by RadSim)
- *Satellite azimuth angle* (degrees)
- *Solar zenith angle* (degrees)
- *Solar azimuth angle* (degrees)
- *Year* (integer)
- *Month* (integer, 1-12)
- *Day* (integer, day of month)
- *Hour* (integer, 0-23)
- *Minute* (integer, 0-59)
- *Footprint semi-major axis* (km)
- *Footprint semi-minor axis* (km)

The first three 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 (`addsolar=T`): note both solar angles must be specified or neither. Azimuth angles are measured clockwise, with due north being zero and due east being $+90^\circ$. 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, 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).

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

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

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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 (*file type 0*)
- GRIB files:
 - data from ECMWF including CAMS (*file type 1*)
 - data from the DWD ICON model (*file type 6*)
 - data from the HARMONIE model (*file type 7*)
 - data from JMA (*file type 8*)
- NetCDF files:
 - data from ECMWF (excluding CAMS) (*file type 5*)
- 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*)

Required Fields


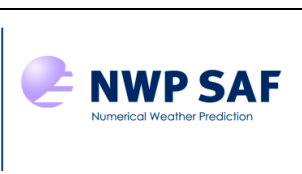
Regardless of the data source, the following fields must be available in the file, either directly or in an equivalent form that can be converted easily. Available conversions are described in section 5.3. Further details on file contents can be found in the following subsections. If optional features 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.

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

- Pressure
- Temperature
- Specific humidity

Required surface fields

- Pressure
- Temperature (at 2m)
- Specific humidity (at 2m)
- 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)

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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 ozone_data=T (enabling variable ozone in simulations)

Required on 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/coefficient-download/#RTTOV_rtcoef_coefficients

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

Required on levels:

- Specific liquid water content (cloud-liquid-water)

Option run_scatt=T (use of microwave scattering code)

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

- Specific liquid water content (cloud-liquid-water)
- Specific ice content (cloud-ice-water)
- Rain rate (optional)
- Snow rate (optional)
- Area cloud fraction

Option ir_addclouds=T (visible/infrared cloud scattering)

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

- Specific liquid water content (cloud-liquid-water)
- Specific ice content (cloud-ice-water)
- Area cloud fraction

Optional for ICON GRIB files:

- Cloud liquid water particle size
- Cloud ice water particle size
- Density

Option `ir_addaerosols_cams=T` (visible/infrared aerosol scattering for CAMS species)

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

- Sea salt size bin 1
- Sea salt size bin 2
- Sea salt size bin 3
- Dust size bin 1
- Dust size bin 2
- Dust size bin 3
- Hydrophilic organic matter
- Hydrophobic black carbon
- Sulphate

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 (or 408)	Pressure levels
409	Surface pressure
24	Surface skin temperature
3209	10m wind (u-component)
3210	10m wind (v-component)
3236	Temperature at 1.5m
3245 or 3237	Relative or specific humidity at 1.5m

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
60	Ozone (mixing ratio)	Required for <i>ozone_data</i>
254	Specific liquid water content (qcl)	Required for <i>clw_data</i> , <i>run_scatt</i> , <i>ir_addclouds</i>
12	Specific ice content (qcf)	Required for <i>run_scatt</i> (see note 1) and <i>ir_addclouds</i>
186	Rain rate	Optional for <i>run_scatt</i> (see note 2)
187	Snow rate	Optional for <i>run_scatt</i> (see note 1)
265	Area cloud fraction	Optional for <i>run_scatt</i> , <i>ir_addclouds</i> but strongly recommended (see note 3).
266	Bulk cloud fraction	Optional for <i>run_scatt</i> , <i>ir_addclouds</i> only if area cloud fraction unavailable (see note 3).

Notes:

1. 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 v13 hydrotables if running with the *run_scatt* option. The capability to use snow rate has been included but is not currently applicable.
2. Rain is not included in the liquid water field, it has to be added separately. Although desirable, it is not considered compulsory - the simulation can still be run without a rain-rate field if it's not available.
3. 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 and DWD ICON 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 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)
167	Temperature at 2m
168	Dewpoint temperature at 2m

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
203	Ozone (mixing ratio)	Required for <i>ozone_data</i>
246	Specific liquid water content (qcl)	Required for <i>clw_data</i> , <i>run_scatt</i> , <i>ir_addclouds</i> (see section 5.9)
247	Specific ice content (qcf)	Required for <i>run_scatt</i> and <i>ir_addclouds</i> (see section 5.9)
75	Rain (mixing ratio)	Optional for <i>run_scatt</i>
76	Snow (mixing ratio)	Optional for <i>run_scatt</i>
248	Area cloud fraction	Required for <i>run_scatt</i> , <i>ir_addclouds</i>
210001	CAMS sea salt size bin 1 (mixing ratio)	Optional for <i>ir_addaerosols_cams</i> (see section 5.10)
210002	CAMS sea salt size bin 2 (mixing ratio)	Optional for <i>ir_addaerosols_cams</i>
210003	CAMS sea salt size bin 3 (mixing ratio)	Optional for <i>ir_addaerosols_cams</i>
210004	CAMS dust size bin 1 (mixing ratio)	Optional for <i>ir_addaerosols_cams</i>
210005	CAMS dust size bin 2 (mixing ratio)	Optional for <i>ir_addaerosols_cams</i>
210006	CAMS dust size bin 3 (mixing ratio)	Optional for <i>ir_addaerosols_cams</i>
210007	CAMS hydrophilic organic matter (mixing ratio)	Optional for <i>ir_addaerosols_cams</i>
210010	CAMS hydrophobic black carbon (mixing ratio)	Optional for <i>ir_addaerosols_cams</i>
210011	CAMS sulphate (mixing ratio)	Optional for <i>ir_addaerosols_cams</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)

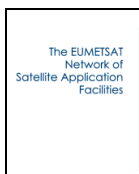
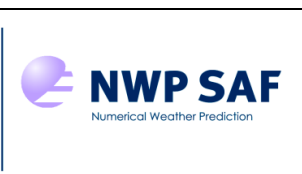
Notes:

1. Pressure levels are usually not included explicitly and need to be reconstructed from the surface pressure field and a set of 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 GRIB data (*file type 6*)

For ICON 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 *ParamId* key values. Multiple parameter IDs are available for some variables: depending on environment variable settings for ecCodes definitions, different parameter IDs can be

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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 (mandatory, 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)
167 / 500011	2t / t_2m	t_2m	Temperature at 2m
168 / 500017	2d / td_2m	td_2m	Dewpoint temperature at 2m
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)

