

RTTOV-9 TOP LEVEL DESIGN

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 2008, EUMETSAT, All Rights Reserved.

Change record			
Version	Date	Author / changed by	Remarks
0.1	29/06/07	R. Saunders	Initial draft to update
1.0	19/02/08	R. Saunders	Updated prior to delivery

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

1. Introduction and scope

The purpose of this report is to document the main functional elements of RTTOV-9. 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-9 on to a users computer system. Scientific and validation documentation can be found in the RTTOV-9 scientific and validation report published on the NWP-SAF web site on the RTTOV-9 page at:

<http://www.metoffice.gov.uk/research/interproj/nwpsaf/rtm/> 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-9 design

2.1 RTTOV-9 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 on the following platforms: DELL PC, AMD Athlon, SGI, IBM, NEC, SUN, and various compiler options for these are supplied in the Makefile.

The RTTOV-9 package delivered to the user consists of :

- a self contained library of Fortran90 subroutines, interface and modules
- a makefile to compile the code
- a set of main programs for testing various aspects
- a set of testing scripts for testing various aspects of the code
- a set of coefficient files (note AIRS and IASI coefficients are not included 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-9 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...). 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.

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

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. He calls the RTTOV coefficient file ingest module *rttov_read_coef* 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_coef* reads the data and allocates memory for the storage. *rttov_dealloc_coef* will allow the user to de-allocate memory for the coefficients. The *rttov_ascii2bin_coef* module enables the user to create a coefficient set in a binary form which allow for more efficient input in RTTOV-9 especially for large coefficient files for advanced IR sounders.

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 indices for each calculated radiance. The user has to provide a list of channel indices for each calculated radiance.

2.2.4 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.

2.2.5 Emissivities

The model is able to compute IR (over ocean) and MW emissivities (over ocean and approximately over 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 emissivity models used in RTTOV-9 are SSIREM (IR) and FASTEM version 3 (MW). The emissivities computed by the model are returned in the output arrays from RTTOV.

2.2.6 Physical units

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

- Pressure: *hPa*
- Atmospheric temperature: *Kelvin*
- Radiances: *mW/cm⁻¹/ster/m²*
- Equivalent black body brightness temperature: *Kelvin*
- 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/m³*
- rain, solid precipitation: *kg/kg*
- Aerosol layer mean number density: *cm⁻³*
- Angles: *degrees (0°-360°) East = 90°*.
- Elevation: *km*
- Latitude: *degrees (-90° to +90°)*
- Wind fetch: *km*

2.2.7 Input and output

The *rttov-9* module accepts as input:

- The total number of calculated brightness temps or radiances per call
(ie `channels × nprof`) = `nchannels`

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

- The number of profiles per call - `nprofiles`
- The set of required channel indices (should match coeff file indices) - `channels`
- List of profile indices for each channel and profile (size `nchannels`)- `lprofiles`
- Input atmospheric profiles - `profiles` (size `nprofiles`)
- a set of surface emissivities (optional) (size `nchannels`) - `emissivity`
- a logical array `calcemiss` for selecting internal emissivity calculation if true (size `nchannels`)
- Logical flag to interpolate user profile levels to coefficient levels - `addinterp`
- Logical flag to allow aerosol optical depth computation - `laerosl`
- Logical flag to allow IR cloud optical depth computation - `lclouds`
- Array of coefficients for clear sky optical depth calculations - `coef`
- Array of coefficients for IR cloudy sky optical depth calculations - `coef_scatt_ir`
- Array of scattering optical depth properties - `optp`

Output parameters:

- Error status returned - `rttov_errorstatus(nprofiles)`
- Transmission computed including
 - surface to space transmittances (`nchannels`)
 - layer to space transmittances (`levels, nchannels`)
- Top of atmosphere radiances computed including:
 - Clear sky radiances and brightness temperatures (`nchannels`)
 - Clear + cloudy (all sky) radiances and brightness temps (`nchannels`)
 - Clear sky downwelling radiance (`nchannels`)
 - Reflected clear sky downwelling radiance (`nchannels`)
 - Clear sky radiance without reflection term (`nchannels`)
 - Overcast radiances at cloud top specified by user (`nchannels`)
 - Overcast radiance from cloud tops placed at on each level (`levels, nchannels`)
 - + others documented in `rttov_types.F90` and users guide.
- Surface emissivity computed internally by ISEM or FASTEM (`nchannels`)

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.

