

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

# **NWP SAF**

## **RTTOV Version 12 Top Level Design**

Version 1.1

25<sup>th</sup> January 2017

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

## RTTOV Version 12 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 29 June 2011, 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, KNMI and Météo France.

**Copyright 2017, EUMETSAT, All Rights Reserved.**

<b>Change record</b>				
<b>Version</b>	<b>Date</b>	<b>Author</b>	<b>Approved</b>	<b>Remarks</b>
1.0	15/11/16	JAH/DR		Initial draft
1.1	25/11/17	JAH		Updates after DRI

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

## Table of Contents

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>4</b>
<b>1.1</b>	<b>Reference documents .....</b>	<b>4</b>
<b>1.2</b>	<b>Design overview.....</b>	<b>4</b>
<b>2.</b>	<b>SOFTWARE FUNCTIONS AND MODULE/SUBROUTINE DESIGN.....</b>	<b>8</b>
<b>2.1</b>	<b>User-level functionality.....</b>	<b>8</b>
<b>2.2</b>	<b>RTTOV direct model design .....</b>	<b>8</b>
<b>2.3</b>	<b>RTTOV-SCATT direct model design .....</b>	<b>9</b>
<b>2.4</b>	<b>Tangent linear (TL), adjoint (AD) and Jacobian (K) models .....</b>	<b>9</b>
<b>2.5</b>	<b>Calling trees.....</b>	<b>9</b>
<b>3.</b>	<b>IMPLEMENTATION ASPECTS .....</b>	<b>16</b>
<b>3.1</b>	<b>External libraries .....</b>	<b>16</b>
<b>3.2</b>	<b>Compiler requirements.....</b>	<b>16</b>
<b>3.3</b>	<b>Required and optional third-party software .....</b>	<b>17</b>
<b>3.4</b>	<b>Build method .....</b>	<b>17</b>
<b>3.5</b>	<b>Run-time considerations.....</b>	<b>17</b>

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

## 1. INTRODUCTION

This document defines the top level design for Version 12 of RTTOV, in accordance with the requirements of the NWP SAF.

### 1.1 Reference documents

[RD-1] NWPSAF-MO-DS-029 RTTOV v12 Product Specification

[RD-2] NWPSAF-MO-UD-037 RTTOV v12 User Guide

[RD-3] NWPSAF-MO-TV-038 RTTOV v12 Test Plan

### 1.2 Design overview

#### Managing computer resources

The major requirement for RTTOV v12 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.

#### 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 memory for the coefficients. The *rttov\_conv\_coef* executable enables the user to convert coefficients 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.

The coefficient file format for v12 has changed since v11. Executables are available with RTTOV v12 to convert RTTOV v10/v11-format files to v12 format and vice versa. These work with both ASCII and HDF5 format files.

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

## 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. 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.

## Management of instrument channels

The user provides the model with a list of channel numbers for each calculated radiance. 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 is able to 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 v12 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 distinct 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 $\mu$ m, while for channels above 3 $\mu$ m a Lambertian BRDF is calculated from the surface emissivity. In both cases the same value is used for the direct and diffuse BRDFs.

In addition, land surface 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 and CAMEL 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.

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

In all cases the emissivities and reflectances 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). The **rttov\_parallel\_direct** subroutine has the same arguments as **rttov\_direct** plus an optional argument **nthreads** to specify the number of threads.

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 <sup>-1</sup> /sr/m <sup>2</sup> , degK, and BRF/unitless).
Type(rttov_radiance2)	Intent(inout), optional	radiance2	Secondary output radiances (mW/cm <sup>-1</sup> /sr/m <sup>2</sup> ).
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 <b>opts%rt_ir%addaerosl</b> and <b>opts%rt_ir%user_aer_opt_param</b> are true.
Type(rttov_opt_param)	Intent(in), optional	cld_opt_param	Cloud optical parameter input profiles if <b>opts%rt_ir%addclouds</b> and <b>opts%rt_ir%user_cld_opt_param</b> 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 <b>opts%rt_ir%pc%addradrec</b> is .true.

