

# RTTOV v11 TOP LEVEL DESIGN

NWPSAF-MO-DS-022

*James Hocking, David Rundle  
Met Office*

This documentation was developed within the context of the EUMETSAT Satellite Application Facility on Numerical Weather Prediction (NWP SAF), under the Cooperation Agreement dated 1 December, 2006, 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 2013, EUMETSAT, All Rights Reserved.**

Change record			
Version	Date	Author / changed by	Remarks
0.1	29/04/13	JH/DR	Initial draft
1.0	16/05/13	JH	Post-DRI.

## 1. Introduction and scope

The purpose of this report is to document the main functional elements of RTTOV v11 and to describe the sequence of internal subroutine calls. A users guide is available for an overview and installation of RTTOV v11 on to a users computer system. Scientific and validation documentation can be found in the RTTOV v11 scientific and validation report published on the NWP-SAF web site on the RTTOV v11 page at:

[http://research.metoffice.gov.uk/research/interproj/nwpsaf/rtn/rtn\\_rtov11.html](http://research.metoffice.gov.uk/research/interproj/nwpsaf/rtn/rtn_rtov11.html) along with all the other documentation. The code, scripts, test files and coefficient files are available either via ftp (as a compressed UNIX tar file) or on a CD-ROM from ECMWF data services.

## 2. RTTOV v11 design

### 2.1 RTTOV v11 package overview

The model is only available in standard Fortran 90 code and unix scripts are prepared to run the test code supplied with the model. The FORTRAN 90 standard defined in Andrews *et. al.* (1995) 'European Standards for Writing and Documenting Exchangeable Fortran 90 Code' has been followed wherever possible. The subroutine headers contain documentation on the input/output variables. All user-level subroutine interfaces are described in the user guide: this is intended to be the definitive reference for the RTTOV user interface. An overall routine calling tree is given in Figure 1. The code has been thoroughly tested for thread-safety within an operational parallel environment. It will be tested at least on the following platforms: IBM power 7, Linux Intel PC, Apple Mac and various compiler options for these which are supplied in the build/ directory.

The RTTOV v11 package delivered to the user consists of :

- a self contained library of Fortran90 subroutines and modules
- a perl script to generate Makefiles for the code
- a set of main programs for testing various aspects of the code
- a few utility routines
- a set of testing scripts for testing various aspects of the code
- a set of coefficient files (note AIRS, IASI and CrIS coefficients are not included due to their size but are available on the web site)
- a set of input files required for testing
- a set of reference test results
- a set of documentation

All the above are provided as a gzipped tar file to the user.

## 2.2 Functionality

### 2.2.1 Managing computer resources

The major requirement for RTTOV v11 is to allow the user to be able to run the code without limits for the number of channels, profiles and instruments in the same run. This has been established 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 of the arrays (sequence of allocation for all derived arrays of one derived type). 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...). 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.

## **2.2.2 Managing coefficient files**

The user manages the RTTOV coefficients within his 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 characteristics of the satellite and instrument, or with a filename and with a list of selected channels, or all channels are considered. *rttov\_read\_coefs* reads the data and allocates memory for the storage. *rttov\_dealloc\_coefs* will allow the user to de-allocate memory for the coefficients. The *rttov\_conv\_coef* program enables the user to create a coefficient set in a binary or HDF5 form which allow for more efficient input in RTTOV v11 especially for large coefficient files for advanced IR sounders.

RTTOV v11 coefficient files are compatible with RTTOV v10, but are in a slightly different format than earlier versions of RTTOV (v7/8/9) coefficient files. If the user wants to convert an old RTTOV v7/8/9 coefficient file to RTTOV v10/v11 format then he can run the program *rttov789\_conv\_coef* to create ascii coefficient files which can be read by the new RTTOV v11 code.

## **2.2.3 Management of atmospheric profiles**

The user should provide RTTOV with atmospheric profiles and surface variables in the form of atmospheric structures. The values of the profile will be checked according to the limits given by the coefficient file. The user has to provide a list of profile and channel indices for each calculated radiance.

## **2.2.3 Management of instrument channels**

The user has to provide 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. The user will be informed if an invalid channel number is provided.

## **2.2.4 Surface Emissivities**

The model is able to compute IR (over ocean and land) and MW emissivities (over ocean and land) according to some surface input parameters. The user has to provide a logical array indicating for which channel and profile he wants the model to calculate emissivities. The ocean surface emissivity models used in RTTOV v11 are ISEM for infrared radiances and a model developed by Matricardi (2010) for the PC scores and FASTEM versions 1-5 for the microwave radiances. Over the land an infrared (Borbas et. al. 2010) and microwave (Aires et. al. 2010) emissivity atlas can be used. The emissivities computed by the model are returned in the output arrays from RTTOV.

## **2.2.5 Surface Reflectances**

For solar-affected simulations a surface bi-directional reflectance (BRDF) must be supplied. The model can compute BRDF values over ocean in the presence of sun-glint according to a model implemented by Matricardi (2003). The user has to provide a logical array indicating for which channel and profile he wants the model to calculate BRDFs. For short-wave IR channels (with wavelengths greater than 3 microns), the BRDF may be calculated from the surface emissivity. Over land for visible and near-IR channels (0.4 – 2.5 microns) a BRDF atlas described in Vidot and Borbas (2013) may be used. The BRDF values computed/used by the model are returned in the output arrays from RTTOV.