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-9 structures is given in Annex K of the RTTOV-9 user guide.

4. RTTOV modules design overview

The RTTOV-9 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-9 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-9 there are three

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

options for predictors (so called RTTOV-7, RTTOV-8 and RTTOV-9 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 deallocate the radiance derived type

The calling trees are given below in Figure 2 .

```
Call rttov_setup (&
  & setup_errorstatus, &! out
  & Err_unit, &! in
  & verbosity_level, &! in
  & nrttovid, &! in
  & laerosl, &! in
  & lclouds, &! in
  & coef, &! out
  & coef_scatt_ir, &! out
  & optp, &
  & instrument, &! in
  & ichan ) ! in Optional
```

Calls routines below to simplify things for user so user should only need to call above unless more complex setups are required (e.g. to specify coeff directory as well as name).

```
rttov_error_report( errorstatus(inst), errMessage, NameOfRoutine )
```

Sets up error message handling

```
rttov_readcoeffs (&
  & errorstatus, &! out
  & aerosl, &
  & clouds, &
  & coef, &! out
  & coef_scatt_ir, &! out
  & optp, &
  & instrument, &! in Optional
  & kmyproc, &! in Optional
  & kioproc, &! in Optional
  & file_id, &! in Optional
  & file_id_scatt_ir, &! in Optional
  & file_id_optp, &! in Optional
  & channels ) ! in Optional
```

Read an ASCII or binary coefficient file and allocates coef structure arrays

```
rttov_initcoeffs (
  & errorstatus, &! out
  & laerosol, &! in
  & lcloud, &! in
  & coef, &! inout
  & coef_scatt_ir, &! inout
```

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

```

& optp,      &! Inout
& knproc,    &! in Optional
& kmyproc,   &! in Optional
& kioproc    )! in Optional

```

Initialises coefficient files.

```

rttov_dealloc_coef(errorstatus, coef, coef_scatt_ir ,optp)

```

Deallocate the coef structure arrays

```

rttov_alloc_prof(errorstatus,nprof,profiles,nlevels, &
& coef_scatt_ir,asw,addaerosl,addclouds,init,lob)

```

Allocate and deallocate profile structure according to asw

```

rttov_alloc_rad(errorstatus,nchannels,radiance,coef,asw)

```

Allocate and deallocate the radiance structure according to asw

Also supplied :

```

rttov_writecoef (&
& errorstatus, &! out
& coef,      &! in
& coef_scatt_ir, &! in
& optp ,     &! in
& file_id,   &! in
& lbinary,   &! in Optional
& aerosl,    &! in Optional
& clouds    )! in Optional

```

Write the coefficient coef structure to file in binary or ASCII form according to the logical lbinary.

4.2 RTTOV (Forward Model) main functions

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 predictors
- modules to calculate optical depth and transmittances
- modules to calculate surface emissivity
- module to interpolate optical depths on to user levels
- module to integrate the radiative transfer equation

```

rttov_direct( &
& errorstatus, &! out
& nprofiles,   &! in
& nchannels,   &! in
& channels,    &! in
& lprofiles,   &! in
& addinterp,   &! in
& profiles,    &! in
& coef,        &! in
& coef_scatt_ir, &
& optp,        &
& addsun,      &! in
& addaerosl,   &! in
& addclouds,   &! in
& calcemis,    &! in
& emissivity,  &! inout
& transmission, &! inout
& radiancedata ) ! inout

```

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

Perform the radiative transfer calculations for one coefficient set and nprofiles. Allocates memory for all local structures and de-allocate before return. *rttov_direct* calls the following routines:

```
Call rttov_intavg_prof( &
  & nprofiles,           &
  & profiles(1)%nlevels, & !in
  & profiles_COEF(1)%nlevels, & !in
  & profiles,           & !in  source variables
  & profiles_COEF,     & !inout target variables
  & errorstatus)      !out errorstatus
```