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

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

Type	In/Out	Variable	Description
Integer	Intent(out)	errorstatus	Return code.
Type(rttov_options_scatt)	Intent(in)	opts_scatt	Options to control aspects of RTTOV.
Integer	Intent(in)	nlevels	Number of input profile levels.
Type(rttov_chanprof)	Intent(in)	chanprof(nchanprof)	Chanprof structure.
Integer	Intent(in)	frequencies(nchanprof)	Frequency indices.
Type(rttov_profile)	Intent(in)	profiles(nprof)	Profiles structure.
Type(rttov_profile_cloud)	Intent(in)	cld_profiles(nprof)	Cloud profile structure.
Type(rttov_coefs)	Intent(in)	coef_rttov	RTTOV coefficient structure.
Type(rttov_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(rttov_emissivity)	Intent(inout)	emissivity(nchanprof)	Input/output emissivities.
Type(rttov_radiance)	Intent(inout)	radiance	Output radiances (mW/cm <sup>2</sup> /sr/m <sup>2</sup> & degK).
Real	Intent(out), optional	cfrac	Cloud fraction actually used (diagnostic).

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 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. Similarly an additional coefficient file is required for PC-RTTOV simulations.

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].

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

## 2. SOFTWARE FUNCTIONS AND MODULE/SUBROUTINE DESIGN

### 2.1 User-level functionality

RTTOV v12 provides the following primary user-level functions:

- Subroutine to read RTTOV coefficient files.
- Subroutine to check that user-specified options are consistent with the loaded coefficients.
- 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 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 HT-FRTC is selected, skips standard RTTOV code path and calls the interface subroutine for HT-FRTC.
- Performs simple checks on inputs to catch inconsistent or invalid options and arguments.
- Allocate and initialise internal data structures.
- Convert input profiles to the units used internally by RTTOV.
- If interpolation is required, interpolate input profiles onto RTTOV coefficient levels.
- If the *do\_checkinput* boolean is true, check profiles on coefficient levels for unphysical input. If the *verbose* boolean is true and the *apply\_reg\_limits* boolean is false the input profiles are also compared against the regression limits and warnings are output if the limits are exceeded.
- If the *apply\_reg\_limits* boolean is true, the interpolated profiles are clipped 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.
- Various auxiliary profile calculations are performed (for example calculations related to profile geometry).
- Calculation of gas absorption optical depth predictors on coefficient levels.
- Calculation of gas absorption optical depths on coefficient levels.
- If interpolation is required, interpolation of optical depths onto input levels.
- If visible/IR scattering simulations are enabled:
  - calculation of cloud columns (“streams”) due to cloud overlap assumptions
  - calculation of absorption and scattering optical depths due to aerosol/cloud
- Calculation of atmospheric transmittances.
- Calculate surface emissivities (where *calcemis(:)* is true).
- For solar simulations, calculate surface BRDFs (where *calcrefl(:)* is true).
- Integrate the radiative transfer equation\*.



<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

- 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.

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\_initscatt* to perform some preparatory calculations for the delta-Eddington solver. These include calculations related to the input profiles and calculations of the scattering properties based on the input Mitable file.
- Calls *rttov\_eddington* which integrates the radiative transfer equation.

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 actually 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 v12 direct model, for the *rttov\_direct* subroutine and for the *rttov\_scatt* subroutine respectively. Only subroutines called from the top-level subroutine are shown.