## **2.2.5 Physical units**

Inside the code and for the user interface physical units used are:

- Pressure: *hPa*
- Atmospheric temperature: *Kelvin*
- Radiances:  $mW/cm^{-1}/sr/m^2$
- Equivalent black body brightness temperature: *Kelvin*
- PCscores:  $mW/cm^{-1}/sr/m^2$
- Gas mixing ratio: *ppmv*
- Cloud cover, cloud fraction: *(0-1) 0 for cloud free*
- Surface wind vector u and v in *m/s*.
- Cloud liquid water, ice water  $g/m^3$
- rain, solid precipitation flux:  $(kg/(m^2))/s$
- Aerosol layer mean number density:  $cm^{-3}$
- Angles (solar and satellite): *degrees (0°-360°) East = 90°*
- Elevation: *km*
- Latitude/Longitude: *degrees (-90° to +90°/0-360°)*
- Wind fetch: *km*
- Earth magnetic field strength: *Gauss*
- Surface snow coverage fraction: *Unitless (0-1) 0 for snowfree*
- Salinity ‰ (*parts per thousand*)

### 2.2.6 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), and **nprof** is the size of the **profiles(:)** array (i.e. the number of profiles to process in each call to 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(profile_type)	Intent(in)	profiles(nprof)	Profiles structure.
Type(rttov_coefs)	Intent(in)	coefs	RTTOV coefficient structure.
Type(transmission_type)	Intent(inout)	transmission	Output transmittances (0-1).
Type(radiance_type)	Intent(inout)	radiancedata	Primary output radiances (mw/cm <sup>-1</sup> /sr/m <sup>2</sup> & degK).
Type(radiance2_type)	Intent(inout), optional	radiancedata2	Secondary (optional) output radiances from forward model (mw/cm <sup>-1</sup> /sr/m <sup>2</sup> & degK).
Logical	Intent(in), optional	calcemis(nchanprof)	.true. if RTTOV should calculate surface emissivity using ISEM/FASTEM or .false. if user is supplying an emissivity value (e.g. from the atlas).
Type(rttov_emissivity)	Intent(inout), optional	emissivity(nchanprof)	Input/output emissivities.
Logical	Intent(in), optional	calcrefl(nchanprof)	.true. if RTTOV should calculate surface BRDF or .false. if user is supplying a BRDF value (e.g. from the atlas).
Type(rttov_reflectance)	Intent(inout), optional	reflectance(nchanprof)	Input/output BRDFs.
Type(rttov_opt_param)	Intent(in), optional	aer_opt_param	Profiles of optical parameters for IR aerosol simulations.
Type(rttov_opt_param)	Intent(in), optional	cld_opt_param	Profiles of optical parameters for IR cloud simulations.
Type(rttov_pccomp)	Intent(inout), optional	pccomp	Structure to hold output PC scores and reconstructed radiances.
Integer	Intent(in), optional	channels_rec(:)	Channels for which to compute reconstructed radiances if <b>opts % rt_ir % pc % addradrec</b> is .true.

The only input files required are the coefficient files. For the clear air code only one coefficient file per sensor is required but for the cloud, aerosol and rain scattering options additional coefficient files must be read which contain the scattering coefficients and optical parameters. Similar structures are required for the TL/AD and K codes.

### 3. RTTOV structures

Structures are used to define new derived data types that make the code lighter and easier to read. One structure stores related data like all information needed to describe an atmospheric profile. All variable arrays are allocated at the run time according to the environment, there is no need to declare parameters for the maximum size of arrays.

The full definition of the RTTOV v11 structures is given in the RTTOV v11 user guide.

## 4. RTTOV module/subroutine design overview

The RTTOV v11 model comprises various modules and subroutines to interface with the users code including subroutines for coefficient ingest, for allocating/deallocating RTTOV derived types, for accessing the emissivity and BRDF atlases, for each of the direct model, TL, AD, K and RTTOV\_SCATT codes, and several ancillary subroutines which may be helpful for users. These are described in more detail below and the interfaces are defined in the RTTOV v11 user guide.

The update of new coefficients or instrument parameters is managed by code not included with the RTTOV package. The coefficient files are delivered in ASCII format for portability. New science in the coefficient generation may induce changes inside the RTTOV code. For RTTOV v11 there are three options for predictors (so called RTTOV v7, RTTOV v8 and RTTOV v9 predictors). The selection of which code is run is managed by flags in the coefficient file.

### 4.1 RTTOV initialisation functions

In all RTTOV functions, local arrays are dynamic arrays for which dimensions are given by arguments (levels, channels, profiles, predictors).

The subroutines for the coefficient and atlas management and derived type allocation and initialisation are:

- subroutine to read the coefficients
- subroutine to de-allocate the coefficient derived type
- subroutine to allocate and de-allocate the profiles derived type
- subroutine to allocate and de-allocate the radiance derived types
- subroutine to allocate and de-allocate the transmittance derived type
- subroutine to allocate and de-allocate the PC score and reconstructed radiance derived type
- subroutines to initialise/read/de-allocate the emissivity atlases
- subroutines to initialise/read/de-allocate the BRDF atlases
- subroutine to allocate and de-allocate the optical parameter derived types
- subroutines to perform calculation of the “b” parameter for IR scattering simulations

The calling trees are given below in Figure 2 .

To read and initialise coefficients

```
call rttov_read_coefs (  
    err,  
    coefs,  
    opts,  
    channels,  
    channels_rec,  
    form_coef,  
    form_scaer,  
    form_scclld,  
    form_pccoef,  
    file_coef,  
    file_scaer,  
    file_scclld,  
    file_pccoef,  
    file_id_coef,  
    file_id_scaer,  
    file_id_scclld,  
    file_id_pccoef,  
    instrument)
```

Deallocate the coef structure arrays

```
call rttov_dealloc_coefs (err, coefs)
```

Allocate and deallocate profile structure according to asw

```
call rttov_alloc_prof (  
    err,  
    nprof,  
    profiles,  
    nlevels,  
    opts,  
    asw,  
    coefs,  
    init)
```

Allocate and deallocate the radiance structure according to asw

```
call rttov_alloc_rad (  
    err,  
    nchannels,  
    radiance,  
    nlayers,  
    asw,  
    radiance2,  
    init)
```

Allocate transmittance structure

```
call rttov_alloc_transmission (  
    err,  
    transmission,  
    nlayers,  
    nchannels,  
    asw,  
    init)
```

## 4.2 RTTOV (Forward Model) main functions

Perform the radiative transfer calculations for one coefficient set and number of profiles as specified in chanprof. Allocates memory for all local structures and deallocates before return.

