AtmosphereModel

5.1

Generated by Doxygen 1.8.14

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Module Documentation

6.1 Models

Modules

- Environment

6.1.1 Detailed Description

6.2 Environment

Modules

- Atmosphere

6.2.1 Detailed Description

6.3 Atmosphere

Modules

- BaseAtmosphere

Files

- file atmosphere_messages.hh

  Implement atmosphere_messages.
- file atmosphere.hh

  General base class for atmosphere models.
- file base_atmos/include/class_declarations.hh

  Forward declarations of classes defined for JEOD 2.0 Atmosphere.
- file wind_velocity_base.hh

  General base class for wind velocity models.
- file atmosphere_messages.cc

  Implement atmosphere_messages.
- file atmosphere_state.cc

  Implementation of the base atmosphere-state model.
- file wind_velocity.cc

  General base class for wind velocity models.
- file wind_velocity_base.cc

  General base class for wind velocity models.
- file MET/include/class_declarations.hh

  Forward declarations of classes defined for JEOD 2.0 Atmosphere.
- file MET_atmosphere.hh

  Implement the MET atmosphere using the atmosphere framework.
- file MET_atmosphere_state.hh

  Implement the MET atmosphere state using the atmosphere framework.
- file MET_atmosphere_state_vars.hh

  Implement the MET atmosphere state variables using the atmosphere framework.
- file MET_atmosphere.cc

  Implementation of MET atmosphere model.
- file MET_atmosphere.cc

  Implementation of MET atmosphere model.
- file MET_atmosphere_state_vars.cc

  Implementation of MET atmosphere model.

Namespaces

- jeod

  Namespace jeod.

Macros

- #define PATH "environment/atmosphere/base_atmos"
- #define _USE_MATH_DEFINES_

6.3.1 Detailed Description

6.3.2 Macro Definition Documentation

6.3.2.1 _USE_MATH_DEFINES_

```
#define _USE_MATH_DEFINES_
```

Definition at line 39 of file MET_atmosphere.cc.

6.3.2.2 PATH

```
#define PATH "environment/atmosphere/base_atmos"
```

Definition at line 28 of file atmosphere_messages.cc.

6.4 BaseAtmosphere

Files

- file atmosphere.hh

  General base class for atmosphere models.
- file wind_velocity.hh

  A wind velocity model based on winds caused by rotation of the planet.

Namespaces

- jeod

  Namespace jeod.

6.4.1 Detailed Description

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class Atmosphere

  A generic base class for atmospheres.
- class AtmosphereMessages

  Describes messages used in the Atmosphere model.
- class AtmosphereState

  A generic base class for atmosphere state, containing common atmosphere state parameters, i.e.
- class METAtmosphere
- class METAtmosphere_solar_max_default_data
- class METAtmosphere_solar_mean_default_data
- class METAtmosphere_solar_min_default_data
- class METAtmosphereChemical

  The chemical composition of the MET Atmosphere.
- class METAtmosphereState

  The MET specific implementation of AtmosphereState.
- class METAtmosphereStateVars

  The data variables component of the MET specific implementation of AtmosphereState.
- class METAtmosphereThermal

  The Thermal aspect of the computation.
- class WindVelocity

  A generic wind velocity implementation.
- class WindVelocity_wind_velocity_default_data
- class WindVelocityBase

  The generic base class for wind velocity classes.

7.1.1 Detailed Description

Namespace jeod.

Chapter 8

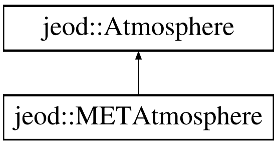
Data Structure Documentation

8.1 jeod::Atmosphere Class Reference

A generic base class for atmospheres.

```
#include <atmosphere.hh>
```

Inheritance diagram for jeod::Atmosphere:



Public Member Functions

- Atmosphere ()=default
- virtual ∼Atmosphere ()=default
- Atmosphere & operator= (const Atmosphere &rhs)=delete
- Atmosphere (const Atmosphere &rhs)=delete
- virtual void update_atmosphere (const PlanetFixedPosition ∗position, AtmosphereState ∗state)=0

  A pure virtual function for updating the atmosphere, and inserting.

Data Fields

- bool active {true}

  If true the atmosphere state will calculate, if false it will not.

Friends

- class InputProcessor
- void init_attrjeod__Atmosphere ()

8.1.1 Detailed Description

A generic base class for atmospheres.