Interpolate user profile levels to coefficient levels

```
rttov_checkinput( &
  & profiles( i ),    & !in
  & coef,             & !in
  & errorstatus(i) ) !out
```

One profile at a time (external loop on profiles) checks the input profile on COEF levels according to the limits given by the coefficient set.

```
rttov_profaux( &
  & addclouds,       &
  & profiles(i),    & ! in
  & coef,           & ! in
  & aux_prof(i))  ! inout
```

One profile at a time (external loop on profiles) calculates some variables related to input profile, like nearest profile level for surface, or Debye terms (MW only). This has to be done on user and coef levels.

```
Call rttov_alloc_raytracing( alloc_status(1), nprofiles, raytracing, profiles(1)%nlevels, 1_jpim, blob =
raytracing_blob )
```

Allocates space for geometrical path arrays

```
Call rttov_alloc_predictor( alloc_status(1), nprofiles, predictors, coef, 1_jpim, profiles(:)%addsolar,
blob = predictors_blob )
```

Allocates space for predictor arrays

```
rttov_setgeometry(&
  & nprofiles,    & ! in
  & profiles,    & ! in
  & aux_prof,    & ! in
  & coef,        & ! in
  & angles,      & ! out
  & raytracing ) ! out
```

Calculates the auxiliary path geometry variables according to the input profile.

```
rttov_setpredictors_7( &
  & nprofiles,    & ! in
  & profiles(i),  & ! in
  & angles(i),    & ! in
  & coef,         & ! in
  & aux_prof(i),  & ! in
  & predictors(i), & ! inout (in because of mem allocation)
```

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

& raytracing(i)) ! inout

or *rttov_setpredictors_8, rttov_setpredictors_9, rttov_setpredictors_9_solar*

One profile at a time (external loop on profiles) calculates the RTTOV-7 or 8 or 9 predictors according to the input profile and geometry and optionally for RTTOV-9 predictors for the solar path also.

rttov_alloc_opdp_path(alloc_status(1),opdp_path,profiles(1)%nlevels, nchannels,1_jpim, blob = opdp_path_blob)

Allocates space for optical depth parameters

```
rttov_opdep(      &
  & nchannels,      &! in
  & nprofiles,      &! in
  & coef%nlayers,   &! in
  & channels,       &! in
  & lprofiles,      &! in
  & predictors,     &! in
  & aux_prof_COEF,  &! in
  & coef,           &! in
  & ircl,           &!
  & opdp_path_COEF, &! inout
  & opdp_ref_COEF) ! inout
```

Computes clear sky optical depths or *rttov_opdep9* or *rttov_opdep9_solar*.

```
rttov_intavg_chan( &
  & nchannels,      & !in
  & sun,            & !in
  & profiles_COEF(1)%nlevels, & !in
  & profiles(1)%nlevels, & !in
  & nprofiles,      &
  & lprofiles,      &
  & profiles_COEF,  & !inout
  & profiles,       & !inout only p and addsolar needed
  & opdp_path_COEF, & !inout source variable
  & opdp_path,      & !inout target variable
  & errorstatus)   !out
```

Interpolates optical depths on coef levels to user levels.

Call *rttov_alloc_ircl(alloc_status(1), nprofiles, ircl, profiles(1)%nlevels, 1_jpim, addaerosl, addclouds, blob = ircl_blob)*

Allocate space for IR cloudy radiance calculations.

```
rttov_cldstr(&
  & nprofiles,      &! in
  & profiles,       &! in
  & coef,           &! in
  & ircl,           &! out
  & addclouds,     &! in
  & nstreams)      ! out
```

For scattering calculations compute the number of streams, the area coverage of each stream and the cloud distribution in each stream.