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

ParamId(s)	ShortName(s)	ICON var name	Description	Usage notes
203 / 500242	o3	o3	Ozone (mixing ratio)	Required for <i>ozone_data</i>
54 / 500001	pres / p	pres_ifc	Pressure half-levels	Required for <i>run_scatt</i> (see note 2)
503056	qc_dia	tot_qc_dia	Total specific liquid water content	Required for <i>clw_data</i> , <i>run_scatt</i> , <i>ir_addclouds</i> (see section 5.9)
503057	qi_dia	tot_qi_dia	Total specific ice content	Required for <i>run_scatt</i> , <i>ir_addclouds</i> (see section 5.9)
260020 / 500102	rwmr / qr	qr	Rain (mixing ratio)	Optional for <i>run_scatt</i>
260021 / 500103	snmr / qs	qs	Snow (mixing ratio)	Optional for <i>run_scatt</i>
260257 / 500098	ccl / clc	clc	Cloud cover	Required for <i>run_scatt</i> , <i>ir_addclouds</i>
500779 / 503566	recloud	reff_qc	Cloud liquid water effective radius	Optional for <i>ir_addclouds</i> with <i>ircloud_clw_scheme</i> = 2 (see note 3)
500781 / 503568	reice	reff_qi	Cloud ice water effective radius	Optional for <i>ir_addclouds</i> with <i>ircloud_ice_scheme</i> = 1 (see note 3)
3089 / 500545	den	rho	Density	Optional for <i>ir_addclouds</i> with <i>ircloud_clw_scheme</i> = 2 (see note 4)

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 full (main) pressure levels are required for all simulations. The pressure half-levels are required only for RTTOV-SCATT (MW scattering) simulations. The two

fields use the same parameter IDs but are distinguished by the *typeOfLevel* attribute which is “*general/VerticalLayer*” for full levels and “*general/Vertical*” for half-levels.

- For cloud scattering simulations (*ir_addclouds* true) using cloud liquid or ice optical properties that explicitly depend on particle size (*ircloud_clw_scheme=2*, *ircloud_ice_scheme=1*), the ICON cloud liquid or ice effective radius fields may be present in the file. To enable use of them, the *ircloud_use_model_clw_deff* and/or *ircloud_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.
- 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 (*ircloud_clw_scheme = 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 GRIB data (file type 7)

For HARMONIE 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 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.

indicatorOfParameter	indicatorOfTypeOfLevel	level	Description
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)
11	105	2	Temperature at 2m
17	105	2	Dewpoint temperature at 2m
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.

indicatorOfParameter	indicatorOfTypeOfLevel	level	Description	Usage notes
66	105	0	Snow depth	Optional, only used with IR land surface emissivity atlases, see note 1
76	109	-	Specific liquid water content (qcl)	Required for <i>clw_data</i> , <i>run_scatt</i> , <i>ir_addclouds</i> (see section 5.9)

58	109	-	Specific ice content (qcf)	Required for <i>run_scatt</i> and <i>ir_addclouds</i> (see section 5.9)
181	109	-	Rain (mixing ratio)	Optional for <i>run_scatt</i>
184	109	-	Snow (mixing ratio)	Optional for <i>run_scatt</i>
71	109	-	Area cloud fraction	Required for <i>run_scatt</i> , <i>ir_addclouds</i>
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:


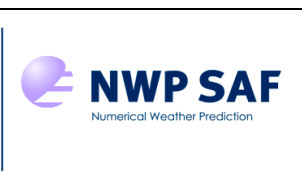
1. Snow depth (in metres) is optional. If present the RTTOV *snow_fraction* profile variable is set to *snow_depth* / 0.1, this value being clipped to lie in the interval [0,1]. This simple linear relationship where 10cm of snow implies full (100%) snow cover is broadly similar to the relationships used in the HARMONIE 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: cloud 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
133 / 174096	heightAboveGround	Specific humidity at 2m
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)

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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 *ozone_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*. In addition, the atmospheric data file must contain the *level* dimension. All atmospheric variables must have dimensions (*time*, *level*, *latitude*, *longitude*) and all surface (single-level) variables must have dimensions (*time*, *latitude*, *longitude*). Note that even if the files contain data for only a single validity/data time, the *time* dimension must be present (and in that case 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* 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.

The integer *time* variable, containing the values of the *time* dimension, 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”

RadSim relies on the attribute having this format 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 *ParamId* key value is given in brackets.


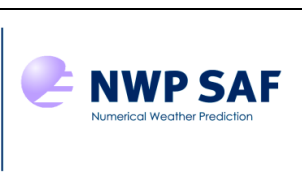
Variable name (GRIB ParamID)	Description (<i>atm = atmospheric file, sfc = surface file</i>)
t (130)	Temperature, <i>atm</i>
q (133)	Specific humidity, <i>atm</i>
p (54)	Pressure levels, <i>atm</i> (optional, see note 1 below)
sp (134)	Surface pressure, <i>sfc</i>
skt (235)	Surface skin temperature, <i>sfc</i>
u10 (165)	10m wind (u-component), <i>sfc</i>
v10 (166)	10m wind (v-component), <i>sfc</i>
t2m (167)	Temperature at 2m, <i>sfc</i>
d2m (168)	Dewpoint temperature at 2m, <i>sfc</i>
siconc (31)	Sea-ice fraction, <i>sfc</i>
z (129)	Geopotential (for orography), <i>sfc</i>
lsm (172)	Land/sea mask (land fraction), <i>sfc</i>

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

Variable name (GRIB ParamID)	Description (<i>atm = atmospheric file, sfc = surface file</i>)	Usage notes
o3 (203)	Ozone (mixing ratio), <i>atm</i>	Required for <i>ozone_data</i>
clwc (246)	Specific liquid water content (qcl), <i>atm</i>	Required for <i>clw_data</i> , <i>run_scatt</i> , <i>ir_addclouds</i> (see section 5.9)
ciwc (247)	Specific ice content (qcf), <i>atm</i>	Required for <i>run_scatt</i> and <i>ir_addclouds</i> (see section 5.9)
crwc (75)	Rain (mixing ratio), <i>atm</i>	Optional for <i>run_scatt</i>
cswc (76)	Snow (mixing ratio), <i>atm</i>	Optional for <i>run_scatt</i>
cc (248)	Area cloud fraction, <i>atm</i>	Required for <i>run_scatt</i> , <i>ir_addclouds</i>
tcc (164)	Total cloud cover, <i>sfc</i>	Not used in simulations (see note 2)
lcc (186)	Low cloud cover, <i>sfc</i>	Not used in simulations (see note 2)
mcc (187)	Medium cloud cover, <i>sfc</i>	Not used in simulations (see note 2)
hcc (188)	High cloud cover, <i>sfc</i>	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. Note that the netCDF files do not contain these coefficients: RadSim has coefficients for the ECMWF 60, 91 and 137 levels, but pressure levels for other vertical grids must be specified explicitly in the file. Note that reading explicit pressure half-levels is not currently supported so RTTOV-SCATT simulations can only be run on standard ECMWF levels for which the pressure half-levels can be calculated.
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.

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

4.3.4 NWP-SAF 60L, 91L and 137L profile datasets (*file types 2, 3, 4*)

These datasets contain sufficient information to run the optional visible/IR or MW cloudy (scattering) simulations if desired. Note that the 60L dataset does not contain rain or snow data.

The 91L and 137L 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 dataset includes 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.