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

**Calling tree for user program calling RTTOV v12**

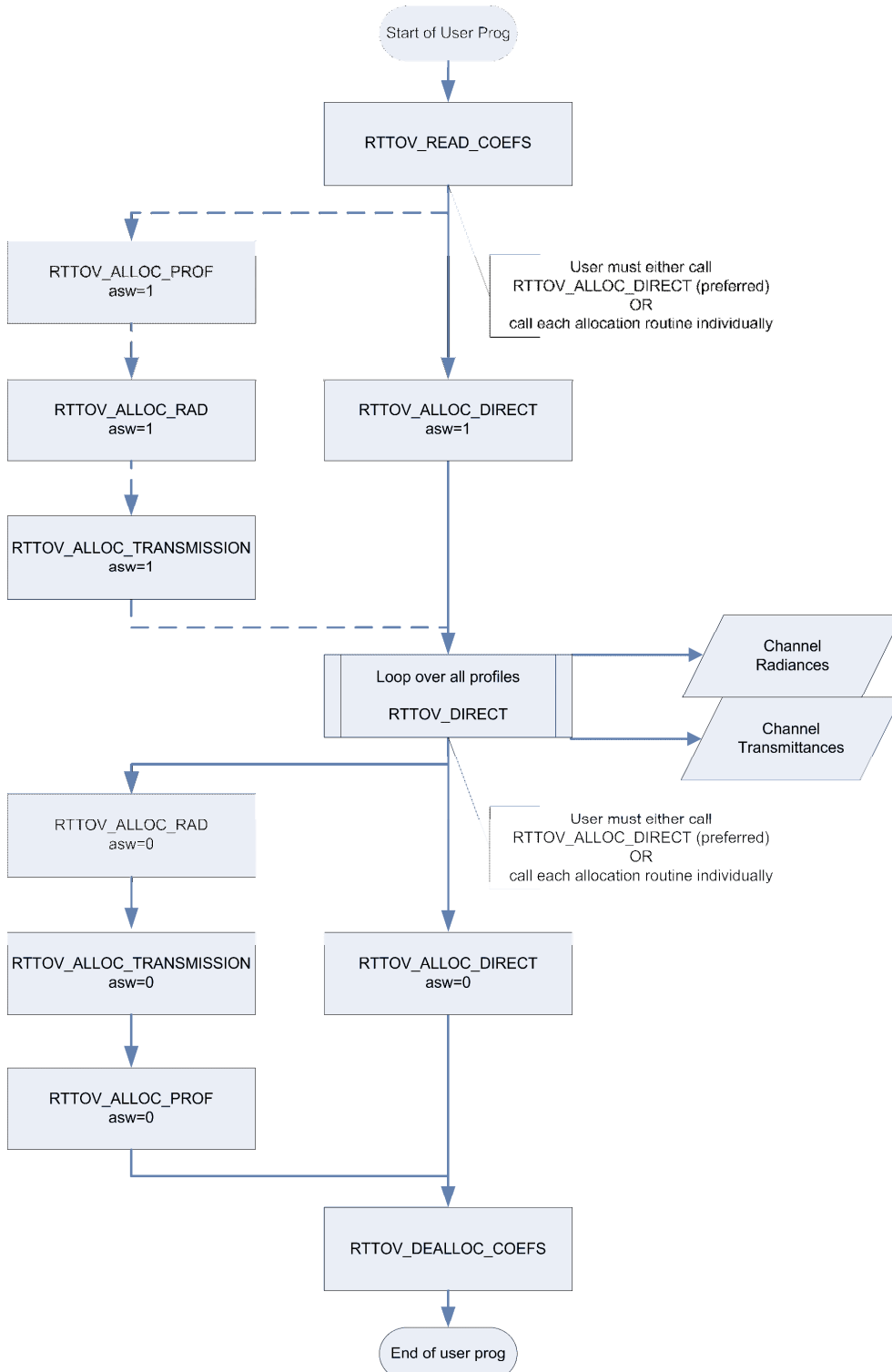