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

Call *rttov_alloc_trans_scatt_ir(alloc_status(1), transmission_scatt_ir, nchannels, profiles(1) %nlevels, 1_jpim)*

Allocate space for scattering IR optical depths

```
rttov_opdpsscattir(
    & nchannels      , &
    & nprofiles     , &
    & profilesf%nlevels , &
    & channels      , &
    & lprofiles     , &
    & aux_prof      , &
    & profiles      , &
    & coef          , &
    & coef_scatt_ir , &
    & raytracing    , &
    & transmission_scatt_ir , &
    & transmission_scatt_ir_stream, &
    & optp          , &
    & ircl          , &
    & addaerosl     , &
    & addclouds)
```

Calculates the scattering optical depths for IR cloud and aerosol scattering.

Call *rttov_alloc_transmission(alloc_status(1), transmission, profiles(1) % nlevels, nchannels, & 1_jpim, nstreams = nstreams, calc = .true.)*

Allocate transmission arrays

```
rttov_transmit(
    & addaerosl,    &! in
    & addclouds,   &! in
    & nchannels,   &! in
    & nprofiles,   &! in
    & profiles(1)%nlevels, &! in
    & channels,    &! in
    & lprofiles,   &! in
    & aux_prof,    &! in
    & coef,        &! in
    & ircl,        &! in
    & opdp_path,   &! in
    & od_layer,    &! out
    & transmission, &! out
    & transmission_scatt_ir_stream, &! out
    & tau_ref,     &! out
    & tau_ref_surf, &! out
    & tau_surf,    &! out
    & tau_layer)
```

To calculate transmittances for a number of channels and profiles from every pressure level to space. Also *rttov_transmit_9* and *rttov_transmit_9_solar* can be called specifically for rtov-9 simulations.

```
rttov_calcemis_ir( &
    & profiles,    &! in
    & angles,      &! in
    & coef,        &! in
    & nchannels,   &! in
```

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

```

& nprofiles,      &! in
& channels,      &! in
& lprofiles,     &! in
& calcemis,     &! in
& emissivity )  ! inout

```

Calculation of IR surface emissivity using SSIREM.

```

rttov_calcemis_mw( &
  & profiles,      &! in
  & angles,       &! in
  & coef,         &! in
  & nchannels,    &! in
  & nprofiles,    &! in
  & channels,     &! in
  & lprofiles,    &! in
  & transmission, &! in
  & calcemis,     &! in
  & emissivity,   &! inout
  & reflectivity, &! out
  & errorstatus ) ! inout

```

Calculation of MW surface emissivity using FASTEM.

```

rttov_refsun( &
  & nprofiles,    &
  & profiles,     &
  & coef,         &
  & aux_prof,     &
  & sunglint,     &
  & raytracing)

```

Compute the fraction of solar radiance that is reflected by a wind roughened water surface.

```

rttov_fresnel(&
  & nchannels,    &
  & channels,     &
  & nprofiles,    &
  & lprofiles,    &
  & profiles,     &
  & coef,         &
  & sunglint,     &
  & fresnrefl)

```

Compute fresnel reflection of sea surface for solar beam

```

rttov_integrate( addcosmic,      &! in
  & addaerosl,                   &! in
  & addclouds,                   &! in
  & nchannels,                   &! in
  & nprofiles,                   &! in
  & nstreams,                    &! in
  & angles,                      &! in
  & channels,                    &! in
  & lprofiles,                   &! in
  & emissivity,                 &! in
  & reflectivity,               &! in

```

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

```

& fresnrefl,      &! in
& sunglint,      &! in
& transmission,  &! in
& transmission_scatt_ir_stream, &
& profiles,      &! in
& aux_prof,      &! in
& coef,          &! in
& raytracing,    &! in
& ircl,          &! in
& radiancedata,  &! inout
& auxrad ,       &
& auxrad_stream )! inout

```

Integration of the radiative transfer equation for the list of channels according to the profiles, geometry and transmittances.

```

rttov_calcrad( &
& addcosmic, &!in
& nchannels, &!in
& nprofiles, &!in
& channels, &!in
& lprofiles, &!in
& profiles, &!in
& coef, &!in
& auxrad%cosmic, &!out
& auxrad%skin, &!out
& auxrad%surfair, &!out
& auxrad%layer )!ou

```

To calculate layer radiances.

```

rttov_calcbt( &
& nchannels, &! in
& channels, &! in
& coef, &! in
& rad )! inout

```

To compute brightness temperature from radiance.

```

rttov_ErrorReport (errorstatus(inst), errMessage, NameOfRoutine)
Report any errors.

