

RTTOV v10 TOP LEVEL DESIGN

NWPSAF-MO-DS-019

Roger Saunders
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 2010, EUMETSAT, All Rights Reserved.

Change record			
Version	Date	Author / changed by	Remarks
0.1	27/10/10	R. Saunders	Initial draft to update
1.0	25/11/10	R. Saunders	Updated prior to delivery
1.1	09/12/10	J. Hocking	Response to DRI comments

1. Introduction and scope

The purpose of this report is to document the main functional elements of RTTOV v10. It shows the modular structure of the software, including the function of each module and the data-flow between modules. A users guide is available for an overview and installation of RTTOV v10 on to a users computer system. Scientific and validation documentation can be found in the RTTOV v10 scientific and validation report published on the NWP-SAF web site on the RTTOV v10 page at:

http://research.metoffice.gov.uk/research/interproj/nwpsaf/rtm/rtm_rtov10.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 v10 design

2.1 RTTOV v10 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 and so the user should study these headers to get details of the interface between modules. An overall routine calling tree is given in Figure 1. The code will be optimised for vector machines and also made threadsafe for MPP platforms. It will be tested at least on the following platforms: IBM power 6, NEC-SX8, SUN, Linux Intel PC, Apple Mac and various compiler options for these which are supplied in the Makefile.

The RTTOV v10 package delivered to the user consists of :

- a self contained library of Fortran90 subroutines and modules
- a perl script to compile 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 v10 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 module *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 form which allow for more efficient input in RTTOV v10 especially for large coefficient files for advanced IR sounders.

RTTOV v10 coefficient files are in a slightly different format than previous versions of RTTOV coefficient files. If the user wants to convert an old RTTOV v7/8/9 coefficient file to RTTOV v10 format then he can run the program *rttov789_conv_coef* to create ascii coefficient files which can be read by the new RTTOV v10 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 v10 are SSIREM for infrared radiances and a model developed by Matricardi (2010) for the PC scores and FASTEM version 4 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 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 (or Kg/Kg is still allowed)*
- 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: $(\text{kg}/\text{m}^2)/\text{s}$
- Aerosol layer mean number density: cm^{-3}
- Angles (solar and satellite): *degrees* (0° - 360°) East = 90°
- Elevation: *km*
- Latitude/Longitude: *degrees* (-90° to $+90^\circ$ / 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

In the following table, **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.

The RTTOV v10 module accepts as input and output the following:

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.
Logical	Intent(in)	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).
Real	Intent(in)	emissivity(nchanprof)	Input emissivities.
Real	Intent(inout)	emissivity_out(nchanprof)	Emissivities used in RTTOV calculations.
Type(transmission_type)	Intent(inout)	transmission	Output transmittances (0 - 1).
Type(radiance_type)	Intent(inout)	radiancedata	Output radiances ($\text{mw}/\text{cm}^{-1}/\text{sr}/\text{m}^2$ & degK).
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 opts %addradrec is .true.

The only input files required are the coefficient files that are assumed to be in the same directory as the program is run or better a link of the same name as the coefficient file. For the clear air code only 1 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 v10 structures is given in the RTTOV v10 user guide.

4. RTTOV modules design overview

The RTTOV v10 model comprises up to 8 modules to interface with the users code, 1 for the coefficient ingest, one for each of the direct model ,TL, AD, K and RTTOV_SCATT codes and 3 utility routines for allocating/deallocating the memory. These are described in more detail below and the interfaces are defined in the RTTOV v10 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 v10 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 Setup functions

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

The modules for the coefficient management are:

- module to read the coefficients
- module to de-allocate the coefficient derived type
- module to allocate and de-allocate the profiles derived type
- module to allocate and de-allocate the radiance derived type

The calling trees are given below in Figure 2 .

