

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

NWP SAF

RTTOV Version 13 Top Level Design

Version 1.1

19th October 2020

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

RTTOV Version 13 Top Level Design

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 2020, EUMETSAT, All Rights Reserved.

Change record				
Version	Date	Author	Approved	Remarks
1.0	07/09/2020	JH		Initial version
1.1	19/10/2020	JH		Updates after DRR

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

Table of Contents

1.	INTRODUCTION.....	4
1.1	Reference documents	4
1.2	Design overview.....	4
2.	SOFTWARE FUNCTIONS AND MODULE/SUBROUTINE DESIGN.....	9
2.1	User-level functionality.....	9
2.2	RTTOV direct model design	9
2.3	RTTOV-SCATT direct model design	10
2.4	Tangent linear (TL), adjoint (AD) and Jacobian (K) models	11
2.5	Calling trees.....	11
3.	IMPLEMENTATION ASPECTS	20
3.1	External libraries	20
3.2	Compiler requirements.....	21
3.3	Required and optional third-party software	21
3.4	Build method	21
3.5	Run-time considerations.....	21

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

1. INTRODUCTION

This document defines the top level design for Version 13 of RTTOV, in accordance with the requirements of the NWP SAF.

1.1 Reference documents

[RD-1] NWPSAF-MO-DS-039 RTTOV v13 Product Specification

[RD-2] NWPSAF-MO-UD-046 RTTOV v13 User Guide

[RD-3] NWPSAF-MO-TV-044 RTTOV v13 Test Plan

1.2 Design overview

Managing computer resources

The major requirement for RTTOV v13 is to allow the user to be able to run the code without limits on the number of channels, profiles and instruments in the same run. This is achieved by using arrays of derived types and dynamic memory allocation. Derived types reduce the number of arguments to be passed to the subroutines and facilitates the memory allocation for the arrays. The remaining limits are given by:

- The external RTTOV coefficient file which contains the number of fixed pressure levels on which the optical depth computation is performed.
- The size of the computer physical memory and sufficient disk space to store the files.

The user can allocate all the derived types which are arguments of the subroutines (coefficients, profiles, radiances, channels, PCscores, etc). In this way the user can run inside the same main program calls for different instruments for which the user input and reference profiles can be on different pressure levels.

Configuring simulations

All aspects of RTTOV simulations are configured at run-time by setting variables in the *rttov_options* derived type. Many options are included, for example, whether solar radiation is enabled, and whether clouds and/or aerosols are included in visible/IR simulations. RTTOV-SCATT allows a subset of RTTOV options to be user-specified, and RTTOV-SCATT calls are configured via the *rttov_options_scatt*. Users should set the broadest range of options required for their simulations before reading coefficients or allocating data structures, to ensure that all relevant data are read in and all necessary arrays are allocated. This is because only the necessary data are read in and allocated to keep memory usage as small as possible for a given type of simulation. Options can subsequently be turned off when running RTTOV if desired.

Managing coefficient files

The user manages the RTTOV coefficients within their program and is responsible for allocating the maximum number of instruments for the run. The user calls the RTTOV coefficient file ingest subroutine *rttov_read_coefs* with the IDs of the satellite and instrument, or with a filename, and optionally with a list of selected channels to read (otherwise all channels are read). *rttov_read_coefs* reads the data and allocates memory for the storage. The *rttov_dealloc_coefs* subroutine deallocates

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

memory for the coefficients. For visible/IR cloud/aerosol simulations (including the MFASIS fast visible cloud parameterisation) and/or PC-RTTOV simulations, additional optical property, look up table or coefficient files are required, and these are ingested in the same call to *rttov_read_coefs* as the RTTOV gas optical depth coefficient file. The *rttov_conv_coef* executable enables the user to convert all these file types between ASCII, Fortran unformatted (“binary”) and HDF5 formats and allows the extraction of a subset of channels. This enables greater efficiency when using the large coefficient files required by hyperspectral sounders.

RTTOV v13 implements a new gas optical depth parameterisation and this requires changes in the format of the gas optical depth coefficient files. All RTTOV v12-compatible optical depth coefficient files are supported, and executables remain available to convert older v10/v11 optical depth coefficient files to v12 format and vice versa in either ASCII or HDF5 format.

RTTOV-SCATT hydrotables are distributed in ASCII format and the *rttov_ascii2bin_scattcoef.exe* executable can be used to convert these to Fortran unformatted format. HDF5 is not yet supported for the hydrotables. The hydrotables are read in using the *rttov_read_scattcoeffs* subroutine and the data structure is deallocated using the *rttov_dealloc_scattcoeffs* subroutine.

HTFRTC input files are available to download in NetCDF and ASCII format. There is no facility to convert between these formats. HTFRTC coefficients are read in using the *rttov_read_coefs_htfrtc*.

The *rttov_user_options_checkinput* subroutine is provided to check compatibility between the selected options and coefficients that have been read in. This can be useful for debugging simulations as it warns not only of strictly inconsistent option settings, but also questionable settings which will not cause errors, but might indicate that the user is not simulating what they expect.

Management of atmospheric profiles

The user provides RTTOV with atmospheric profiles and surface variables via the *rttov_profile* derived type. Pressure levels are not fixed and need to be provided by the user along with the profile: the only limitation is that all profiles passed to RTTOV in a single call have the same number of pressure levels, though the pressures themselves may vary between profiles. The *rttov_user_profile_checkinput* subroutine allows users to check the validity of an input profile before calling RTTOV. Alternatively, RTTOV can check the profiles internally. This checking can include comparisons of the profile to the limits derived from the minimum/maximum value envelopes of the profile set with which the RTTOV coefficients are trained. The min/max profile envelopes are contained in the coefficient files.

In addition to the *rttov_profile* derived type, the *rttov_profile_cloud* derived type is used to provide additional cloud and hydrometeor inputs for RTTOV-SCATT (the MW scattering model), and the *rttov_opt_param* derived type can be used to provide explicit profiles of optical properties for visible/IR scattering simulations.

The user provides a list of profile and channel indices for each calculated radiance.

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

Management of instrument channels

The user provides the model with a list of channel numbers for each profile for which radiances are to be simulated. The definition of the channel numbers is provided in the coefficient file for each instrument. An error is thrown if an invalid channel number is provided.

Surface emissivities and reflectances

RTTOV can compute IR (over ocean) and MW (over ocean and land/sea-ice) emissivities according to some surface input parameters. The user provides a logical array indicating for which channels and profiles the model should calculate emissivities. The ocean surface emissivity models available in RTTOV v13 are ISEM or IREMIS for infrared radiances and FASTEM versions 1-6 or TESSEM2 for the microwave radiances. For infrared radiances RTTOV provides a fixed value of emissivity over land and sea-ice.