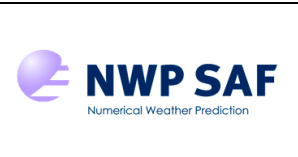
To specify these datasets (in particular the 91L or 137L datasets) the *model_datafile* variable should point to the relevant *.atm file. RadSim will then look for the associated *.sfc file in the same directory.

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

4.3.5 Temporal interpolation

Temporal interpolation is currently supported for ECMWF GRIB and netCDF data, DWD ICON GRIB data, HARMONIE 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. 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 1000 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.

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

All model fields are linearly interpolated in time. 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_rttov13*). Look in 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_rttov13/rttovXpredYL/* where X is the predictor version (7, 8 or 9 being the older predictors and 13 being the new ones introduced in RTTOV v13) and Y is the number of levels (usually 54 for most sensors, but also 101 for hyperspectral sounders). The v13 predictor files are generally recommended, but RTTOV v13 retains support for the older predictor versions.

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

```
rttov_coeffs_dir=/path_to_rttov13/rtcoef_rttov13/rttov13pred54L
```

You can specify the explicit filename for the coefficient file in the *rttov_coeff_file* configuration namelist variable: RadSim will look for this file in the *rttov_coeffs_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/H5* (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 new configuration variable *rttov_coeffs_options* is available which can be used to specify any additional text that appears in the filename between “instrument” and the file extension (“.dat”/“.H5”) such as “_o3co2” or “_ironly”.

Depending on the type of simulation, additional coefficient files may be required: visible/IR cloud and/or aerosol coefficient files, MFASIS LUT files, MFASIS-NN coef files, MW RTTOV-SCATT hydrotable files, PC-RTTOV coefficient files and HTFRTC 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 - in particular those for PC-

<p>The EUMETSAT Network of Satellite Application Facilities</p>		<h2 style="text-align: center;">Radiance Simulator v3.2 User Guide</h2>	<p>Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023</p>
---	---	---	--

RTTOV - 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_coeffs_type* variable specifies the file extension used when RadSim constructs the filenames. It can be set to “.H5” to read HDF5 input files instead of ASCII files (default: “.dat”). This applies to all input files except MFASIS LUTs (always assumed to be HDF5 format), hydrotable files (always assumed to be ASCII format), and HTFRTC input files (always assumed to be NetCDF format).

Unless running HTFRTC simulations, the the *rttov_coeffs_dir* variable is mandatory and an RTTOV optical depth coefficient file must be supplied for all simulation types.

For visible/IR cloud simulations, a cloud coefficients file must also be supplied: it must contain optical properties for the same set of channels as the optical depth coefficient file. For MFASIS LUT simulations, an MFASIS LUT file is required in addition to the cloud optical property file. Depending on the chosen cloud liquid water optical properties, the Deff or OPAC LUT is required. For MFASIS-NN simulations, an MFASIS-NN coef file is required in addition to the cloud optical property file.

For aerosol simulations using CAMS fields, an RTTOV CAMS aerosol coefficients file must also be supplied: it must contain optical properties for the same set of channels as the optical depth coefficient file.

For MW scattering simulations (using RTTOV-SCATT), the hydrotable (containing the hydrometeor optical properties) determines how the input fields are input to RTTOV. 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 v13.2 provides a new treatment of polarisation, specified by setting the new *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_coeff_file* config namelist option, otherwise the default filename will be assumed.

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

For HTFRTC simulations, two files must be supplied: the static HTFRTC coefficient file, and the sensor-specific HTFRTC file for the instrument being simulated. In this case no RTTOV optical depth coefficient directory/file is required.

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>-<datetime>.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 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


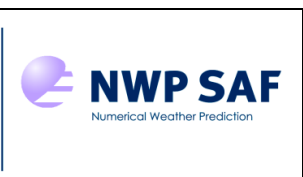
- *platform, satid, instrument* – character strings identifying the simulated instrument.
- *channels* – integer array of the channels used.
- *validity_time* – 5 element integer array containing year, month, day, hour, minute. This is not written if the profile data have multiple validity times.
- *data_time* – 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, halflevels* 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

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

- *rtsurf* – RTTOV surface type (land, sea or sea-ice)
- *qcflags* – quality control flags (see section 5.13 below for more details)
- *qcrttov* – rtov error status values
- *qcinfo* – additional quality control information
- *bt* OR *radiance* – simulated top-of-atmosphere brightness temperatures or radiances; only one of these is written


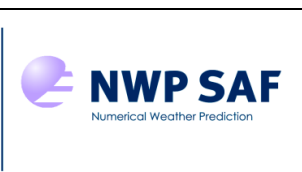
Fields that may be present depending on the options chosen:

- *refl* – simulated top-of-atmosphere reflectances (if solar radiation enabled, but not written if *write_radiances* is true)
- *emiss* – surface emissivities
- *brdf* – surface BRDFs (if solar radiation enabled)
- *tjac*, *qjac*, *o3jac* – temperature, humidity and ozone Jacobians
- *tskinjac*, *wind10mujac*, *wind10mvjac*, *emissjac* – 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* – altitudes of pressure 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)
- *nprof_per_footprint* – the number of profiles averaged in each footprint

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)
- *lon* - longitude (degrees, 0-360)
- *lsm* – land-sea mask (0-1)
- *zsurf* – surface height (m)
- *seaice* – sea-ice fraction (0-1)
- *pstar* – surface pressure (Pa)


		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

- *t2* – temperature at 2m (K)
- *q2* – specific humidity at 2m (kg/kg)
- *tskin* – surface skin temperature (K)
- *u10* – 10m wind u-component (m/s)
- *v10* – 10m wind v-component (m/s)
- *p* – pressure 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:

- *ph* – pressure at half-levels (Pa)
- *z* – height at pressure 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)
- *cams_sea_salt1* – CAMS sea salt size bin 1 (kg/kg)
- *cams_sea_salt2* – CAMS sea salt size bin 2 (kg/kg)
- *cams_sea_salt3* – CAMS sea salt size bin 3 (kg/kg)
- *cams_dust1* – CAMS dust size bin 1 (kg/kg)
- *cams_dust2* – CAMS dust size bin 2 (kg/kg)
- *cams_dust3* – CAMS dust size bin 3 (kg/kg)
- *cams_hphil_omat* – CAMS hydrophilic organic matter (kg/kg)
- *cams_hphob_bcar* – CAMS hydrophobic black carbon (kg/kg)
- *cams_sulphate* – CAMS sulphate (kg/kg)
- *o3* – ozone profiles (kg/kg)
- *co2* – CO₂ profiles (kg/kg) if the *co2_max_ppmv* option is used
- *density* – air density (kg/m³)
- *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 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 can be a significantly smaller output file that contains enough data to make geographical plots of simulation results.

<p>The EUMETSAT Network of Satellite Application Facilities</p>		<p>Radiance Simulator v3.2 User Guide</p>	<p>Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023</p>
---	---	---	--

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, which is the approximate file size limit on some older operating systems.

Output files are always netCDF-4 classic format using internal compression for array variables. This allows larger files to be created on those systems that support them.