To read and initialise coefficients

```
Call rttov_setup (      &  
  & setup_errorstatus, &! out  
  & Err_unit,          &! in  
  & verbosity_level,  &! in  
  & opts,              &! in  
  & coefs(nrttovid),  &! out  
  & instrument)      ! in
```

To set up error message handling

```
Call rttov_errorhandling(Err_unit, verbosity_level)
```

Deallocate the coef structure arrays

```
Call rttov_dealloc_coefs (errorstatus, coefs(nrttovid))
```

Allocate and deallocate profile structure according to asw

```
Call rttov_alloc_prof(      &  
  & errorstatus,          &  
  & nprof,                &  
  & profiles,            &  
  & nlevels,              &  
  & opts,                 &  
  & asw,                  &  
  & coefs = coefs(nrttovid), &  
  & init = .true.       )
```

Allocate and deallocate the radiance structure according to asw

```
Call rttov_alloc_rad (errorstatus,nchannels,radiance,nlevels-1,asw)
```

Allocate transmittance structure

```
Call rttov_alloc_transmission( &  
  & errorstatus,          &  
  & transmission,        &
```

```
& nlevels-1,      &
& nchannels,     &
& asw,           &
& init = .true.)
```

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 de-allocate before return.

The direct model module contains sequentially:

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

The basic call to RTTOV looks like this:

```
call rtov_direct(      &
& rtov_errorstatus, &! out error flag
& chanprof,         &! in channel and profile index structure
& opts,             &! in options structure
& profiles,         &! in profile array
& coefs(nrttovid), &! in coefficients strucutre
& calcemis,        &! in flag for internal emissivity calc
& emissivity_in,   &! in input emissivities per channel
& emissivity_out,  &! out emissivities used by RTTOV per channel
& transmission,    &! inout computed transmittances
& radiance)        ! inout computed radiances
```

Or including all the optional variables:

```
Call rtov_direct( &
& rtov_errorstatus, &! out error flag
& chanprof,         &! in channel and profile index structure
& opts,             &! in options structure
& profiles,         &! in profile array
& coefs(nrttovid), &! in coefficients strucutre
& calcemis,        &! in flag for internal emissivity calc
& emissivity_in,   &! in input emissivities per channel
& emissivity_out,  &! out emissivities used by RTTOV per channel
& transmission,    &! inout computed transmittances
& radiance         & ! inout computed radiances
& traj,            & ! Optional array to hold internal RTTOV variables
& pccomp,          & ! PC scores
& channels_rec,    & ! channels to compute reconstructed radiances
```

rtov_direct calls the following routines (N.B. only main functional routines listed here):

Check PC channels are valid if required.

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

Make quick escape if target values are out of bounds

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

Check profile variables are within limits

```
Call rtov_checkinput( &
  & opts, &
  & traj0%profiles_COEF, &
  & coefs%coef, &
  & coefs%coef_pccomp, &
  & err)
```

Determine cloud top, surface levels, ice cloud parameters

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

Set up common geometric variables for the RT integration

```
Call rtov_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 rtov_setpredictors_7( &
    & opts, &
    & traj0%profiles_COEF, &
    & traj0_sta%angles_COEF, &
    & coefs%coef, &
    & traj0%aux_prof_COEF, &
    & traj0%predictors, &
    & traj0%raytracing_COEF)! inout (in because of mem allocation)
ELSE IF (coefs%coef%fmv_model_ver == 8) THEN
```

```

CALL rtov_setpredictors_8( &
  & opts,          &
  & traj0%profiles_COEF, &
  & traj0%sta%angles_COEF, &
  & coefs%coef,    &
  & traj0%aux_prof_COEF, &
  & traj0%predictors, &
  & traj0%raytracing_COEF)! inout (in because of mem allocation)
ELSE IF (coefs%coef%fmv_model_ver == 9) THEN
CALL rtov_setpredictors_9( &
  & opts,          &
  & traj0%profiles_COEF, &
  & traj0%sta%angles_COEF, &
  & coefs%coef_pccomp, &
  & coefs%coef,    &
  & traj0%predictors, &
  & traj0%raytracing_COEF)! inout (in because of mem allocation)
IF (opts%addsolar) THEN
CALL rtov_setpredictors_9_solar( &
  & opts,          &
  & traj0%profiles_COEF, &
  & coefs%coef,    &
  & traj0%predictors, &
  & traj0%raytracing_COEF)
ENDIF