Similarly, RTTOV can compute sea surface reflectances (BRDFs) due to sun glint for solar-affected simulations. This involves a dynamic BRDF for the direct solar beam which accounts for sun-glint, and a separate Lambertian diffuse BRDF for downward scattered radiation. For visible/near-infrared channels the diffuse BRDF is interpolated from fixed reflectance spectra while for shortwave infrared channels it is computed from the surface emissivity. The user provides a logical array indicating for which channels and profiles the model should calculate reflectances. For land and sea-ice RTTOV provides crude fixed values for surface reflectance for channels below $3\mu\text{m}$, while for channels above $3\mu\text{m}$ a Lambertian BRDF is calculated from the surface emissivity. In both cases the same value is used for the direct and diffuse BRDFs.

The user may optionally provide input emissivities and direct BRDFs for any channel, and the user may also optionally provide the diffuse BRDF for channels below $3\mu\text{m}$. For land surfaces, emissivity and BRDF atlases are available which can provide emissivity and reflectance values to be passed into RTTOV. The available atlases are the University of Wisconsin, CAMEL 2007, and CAMEL climatology emissivity atlases in the infrared, the TELSEM2 and CNRM emissivity atlases in the microwave, and the CMS BRDF atlas for visible and near-IR channels.

In all cases the emissivities and reflectances (both direct BRDF and diffuse BRDF) computed/used by RTTOV are returned in the output arrays from RTTOV. References for the emissivity models and atlases are given in the user guide [RD-2].

Input and output

The table below lists the arguments accepted by the **rttov_direct** subroutine. Here **nchanprof** is the size of the **chanprof(:)** array (i.e. the total number of radiances to compute), **nprof** is the size of the **profiles(:)** array (i.e. the number of profiles to process in each call to RTTOV), and **nchannelsrec** is the number of reconstructed radiances to compute (for PC-RTTOV or HTFRTC). The **rttov_parallel_direct** subroutine has the same arguments as **rttov_direct** plus an optional argument **nthreads** to specify the number of threads.

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

Type	In/Out	Variable	Description
Integer	Intent(out)	errorstatus	Return code.
Type(rttov_chanprof)	Intent(in)	chanprof(nchanprof)	Chanprof structure.
Type(rttov_options)	Intent(in)	opts	RTTOV options structure
Type(rttov_profile)	Intent(in)	profiles(nprof)	Profiles structure.
Type(rttov_coefs)	Intent(in)	coefs	RTTOV coefficient structure.
Type(rttov_transmission)	Intent(inout)	transmission	Output transmittances (0-1).
Type(rttov_radiance)	Intent(inout)	radiance	Output radiances (mW/cm ⁻¹ /sr/m ² , degK, and BRDF/unitless).
Type(rttov_radiance2)	Intent(inout), optional	radiance2	Secondary output radiances (mW/cm ⁻¹ /sr/m ²).
Logical	Intent(in), optional	calcemis(nchanprof)	.true. if RTTOV should calculate/select surface emissivity internally or .false. if user is supplying an emissivity value (e.g. from an atlas).
Type(rttov_emissivity)	Intent(inout), optional	emissivity(nchanprof)	Input/output emissivities.
Logical	Intent(in), optional	calcrefl(nchanprof)	.true. if RTTOV should calculate/select surface BRDF internally or .false. if user is supplying a BRDF value (e.g. from the atlas).
Type(rttov_reflectance)	Intent(in), optional	reflectance(nchanprof)	Input/output reflectances.
Type(rttov_opt_param)	Intent(in), optional	aer_opt_param	Aerosol optical parameter input profiles if opts%rt_ir%addaerosl and opts%rt_ir%user_aer_opt_param are true.
Type(rttov_opt_param)	Intent(in), optional	cld_opt_param	Cloud optical parameter input profiles if opts%rt_ir%addclouds and opts%rt_ir%user_cld_opt_param are true.
Type(rttov_traj)	Intent(inout), optional	traj	Trajectory structure to hold temporary data: may improve performance on some architectures for certain types of simulation.
Type(rttov_pccomp)	Intent(inout), optional	pccomp	Structure to hold output PC scores and reconstructed radiances.
Integer	Intent(in), optional	channels_rec(nchannelsrec)	Channels for which to compute reconstructed radiances if opts%rt_ir%pc%addradrec is .true.

The table below lists the arguments accepted by the **rttov_scatt** subroutine which is the RTTOV-SCATT (MW scattering) direct model.

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

Type	In/Out	Variable	Description
Integer	Intent(out)	errorstatus	Return code.
Type(rtov_options_scatt)	Intent(in)	opts_scatt	Options to control aspects of RTTOV-SCATT.
Integer	Intent(in)	nlevels	Number of input profile levels.
Type(rtov_chanprof)	Intent(in)	chanprof(nchanprof)	Chanprof structure.
Integer	Intent(in)	frequencies(nchanprof)	Frequency indices.
Type(rtov_profile)	Intent(in)	profiles(nprof)	Profiles structure.
Type(rtov_profile_cloud)	Intent(in)	cld_profiles(nprof)	Cloud profile structure.
Type(rtov_coefs)	Intent(in)	coef_rtov	RTTOV coefficient structure.
Type(rtov_scatt_coef)	Intent(in)	coef_scatt	RTTOV_SCATT coefficient structure.
Logical	Intent(in)	calcemis(nchanprof)	.true. if RTTOV should calculate surface emissivity or .false. if user is supplying an emissivity value (e.g. from the atlas).
Type(rtov_emissivity)	Intent(inout)	emissivity(nchanprof)	Input/output emissivities.
Type(rtov_radiance)	Intent(inout)	radiance	Output radiances (mW/cm ² /sr/m ² & degK).
Real	Intent(out), optional	cfrac	Cloud fraction actually used (diagnostic).
Type(rtov_scatt_emis_retrieval_type)	Intent(inout), optional	emis_retrieval_terms	Structure containing radiance and transmittance data necessary for dynamic surface emissivity retrievals.
Type(rtov_reflectivity)	Intent(inout), optional	reflectivity	If present, activates radar simulations, and holds calculated reflectivities.

The interfaces for the TL, AD and K models are similar to those for the direct model calls shown above. They have some additional arguments for the TL and AD/K variables, but these use the same derived types as the direct model.

The only input files required are the gas optical depth coefficient files. For clear-sky simulations only one coefficient file per sensor is required but there are additional coefficient files containing the scattering optical properties for simulations with cloud, aerosol and hydrometeors. The MFASIS fast visible cloud model requires a look-up table for the sensor being simulated. Similarly, an additional coefficient file is required for PC-RTTOV simulations. The HTFRTC model requires its own specific input coefficient files, and in this case no RTTOV coefficient files are required.

Subroutines are provided for users to (de)allocate and initialise all the derived types listed in the tables above at runtime. Detailed descriptions of the derived types are available in the user guide [RD-2].