The direct model subroutine calls sequentially:

- subroutine to compare the input profile with the coefficient limits
- subroutine to calculate some auxiliary parameters from the input profile, like geometry angles
- subroutine to interpolate user input profile on to RTTOV levels
- subroutines to calculate the optical depth and PC scores predictors
- subroutines to calculate optical depth and transmittances and PC scores
- subroutines to calculate surface emissivity
- subroutine to calculate surface BRDF
- subroutine to interpolate optical depths on to user levels
- subroutine to integrate the radiative transfer equation

The basic call to RTTOV looks like this:

```
call rttov_direct (  
    errorstatus,    ! out    return code  
    chanprof,      ! in     channel/profile index structure  
    opts,          ! in     options structure  
    profiles,      ! in     profile array  
    coefs,         ! in     coefficients structure  
    transmission, ! inout  computed transmittances  
    radiancedata, ! inout  computed primary radiances  
    calcemis,     ! in     flag for internal emissivity calc  
    emissivity,   ! inout  input/output emissivities  
    calcrefl,     ! in     flag for internal BRDF calc
```

reflectance) ! inout input/output surface BRDFs

Or including all the optional variables:

```
call rttov_direct (
  errorstatus,      ! out   return code
  chanprof,        ! in    channel/profile index structure
  opts,            ! in    options structure
  profiles,        ! in    profile array
  coefs,           ! in    coefficients structure
  transmission,    ! inout computed transmittances
  radiancedata,    ! inout computed primary radiances
  radiancedata2,   ! inout computed secondary radiances
  calcemis,        ! in    flag for internal emissivity calc
  emissivity,      ! inout input/output emissivities
  calcrefl,        ! in    flag for internal BRDF calc
  reflectance,     ! inout input/output surface BRDFs
  aer_opt_param,   ! in    profiles of aerosol optical parameters
  cld_opt_param,   ! in    profiles of cloud optical parameters
  traj,           ! inout optional structure to hold internal data
  pccomp,         ! inout computed PC scores
  channels_rec)   ! in    channel list for reconstructed PC radiances
```

*rttov\_direct* calls the following routines (N.B. only main functional routines listed here):

Check PC channels are valid if required.

```
Call rttov_checkpcchan( &
  & nprofiles,      &
  & nchannels,     &
  & opts,          &
  & chanprof,      &
  & coefs,         &
  & ERR            )
```

Set up internal structures and make quick escape if target values are out of bounds

```
Call rttov_check_traj( &
  & ERR,           &
  & nprofiles,     &
  & nchannels,     &
  & opts,          &
  & nlevels,       &
  & coefs,         &
  & l_jpim,        &
  & traj0 = traj0, &
  & traj1 = traj1, &
  & traj2 = traj)
```

Do interpolation of profile variables from user levels to RTTOV coef levels

```
Call rttov_intavg_prof( &
  & opts,          &
  & nlevels,       &
  & coefs%coef%nlevels, &
  & profiles,      &
  & traj0%profiles_COEF)!inout target variables
```

Check profile variables are within limits

```
Call rttov_checkinput( &
  & opts,          &
```



```
& traj0%profiles_COEF, &
& coefs%coef, &
& coefs%coef_pccomp, &
& err)
```

Determine cloud top, surface levels, ice cloud parameters

```
Call rttov_profaux( &
& opts, &
& profiles, &
& coefs%coef, &
& traj0%aux_prof)
```

Set up common geometric variables for the RT integration

```
Call rttov_setgeometry( &
& opts, &
& profiles, &
& traj0%aux_prof, &
& coefs%coef, &
& traj0_sta%angles, &
& traj0%raytracing)
```

Calculate atmosphere/solar predictors for different versions of RTTOV predictors – 7/8/9

```
IF (coefs%coef%fmv_model_ver == 7) THEN
CALL rttov_setpredictors_7( &
& opts, &
& traj0%profiles_COEF, &
& traj0_sta%angles_COEF, &
& coefs%coef, &
& traj0%aux_prof_COEF, &
& traj0%predictors, &
& traj0%predictors%path1, &
& traj0%raytracing_COEF)! inout (in because of mem allocation)
ELSE IF (coefs%coef%fmv_model_ver == 8) THEN
CALL rttov_setpredictors_8( &
& opts, &
& traj0%profiles_COEF, &
& traj0_sta%angles_COEF, &
& coefs%coef, &
& traj0%aux_prof_COEF, &
& traj0%predictors, &
& traj0%predictors%path1, &
& traj0%raytracing_COEF)! inout (in because of mem allocation)
ELSE IF (coefs%coef%fmv_model_ver == 9) THEN
IF (traj0_sta%dothermal) THEN
CALL rttov_setpredictors_9( &
& opts, &
& traj0%profiles_COEF, &
& traj0_sta%angles_COEF, &
& coefs%coef_pccomp, &
& coefs%coef, &
& traj0%predictors, &
& traj0%predictors%path1, &
& traj0%raytracing_COEF)! inout (in because of mem allocation)
ENDIF
IF (traj0_sta%dosolar) THEN
CALL rttov_setpredictors_9_solar( &
& opts, &
```

```

    & traj0%profiles_COEF, &
    & coefs%coef, &
    & traj0%predictors%path2, &
    & traj0%raytracing_COEF)

```

ENDIF

ENDIF

Compute atmospheric and solar optical depths for different versions of predictors.

```

IF (coefs%coef%fmv_model_ver == 9) THEN

```

```

  IF (traj0%sta%dothermal) THEN

```

```

    ! Calculate thermal path1 optical depths for all thermal channels

```

```

    CALL rttov_opdep_9( &
      & coefs%coef%nlayers, &
      & chanprof, &
      & traj0%sta%thermal, &
      & traj0%predictors, &
      & traj0%predictors%path1, &
      & traj0%aux_prof_COEF, &
      & coefs%coef, &
      & coefs%coef%thermal, &
      & traj0%opdp_path_COEF%atm_level, &
      & traj0%sta%thermal_path1%opdp_ref_COEF)

```

ENDIF

```

  IF (traj0%sta%dosolar) THEN

```

```

    ! Calculate solar path2 optical depths for all solar channels

```

```

    CALL rttov_opdep_9_solar( &
      & coefs%coef%nlayers, &
      & chanprof, &
      & traj0%sta%solar, &
      & traj0%predictors%path2, &
      & coefs%coef, &
      & coefs%coef%solar, &
      & traj0%opdp_path_COEF, &
      & traj0%sta%solar_path2%opdp_ref_COEF)

```

ENDIF

```

ELSE

```

```

  CALL rttov_opdep( &
    & coefs%coef%nlayers, &
    & chanprof, &
    & traj0%predictors, &
    & traj0%predictors%path1, &
    & traj0%aux_prof_COEF, &
    & coefs%coef, &
    & coefs%coef%thermal, &
    & traj0%opdp_path_COEF, &
    & traj0%sta%thermal_path1%opdp_ref_COEF)