```

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

```

IF (coefs%coef%fmv_model_ver == 9) THEN
CALL rtov_opdep_9( &
  & coefs%coef%nlayers, &
  & chanprof,          &
  & traj0%predictors, &
  & traj0%aux_prof_COEF, &
  & coefs%coef,    &
  & traj0%opdp_path_COEF, &
  & traj0%sta%opdp_ref_COEF)
IF (opts%addsolar) THEN
CALL rtov_opdep_9_solar( &
  & coefs%coef%nlayers, &
  & chanprof,          &
  & profiles,         &
  & traj0%predictors, &
  & coefs%coef,    &
  & traj0%opdp_path_COEF, &
  & traj0%sta%opdpsun_ref_COEF)
ENDIF
ELSE
CALL rtov_opdep( &
  & coefs%coef%nlayers, &
  & chanprof,          &
  & traj0%predictors, &
  & traj0%aux_prof_COEF, &
  & coefs%coef,    &
  & traj0%opdp_path_COEF, &
  & traj0%sta%opdp_ref_COEF)
ENDIF

```

Now convert optical depths from RTTOV coef levels to user levels

```
CALL rtov_intavg_chan( &
  & opts%lgradp,          &
  & traj0_sta%sun,       &
  & 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( &
  & profiles,            &
  & opts%cldstr_threshold, &
  & traj0%ircld,        &
  & traj0_dyn%nstreams)
```

Calculate optical depths of aerosols and/or clouds

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

Calculate transmittances (also similar routine needed for solar path)

```
CALL rtov_transmit( &
  & opts%addaerosl,      &
  & opts%addclouds,      &
  & profiles(1)%nlayers, &
  & chanprof,            &
  & traj0%aux_prof,      &
  & coefs%coef,          &
  & traj0%ircld,         &
  & traj0%opdp_path,     &
  & traj0_sta%od_level,  &
  & transmission,        &
  & traj0_dyn%transmission_aux, &
  & traj0_dyn%transmission_scatt_ir_stream, &
  & traj0_sta%tau_ref,   &
  & traj0_sta%tau_ref_surf, &
  & traj0_sta%tau_surf,  &
  & traj0_sta%tau_level)
```

Compute IR surface emissivity

```
CALL rtov_calcemis_ir( &
  & profiles,            &
  & traj0_sta%angles,    &
```

```
& coefs%coef,      &
& opts%addpc,      &
& coefs%coef_pccomp, &
& chanprof,        &
& calcemis,        &
& emissivity_out,  &
& errorstatus)
```

Compute MW surface emissivity

```
CALL rtov_calcemis_mw( &
& profiles,          &
& traj0_sta%angles, &
& coefs%coef,       &
& chanprof,         &
& traj0_dyn%transmission_aux, &
& calcemis,         &
& emissivity_out,   &
& traj0%reflectivity, &
& errorstatus)
```

Compute Fresnel reflectance and sun glint

```
CALL rtov_refsun( &
& profiles,          &
& coefs%coef,       &
& traj0%aux_prof,   &
& opts%addsolar,    &
& traj0%sunlint,    &
& traj0%raytracing)
```

```
CALL rtov_fresnel( &
& chanprof,         &
& profiles,         &
& coefs%coef,       &
& opts%addsolar,    &
& traj0%sunlint,    &
& traj0%fresnrefl)
```

Integrate the radiative transfer equation

```
CALL rtov_integrate( &
& addcosmic,        &
& opts,             &
& traj0_dyn%nstreams, &
& chanprof,         &
& emissivity_out,   &
& traj0%reflectivity, &
& traj0%fresnrefl,  &
& traj0%sunlint,    &
& traj0_dyn%transmission_aux, &
& traj0_dyn%transmission_scatt_ir_stream, &
& profiles,         &
& traj0%aux_prof,   &
& coefs%coef,       &
& traj0%raytracing, &
& traj0%ircl,       &
& radiancedata,     &
& 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( &
  & traj0_sta%chanprof_in, &
  & traj0_sta%chanprof_pc, &
  & pccomp,                &
  & coefs%coef_pccomp)
```