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

2. SOFTWARE FUNCTIONS AND MODULE/SUBROUTINE DESIGN

2.1 User-level functionality

RTTOV v13 provides the following primary user-level functions:

- Subroutine to read RTTOV coefficient files, cloud/aerosol optical property files, MFASIS look-up tables and PC-RTTOV coefficient files.
- Subroutine to check that user-specified options are consistent with the loaded coefficients.
- Subroutine to read HFTRTC coefficient files.
- Subroutine to read RTTOV-SCATT hydrotables.
- Subroutine to check validity of a user profile.
- Subroutines to (de)allocate and initialise RTTOV input/output structures.
- Subroutines to call the direct, tangent linear (TL), adjoint (AD) and Jacobian (K) models.
- Subroutines to call the direct, TL, AD and K models using multiple threads (OpenMP).
- Subroutines to call the RTTOV-SCATT (MW scattering) direct, TL and AD/K models.
- Subroutines to call the RTTOV-SCATT direct, TL and AD/K models using multiple threads.
- Subroutines to initialise and deallocate the emissivity and BRDF atlases.
- Subroutines to access initialised emissivity and BRDF atlases.

In addition, there are a number of ancillary subroutines which may be helpful for users. Section 6 of the user guide [RD-2] provides a full list of the user level subroutines and the interfaces to all user-level subroutines are given in the user guide Annexes.

2.2 RTTOV direct model design

The *rttov_direct* (direct model) subroutine performs the following tasks:

- If HTFRTC is selected, skips standard RTTOV code path and calls the interface subroutine for HTFRTC.
- Initialises primary outputs.
- Performs checks on inputs to catch inconsistent or invalid options and arguments.
- Allocates and initialises internal data structures.
- Converts input profiles to the units used internally by RTTOV.
- If the *do_checkinput* Boolean option is true, checks input profiles for unphysical/invalid values.
- Carries out ancillary calculations related to profile geometry.
- If interpolation is required, interpolates input profiles onto RTTOV coefficient levels.
- If the *do_checkinput* Boolean is true, checks profiles on coefficient levels against the optical depth regression limits, and if the *verbose* Boolean is true and the *apply_reg_limits* Boolean is false, outputs warnings when limits are exceeded.
- If the *apply_reg_limits* boolean is true, clips the (interpolated) profiles to the regression limits. This clipping is applied to any extrapolated profile data at the top of the profile when the RTTOV interpolator is used.
- If interpolation is required, performs ancillary calculations related to profile geometry on the interpolated profiles.
- Calculates gas absorption optical depth predictors on coefficient levels.
- Calculates gas absorption optical depths on coefficient levels.

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

- If interpolation is required, interpolates gas absorption optical depths onto input levels.
- If requested, calculates optical depth due to cloud liquid water absorption for microwave sensors.
- If requested, calculates Rayleigh extinction for visible/near-IR channels.
- If visible/IR cloud simulations requested, calculates cloud columns due to selected cloud overlap assumption.
- Allocates internal data structures which depend on the number of cloud columns.
- If PC-RTTOV aerosol simulations requested, applies regression limits for input aerosol profiles.
- If visible/IR cloud/aerosol simulations requested, calculate optical properties and cloud/aerosol optical depths required for the chosen solver(s).
- Calculates atmospheric transmittances.
- Calculates surface emissivities (where *calcemis(:)* is true).
- For solar simulations, calculates surface BRDFs (where *calcrefl(:)* is true).
- Integrate the radiative transfer equation*.
- If selected, calculate the non local thermodynamic equilibrium (NLTE) bias correction.
- For PC-RTTOV, calculate PC scores and if requested the reconstructed radiances and brightness temperatures.
- For all other simulation types, calculate the top of atmosphere brightness temperatures and reflectances for each channel as appropriate.
- Deallocate all internally allocated memory.

*The integration of the radiative transfer equation (RTE) is carried out by *rttov_integrate* for most simulation types. This implements the linear-in-tau approximation in a straightforward numerical integration of the RTE. For visible/IR scattering simulations where the Discrete Ordinates Method (DOM) solver has been selected by the user, separate subroutines (*rttov_dom_setup_profile* and *rttov_dom*) are called, first to prepare the input profiles and then to execute the DOM algorithm to solve the RTE. These two subroutines are called separately for the emission and solar source terms (as required) for reasons of efficiency. Even when DOM is used, *rttov_integrate* is called because the Rayleigh single-scattering implemented therein is applied to all solar simulations unless the user requests Rayleigh multiple scattering to be included in the DOM calculation. Finally, if the user selects MFASIS as the visible solver, the top of atmosphere reflectances are computed instead by the *rttov_mfasis* subroutine.

Figure 2 below gives the calling tree for *rttov_direct* which follows the above description.

2.3 RTTOV-SCATT direct model design

The *rttov_scatt* (MW scattering direct model) subroutine performs the following tasks:

- Calls *rttov_direct* to compute gas absorption optical depths.
- Calls *rttov_iniscatt* to perform some preparatory calculations for the delta-Eddington solver and/or calculations for the active simulator. These include calculations related to the input profiles and calculations of the scattering properties based on the input hydrotable file.
- For passive simulations, calls *rttov_eddington* which integrates the radiative transfer equation.
- For passive simulations, if requested, computes quantities required for dynamic surface emissivity retrievals.

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

- Calculates brightness temperatures from radiances.

These tasks are summarised in Figure 3 below.

2.4 Tangent linear (TL), adjoint (AD) and Jacobian (K) models

The TL model is based on the forward model. The functionality of each subroutine is the same as in the direct model, but they calculate the tangent linear as an additional output.

The AD and K first call the direct model to get the forward “trace” and then respectively the AD or K subroutines are called in the reverse sequence order.

The user guide [RD-2] provides details on what the TL, AD and K models compute and how they should be called. The same routine names as for the forward model are used but appended with “_tl”, “_ad”, and “_k”.

2.5 Calling trees

Figures 1-3 below show the calling trees for a user program calling the RTTOV v13 direct model, for the *rttov_direct* subroutine and for the *rttov_scatt* subroutine respectively. Only subroutines called from the top-level subroutine are shown.