```

4.3 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

```

rttov_scatt
& errorstatus, &! out
& klevg, &! in
& nchannels, &! in
& kproma, &! in

```

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

```

& channels,      &! in
& frequencies,  &! in
& lprofiles,    &! in
& profiles_d1,  &! inout
& cld_profiles_d1, &! in
& coef_rttov,  &! in
& coef_scatt,  &! in
& calcemiss,   &! in
& emissivity_d1, &! inout
& radiance_d1, &! inout
& cfrac )      ! out, diagnostic only

```

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 radiance_d1%tb_clear = clear-sky Tbs.

rttov_scatt calls:

rttov_direct

Perform the radiative transfer calculations for one coefficient set and nprofiles.

rttov_iniscatt

```

& errorstatus, &! out
& nlevels,     &! in
& nchannels,   &! in
& nprofiles,   &! in
& channels,    &! in
& frequencies, &! in
& lprofiles,   &! in
& profiles,    &! in
& cld_profiles, &! in
& coef_rttov,  &! in
& coef_scatt,  &! in
& transmission, &! in
& calcemiss,   &! in
& angles,      &! out
& scatt_aux)   ! inout

```

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

rttov_eddington

```

& nlevels,     &! in
& nchannels,   &! in
& nprofiles,   &! in
& lprofiles,   &! in
& angles,      &! in
& profiles,    &! in
& scatt_aux,   &! in
& cld_bt)

```

To compute Eddington approximation to RT.

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

NWP SAF	RTTOV9 Top Level Design	Doc ID : NWPSAF-MO-DS-016 Version : 1.0 Date : 19/2/08
----------------	--------------------------------	--

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 RTTOV9.1

Flow diagrams for a user program calling RTTOV-9 and for the main *rtsetup* and *rttov_direct* routines are given below to show all subroutines that are called.