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. radiance_d1\%tb = cloud-affected Tbs $\text{radiance_d1\%tb_clear}$ = clear-sky Tbs.

```
Call rtov_scatt( &
  & errorstatus, & ! out error flag
  & nlevels,    & ! in number of levels
  & chanprof,   & ! In channel and profile index structure
  & frequencies, & ! In MW channel frequencies
  & profiles,   & ! In Profile structure
  & cld_profiles, & ! In Cloud_Profile structure
  & coef_rtov,  & ! In Clear air RTTOV coeffs
  & coef_scatt, & ! In RTTOV_SCATT coeffs (mie table)
  & calcemiss, & ! In logical array for surface emissivity calculation
  & emissivity_in, & ! Inout input surface emissivity
  & radiance,   & ! Inout Output radiance structure)
```

rtov_scatt calls:

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

```
Call rtov_direct( & !
  & errorstatus, & ! out
  & chanprof,   & ! in
  & opts,       & ! in
  & profiles,   & ! in
  & coef_rtov,  & ! in
  & calcemiss,  & ! in
  & emissivity_in, & ! in
  & emissivity_out, &
  & transmission, & ! inout)
```

& radianc) *! 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
  & usenewcld, &! in
  & userfrac, &! 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 modules. They are all delivering 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 RTTOV10

Flow diagrams for a user program calling RTTOV v10 and for the main *rtov_setup* and *rtov_direct* routines are given below to show all subroutines that are called.