Calling tree for user program calling RTTOV v13

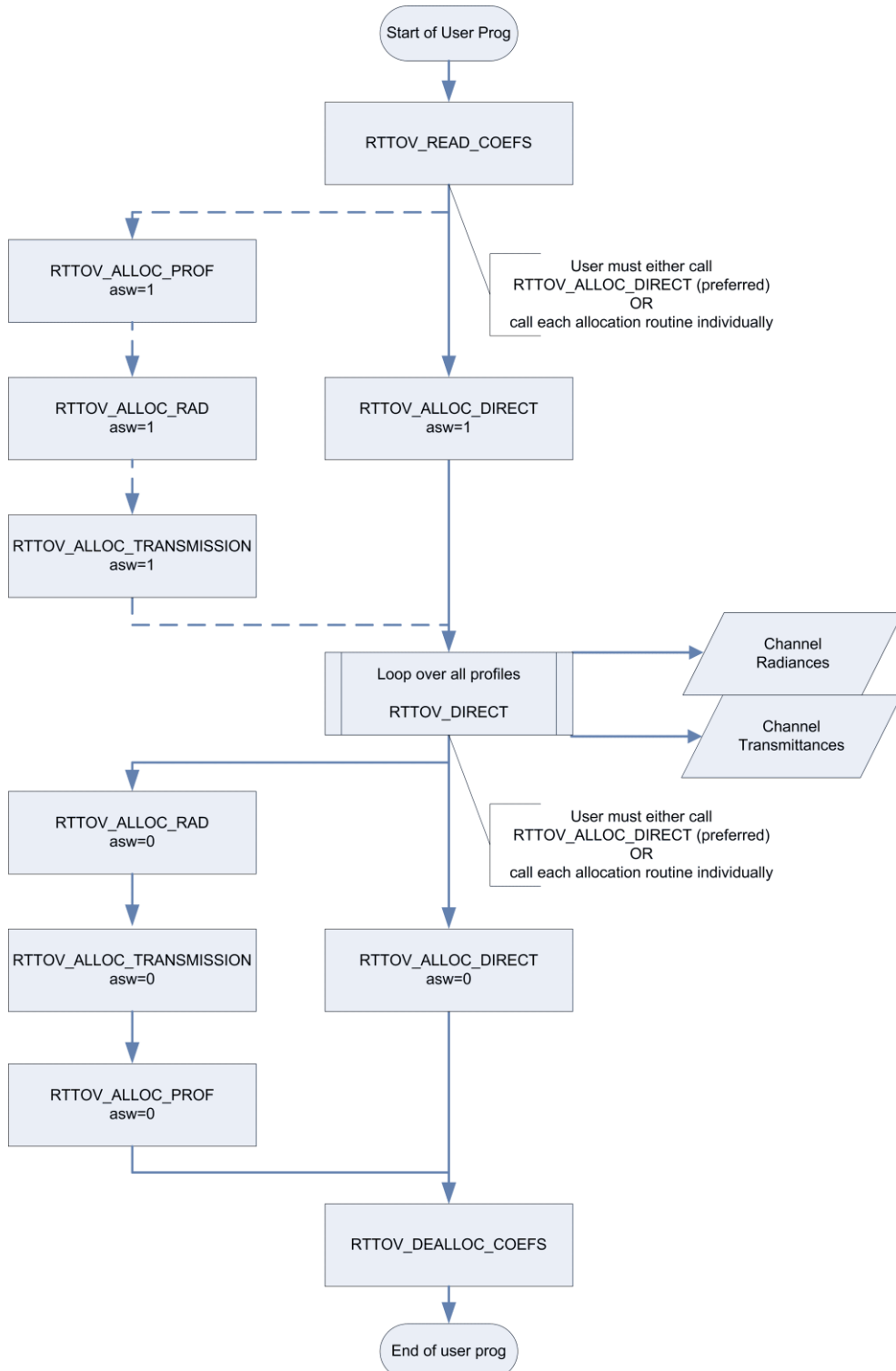