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 in-flight by specifying observation positions and the viewing angle, rather than doing the calculations at model grid points and assuming a fixed view angle. The model fields must be defined for all validity times included in the model input file.

For Met Office UM fields, HARMONIE 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.


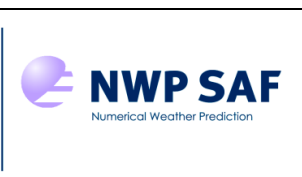
Model fields are interpolated horizontally to observation positions, and, if temporal interpolation is enabled, an additional linear interpolation to the date/time of the observation is performed. In the case of surface type and surface height, if there are values in the observation data file then they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

For data on a regular lat-lon grid, bilinear interpolation is used for the horizontal interpolation. For data on an unstructured grid, the data from the nearest grid point are used: a more sophisticated interpolation may be implemented for DWD ICON data in the future.

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.

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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:

$$b = (1 - w_x)(1 - w_y)a_1 + w_x(1 - w_y)a_2 + (1 - w_x)w_y a_3 + w_x w_y a_4$$

where the weights w_x and w_y are:

$$w_x = (x_p - x)/dx$$

$$w_y = (y_p - y)/dy$$

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 ($t_1 \leq t_p \leq t_2$) is given by:


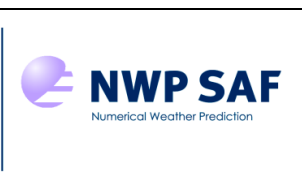
$$c = (1 - w_t)b_1 + w_t b_2$$

Where the weight w_t is: $w_t = (t_p - t_1)/(t_2 - t_1)$

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 $3\mu\text{m}$) 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

		<h1 style="text-align: center;">Radiance Simulator v3.2 User Guide</h1>	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

is calculated over the grid points within the footprint for channels at wavelengths below 5 μ m (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.


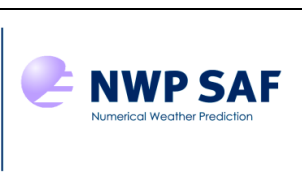
The footprint simulation can be used with temporal interpolation: the model profiles at grid points within the footprint are interpolated in time and then averaged over the footprint.

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, missing emissivity/BRDF values from the atlases, and MFASIS quality flags (see section 5.13). The *qcrttov* errorstatus has the value of the last profile simulated for any footprint.

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.

RadSim v3.1 introduced a new feature whereby the footprint data can be written out and then read in for subsequent runs which can result in faster run-times. This only applies 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

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

to the specified netCDF file (NB this file is overwritten if it already exists) in the specified *output_dir*. On subsequent runs where the footprints are the same, the footprint data can be read by specifying the full path to 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 currently used for interpolation. For all other data, the bilinear interpolation code requires that model data points 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:

$$\theta_z = \pm(\lambda_p + \pi)$$

$$\theta_y = \pm(\pi/2 - \varphi_p)$$

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

The z-axis rotation is then:

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

The y-axis rotation:

$$\varphi_y = \sin^{-1}(\sin \varphi \cos \theta_y - \cos \lambda \cos \varphi \sin \theta_y)$$

$$\lambda_y = d \cos^{-1}(\cos \lambda \cos \varphi \cos \theta_y + \sin \varphi \sin \theta_y)$$

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 usual convention for 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 = reference pressure (=1000 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:

$\varepsilon = R_d/R_v$ (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.

5.3.3 Rain/snow flux conversion

RTTOV v13 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 and UM fields and 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:


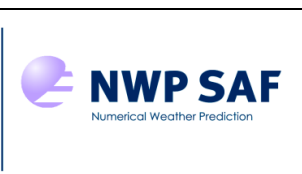
$$m = \left(\frac{af}{\rho}\right)^b \left(\frac{Rt}{M_{air}p}\right)$$

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:

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

5.4 Pressure levels

All level-dependent fields are defined on the same set of pressure levels for each profile. RadSim will use input pressure levels if they are available (this is mandatory for DWD ICON data), but often these must be calculated from other data. If the data source is a Met Office UM fieldsfile or an ECMWF GRIB file, sets of coefficients are provided within the file in each case for this purpose. The relevant documentation should be referred to for the

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

exact methods. For the NWP SAF diverse profile datasets which originate from the ECMWF model, the coefficients are contained within the code. These same coefficients are used for ECMWF netCDF files as the files do not contain the coefficients: for other vertical grids, the netCDF file must contain the pressure levels explicitly. The coefficients for computing HARMONIE pressure levels are available in the code for 65 level fields. The coefficients for computing JMA pressure levels are available in the code for 60 and 100 level fields. For model fields on a different number of levels, the coefficients must be included in the GRIB file.

If running the MW scattering code, RTTOV requires, in addition, a set of *half-level* pressures and cloud profile fields are defined on layers bounded by these half-levels. In the case of ECMWF and HARMONIE pressure levels, half-levels are computed as a by-product of calculating the full levels and retained for use with RTTOV-SCATT. Met Office UM *theta* levels are obtained by interpolation from intermediate *rho* levels. The *rho* levels are used as the half-levels in RTTOV-SCATT in this case. For DWD ICON data, the half-level pressures must be included explicitly in the model data file as for the pressure full levels. Currently, reading pressure half-levels is not supported from netCDF files, and these files do not contain the coefficients for calculating pressure levels, so for RTTOV-SCATT simulations using fields in netCDF format, the fields must be on 60, 91 or 137 levels, for which the coefficients are stored in the code.


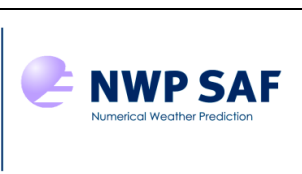
5.5 Surface type

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).

		<h2>Radiance Simulator v3.2 User Guide</h2>	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

5.6 Emissivity and BRDF values

For sea points, 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, and values are provided inside RTTOV for IR emissivity and for solar BRDF.


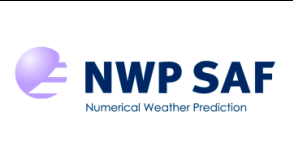
Alternatively, there are emissivity and BRDF atlases available (see the RTTOV Users Guide for details) for both MW and visible/IR over land and sea-ice. They are not used by default, but are highly recommended, particularly for microwave emissivities and for BRDF values, as the fixed values are unlikely to be very accurate given there is significant variation at those frequencies.

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 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 old 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

		<h1 style="text-align: center;">Radiance Simulator v3.2 User Guide</h1>	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

remains set to zero. Where the computed zenith angle would exceed 90° (i.e. the location is below the horizon), RadSim sets the 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 *addsolar* configuration namelist variable to true. A solar-enabled 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 `rttov13pred*L/` or `rttov9pred*L/` subdirectory of the `rtcoef_rttov13/` 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.