Definition at line 78 of file atmosphere.hh.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 Atmosphere() [1/2]

```
jeod::Atmosphere::Atmosphere ( )  [default]
```

8.1.2.2 ∼Atmosphere()

```
virtual jeod::Atmosphere::∼Atmosphere ( )  [virtual], [default]
```

8.1.2.3 Atmosphere() [2/2]

```
jeod::Atmosphere::Atmosphere (
            const Atmosphere & rhs )  [delete]
```

8.1.3 Member Function Documentation

8.1.3.1 operator=()

```
Atmosphere& jeod::Atmosphere::operator= (
            const Atmosphere & rhs )  [delete]
```

8.1.3.2 update_atmosphere()

```
virtual void jeod::Atmosphere::update_atmosphere (
            const PlanetFixedPosition * position,
            AtmosphereState * state )  [pure virtual]
```

A pure virtual function for updating the atmosphere, and inserting.

Parameters

in	*position*	planet fixed position
out	*state*	The AtmosphereState

Implemented in jeod::METAtmosphere.

Referenced by jeod::AtmosphereState::update_state().

8.1.4 Friends And Related Function Documentation

8.1.4.1 init_attrjeod__Atmosphere

```
void init_attrjeod__Atmosphere ( )  [friend]
```

8.1.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 80 of file atmosphere.hh.

8.1.5 Field Documentation

8.1.5.1 active

```
bool jeod::Atmosphere::active {true}
```

If true the atmosphere state will calculate, if false it will not.

trick_units(–) activity-control flag.

Definition at line 84 of file atmosphere.hh.

The documentation for this class was generated from the following file:

- atmosphere.hh

8.2 jeod::AtmosphereMessages Class Reference

Describes messages used in the Atmosphere model.

```
#include <atmosphere_messages.hh>
```

Public Member Functions

- AtmosphereMessages ()=delete
- AtmosphereMessages (const AtmosphereMessages &rhs)=delete
- AtmosphereMessages & operator= (const AtmosphereMessages &rhs)=delete

Static Public Attributes

- static const char ∗ initialization_error = "environment/atmosphere/base_atmos" "initialization_error"

  Indicates an error during initialization.
- static const char ∗ framework_error = "environment/atmosphere/base_atmos" "framework_error"

  Indicates an error during use of the generic framework.
- static const char ∗ framework_warning = "environment/atmosphere/base_atmos" "framework_warning"

  Indicates a warning associated with the generic framework.
- static const char ∗ numerical_warning = "environment/atmosphere/base_atmos" "numerical_warning"

  Indicates a warning associated with numerical values.

Friends

- class InputProcessor
- void init_attrjeod__AtmosphereMessages ()

8.2.1 Detailed Description

Describes messages used in the Atmosphere model.

Definition at line 76 of file atmosphere_messages.hh.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 AtmosphereMessages() [1/2]

```
jeod::AtmosphereMessages::AtmosphereMessages ( )  [delete]
```

8.2.2.2 AtmosphereMessages() [2/2]

```
jeod::AtmosphereMessages::AtmosphereMessages (
             const AtmosphereMessages & rhs )  [delete]
```

8.2.3 Member Function Documentation

8.2.3.1 operator=()

```
AtmosphereMessages& jeod::AtmosphereMessages::operator= (
             const AtmosphereMessages & rhs )  [delete]
```

8.2.4 Friends And Related Function Documentation

8.2.4.1 init_attrjeod__AtmosphereMessages

```
void init_attrjeod__AtmosphereMessages ( )  [friend]
```

8.2.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 78 of file atmosphere_messages.hh.

8.2.5 Field Documentation

8.2.5.1 framework_error

```
const char * jeod::AtmosphereMessages::framework_error = "environment/atmosphere/base_atmos"
"framework_error"  [static]
```

Indicates an error during use of the generic framework.

trick_units(–)

Definition at line 91 of file atmosphere_messages.hh.

Referenced by jeod::WindVelocity::set_omega_scale_table(), jeod::METAtmosphere::update_atmosphere(), and jeod::WindVelocity::update_wind().

8.2.5.2 framework_warning

```
const char * jeod::AtmosphereMessages::framework_warning = "environment/atmosphere/base_atmos"
"framework_warning"  [static]
```

Indicates a warning associated with the generic framework.

trick_units(–)

Definition at line 98 of file atmosphere_messages.hh.