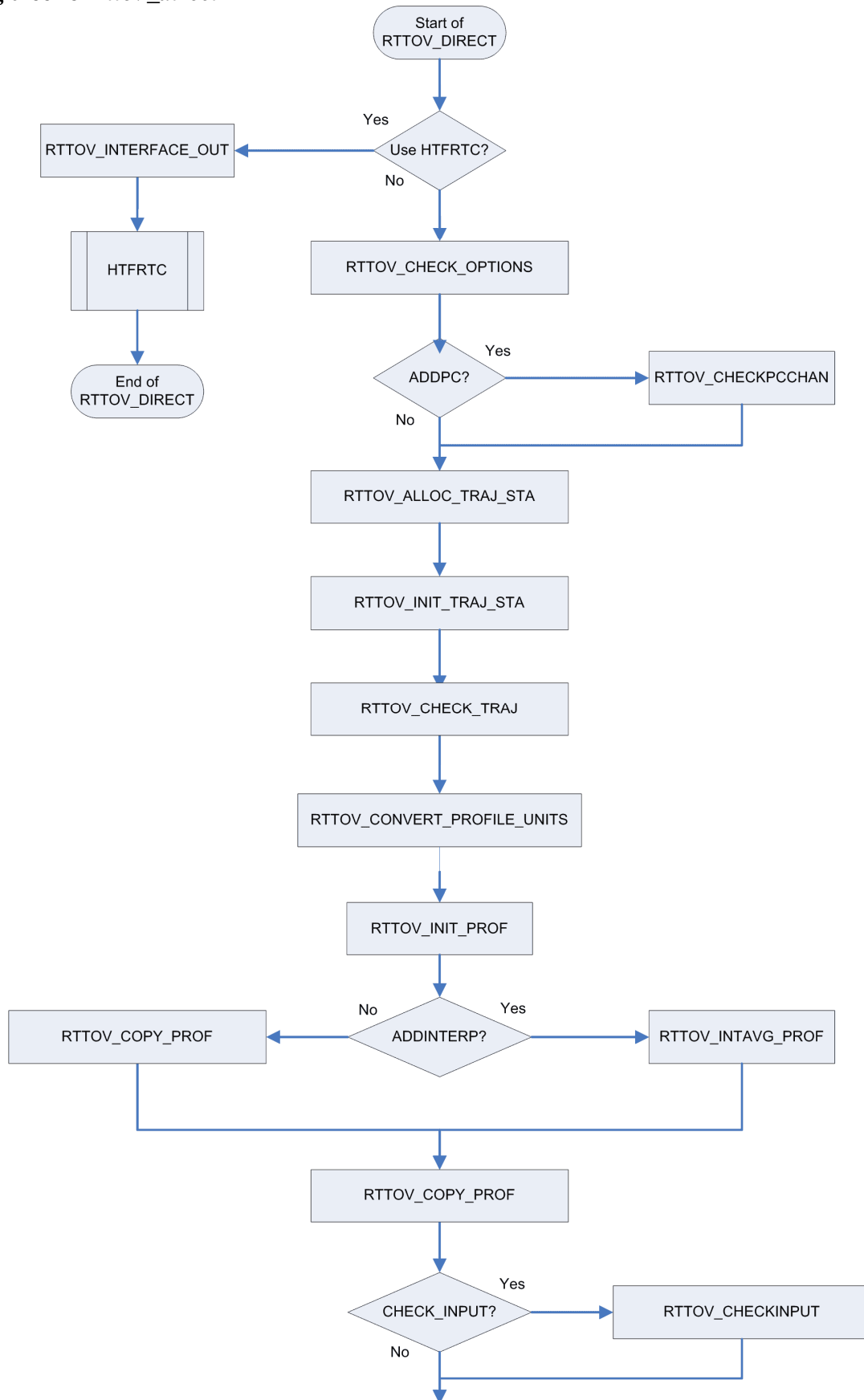