```

ENDIF

Convert optical depths from RTTOV coef levels to user levels

```

CALL rttov_intavg_chan( &
  & opts, &
  & traj0%sta%thermal, &
  & traj0%sta%solar, &
  & coefs%coef%nlevels, &
  & nlevels, &
  & chanprof, &
  & traj0%profiles_COEF, &

```

```
& profiles, &
& traj0%opdp_path_COEF, &
& traj0%opdp_path)
```

If clouds are present calculate the number of streams and the cloud distribution in each stream

```
CALL rtov_cldstr( &
& opts, &
& profiles, &
& opts%rt_ir%cldstr_threshold, &
& traj0%ircld, &
& traj0_dyn%streams)
```

Calculate optical depths of aerosols and/or clouds

```
CALL rtov_opdpsscattir( &
& profiles(1)%nlayers, &
& chanprof, &
& opts, &
& traj0%aux_prof, &
& profiles, &
& aer_opt_param, &
& cld_opt_param, &
& traj0_sta%solar, &
& coefs%coef, &
& coefs%coef_scatt_ir, &
& traj0%raytracing, &
& traj0%transmission_scatt_ir, &
& traj0_dyn%transmission_scatt_ir_stream, &
& coefs%optp, &
& traj0%ircld)
```

Calculate transmittances for “thermal” and “solar” calculations

```
IF (traj0_sta%dothermal) THEN
! Thermal path1 transmittances for all thermal channels
CALL rtov_transmit( &
& opts%rt_ir%addaerosl, &
& opts%rt_ir%addclouds, &
& profiles(1)%nlayers, &
& chanprof, &
& traj0_sta%thermal, &
& traj0%aux_prof, &
& coefs%coef, &
& traj0%ircld, &
& traj0%opdp_path%atm_level, &
& traj0_sta%thermal_path1%od_level, &
& transmission%tau_levels, &
& transmission%tau_total, &
& traj0_dyn%transmission_aux, &
& traj0_dyn%transmission_aux%thermal_path1, &
& traj0_dyn%transmission_scatt_ir_stream, &
& traj0_sta%thermal_path1%tau_ref, &
& traj0_sta%thermal_path1%tau_ref_surf, &
& traj0_sta%thermal_path1%tau_surf, &
& traj0_sta%thermal_path1%tau_level)
ENDIF
```

```
IF (traj0_sta%dosolar) THEN
! Solar path2 transmittances for all solar channels
```

```
CALL rtov_transmit_9_solar( &
  & opts%rt_ir%addaerosl,      &
  & opts%rt_ir%addclouds,     &
  & profiles(1)%nlayers,      &
  & chanprof,                  &
  & traj0_sta%solar,          &
  & traj0%aux_prof,           &
  & coefs%coef,               &
  & traj0%raytracing,         &
  & traj0%ircld,              &
  & traj0%opdp_path,          &
  & traj0_sta%solar_path2,     &
  & traj0_sta%solar_path1,     &
  & transmission,             &
  & traj0_dyn%transmission_aux, &
  & traj0_dyn%transmission_scatt_ir_stream)
ENDIF
```

Compute IR surface emissivity

```
CALL rtov_calcemis_ir( &
  & profiles,                  &
  & traj0_sta%angles,         &
  & coefs%coef,               &
  & opts%rt_ir%pc%addpc,     &
  & coefs%coef_pccomp,       &
  & traj0_sta%thermal,       &
  & chanprof,                 &
  & calcemis,                 &
  & emissivity%emis_out,     &
  & err)
```

Compute MW surface emissivity

```
CALL rtov_calcemis_mw( &
  & opts,                      &
  & profiles,                  &
  & traj0_sta%angles,         &
  & coefs%coef,               &
  & chanprof,                 &
  & traj0_dyn%transmission_aux, &
  & calcemis,                 &
  & emissivity%emis_out,     &
  & traj0%thermrefl,         &
  & err)
```

Compute Fresnel reflectance and sun glint

```
CALL rtov_refsun( &
  & profiles,                  &
  & coefs%coef,               &
  & traj0%aux_prof,           &
  & traj0%sunglint,           &
  & traj0%raytracing)
```

```
CALL rtov_fresnel( &
  & chanprof,                 &
  & profiles,                  &
  & traj0_sta%solar,          &
  & coefs%coef,               &
```

& traj0%*sunlint*, &  
& traj0%*fresnrefl*)

Compute surface BRDF

```
CALL rtov_calcsurfrefl( &
  & coefs%coef, &
  & profiles, &
  & traj0%sunlint, &
  & traj0%fresnrefl, &
  & traj0%sta%solar, &
  & chanprof, &
  & traj0%sta%refl_norm, &
  & calcrefl, &
  & emissivity, &
  & reflectance%refl_out)
```

Integrate the radiative transfer equation

```
CALL rtov_integrate( &
  & addcosmic, &
  & opts, &
  & traj0%dyn%streams, &
  & chanprof, &
  & emissivity, &
  & reflectance, &
  & traj0%sta%refl_norm, &
  & traj0%thermrefl, &
  & traj0%sta%do_lambertian, &
  & traj0%sta%thermal, &
  & traj0%sta%dothermal, &
  & traj0%sta%solar, &
  & traj0%sta%dosolar, &
  & traj0%sta%solar_spec_esd, &
  & traj0%dyn%transmission_aux, &
  & traj0%dyn%transmission_scatt_ir_stream, &
  & profiles, &
  & traj0%aux_prof, &
  & coefs%coef, &
  & traj0%raytracing, &
  & traj0%sta%angles, &
  & traj0%ircld, &
  & radiancedata, &
  & radiancedata2, &
  & traj0%sta%auxrad, &
  & traj0%dyn%auxrad_stream)
```

Compute PC scores

```
CALL rtov_pcscor( &
  & opts, &
  & chanprof, &
  & traj0%sta%chanprof_pc, &
  & pccomp, &
  & coefs%coef_pccomp, &
  & radiancedata)
```

Optionally compute reconstructed radiances

```
CALL rtov_reconstruct( &
  & opts, &
```

```
& traj0_sta%chanprof_in, &
& traj0_sta%chanprof_pc, &
& pccomp, &
& coefs%coef_pccomp)
```

