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/CRMT

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.

Current Forward CRTM Interface

Error_Status = CRM_Forward ( ATMosphere, & GeometryInfo, & ChannelInfo, & RTSSolution )

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.

User Interface

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRTM_ChannelInfo_type</td>
<td>Sensor channel information filled during initialisation.</td>
</tr>
<tr>
<td>CRTM_AtmAbsorption_type</td>
<td>Atmospheric state profile data. Contains clouds and Aerosol structures.</td>
</tr>
<tr>
<td>CRTM_Surface_type</td>
<td>Surface type and state information. Contains SensorData structure.</td>
</tr>
<tr>
<td>CRTM_GeometryInfo_type</td>
<td>Earth location, zenith and azimuth angles.</td>
</tr>
<tr>
<td>CRTM_RTSSolution_type</td>
<td>Radiative transfer results.</td>
</tr>
</tbody>
</table>

Example: Definition of Atmosphere Structure

```plaintext
Type, PUBLIC :: CRTM_AtmAbsorption_type
  ! -- Algorithm-specific members
  REAL( fp_kind ), DIMENSION( : ), POINTER :: Absorber_ID => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), POINTER :: Absorber_Units => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), POINTER :: Asymmetry_Factor => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), POINTER :: Delta_Truncation => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), POINTER :: Optical_Depth => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), Pointer :: Temperature => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), Pointer :: Planck_Factor => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), Pointer :: Planck_Temperature => NULL() ! K
END TYPE CRTM_AtmAbsorption_type
```

Example: Definition of AtmScatter Structure

```plaintext
Type, PUBLIC :: CRTM_AtmScatter_type
  ! -- Algorithm-specific members
  REAL( fp_kind ), DIMENSION( : ), POINTER :: Delta_Scatter => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), Pointer :: Asymmetry_Factor => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), Pointer :: Scaled_Scatter => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), Pointer :: Scaled_Asymmetry => NULL() ! K
END TYPE CRTM_AtmScatter_type
```

Shared Data

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpCoef_type</td>
<td>Channel frequencies, polarisation, Planck function coefficients, etc.</td>
</tr>
<tr>
<td>TauCoef_type</td>
<td>Coefficient data used in the AtmAbsorption functions</td>
</tr>
<tr>
<td>AerosolCoef_type</td>
<td>Coefficient data used in the AerosolScatter functions</td>
</tr>
<tr>
<td>ScatterCoef_type</td>
<td>Coefficient data used in the CloudScatter functions</td>
</tr>
</tbody>
</table>

Example: Definition of AtmScatter Structure

```plaintext
Type, PUBLIC :: CRTM_AtmScatter_type
  ! -- Algorithm-specific members
  REAL( fp_kind ), DIMENSION( : ), POINTER :: Delta_Scatter => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), Pointer :: Asymmetry_Factor => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), Pointer :: Scaled_Scatter => NULL() ! K
  REAL( fp_kind ), DIMENSION( : ), Pointer :: Scaled_Asymmetry => NULL() ! K
END TYPE CRTM_AtmScatter_type
```

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.

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.

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; ice, snow, water, etc)
• Reflectivity (diffuse and direct)
• Radiative Transfer. RTSSolution functions.
• Fixed- and variable-angle multi-stream models