Referenced by jeod::WindVelocityBase::update_wind().

8.2.5.3 initialization_error

```
const char * jeod::AtmosphereMessages::initialization_error = "environment/atmosphere/base_↩
atmos" "initialization_error"  [static]
```

Indicates an error during initialization.

trick_units(–)

Definition at line 86 of file atmosphere_messages.hh.

8.2.5.4 numerical_warning

```
const char * jeod::AtmosphereMessages::numerical_warning = "environment/atmosphere/base_atmos"
"numerical_warning"  [static]
```

Indicates a warning associated with numerical values.

trick_units(–)

Definition at line 103 of file atmosphere_messages.hh.

Referenced by jeod::METAtmosphere::compute_exospheric_temperature().

The documentation for this class was generated from the following files:

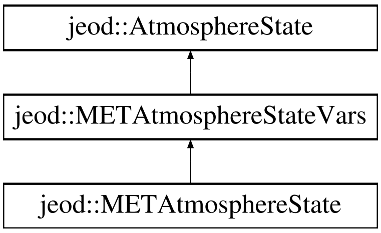
- atmosphere_messages.hh
- atmosphere_messages.cc

8.3 jeod::AtmosphereState Class Reference

A generic base class for atmosphere state, containing common atmosphere state parameters, i.e.

```
#include <atmosphere_state.hh>
```

Inheritance diagram for jeod::AtmosphereState:



Public Member Functions

- AtmosphereState ()=default
- AtmosphereState (Atmosphere &atmos, const PlanetFixedPosition &pfix_pos)
- virtual ∼AtmosphereState ()=default
- AtmosphereState & operator= (const AtmosphereState &rhs)

  AtmosphereState Operator =.
- AtmosphereState (const AtmosphereState &rhs)

  Copy Constructor.
- void update_state (Atmosphere ∗atmos_model_, PlanetFixedPosition ∗pfix_pos_)

  Updates the invoking atmosphere state, using the atmosphere model pointed to by atmos_model, and calculated at the planet fixed position pointed to by pfix_pos.
- virtual void update_state ()

  Updates the invoking atmosphere state, using the atmosphere model pointed to by atmos, and calculated at the planet fixed position pointed to by pfix_pos.
- void update_wind (WindVelocity ∗wind_vel, double inrtl_pos[3], double altitude)

  Updates the wind portion of the invoking atmosphere state, using the wind model pointed to by wind_vel, calculated at the inertial position given by inrtl_pos and the altitude given.

Data Fields

- bool active {true}
- double temperature {}
- double density {}
- double pressure {}
- double wind [3] {}

Protected Attributes

- Atmosphere ∗ atmos {}
- const PlanetFixedPosition ∗ pfix_pos {}

Friends

- class InputProcessor
- void init_attrjeod__AtmosphereState ()

8.3.1 Detailed Description

A generic base class for atmosphere state, containing common atmosphere state parameters, i.e.

pressure, density, temperature, wind velocity

Definition at line 86 of file atmosphere_state.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 AtmosphereState() [1/3]

```
jeod::AtmosphereState::AtmosphereState ( )  [default]
```

8.3.2.2 AtmosphereState() [2/3]

```
jeod::AtmosphereState::AtmosphereState (
            Atmosphere & atmos,
            const PlanetFixedPosition & pfix_pos )
```

Definition at line 34 of file atmosphere_state.cc.

8.3.2.3 ∼AtmosphereState()

```
virtual jeod::AtmosphereState::∼AtmosphereState ( )  [virtual], [default]
```

8.3.2.4 AtmosphereState() [3/3]

```
jeod::AtmosphereState::AtmosphereState (
            const AtmosphereState & rhs )
```

Copy Constructor.

Parameters

in	*rhs*	The AtmosphereState to copy from

Definition at line 45 of file atmosphere_state.cc.

References atmos, density, pfix_pos, pressure, temperature, and wind.

8.3.3 Member Function Documentation

8.3.3.1 operator=()

```
AtmosphereState & jeod::AtmosphereState::operator= (
             const AtmosphereState & rhs )
```

AtmosphereState Operator =.

Returns

The newly copied AtmosphereState

Parameters

in	*rhs*	The AtmosphereState to copy

Definition at line 65 of file atmosphere_state.cc.

References density, pressure, and temperature.

Referenced by jeod::METAtmosphereStateVars::operator=().