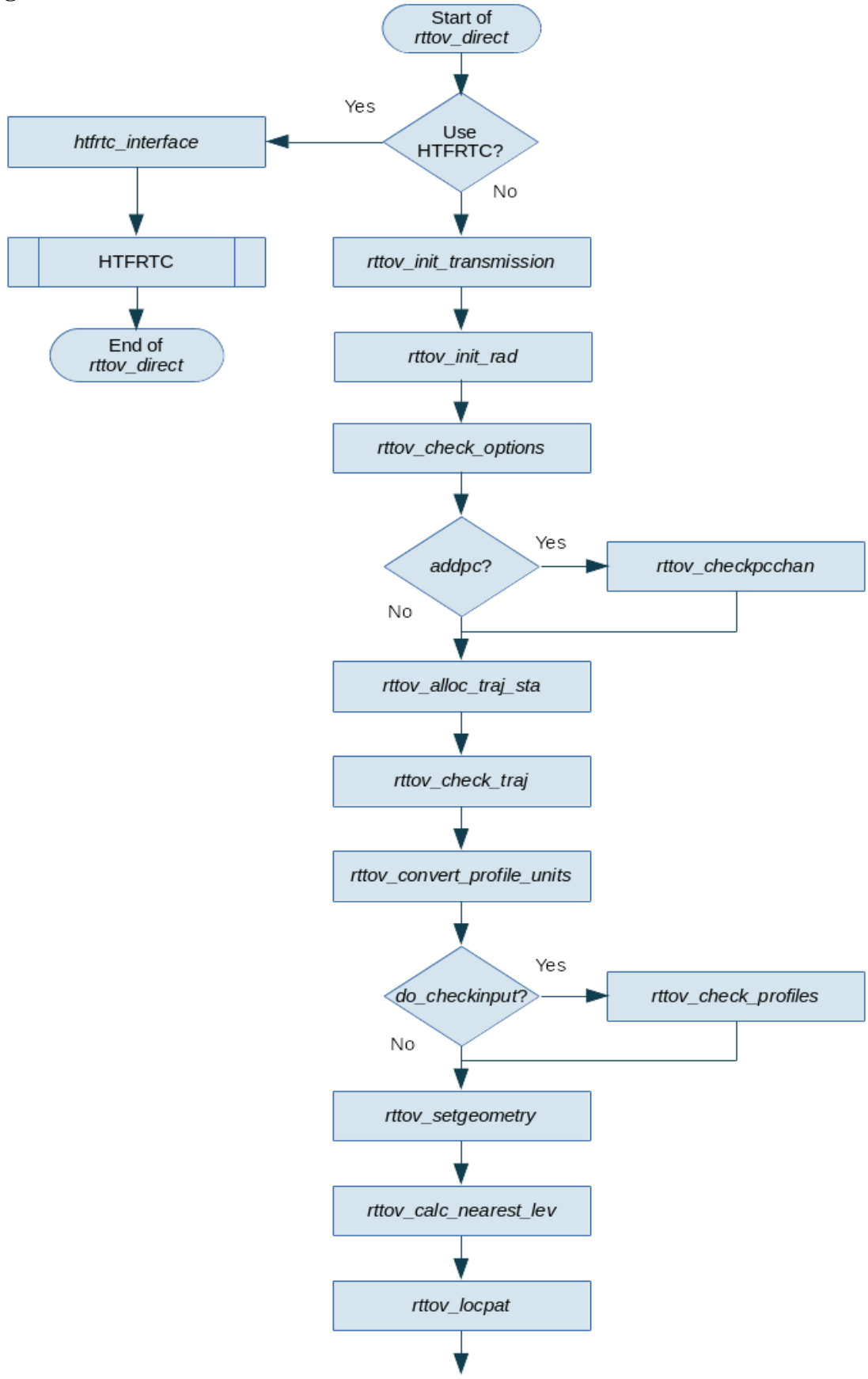
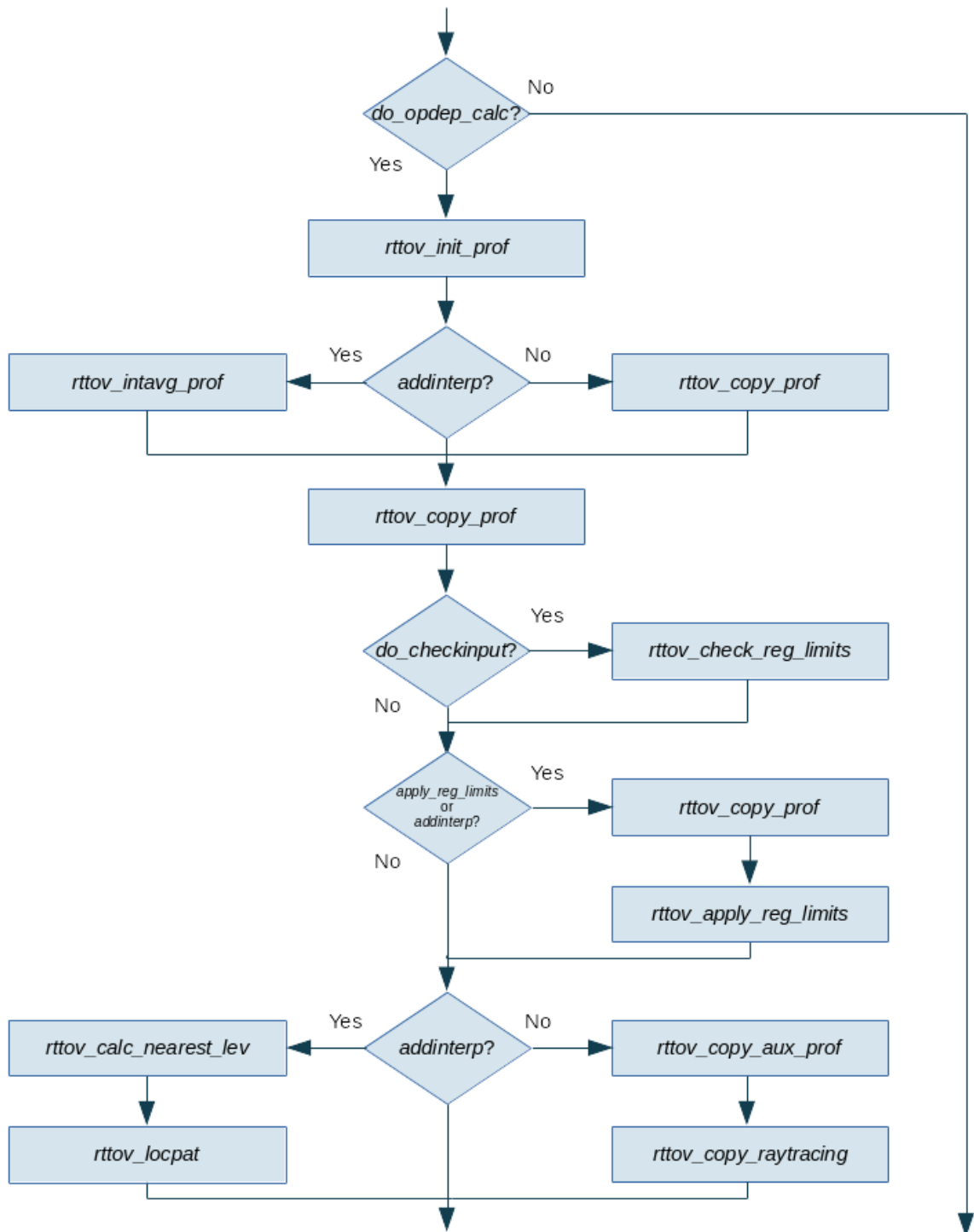