5.1 User calling tree for user program calling RTTOV10

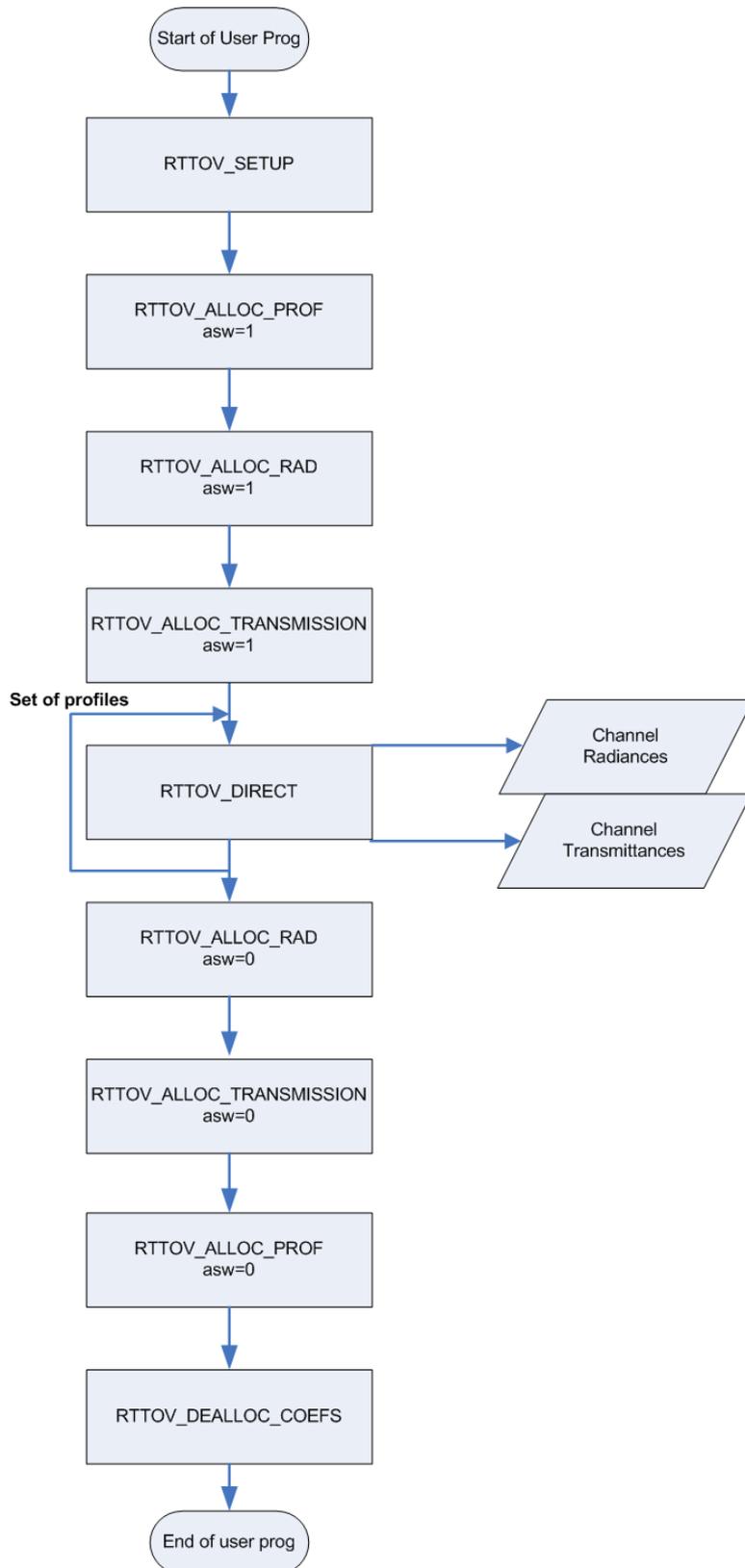


Figure 1. Process diagram of user program calling RTTOV v10 forward model.