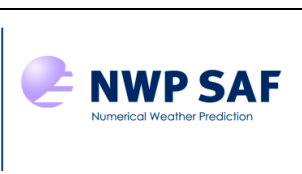
If you supply an obs data file for solar-enabled simulations, then you can again supply fixed angles to use for all simulations using the *sun_zen_angle* and *sun_azi_angle* configuration namelist variables. If you do not specify these, then you must either include both the solar zenith and azimuth angle fields, or the full date/time fields in the obs data file. In the former case, your 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 Visible/IR cloud simulations

For visible/IR cloud simulations, RTTOV requires cloud liquid and ice water concentrations and cloud fractions to be provided on layers bounded by the full pressure levels. NWP models typically provide these fields on the full pressure levels: to make use of them RadSim ignores the top level and assigns data from levels *2:nlevels* to layers *1:nlayers* (where *nlevels* is the number of full pressure levels and *nlayers* = *nlevels*-1). This effectively shifts the cloud profiles upwards by half a layer. This does not affect MW cloud simulations in RTTOV-SCATT as this model ingests *nlevels* of cloud data on the layers bounded by the *nlevels*+1 half-pressure levels.

Cloud and hydrometeor concentrations in the NWP models currently supported by RadSim represent grid-box average values. Within RTTOV and RTTOV-SCATT, the radiative transfer calculations require the concentrations of cloud in the cloudy fraction of each layer. This is obtained by dividing the layer-average cloud concentration by the layer cloud

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

fraction. RTTOV-SCATT calculates an effective cloud fraction for each profile internally which is applied to all cloud layers, so this cloud fraction scaling is carried out within RTTOV-SCATT. For all visible/IR cloud simulations, RTTOV is now also configured to perform the cloud fraction scaling.


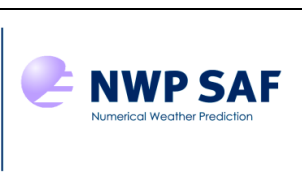
For visible/IR cloud simulations the optical properties for liquid and ice cloud are selected via the *ircloud_clw_scheme* and *ircloud_ice_scheme* configuration namelist variables. All options are available with all model data inputs. When the OPAC cloud liquid water properties are selected, RadSim currently uses only the maritime and continental stratiform particle types, selected according to the surface type of the profile (see section 5.5): 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 *ircloud_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 *ircloud_use_model_clw_deff* and/or *ircloud_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).

Solar scattering simulations are enabled by setting both *ir_addclouds* and *addsolar* 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 refraction is not taken into account, for example). The solver for solar radiation can be specified in the *vis_scatt_model* configuration namelist variable. MFASIS LUT can be selected by setting this to 3, and MFASIS-NN is selected by setting it to 4. A shortcut is provided for MFASIS-NN: setting the *run_mfasis* variable to true automatically sets *ir_addclouds* and *addsolar* to true and selects MFASIS-NN as the solver. MFASIS LUT files are trained using specific cloud liquid water and cloud ice water optical properties, and consistent properties must therefore be specified in the *ircloud_clw_scheme* and *ircloud_ice_scheme* configuration namelist variables. For MFASIS-NN this applies only to the ice properties.

MFASIS LUT simulations require an additional MFASIS LUT file to be provided and MFASIS-NN simulations require an MFASIS-NN coef file. Note that MFASIS LUT and MFASIS-NN currently support only a subset of visible/near-IR channels and any visible/near-IR channels not supported by the selected solver (and with no significant thermal radiation) will have zero radiances.

5.10 Visible/IR CAMS 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 simulations are enabled by setting *ir_addaerosols_cams* to true. Any or all of the supported CAMS species may be present in the GRIB file: all aerosol fields present

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

will be used in the simulations. Aerosol simulations are not currently supported for any other model input.

As for the visible/IR cloud simulations, RTTOV requires aerosol concentrations to be provided on layers bounded by the full pressure levels and as such the top-most level in the CAMS aerosol fields is omitted.

Aerosols can be enabled alongside cloud simulations if all relevant cloud fields are present in the CAMS GRIB file. In this case the MFASIS solvers *cannot* be used for solar radiation as they do not support aerosol simulations. As for clouds, if solar radiation is enabled then the default RTTOV solver is DOM which is computationally expensive.

5.11 Principal Components simulations

RTTOV supports two Principal Components-based models for simulating hyperspectral IR sounders: PC-RTTOV and HTFRTC. These are described in the RTTOV user guide. Note that at the time of writing HTFRTC is non-functional in RTTOV v13.2 due to a bug and is not planned to be fixed. These models provide a more efficient way of simulating many or all channels for these types of instruments. Currently, only non-solar clear-sky simulations are supported by these models in RadSim. 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 output PC scores. The IR emissivity atlases can be used with either model.

To enable PC-RTTOV set the *addpc* 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. The number of PC scores to simulate is specified in the *npcscores* configuration namelist variable. This can range from one up to a sensor-dependent maximum (typically 400 - but see the RTTOV user guide). PC-RTTOV provides a choice among different predictor sets: this is specified in the *ipcreg* variable. This is typically a value in the range 1-3 or 1-4 depending on the sensor (see the RTTOV user guide): the higher the number the more predictors are used which improves the accuracy but increases the run-time.

To enable HTFRTC set the *htfrtc* configuration namelist variable to true. For HTFRTC no RTTOV coefficient files are required. Instead a “static” HTFRTC coefficient file must be supplied along with a sensor-specific file. The number of PC scores to simulate (1-300) can be specified in the *npcscores* variable.

5.12 Running RTTOV

Clear-sky and visible/IR scattering simulations use the *rttov_direct* or *rttov_parallel_direct* subroutines (the latter if *nthreads* is greater than 1), but if Jacobian output has been requested then *rttov_k* or *rttov_parallel_k* is run instead and this will increase the run time. MW scattering simulations use *rttov_scatt* or *rttov_parallel_scatt*. There is currently no option to produce Jacobians from the MW scattering code as RadSim has not been set up to run the adjoint version of RTTOV-SCATT.

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. cloud 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	MFASIS satellite/solar geometry flag
6	MFASIS optical depth/effective diameter flag
7	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 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, the error code is stored in the *qcrttov* variable which is also written to the output file. The RTTOV Users Guide should be consulted to determine what the error codes mean. In this case, brightness temperatures may have been calculated but it will depend on the nature of the error.

Flag 2: If there is a bad emissivity value (outside the range 0-1) where an emissivity atlas is being used then the simulation for that profile is carried out with the fixed emissivity values used when no atlas is used.

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 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 (cf 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°).

Flags 5 and 6: These apply only to MFASIS simulations (LUT or NN). RTTOV radiances have an associated “quality” bit mask comprising a set of flags which provide information about the simulation (see the RTTOV user guide). MFASIS files are trained for a specific range of satellite/solar geometries and for a specific range of liquid and ice cloud properties. Flag 5 is set if any of the RTTOV MFASIS angle-related quality flags are set indicating that the satellite/solar geometry was outside the range used in training MFASIS. Flag 6 is set if the RTTOV MFASIS flag is set indicating that either a liquid or ice cloud optical depth or particle effective diameter was outside the range used in training MFASIS for at least one channel simulated for the given profile.

Flag 7: 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:


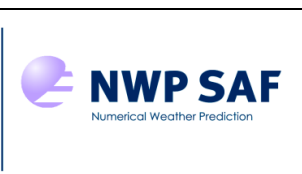
Flag/Bit number	Description
0	Coastline point
1	Empty footprint

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.