Figure 1. Process diagram of user program calling RTTOV v13 forward model.

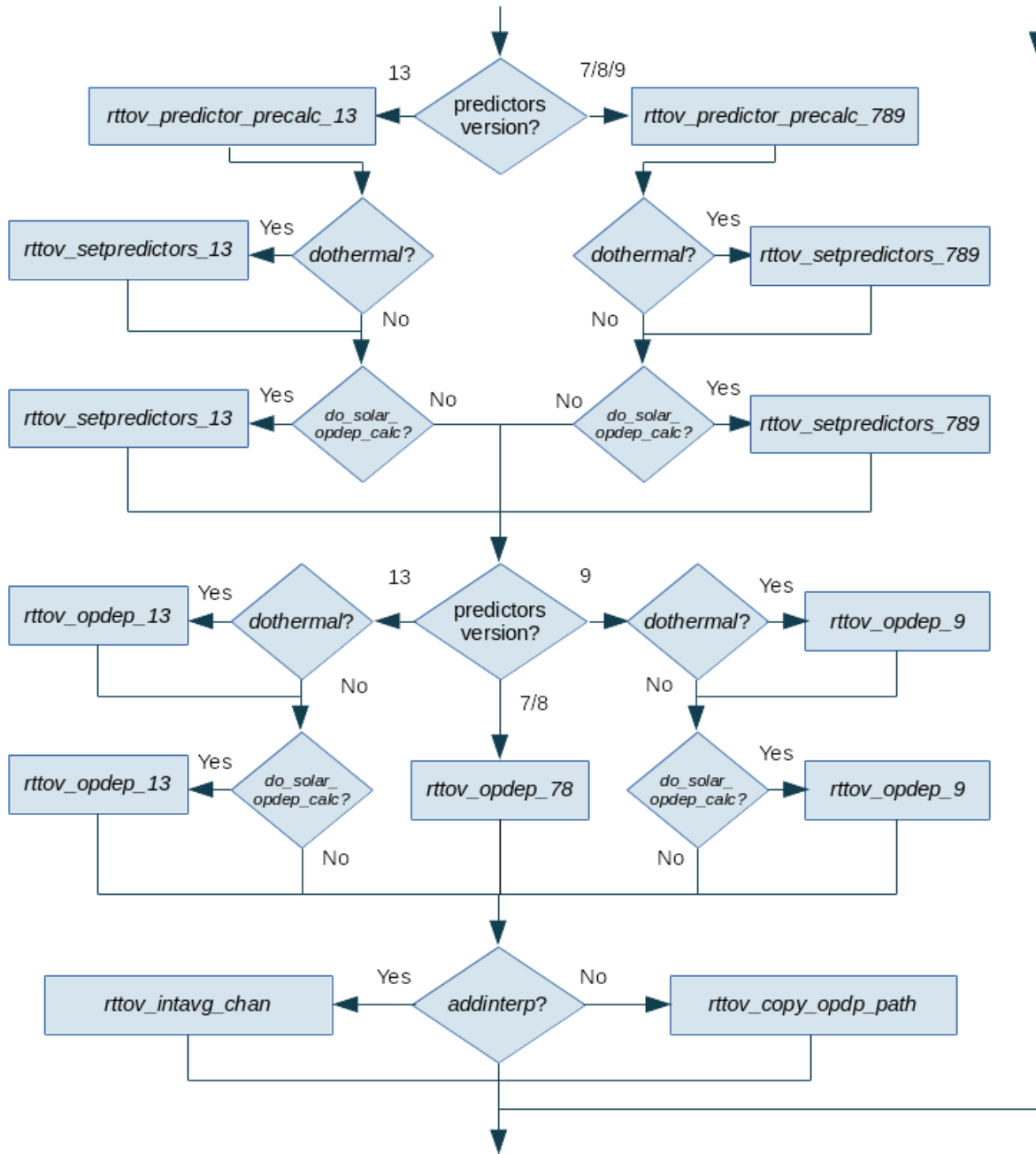
<p>NWP SAF</p>	<p>RTTOV Version 13 Top Level Design</p>	<p>Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020</p>
-----------------------	--	--