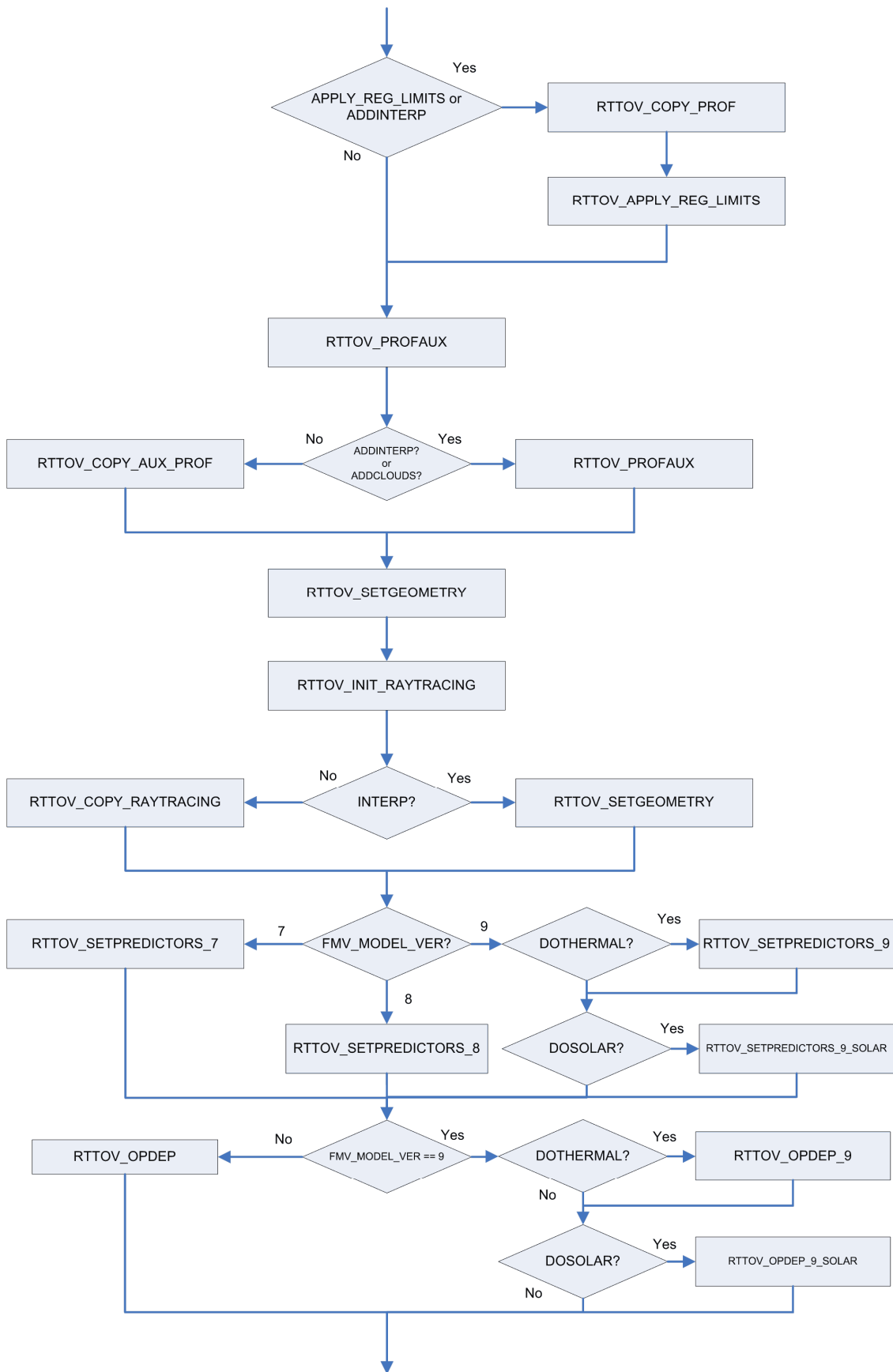
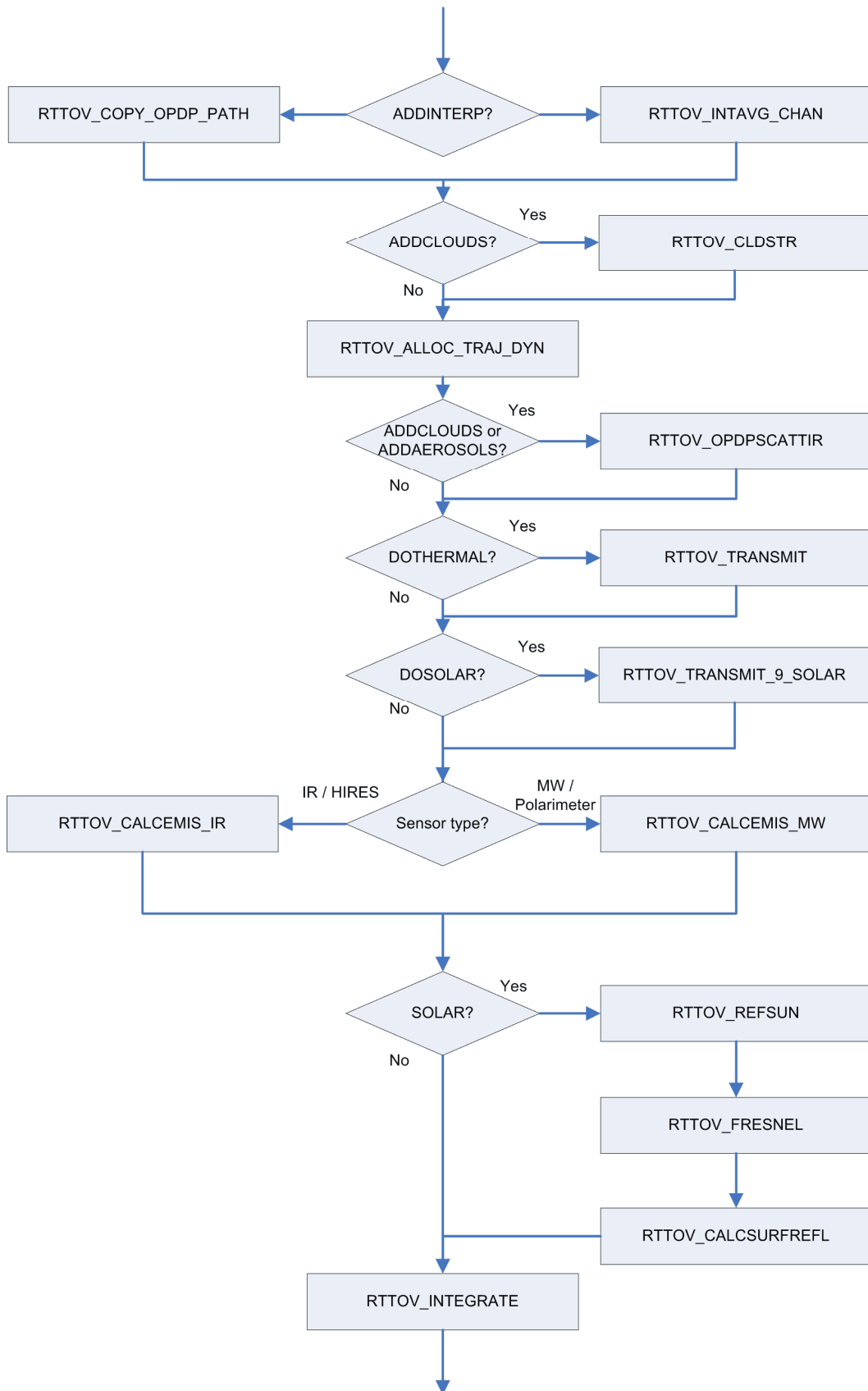