5.1 User calling tree for user program calling RTTOV9.1

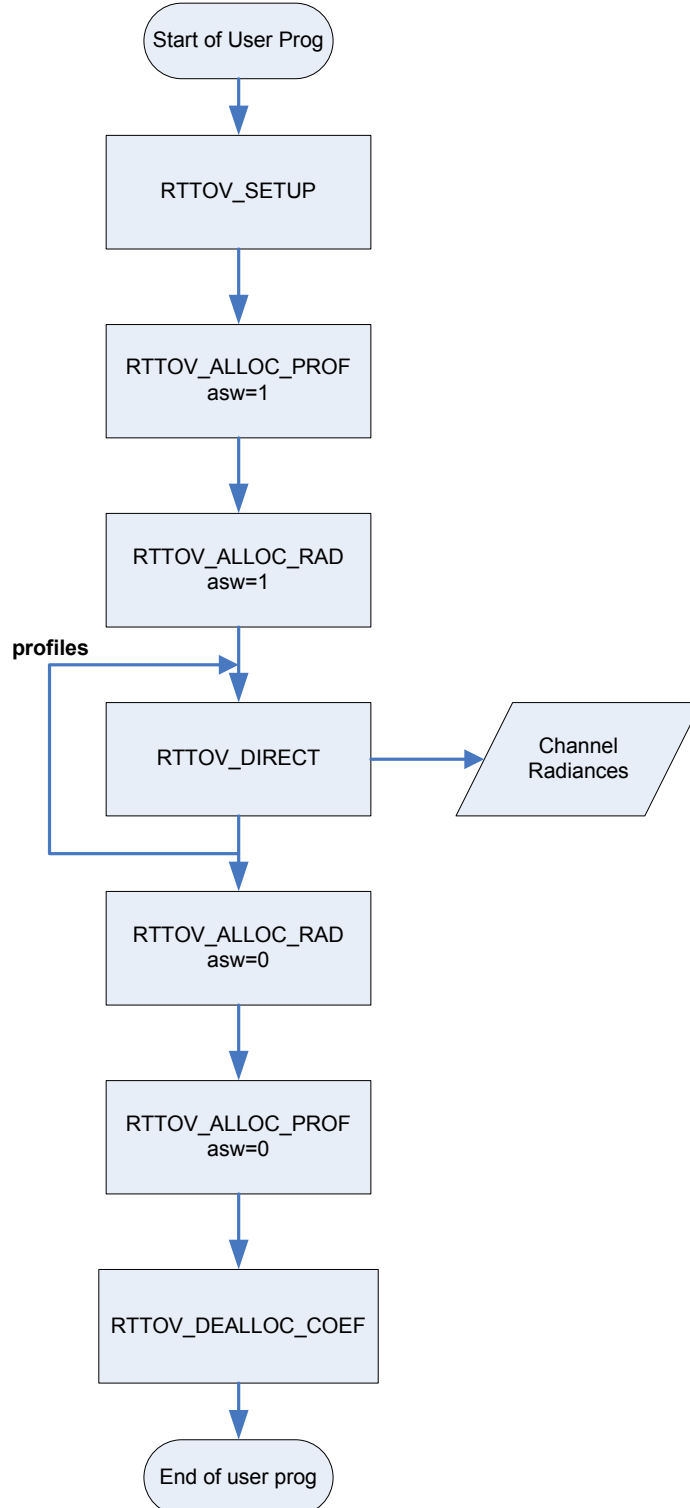


Figure 1. Logical sequence of RTTOV-9 routines