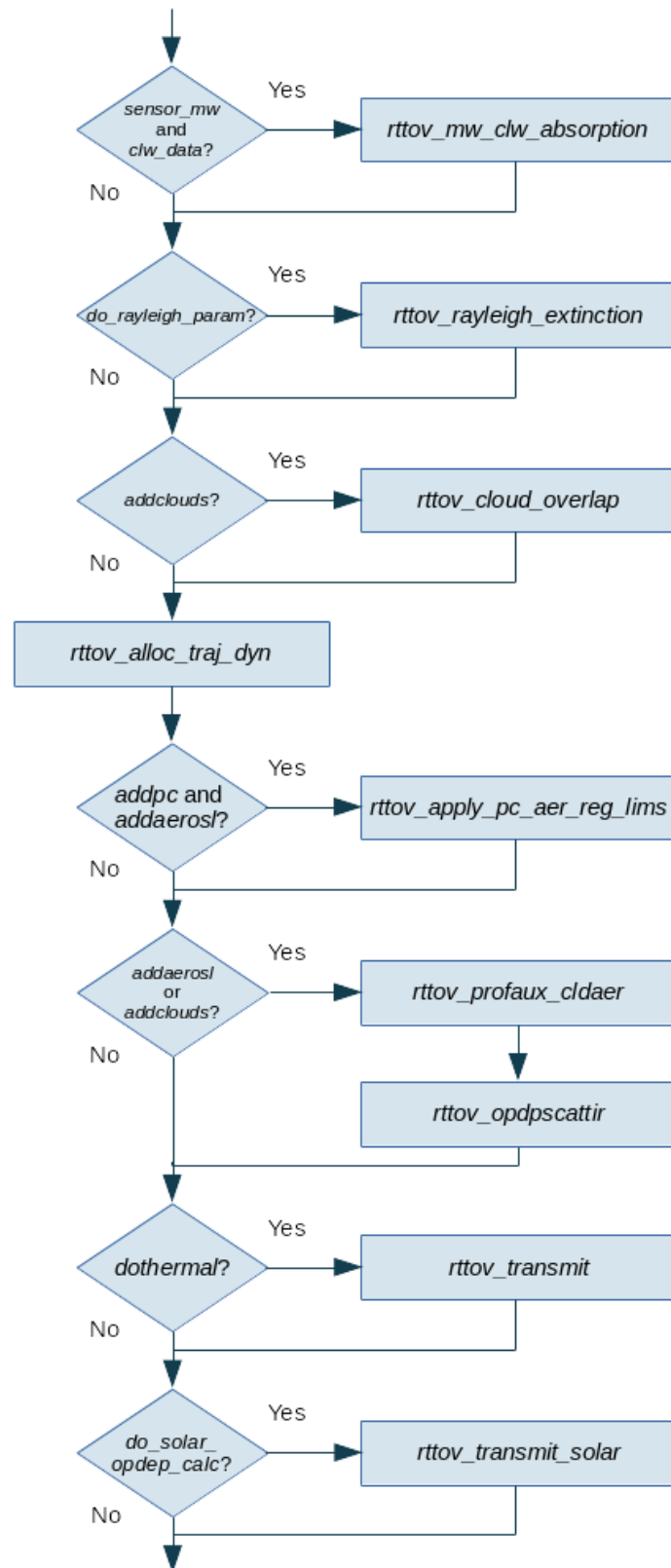
Calling tree for *rttov_direct*

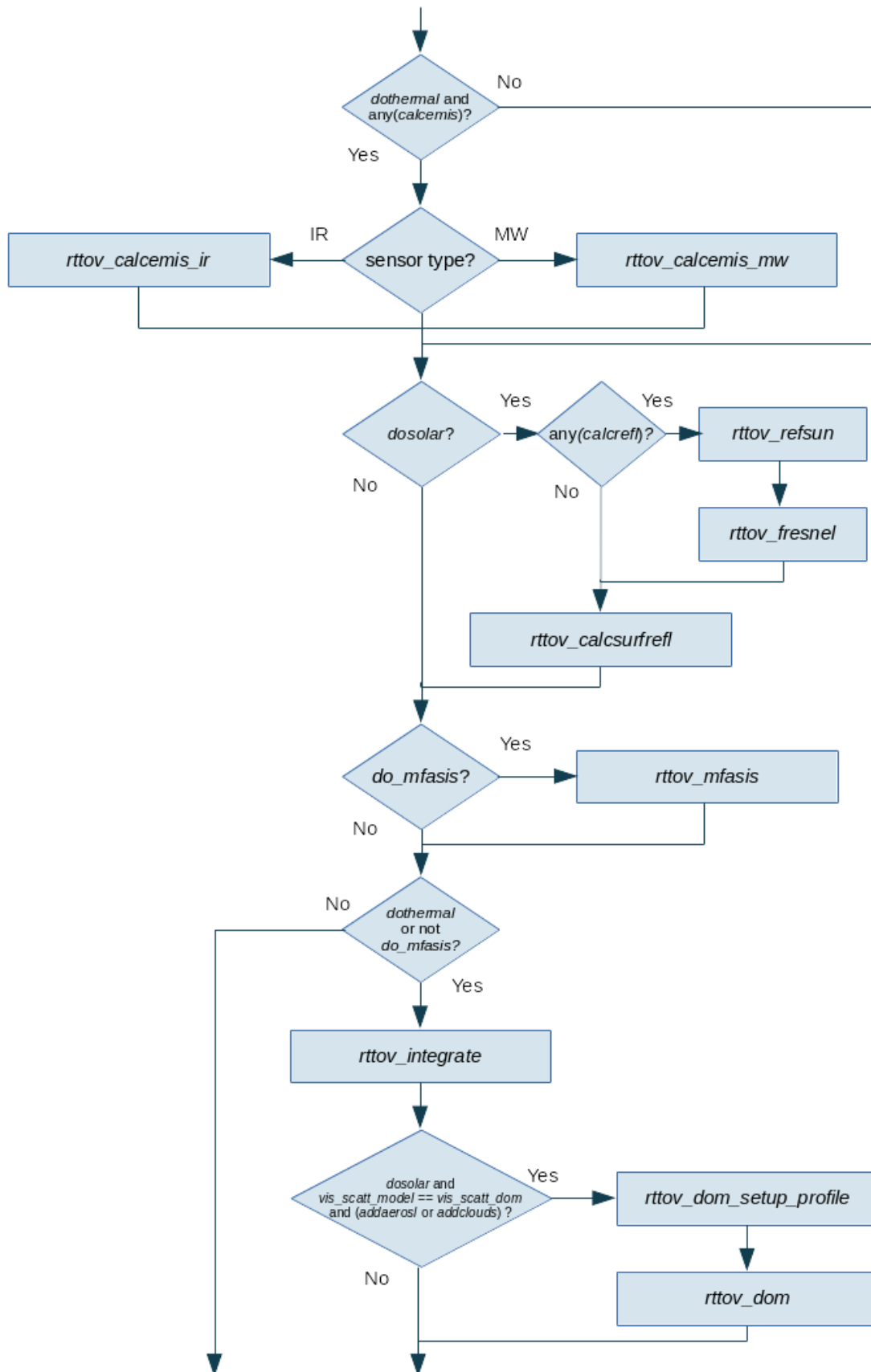




<p>NWP SAF</p>	<p>RTTOV Version 13 Top Level Design</p>	<p>Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020</p>
-----------------------	--	--







<p>NWP SAF</p>	<p>RTTOV Version 13 Top Level Design</p>	<p>Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020</p>
-----------------------	--	--

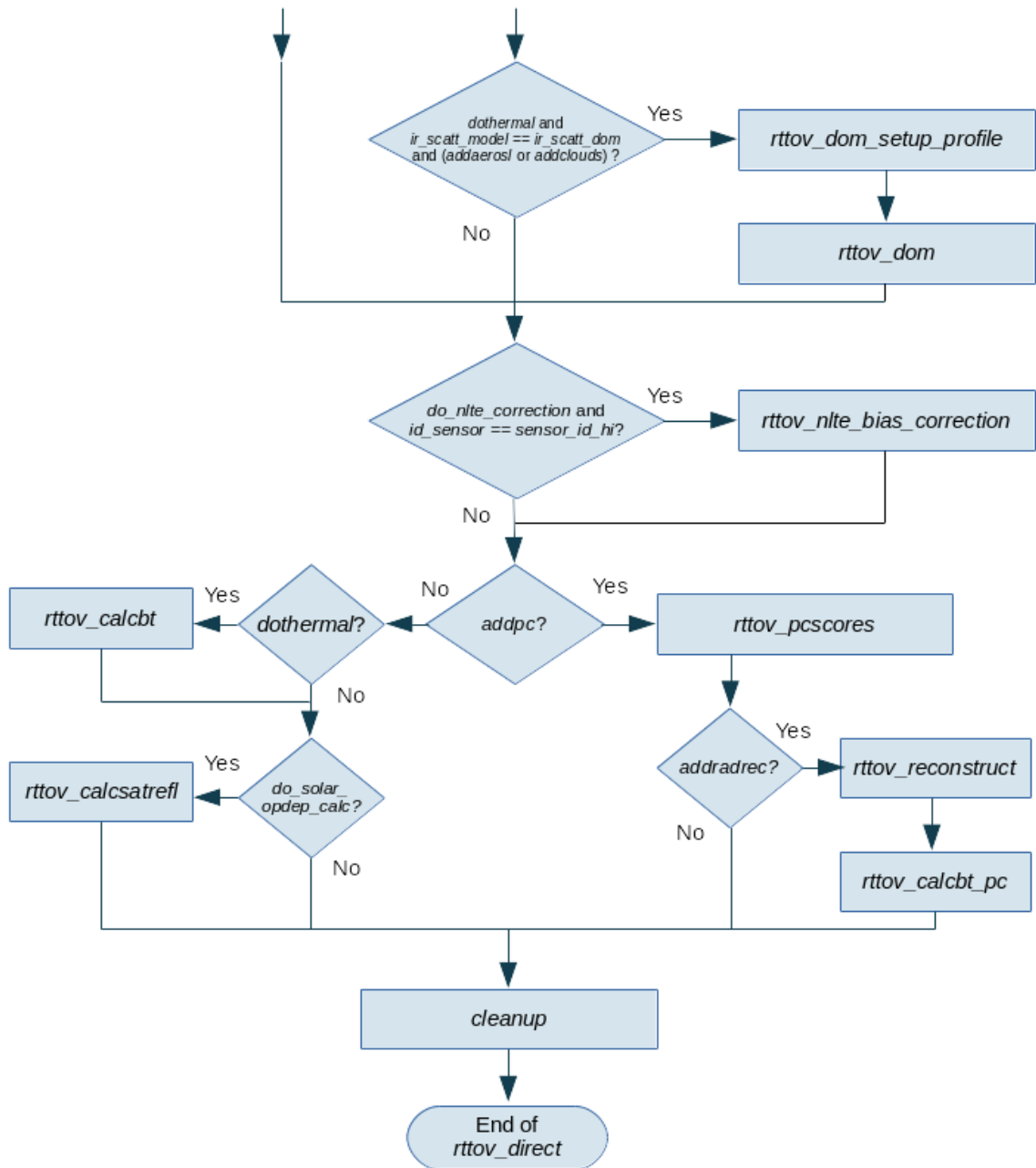


Figure 2. Calling tree for rttov_direct subroutine

<p>NWP SAF</p>	<p>RTTOV Version 13 Top Level Design</p>	<p>Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020</p>
-----------------------	--	--