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 rough guide, some applications require more, some 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.

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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:

$$4 \times 71 \times 1152 \times 1536 \times 8 \text{ bytes} = 3.7\text{GB}$$

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 half-level pressures and 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:

$4 \times 71 \times (1152 \times 1536 + 500000) \times 8 \text{ bytes} = \text{approx } 4.8\text{GB}$

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.

Table: Batching examples

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

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 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


$$15 \times 100000 \times 4 \text{ bytes} = 5.7\text{MB}$$

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.,

$$8461 \times 100000 \times 4 \text{ bytes} = 3.2\text{GB}$$

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.

<p>The EUMETSAT Network of Satellite Application Facilities</p>	 <p>NWP SAF Numerical Weather Prediction</p>	<p>Radiance Simulator v3.2 User Guide</p>	<p>Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023</p>
---	--	---	--

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. Thus, if the array size is set at 500MB, 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 as long as the fundamental limits of the output data file are not violated (see section 4.5.3).

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 <https://nwp-saf.eumetsat.int/site/help-desk/>.

7. 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, [https://doi.org/10.1175/1520-0469\(1994\)051<1823:TMAPOE>2.0.CO;2](https://doi.org/10.1175/1520-0469(1994)051<1823:TMAPOE>2.0.CO;2)

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/rm/docs_rtov13/rttov13_svr.pdf (last accessed 27 Nov 2020).

8. 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
                   [--rttov_coeffs_type .dat|.H5]
                   [--rttov_coeffs_options STRING]
                   [--rttov_coeffs_dir DIR] [--rttov_scaer_dir DIR]
                   [--rttov_scclld_dir DIR] [--rttov_mfasis_cld_dir DIR]
                   [--rttov_mfasis_nn_dir DIR]
                   [--rttov_hydrotablē_dir DIR] [--rttov_pccoeffs_dir DIR]
                   [--rttov_htfrtc_coeffs_dir DIR]
                   [--rttov_coeff_file FILE] [--rttov_scaer_file FILE]
                   [--rttov_scclld_file FILE]
                   [--rttov_mfasis_cld_file FILE]
                   [--rttov_mfasis_nn_file FILE]
                   [--rttov_hydrotable_file FILE]
                   [--rttov_pol_coeff_file FILE]
                   [--rttov_pccoeff_file FILE]
                   [--rttov_htfrtc_static_file FILE]
                   [--rttov_htfrtc_sensor_file FILE] [--obs_datafile FILE]
                   [--read_footprint_file FILE]
                   [--write_footprint_file FILE] [--emiss_atlas_dir DIR]
                   [--brdf_atlas_dir DIR] --platform STRING
                   --satid INTEGER --inst STRING
                   [--channels INTEGER [INTEGER ...]]
                   [--temporal_data T|F] [--enable_footprints T|F]
                   [--use_emiss_atlas T|F] [--emis_atlas_id 1|2|3]
                   [--use_brdf_atlas T|F] [--use_all_atlas_months T|F]
                   [--run_scatt T|F] [--run_mfasis T|F]
                   [--ircloud_use_model_clw_deff T|F]
                   [--ircloud_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]
                   [--seaice_threshold FLOAT] [--coast_threshold FLOAT]
                   [--output_mode 1|2|3] [--max_array_size INTEGER]
                   [--max_profs INTEGER] [--nthreads INTEGER]
                   [--nprofs_per_call INTEGER] [--addpc T|F]
                   [--htfrtc T|F] [--npcscores INTEGER] [--ipcreg INTEGER]
```

```

[--ipcbnd INTEGER] [--addsolar T|F]
[--rayleigh_single_scatt T|F] [--rayleigh_depol T|F]
[--rayleigh_max_wavelength FLOAT]
[--rayleigh_min_pressure FLOAT] [--dom_rayleigh T|F]
[--ir_addaerosols_cams T|F] [--ir_addclouds T|F]
[--ir_scatt_model 1|2] [--vis_scatt_model 1|2|3|4]
[--dom_nstreams INTEGER] [--dom_accuracy FLOAT]
[--dom_opdep_threshold FLOAT]
[--ircloud_clw_scheme 1|2] [--ircloud_ice_scheme 1|2|3]
[--ircloud_icede_param 1|2|3|4]
[--cldcol_threshold FLOAT]
[--fastem_version 0|1|2|3|4|5|6|7]
[--ir_sea_emis_model 1|2] [--solar_sea_brdf_model 1|2]
[--ozone_data T|F] [--clw_data T|F]
[--cc_threshold FLOAT] [--pol_mode 0|1|2]
[--ice_polarisation FLOAT] [--apply_reg_limits T|F]
[--opdep13_gas_clip T|F] [--interp_mode 1|2|3|4|5]
[--addrefrac T|F] [--plane_parallel T|F]
[--do_lambertian T|F] [--lambertian_fixed_angle T|F]
[--do_nlte_correction T|F] [--mw_clw_scheme 1|2|3]
[--mw_clw_cloud_top FLOAT] [--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_emissjac T|F] [--write_radiances T|F]
[--write_emiss 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	<i>character</i>	Name of output configuration namelist file.
--write_config_only	<i>T F, default F, optional</i>	If T write out namelist file and stop. If F write out file and run RadSim.
--radsim_bin_dir	<i>character, default "bin/", optional</i>	Location of RadSim <i>bin/</i> directory. It is not necessary to specify this if calling <i>radsim_run.py</i> 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 (<https://nwp-saf.eumetsat.int/site/software/radiance-simulator/>), 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. 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:


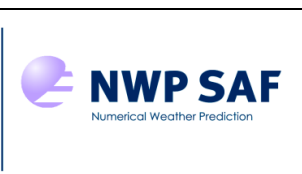
```
$ python radsim_geo_obs.py -h
usage: radsim_geo_obs.py [-h] --output_file FILE --data_file FILE
                        [--sat_lat FLOAT] --sat_lon FLOAT
                        [--sat_height FLOAT]
                        [--nadir_footprint_radius FLOAT] [--scan_time FLOAT]
                        [--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	<i>character</i>	Name of output obs data file.
--data_file	<i>character</i>	Name of input netCDF file containing pixel lat/lon and optionally other data for a specific geostationary sensor.
--sat_lat	<i>float, default 0., optional</i>	Latitude of sub-satellite point in the interval [-90, 90] (degrees).
--sat_lon	<i>float</i>	Longitude of sub-satellite point. The value must lie within the range of longitudes specified in the input netCDF file (degrees).

<code>--sat_height</code>	<i>float, default 35800 km, optional</i>	Satellite height (km).
<code>--nadir_footprint_radius</code>	<i>float, optional</i>	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.
<code>--scan_time</code>	<i>float, optional</i>	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.
<code>--date_time</code>	<i>five integers: year, month, day, hour, minute, optional</i>	Date/time at which the scan begins. Only required if you want to enable temporal interpolation and/or solar radiation in RadSim.
<code>--fixed_date_time</code>	<i>T/F, default F, optional</i>	If T, then all pixels are assigned the date/time specified in <i>--date_time</i> .
<code>--max_sat_zen</code>	<i>float, default 85 degrees, optional</i>	Maximum zenith angle (degrees) to include in output obs data file. It is not recommended to increase beyond the default.
<code>--min_lat</code>	<i>float, optional</i>	Minimum latitude (degrees) to include in output obs data file, valid range [-90,90].
<code>--max_lat</code>	<i>float, optional</i>	Maximum latitude (degrees) to include in output obs data file, valid range [-90,90].
<code>--min_lon</code>	<i>float, optional</i>	Minimum longitude (degrees) to include output obs data file, must lie within range of longitudes in netCDF file.
<code>--max_lon</code>	<i>float, optional</i>	Maximum longitude (degrees) to include output obs data file, must lie within range of longitudes in netCDF file.
<code>--write_lsm</code>	<i>T/F, default F, optional</i>	If true, includes the land/sea mask (land fraction) in the output obs data file. Requires corresponding pixel data in the netCDF file.
<code>--write_orog</code>	<i>T/F, default F, optional</i>	If true, includes the orography in the output obs data file. Requires corresponding pixel data in the netCDF file.
<code>--write_sat_angles</code>	<i>T/F, default T, optional</i>	If true, write satellite zenith and azimuth angles to output file. Recommended true (the default).
<code>--write_scan_data</code>	<i>T/F, default F, optional</i>	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.

		<h2>Radiance Simulator v3.2 User Guide</h2>	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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)
- “**lon**”, 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”
- “**lsm**”, 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:

1. 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.

- 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 “*lsm*” (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_lsm 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 .

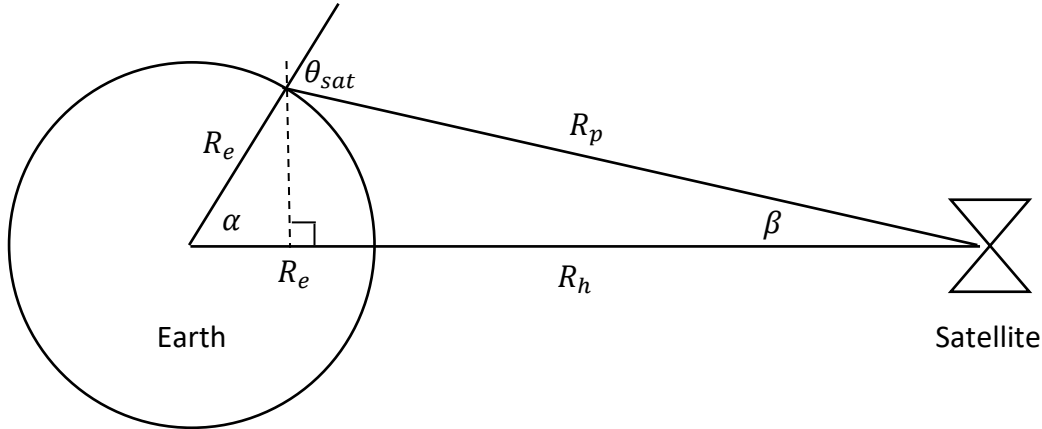


Figure B1: schematic diagram of a satellite illustrating zenith angle calculation.

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.

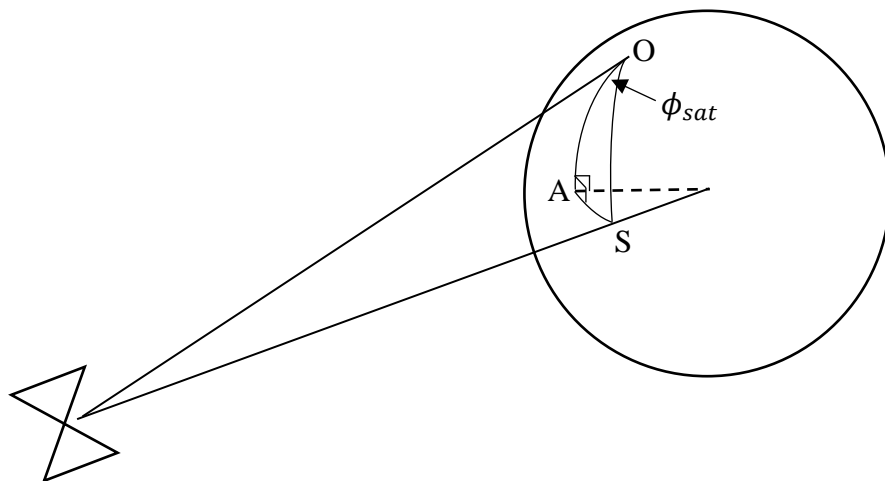


Figure B2: schematic diagram of a geostationary satellite illustrating azimuth angle calculation.

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`, `emiss`, 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:

```
$ src/scripts/radsim_plot_example.py -h
usage: radsim_plot_example.py [-h] --filename FILE [--varname STRING]
                             [--channel INTEGER] [--scatter T|F]
                             [--size FLOAT] [--vmin FLOAT] [--vmax FLOAT]
                             [--cmap STRING] [--title STRING] [--diff FILE]
```

Argument	Value	Description
<code>-h</code>		Print out usage information.
<code>--filename</code>	<i>character</i>	Name of output NetCDF file.
<code>--var_name</code>	<i>character, default 'bt', optional</i>	Name of variable to plot: defaults to brightness temperatures ('bt').
<code>--channel</code>	<i>integer, optional</i>	Channel number to plot. This is mandatory for data stored per-channel (such as <code>bt</code> , <code>refl</code> , <code>radiance</code> , <code>emiss</code> or <code>brdf</code>). The channels are numbered from 1 to <code>n</code> , where <code>n</code> 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 the <code>--channel</code> argument to specify the level to plot. For single-level or surface profile datasets (e.g. sea-ice) you must omit this argument.
<code>--scatter</code>	<i>T/F, default F, optional</i>	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.
<code>--size</code>	<i>float, default 1.5, optional</i>	Size of blobs in scatter plots.

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

<p>The EUMETSAT Network of Satellite Application Facilities</p>		<p>Radiance Simulator v3.2 User Guide</p>	<p>Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023</p>
---	---	---	--

ANNEX D – Example MARS, CDS and ADS requests

There is good documentation for MARS on the web:

<https://software.ecmwf.int/wiki/display/UDOC/MARS+user+documentation>

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.
- 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=ml,
  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_fcst_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' : '1',
    'stream' : 'oper',
    'type' : 'fc',
    'param' : '130/133/203',
    'levtype' : 'ml',
    'levelist': '1/to/137',
    'date' : '2017-07-01',
    'time' : '06',
    'grid' : '1.0/1.0',
    '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')
```


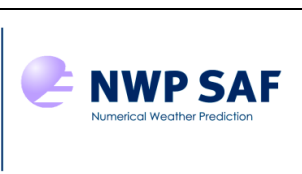
Example 4: ADS request for CAMS data including supported aerosol fields and cloud fields for VIS/IR cloud simulations.

```
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',
            '31', '32', '33',
            '34', '35', '36',
            '37', '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')
```

