

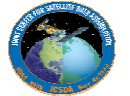


JCSDA Community Radiative Transfer Model (CRTM) Framework



Paul van Delst¹, Yong Han², and Quanhua Liu³
¹CIMSS@NOAA/NCEP/EMC, University of Wisconsin – Madison
²NOAA/NESDIS
³QSS Group, Inc

Joint Center for Satellite Data Assimilation (JCSDA)
Camp Springs MD USA



JCSDA Mission

- Accelerate and improve the quantitative use of research and operational satellite data in weather and climate prediction models

JCSDA Goals

- Reduce from two years to one year the average time for operational implementation of new satellite technology
- Increase uses of current satellite data in NWP models
- Advance the common NWP models and data assimilation infrastructure
- Assess the impacts of data from advanced satellite sensors on weather and climate predictions

What is the CRTM Framework?

- At the simplest level, it's a collection of structure definitions, interface definitions, and stub routines.
- There are User and Developer interfaces, as well as Shared Data interfaces and I/O.
- More detailed information, as well as source code and test data files, is available from:

<http://cimss.ssec.wisc.edu/~paulv/CRTM>

Why do this?

- The radiative transfer problem is split into various components (e.g. gaseous absorption, scattering etc). Each component defines its own structure definition and application modules to facilitate independent development.
- Want to minimise or eliminate potential software conflicts and redundancies.
- Components developed by different groups can "simply" be dropped into the framework. This is an ideal characterisation, as there may be dependencies between components.
- Faster implementation of new science and algorithms.

User Interface

Current Forward CRTM Interface

```
Error_Status = CRTM_Forward( Atmosphere, &
                             Surface, &
                             GeometryInfo, &
                             ChannelInfo, &
                             RTSolution )
```

- All data contained in structures.
- Additional "arguments" can be added as required to the requisite structures – no impact on calling routine.

Allowable dimensionality

L = number of channels; M = number of profiles

INPUTS	OUTPUTS
Atmosphere Surface GeometryInfo	RTSolution
Scalar	L
M	LxM

PUBLIC CRTM Data Structures

Type Name	Description
CRTM_ChannelInfo_type	Sensor channel information filled during initialisation.
CRTM_Atmosphere_type	Atmospheric state profile data. Contains Cloud and Aerosol structures.
CRTM_Surface_type	Surface type and state information. Contains SensorData structure.
CRTM_GeometryInfo_type	Earth location, zenith and azimuth angles.
CRTM_RTSolution_type	Radiative transfer results.

Example: Definition of Atmosphere Structure

```
TYPE, PUBLIC :: CRTM_Atmosphere_type
! -- Dimension values
INTEGER :: Max_Layers = 0 ! K dimension
INTEGER :: n_Layers = 0 ! Kuse dimension
INTEGER :: n_Absorbers = 0 ! J dimension
INTEGER :: Max_Clouds = 0 ! Nc dimension
INTEGER :: n_Clouds = 0 ! NcUse dimension
INTEGER :: Max_Aerosols = 0 ! Na dimension
INTEGER :: n_Aerosols = 0 ! NaUse dimension
! -- Climatology model associated with the profile
INTEGER :: Climatology = INVALID_MODEL
! -- Absorber ID and units
INTEGER, DIMENSION( : ), POINTER :: Absorber_ID => NULL() ! J
INTEGER, DIMENSION( : ), POINTER :: Absorber_Units => NULL() ! J
! -- Profile LEVEL pressure and LAYER quantities
REAL( fp_kind ), DIMENSION( : ), POINTER :: Level_Pressure => NULL() ! K
REAL( fp_kind ), DIMENSION( : ), POINTER :: Pressure => NULL() ! K
REAL( fp_kind ), DIMENSION( : ), POINTER :: Temperature => NULL() ! K
REAL( fp_kind ), DIMENSION( :, : ), POINTER :: Absorber => NULL() ! K x J
! -- Clouds associated with each profile
TYPE( CRTM_Cloud_type ), DIMENSION( : ), POINTER :: Cloud => NULL() ! Nc
! -- Aerosols associated with each profile
TYPE( CRTM_Aerosol_type ), DIMENSION( :, : ), POINTER :: Aerosol => NULL() ! Na
END TYPE CRTM_Atmosphere_type
```