5.2 Calling tree for *rttov_setup*

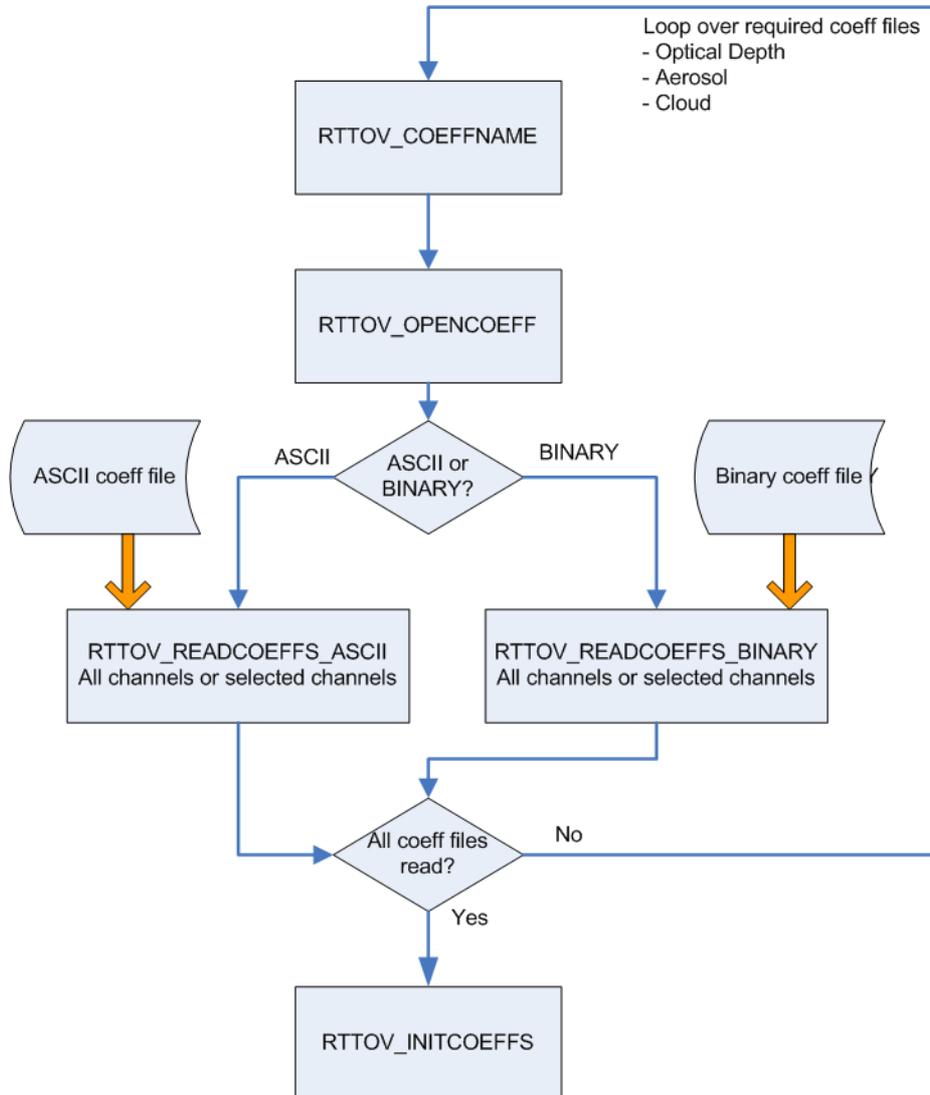
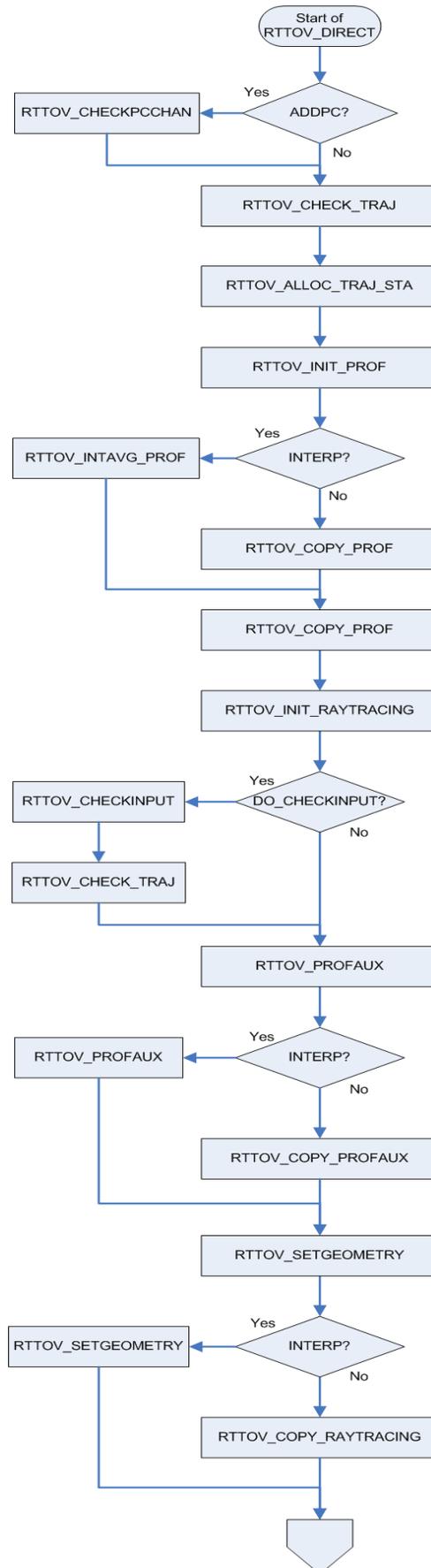