Optionally compute the NLTE bias correction

```
CALL rttov_nlte_bias_correction(coefs%coef, profiles, traj0_sta%angles, chanprof, radiancedata)
```

Calculate satellite-seen BTs, reconstructed BTs and reflectances

```
CALL rttov_calcbt_pc(traj0_sta%chanprof_in, coefs%coef_pccomp, pccomp)
```

```
CALL rttov_calcbt(chanprof, coefs%coef, traj0_sta%thermal, radiancedata)
```

```
CALL rttov_calcsatrefl(chanprof, profiles, traj0_sta%solar_spec_esd, traj0_sta%solar, radiancedata)
```

## 4.2 RTTOV\_scatt main functions

RTTOV\_scatt needs:

- Module to read the Mie coefficients
- Module to interpolate input profiles on NWP pressure levels to fixed coefficients levels
- Module to perform initialisations for delta-eddington
- Module to calculate scattering parameters from Mie tables
- Module to perform delta-eddington calculations

Computes microwave multi-channel radiances and brightness temperatures for many profiles per call in a cloudy and/or rainy sky. `radianced1%tb` = cloud-affected Tbs `radianced1%tb_clear` = clear-sky Tbs.

```
call rttov_scatt (
  errorstatus, ! out   return status
  opts_scatt,  ! in    options structure
  nlevels,     ! in    number of levels
  chanprof,    ! in    channel/profile index structure
  frequencies, ! in    MW channel frequencies
  profiles,    ! in    profile structure
  cld_profiles, ! in    cloud profile structure
  coef_rttov,  ! in    clear-air RTTOV coefficients
  coef_scatt,  ! in    RTTOV_SCATT coefficients (mie table)
  calcemiss,   ! in    logical array for internal emissivity calc
  emissivity,  ! inout  input/output emissivities
  radiance,    ! inout  computed radiances
  cfrac)       ! out   Cloud fraction actually used
```

`rttov_scatt` calls:

1. Gas absorption perform the radiative transfer calculations for one coefficient set and nprofiles.

```
Call rttov_direct(
  & errorstatus, &! out
  & chanprof,    &! in
  & opts,        &! in
  & profiles,    &! in
  & coef_rttov,  &! in
  & transmission, &! inout
  & radiance,    &! inout
  & calcemis = calcemiss, &! in
  & emissivity = emissivity) ! inout
```

Initialisations for delta Eddington. Calculates some variables related to the input precipitation profile

```

Call rtov_iniscatt(      &
  & errorstatus,      &! out
  & nlevels,          &! in
  & nchannels,        &! in
  & nprofiles,        &! in
  & chanprof,         &! in
  & frequencies,     &! in
  & profiles,         &! in
  & cld_profiles,    &! in
  & coef_rtov%coef,  &! in
  & coef_scatt,      &! in
  & transmission,   &! in
  & calcemiss,      &! in
  & opts_scatt%lusercfrac, &! in
  & angles,         &! out
  & scatt_aux)     ! inout

```

To compute Eddington approximation to RT.

```

Call rtov_eddington( &
  & nlevels,      &! in
  & nchannels,    &! in
  & nprofiles,    &! in
  & chanprof(:)%prof, &! in
  & angles,      &! in
  & profiles,    &! in
  & scatt_aux,   &! in
  & cld_bt)     ! out

```

### 4.3 RTTOV Tangent Linear (TL), Adjoint (AD) and Jacobian matrix (K) design

The TL model is based on the forward model, all functionalities are the same for all subroutines. They all deliver the tangent linear as an additional output.

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

See users guide for a definition of what the TL, AD and K routines actually compute. The same routine names as for the forward model are used but appended with ‘\_TL’, ‘\_AD’, and ‘\_K’.

## 5. Flow diagrams for RTTOV11

Flow diagrams for a user program calling RTTOV v11 and for the main *rttov\_read\_coefs* and *rttov\_direct* routines are given below to show all subroutines that are called.