to be called in user program

5.2 Calling tree for *rttov_setup*

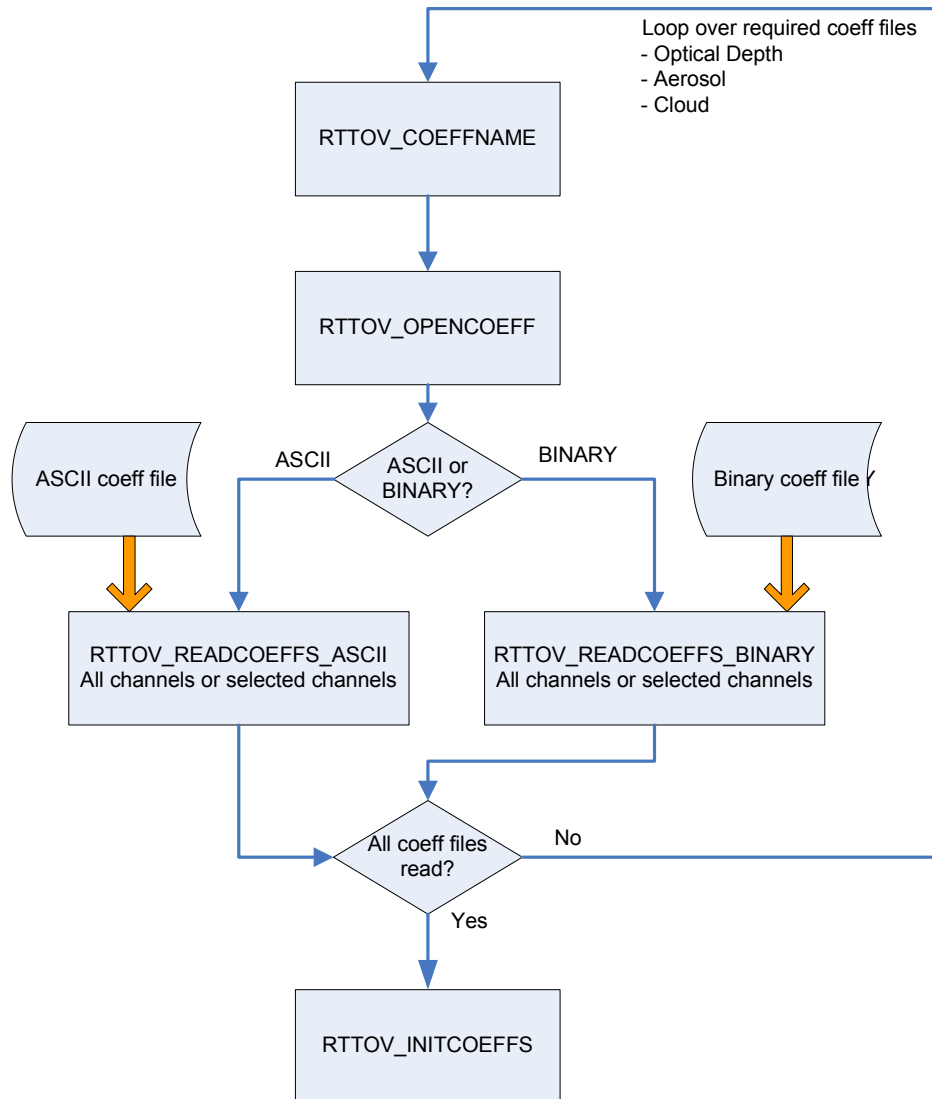
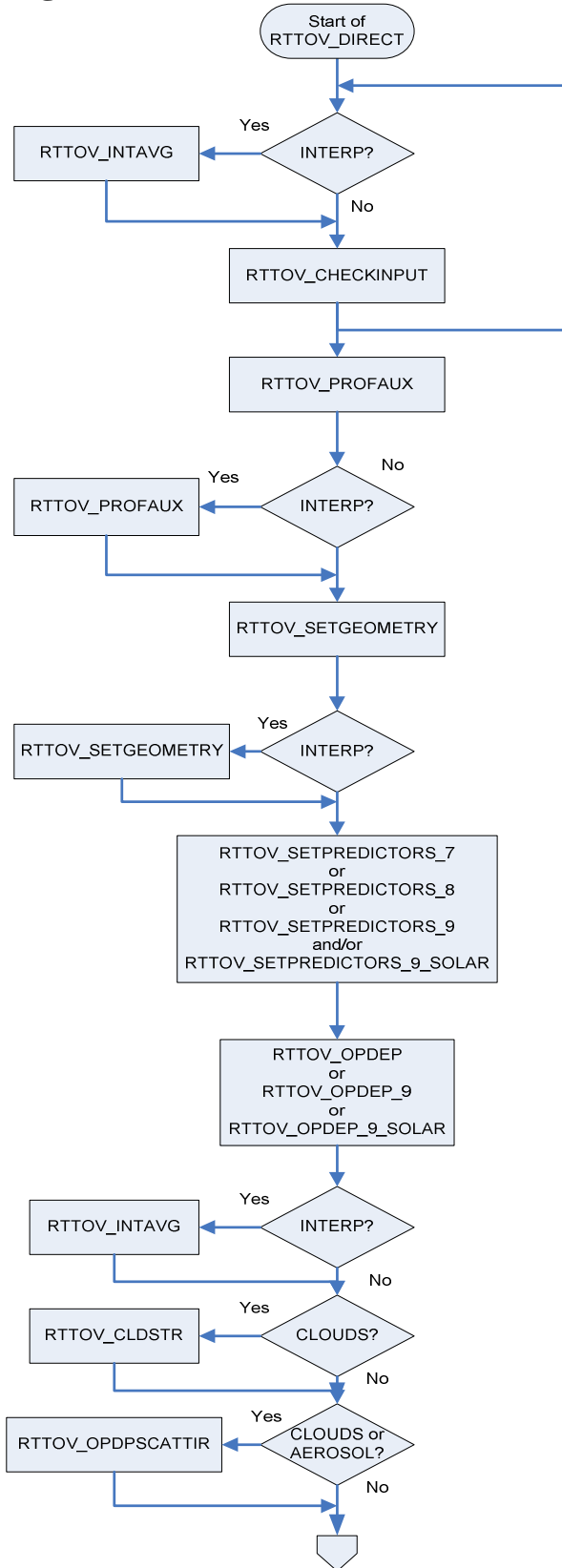