Calling tree for *rttov_scatt*

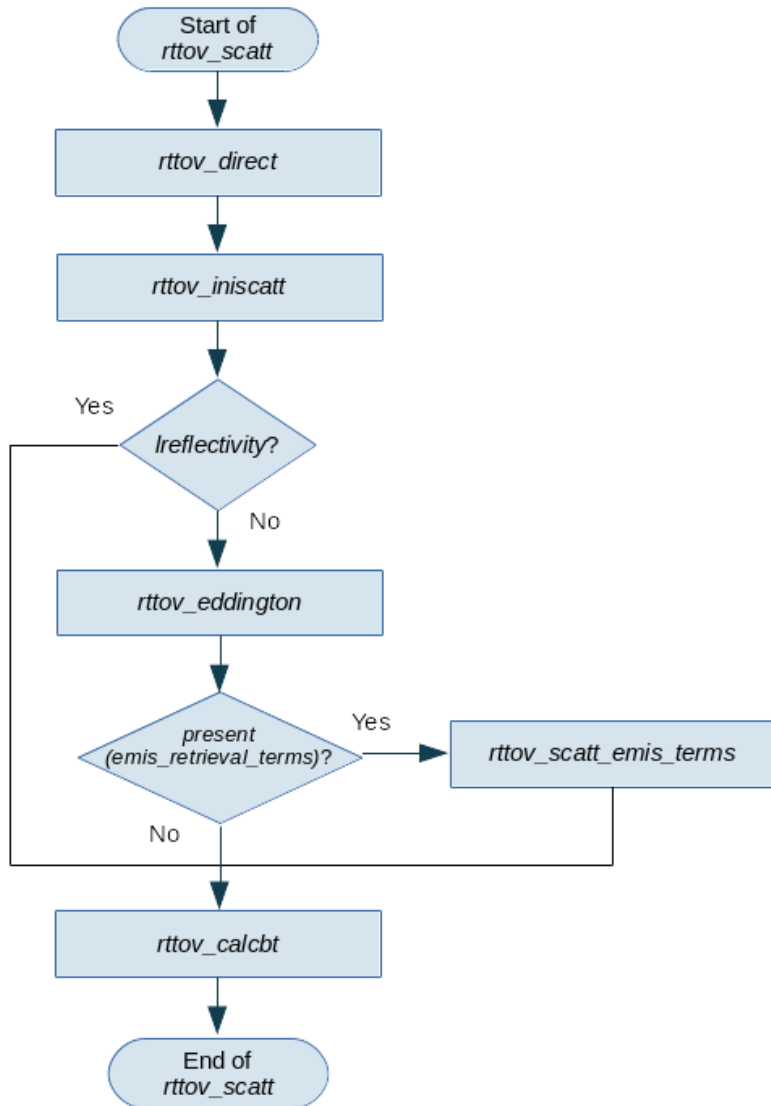


Figure 3. Calling tree for rttov_scatt subroutine

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

3. IMPLEMENTATION ASPECTS

3.1 External libraries

HDF5

RTTOV can be compiled and run without any external dependencies. However, for reasons of efficiency the coefficient files for hyperspectral sounders, the MFASIS look up tables, and the emissivity and BRDF atlases (excluding TELSEM2) are delivered in HDF5 format*. In addition, the RTTOV GUI requires HDF5. To make use of the full capabilities of RTTOV it is therefore necessary to compile against the HDF5 library (v1.8.8 or later, including v1.10.x and v1.12.x).

RTTOV includes a Perl script (*build/Makefile.PL*) which automatically generates the Makefiles: to enable HDF5 support this script is run with the argument *RTTOV_HDF=1*.

To manage compilation with/without HDF5 support enabled, the calls to HDF5 subroutines are switched in/out at compile-time by the presence/absence of the `_RTTOV_HDF` macro which may be supplied to the compiler. The user must edit the file *build/Makefile.local* to provide the compiler with the location of their HDF5 installation. This file already includes the relevant lines which pass this macro to the compiler.

**Users can request ASCII versions of the coefficient files via the helpdesk. Atlas data files are only available in HDF5 format.*

F2PY

The RTTOV GUI and the Python interface to RTTOV require f2py. If the user has f2py installed, they must pass *RTTOV_F2PY=1* to the *build/Makefile.PL* Perl script and then the relevant GUI and wrapper code will be compiled.

NetCDF4

The HTFRTC Principle Components-based radiative transfer model now supports ASCII coefficient files so it has no mandatory external dependencies. However, for greater efficiency, RTTOV can be compiled against the NetCDF4 library in which case the NetCDF HTFRTC coefficient files can be used.

In a similar way to HDF5, the `_RTTOV_NETCDF` macro must be passed to the compiler to switch in the calls to the NetCDF library at compile-time. The user must edit the file *build/Makefile.local* to provide the compiler with the location of their NetCDF4 installation. This file already includes the relevant lines which pass this macro to the compiler.

LAPACK

RTTOV uses several subroutines from the LAPACK library, and the source code for these are included in the package. Some users find this causes problems at compile-time if they are introducing RTTOV into a larger software system in which a version of the LAPACK library is already being linked. Users can choose to exclude the internal RTTOV LAPACK source code, and compile RTTOV against an external LAPACK library. To do this they must pass *RTTOV_USER_LAPACK=1* to the *build/Makefile.PL* script (so that the LAPACK source code is excluded from the Makefiles) and edit the file *build/Makefile.local* to provide the compiler with the location of their LAPACK library.

NWP SAF	RTTOV Version 13 Top Level Design	Doc ID : NWPSAF-MO-DS-040 Version : 1.1 Date : 19.10.2020
----------------	--	---

3.2 Compiler requirements

RTTOV requires a Fortran2008 compiler. The code is mainly Fortran90/95 with some commonly supported features from F2003 and one common F2008 intrinsic. The user guide [RD-2] lists compatible compilers and the full list of platforms/compilers with which RTTOV v13 has been successfully tested is given in the Test Log accompanying the Test Plan [RD-3].

3.3 Required and optional third-party software

RTTOV is self-contained (aside from the external libraries noted above) and so no additional third-party software is required.

3.4 Build method

Compilation of RTTOV involves the following steps

- Edit *build/Makefile.local* to point to the user's HDF5 installation (optional, required to use HDF5 format coefficient files and atlases and the RTTOV GUI)
- Edit *build/Makefile.local* to point to the user's NetCDF4 installation (optional, required for reading NetCDF HTFRTC coefficient files, otherwise read ASCII files)
- Edit *build/Makefile.local* to point to the user's LAPACK library (optional)
- Call *build/Makefile.PL* from within the *src/* directory (optional, required if HDF5 support and/or the RTTOV GUI and/or Python interface are required and/or a user LAPACK library is being used)
- Choose (and if necessary, edit) the appropriate compiler flag file in *build/arch/* (this contains several sets of flags for compilers with which RTTOV has been tested)
- Run *make* to compile RTTOV.

An interactive shell script *build/rttov_compile.sh* is provided to simplify this procedure for users. In this case the user must edit *build/Makefile.local* (if necessary) and then run *rttov_compile.sh*: the script prompts the user for several inputs and then executes the necessary commands to compile RTTOV.

3.5 Run-time considerations

The memory requirements of RTTOV vary a great deal with the kind of simulation being performed. For the more memory-intensive types of simulation (for example visible/IR scattering or PC-RTTOV) the user may find they have to pass fewer profiles to RTTOV in each call than they would for clear-sky simulations, for example.