## 5.1 User calling tree for user program calling RTTOV11

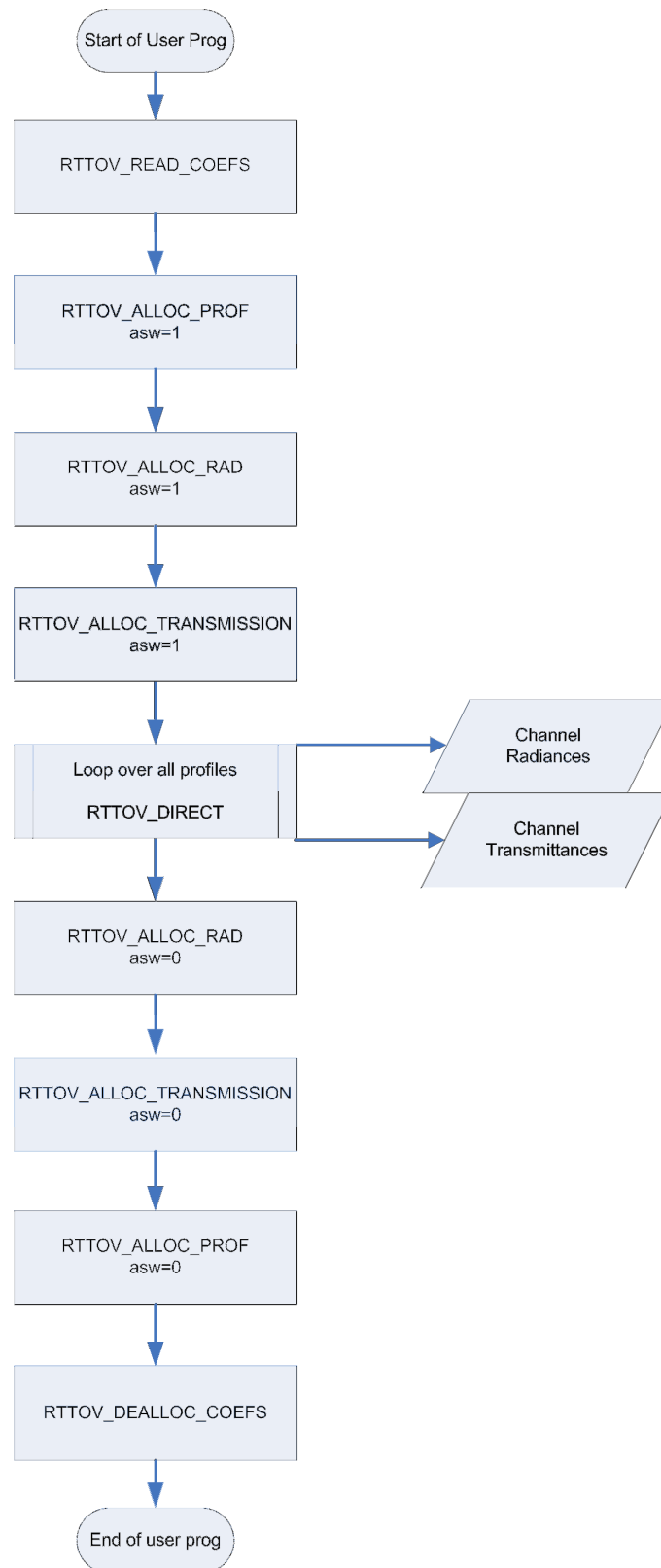
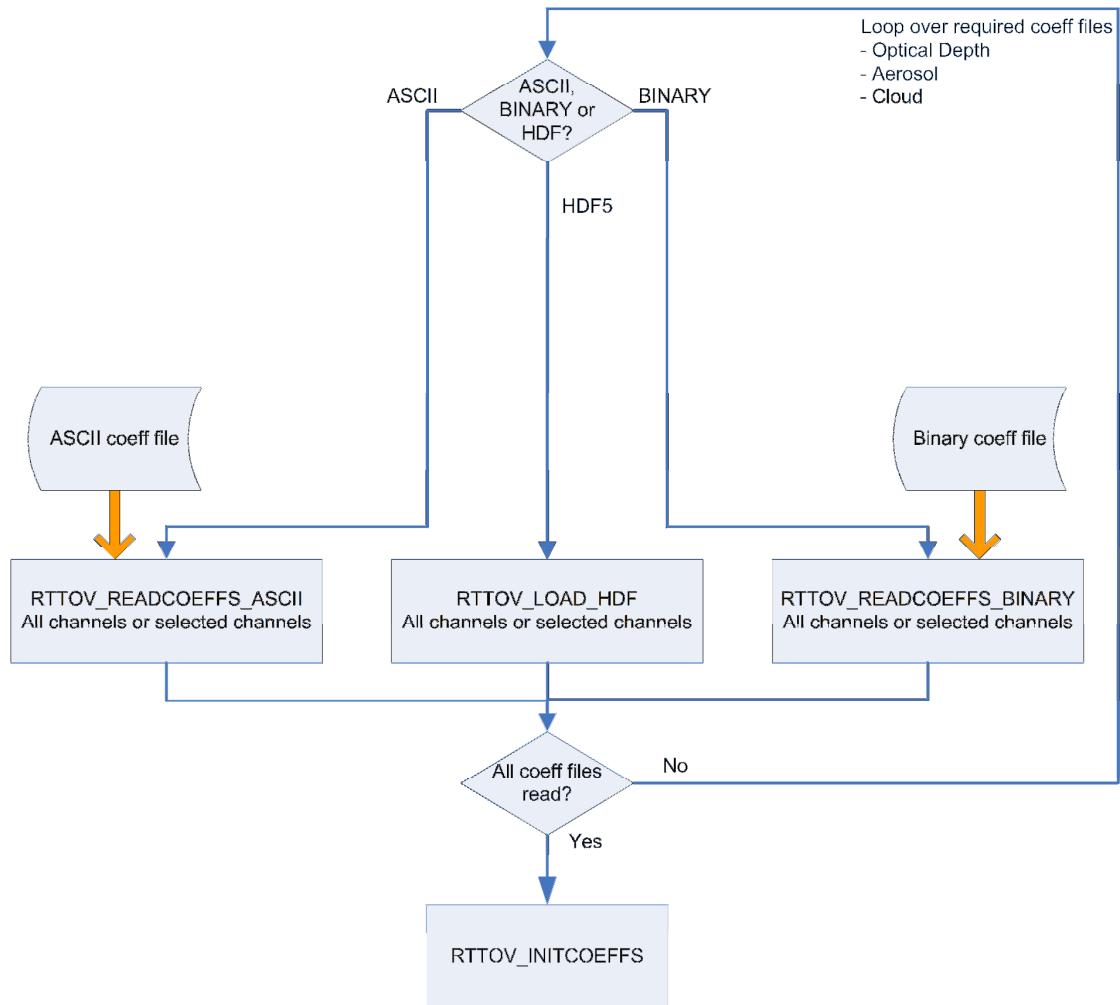


Figure 1. Process diagram of user program calling RTTOV v11 forward model.