Figure 2. Calling tree for RTTOV_SETUP routines

5.3 Calling tree for *rttov_direct*

See next 2 pages



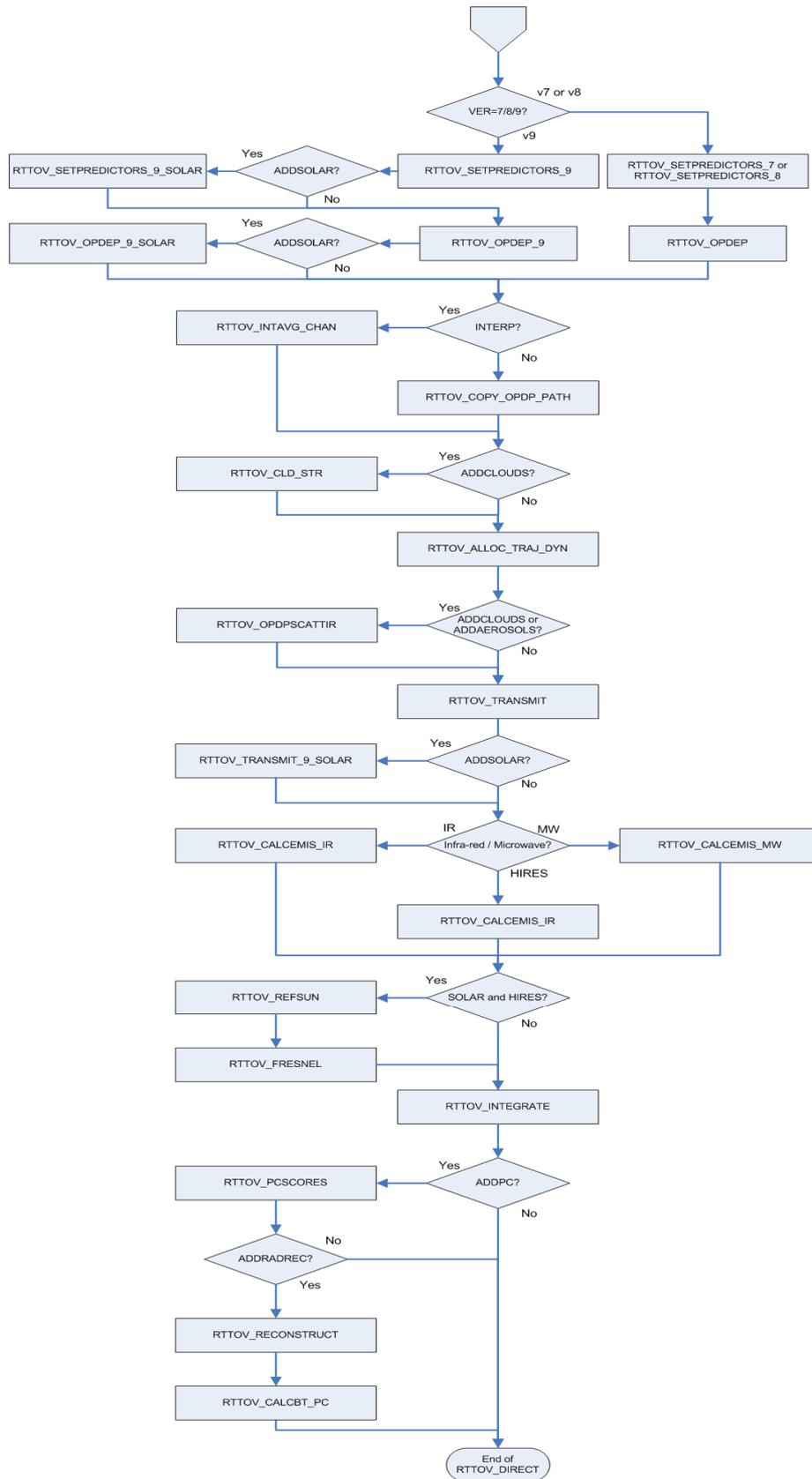


Figure 3. Calling tree for RTTOV_DIRECT routines

5.4 Calling tree for *rttov_scatt*

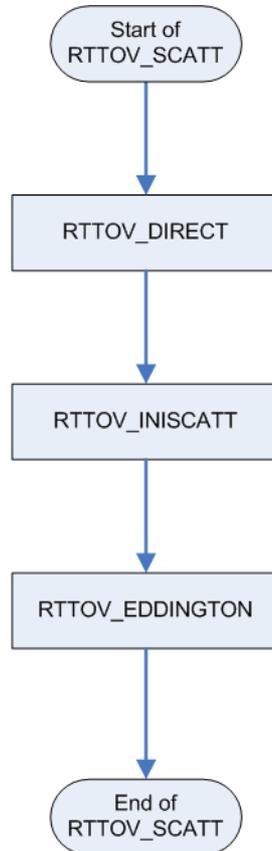


Figure 4. Calling tree for RTTOV_SCATT routines

6. References

Aires, F., C. Prigent, F. Bernardo, C. Jimenez, R. Saunders, P. Brunel 2010 A Tool to Estimate Land Surface Emissivities at Microwaves frequencies (TELSEM) for use in numerical weather prediction schemes. Submitted to Quart. J. Roy. Meteorol. Soc., 2010.

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