Figure 1. Process diagram of user program calling RTTOV v12 forward model.

**Calling tree for *rttov\_direct***







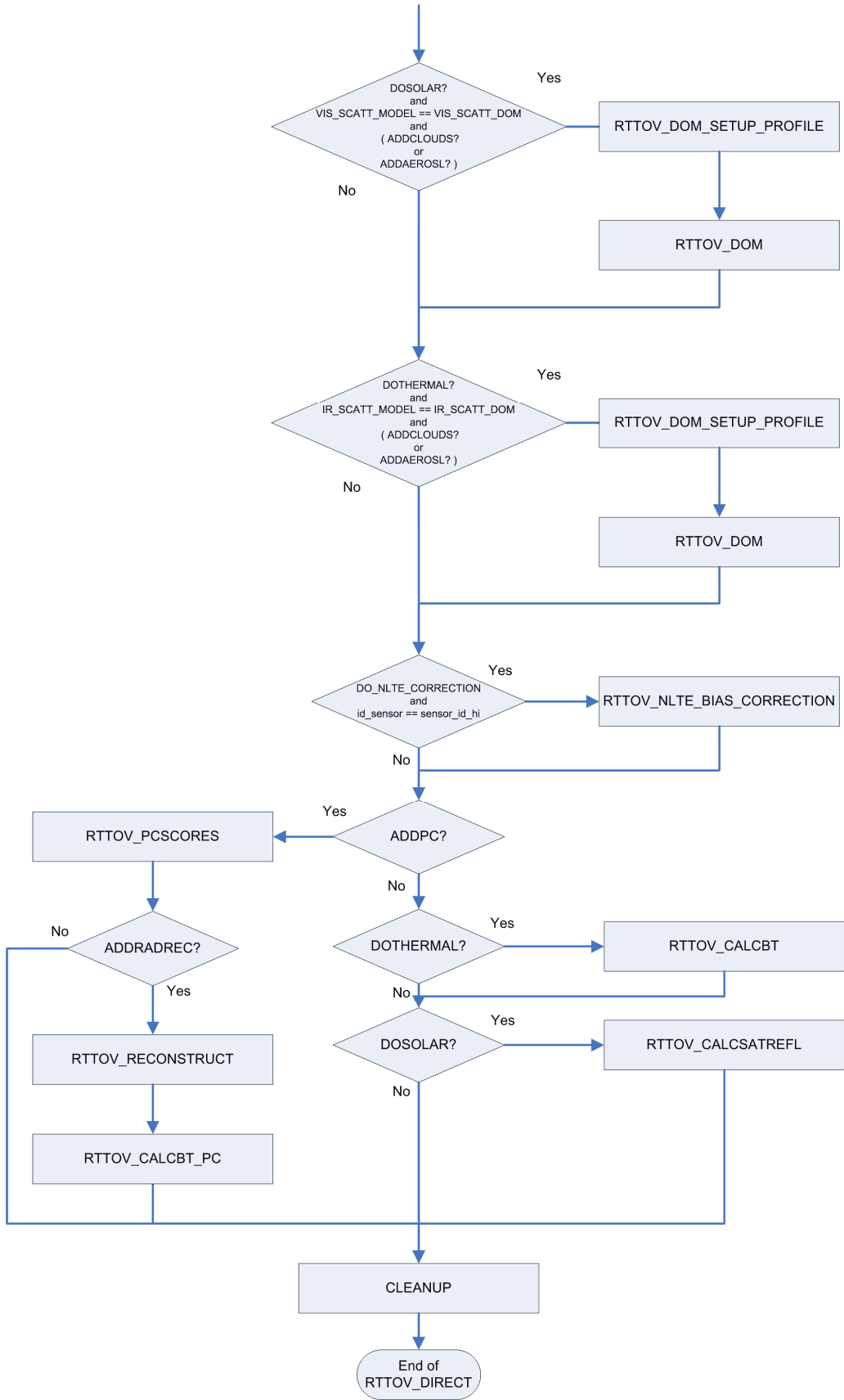
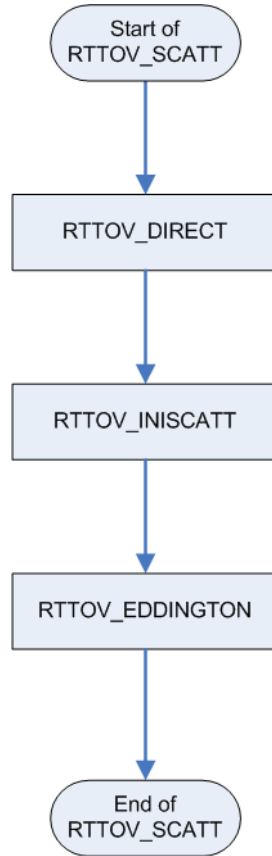


Figure 2. Calling tree for *rtov\_direct* subroutine

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

**Calling tree for *rttov\_scatt***



*Figure 3. Calling tree for rttov\_scatt subroutine*

<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

### 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 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).

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.

##### ***HT-FRTC***

HT-FRTC is a Principle Components-based radiative transfer model developed at the Met Office. RTTOV v12 includes the option to call HT-FRTC through the RTTOV interface for a limited range of simulation types and instruments.

To use this capability users must first download the HT-FRTC package (available from the NWP SAF website) and compile the software. The NetCDF library (v4.1 or later) is also required. To manage compilation of RTTOV with/without HT-FRTC support enabled, the calls to HT-FRTC are switched in/out at compile-time by the presence/absence of the *\_RTTOV\_HTFRTC* 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 HT-FRTC and NetCDF installations. This file already includes the relevant lines which pass this macro to the compiler.

#### 3.2 Compiler requirements

RTTOV requires a Fortran 90 compiler. The user guide [RD-2] lists compatible compilers and the full list of platforms/compilers with which RTTOV v12 has been successfully tested is given in the Test Log accompanying the Test Plan [RD-3].



<b>NWP SAF</b>	<b>RTTOV Version 12 Top Level Design</b>	Doc ID : NWPSAF-MO-DS-030 Version : 1.1 Date : 25.01.2017
----------------	--	---

### 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 HT-FRTC installation and NetCDF installation (optional, required to call HT-FRTC through the RTTOV interface)
- 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)
- 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.