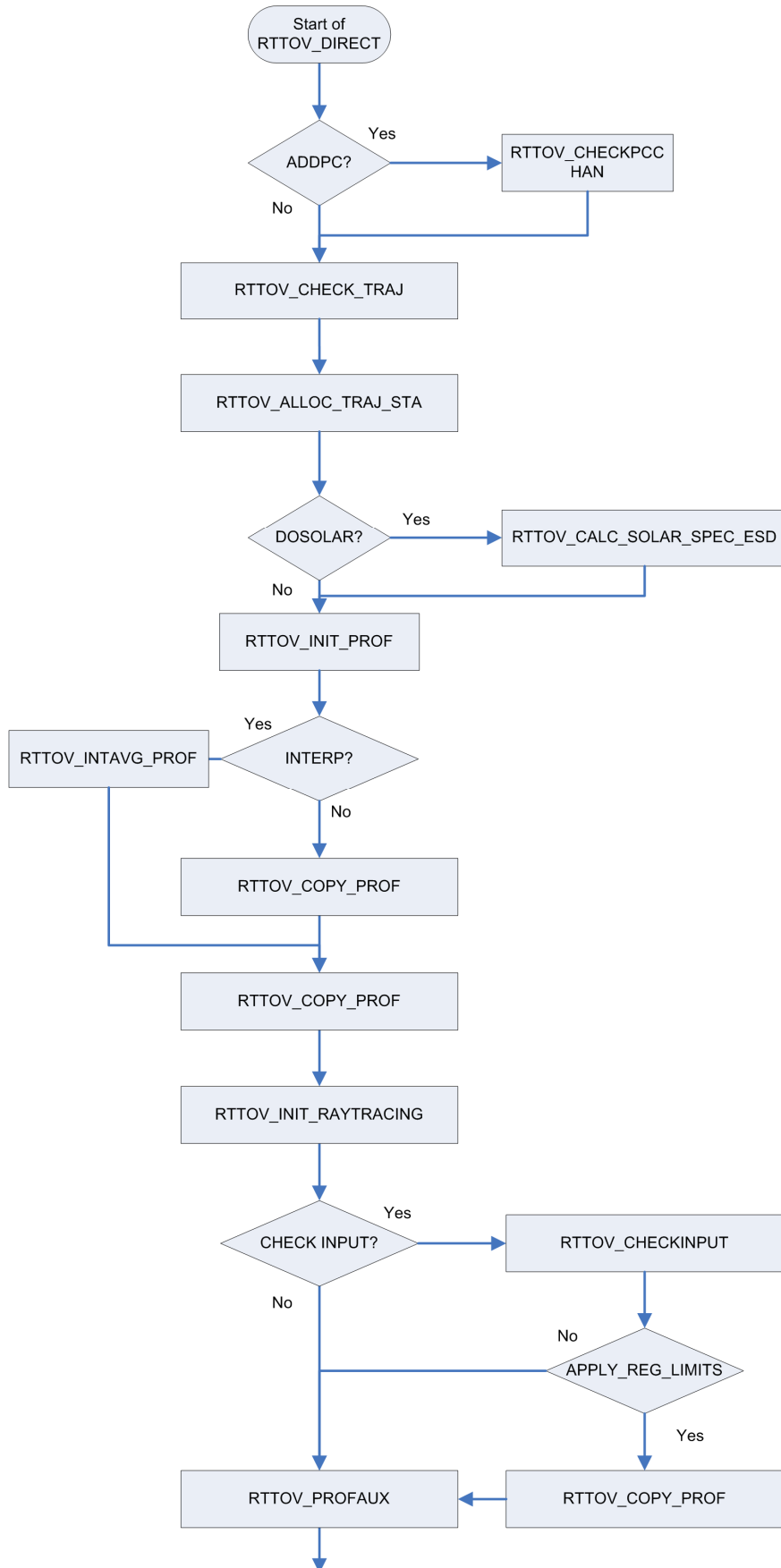
## 5.2 Calling tree for *rttov\_read\_coefs*