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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)

```

if (month ≤ 2) then
  year := year - 1
  month := month + 12
end if
a = year / 100
b = 2 - a + a/4
jd = int(365.25 × (year + 4716)) + int(30.6001 × (month + 1)) + day + b - 1524.5
    + (hour × 3600 + minute × 60 + second) / 86400

```

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


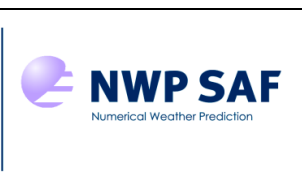
$$nseconds = 21.448 - jc \times (46.8150 + jc \times (0.00059 - jc \times 0.001813))$$

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

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

$$\omega = (125.04 - 1934.136 \times jc) \times deg2rad$$

$$\varepsilon_c = (\varepsilon_m + 0.00256 \times \cos \omega) \times deg2rad$$

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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°)

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

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

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

Calculate the Equation of Time (EoT) in minutes

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

$$c_1 = \sin(2 \times \overline{\Lambda}_{sol} \times \text{deg2rad})$$

$$c_2 = \cos(2 \times \overline{\Lambda}_{sol} \times \text{deg2rad})$$

$$EoT = \left[\frac{(z \times c_1) - (2 \times e \times \sin M) + (4 \times e \times z \times c_2 \times \sin M) - (z^2 \times c_1 \times c_2) - (\frac{5}{4} \times e^2 \times \sin 2M)}{1} \right] \times 4 \times \text{rad2deg}$$

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

$$\Lambda_{sol}^t = \overline{\Lambda}_{sol} + EoC$$

$$\Lambda_{sol}^a = (\Lambda_{sol}^t - 0.00569 - 0.00478 \times \sin \omega) \times \text{deg2rad}$$

Calculate solar declination (δ)

$$\sin \delta = \sin(\varepsilon_c) \times \sin(\Lambda_{sol}^a)$$

<p>The EUMETSAT Network of Satellite Application Facilities</p>		<p>Radiance Simulator v3.2 User Guide</p>	<p>Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023</p>
---	---	---	--

The solar zenith angle is calculated as follows:

Calculate time offset (t_{offset}) 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


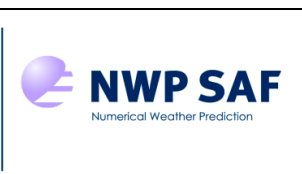
$$h = \left(\frac{t_{sol}}{4} - 180 \right) \times deg2rad$$

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})

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

		Radiance Simulator v3.2 User Guide	Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023
--	--	---	---

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 `ircloud_icede_param` option. In most cases, the RTTOV parameterisations are 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 `ircloud_use_model_clw_deff` and/or `ircloud_use_model_ice_deff` options are true. For cloud liquid water, in cases where the specified model D_{eff} is less than $5 \mu\text{m}$ (and non-zero) it is clipped to $5 \mu\text{m}$. For cloud ice, in cases where the specified model D_{eff} is less than $10 \mu\text{m}$ (and non-zero), the cloud ice concentration is scaled by a factor $(10/D_{\text{eff}})$ to ensure an approximately constant extinction coefficient, and then the diameter is clipped to $10 \mu\text{m}$. 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 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:

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

where D_{eff} is the particle effective diameter (μm), LWC is the cloud liquid water concentration (kg/m^3), 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 $1000 \text{ kg}/\text{m}^3$). 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_{eff} smaller than $10 \mu\text{m}$ are reset to $10 \mu\text{m}$.

The NWP model cloud concentration fields are in units of kg/kg and these are converted to LWC (kg/m^3) by multiplying by the density field from the model data. The above calculation of D_{eff} is consistent with that in RTTOV v13, except that the RTTOV conversion of the cloud units from kg/kg to kg/m^3 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.

<p>The EUMETSAT Network of Satellite Application Facilities</p>		<p>Radiance Simulator v3.2 User Guide</p>	<p>Doc ID : NWPSAF-MO-UD-051 Version : 1.2 Date : 26/07/2023</p>
---	---	---	--

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 $R_e=6371$ km. 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:

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

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):

$$\widehat{OCM} = \alpha \quad \widehat{OCN} = \frac{\pi}{2} - \lambda_o$$

$$\widehat{MCN} = \frac{\pi}{2} - \lambda_m \quad \widehat{NOM} = \phi_m$$

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

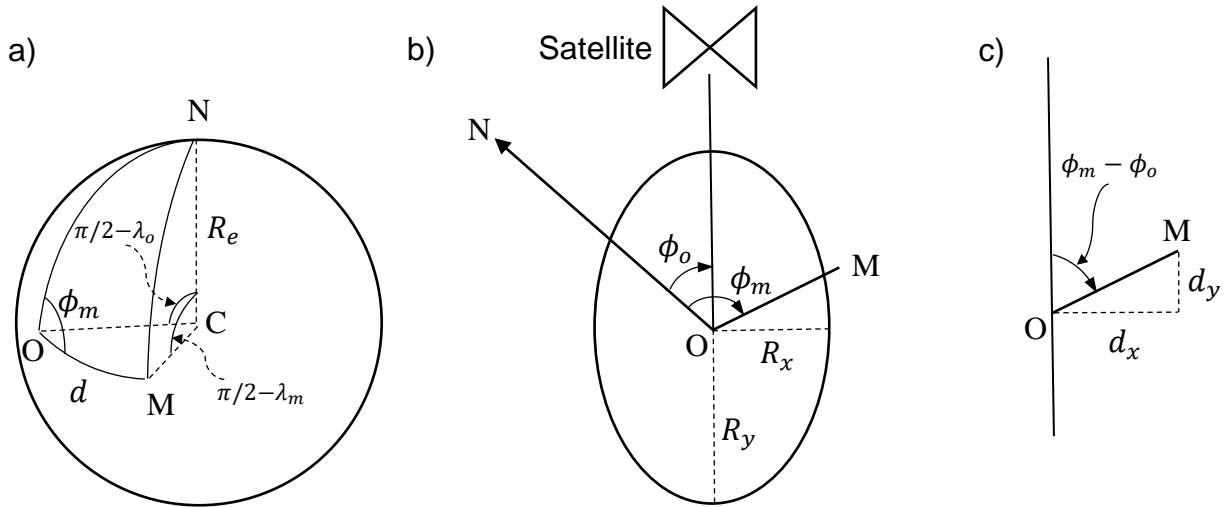


Figure H1: schematic of a satellite observation at location O and a model grid point M .
a) On the spherical Earth with the north pole (N) and Earth centre (C) marked.
b) The observation footprint modelled as an ellipse on the Earth's surface.
c) A detail of b) indicating the distances d_x and d_y that 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:

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

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

$$\begin{aligned} d_x &= R_e \alpha \sin(\phi_m - \phi_o) \\ d_y &= R_e \alpha \cos(\phi_m - \phi_o) \end{aligned}$$

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

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

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).