Developer Interface

The CRTM Components

- Absorption by atmospheric gaseous constituents, e.g. water vapour, ozone, etc. **AtmAbsorption** functions.
 - Compact-OPTRAN is currently used.
 - OPTRAN-v7 has been implemented.
 - OSS has been implemented.
- Scattering and absorption. **AtmScatter** functions.
 - Aerosols
 - Clouds
- Surface Optics. **SfcOptics** functions.
 - Emissivity (land, ocean; μ W, IR; ice, snow, water, etc)
 - Reflectivity (diffuse and direct)
- Radiative Transfer. **RTSolution** functions.
 - Fixed- and variable-angle multi-stream models

INTERNAL CRTM Data Structures

- Not visible via the User Interface
- Developers modify the structure contents as needed
- Some components are mandatory and must be supplied; others are algorithm specific.

Type Name	Description
CRTM_AtmsAbsorption_type	Gaseous absorption optical depths and related parameters.
CRTM_AtmsScatter_type	Scattering parameters such as single scatter albedo, asymmetry factor, optical depths, etc.
CRTM_SfcOptics_type	Surface optical properties such as emissivity and reflectivity.

Example: Definition of AtmsScatter Structure

```
TYPE, PUBLIC :: CRTM_AtmsScatter_type
! -- Dimension values
INTEGER :: n_Layers = 0 ! K dimension
INTEGER :: Max_Legendre_Terms = 0 ! lc dimension
INTEGER :: n_Legendre_Terms = 0 ! lcUse dimension
INTEGER :: Max_Phase_Elements = 0 ! lp dimension
INTEGER :: n_Phase_Elements = 0 ! lpUse dimension

! - Algorithm-specific members
REAL( fp_kind ), DIMENSION( :, :, : ), POINTER :: Phase_Coefficient => NULL()

! - Mandatory members
REAL( fp_kind ), DIMENSION( : ), POINTER :: Optical_Depth => NULL() ! K
REAL( fp_kind ), DIMENSION( : ), POINTER :: Single_Scatter_Albedo => NULL() ! K
REAL( fp_kind ), DIMENSION( : ), POINTER :: Asymmetry_Factor => NULL() ! K
REAL( fp_kind ), DIMENSION( :, : ), POINTER :: Delta_Truncation => NULL() ! K
END TYPE CRTM_AtmsScatter_type
```

Shared Data

The CRTM Shared Data

- Shared Data is the precomputed data that is loaded during the model initialisation. The shared data is loaded into a public data structure that can then be used by application modules.
- Shared data is not visible via the User Interface.
- Needed for:
 - Gaseous absorption functions require regression coefficients (e.g. OPTRAN) or optical depth lookup tables (e.g. OSS).
 - Surface optics functions require regression coefficients (e.g. IRSSEM) or "hinge point" spectra.
 - Scattering functions may require the same (e.g. current aerosol absorption/scattering uses channel based coefficients, but will transition to "hinge point" spectra).
- Getting data into the system is one of the more difficult parts of CRTM development.

Current Shared Data CRTM Data Structures

Type Name	Description
SpcCoeff_type	Channel frequencies, polarisation, Planck function coefficients, etc.
TauCoeff_type	Coefficient data used in the AtmsAbsorption functions.
AerosolCoeff_type	Coefficient data used in the AerosolScatter functions.
ScatterCoeff_type	Coefficient data used in the CloudScatter functions.

- Will need the same for surface optics functions to compute surface emissivities/reflectivities.