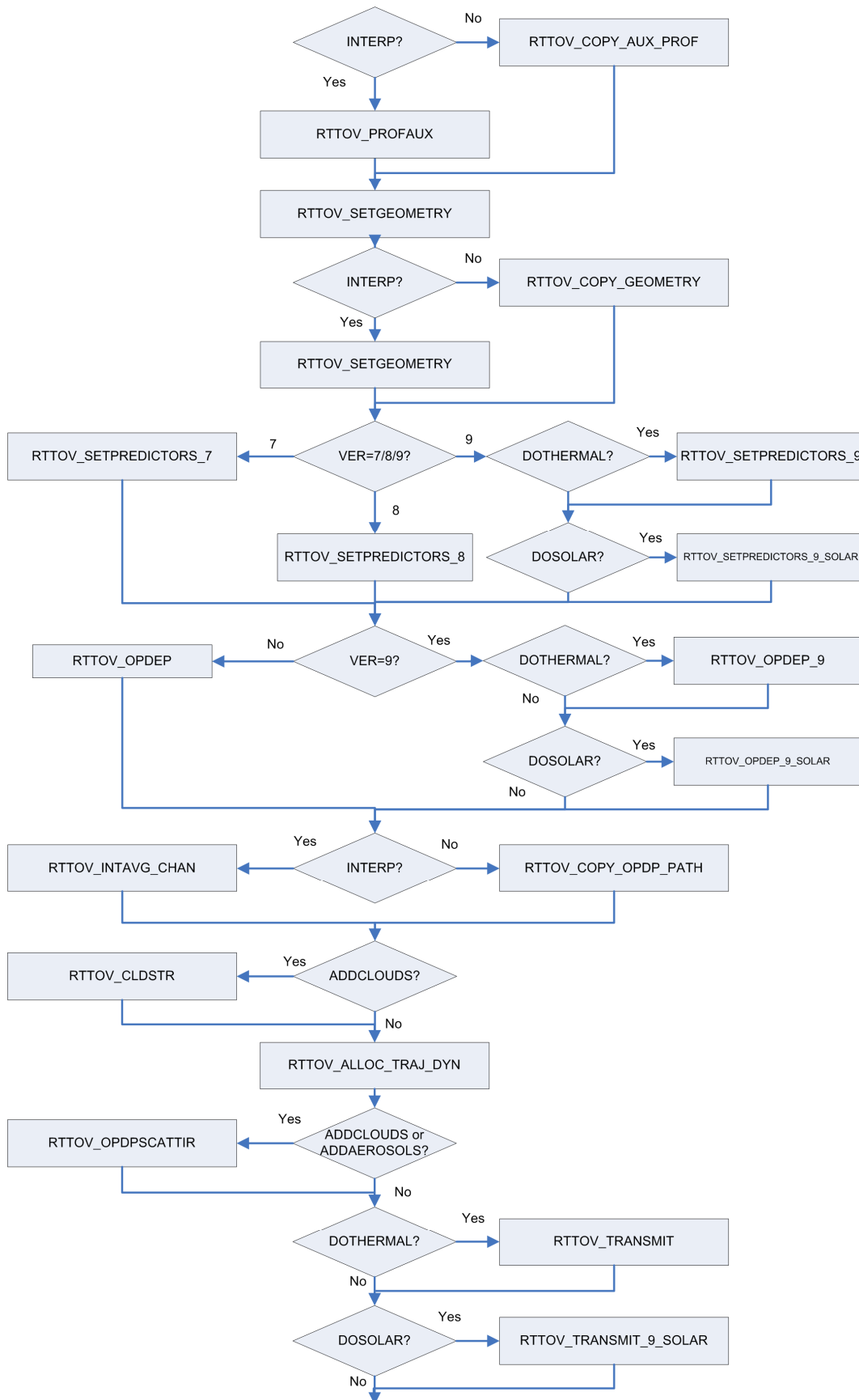


**Figure 2. Calling tree for RTTOV\_READ\_COEFS routines**

### **5.3 Calling tree for *rttov\_direct***

See next 3 pages





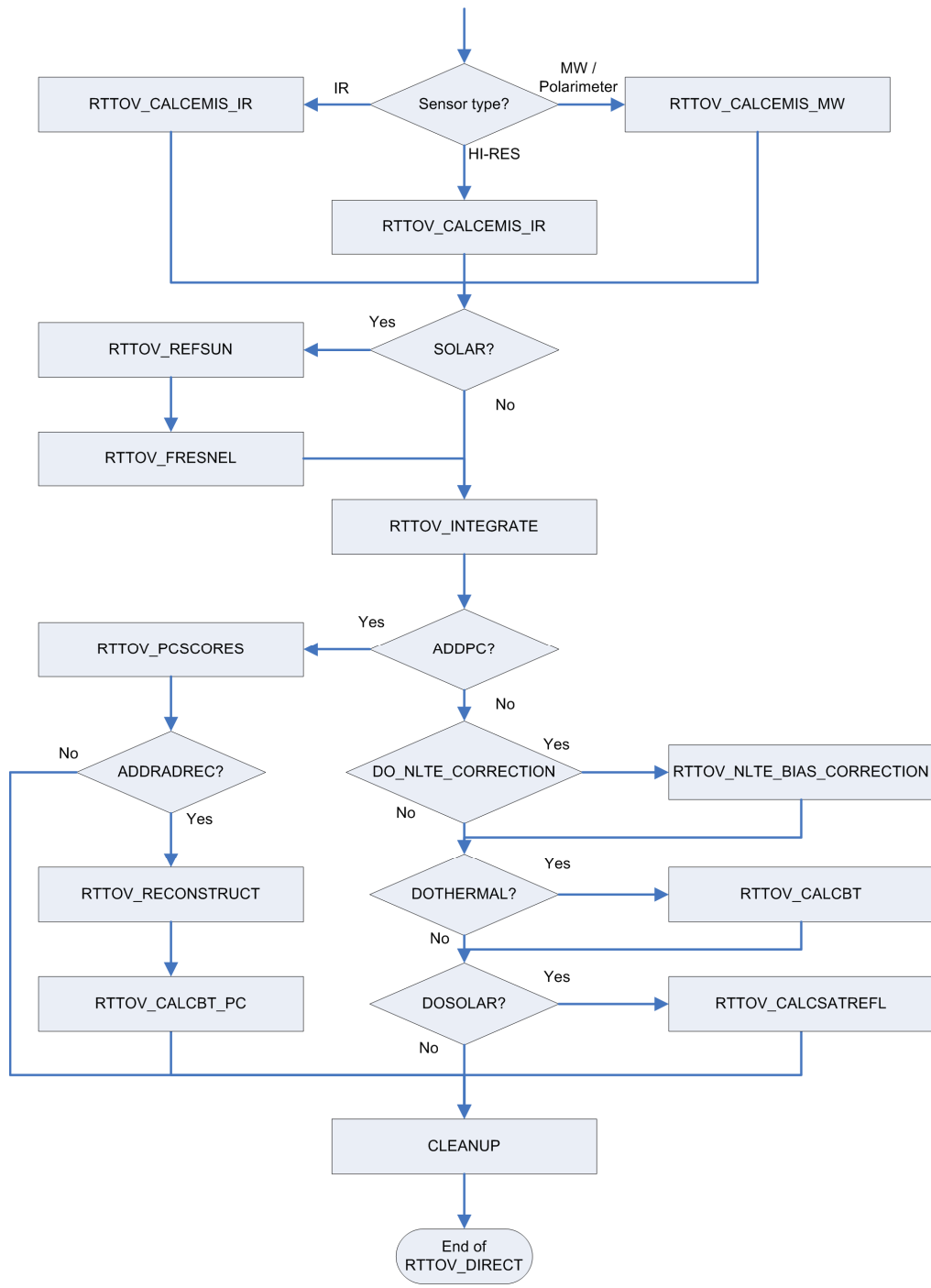
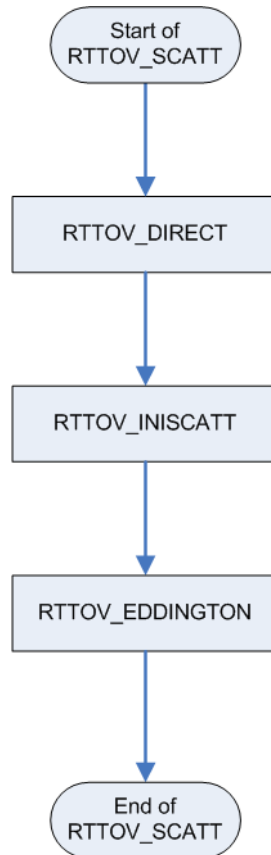


Figure 3. Calling tree for RTTOV\_DIRECT routines

#### 5.4 Calling tree for *rttov\_scatt*



*Figure 4. Calling tree for RTTOV\_SCATT routines*

## 6. References

Aires F., C. Prigent, F. Bernado, C. Jiménez, R. Saunders, and P. Brunel, 2010. A tool to estimate Land Surface Emissivities at Microwaves frequencies (TELSEM) for use in numerical weather production. *Q.J.Royal. Meteorol. Soc.*, **137**, 690-699. doi: 10.1002/qj.803.

Borbas E. and Ruston, B.C. 2010 The RTTOV UWiremis IR land surface emissivity module. Associate Scientist mission report Mission no. AS09-04 Document NWPSAF-MO-VS-042

Matricardi, M. 2003 RTIASI-4, a new version of the ECMWF fast radiative transfer model for the infrared atmospheric sounding interferometer. ECMWF Tech. Memo. 425.

Matricardi, M. 2010 A principal component based version of the RTTOV fast radiative transfer model. ECMWF Tech. Memo. 617.

Vidot J and Borbas E, 2013. Land surface VIS/NIR BRDF atlas for RTTOV-11: Model and Validation against SEVIRI Land SAF Albedo product. *Submitted to Q. J. R. Meteorol. Soc.*