Figure 2. Calling tree for *RTTOV_SETUP* routines

5.3 Calling tree for *rttov_direct*



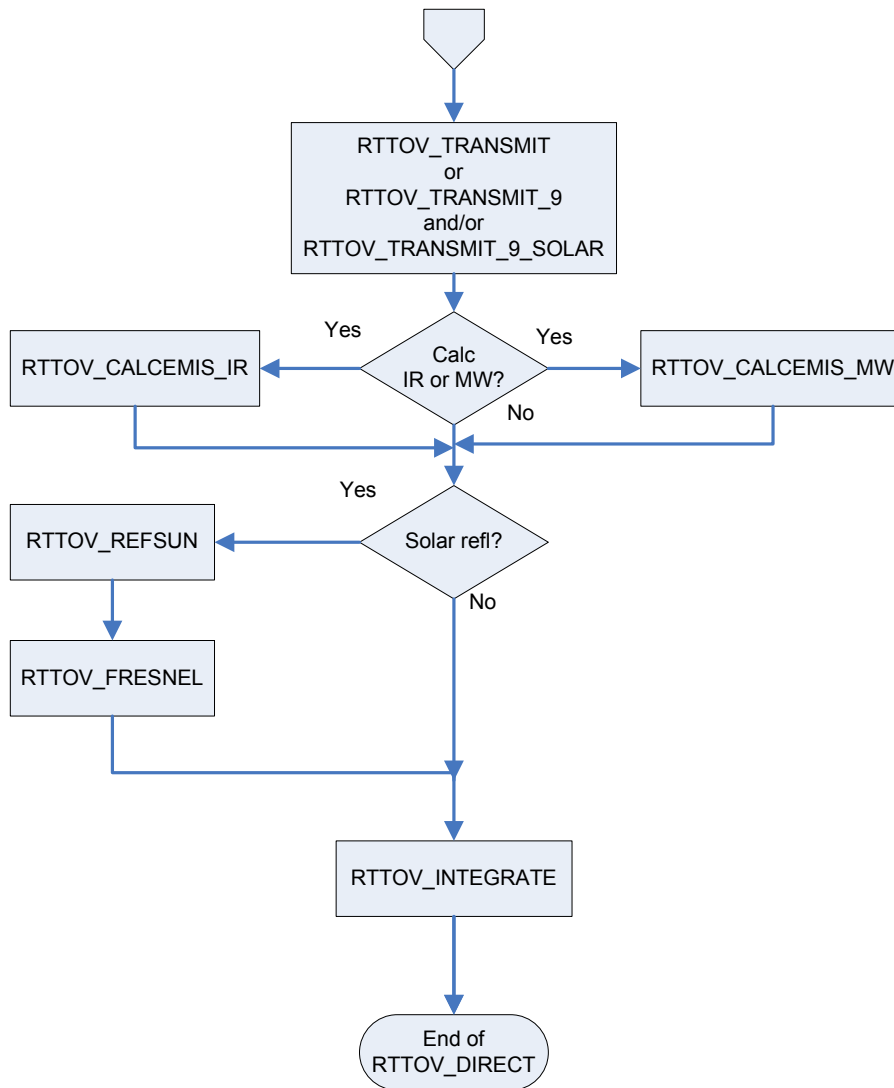


Figure 3. Calling tree for RTTOV_DIRECT routines

5.4 Calling tree for *rttov_scatt*

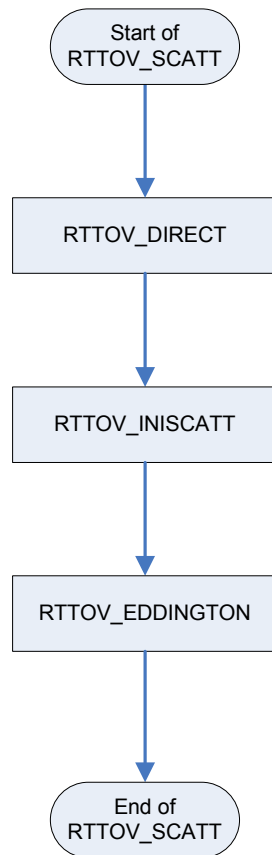


Figure 4. Calling tree for RTTOV_SCATT routines