8.3.3.2 update_state() [1/2]

```
void jeod::AtmosphereState::update_state (
             Atmosphere * atmos_model_,
             PlanetFixedPosition * pfix_pos_ )
```

Updates the invoking atmosphere state, using the atmosphere model pointed to by atmos_model, and calculated at the planet fixed position pointed to by pfix_pos.

Note that any type inheriting from Atmosphere can be sent in for atmos_model.

Parameters

in	*atmos_↩ model_*	Atmosphere model.
in	*pfix_pos_*	Planetary fixed position.

Definition at line 89 of file atmosphere_state.cc.

References active, and jeod::Atmosphere::update_atmosphere().

8.3.3.3 update_state() [2/2]

```
void jeod::AtmosphereState::update_state ( )  [virtual]
```

Updates the invoking atmosphere state, using the atmosphere model pointed to by atmos, and calculated at the planet fixed position pointed to by pfix_pos.

Note that any type inheriting from Atmosphere can used as the Atmosphere pointer but only the values associated with AtmosphereState will be copied back out.

Reimplemented in jeod::METAtmosphereState.

Definition at line 107 of file atmosphere_state.cc.

References active, atmos, pfix_pos, and jeod::Atmosphere::update_atmosphere().

8.3.3.4 update_wind()

```
void jeod::AtmosphereState::update_wind (
             WindVelocity * wind_vel,
             double inrtl_pos[3],
             double altitude )
```

Updates the wind portion of the invoking atmosphere state, using the wind model pointed to by wind_vel, calculated at the inertial position given by inrtl_pos and the altitude given.

Parameters

in	*wind_vel*	Wind velocity model.
in	*inrtl_pos*	Current inertial position. Units: M
in	*altitude*	Geodetic (elliptic) altitude. Units: M

Definition at line 125 of file atmosphere_state.cc.

References active, jeod::WindVelocity::update_wind(), and wind.

8.3.4 Friends And Related Function Documentation

8.3.4.1 init_attrjeod__AtmosphereState

```
void init_attrjeod__AtmosphereState ( )  [friend]
```

8.3.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 88 of file atmosphere_state.hh.

8.3.5 Field Documentation

8.3.5.1 active

```
bool jeod::AtmosphereState::active {true}
```

trick_units(–) Activation flag for computing state.

Definition at line 89 of file atmosphere_state.hh.

Referenced by jeod::METAtmosphereStateVars::METAtmosphereStateVars(), jeod::METAtmosphereStateVars↩ ::operator=(), jeod::METAtmosphereState::update_state(), update_state(), and update_wind().

8.3.5.2 atmos

```
Atmosphere* jeod::AtmosphereState::atmos {}  [protected]
```

Definition at line 100 of file atmosphere_state.hh.

Referenced by AtmosphereState(), and update_state().

8.3.5.3 density

```
double jeod::AtmosphereState::density {}
```

trick_units(kg/m3) total density at altitude

Definition at line 93 of file atmosphere_state.hh.

Referenced by jeod::METAtmosphere::atmos_MET_FAIR5(), AtmosphereState(), jeod::METAtmosphere::compute_seasonal_lat_variation_He(), jeod::METAtmosphere::compute_seasonal_latitude_variation(), jeod::METAtmosphere::jacchia(), operator=(), and jeod::METAtmosphere::update_atmosphere().

8.3.5.4 pfix_pos

```
const PlanetFixedPosition* jeod::AtmosphereState::pfix_pos {}  [protected]
```

Definition at line 101 of file atmosphere_state.hh.

Referenced by AtmosphereState(), jeod::METAtmosphereState::update_state(), and update_state().

8.3.5.5 pressure

```
double jeod::AtmosphereState::pressure {}
```

trick_units(N/m2) Total pressure

Definition at line 95 of file atmosphere_state.hh.

Referenced by AtmosphereState(), operator=(), and jeod::METAtmosphere::update_atmosphere().

8.3.5.6 temperature

```
double jeod::AtmosphereState::temperature {}
```

trick_units(K) Temperature at altitude

Definition at line 91 of file atmosphere_state.hh.

Referenced by AtmosphereState(), jeod::METAtmosphere::jacchia(), operator=(), and jeod::METAtmosphere::update_atmosphere().

8.3.5.7 wind

```
double jeod::AtmosphereState::wind[3] {}
```

trick_units(m/s) Wind velocity

Definition at line 97 of file atmosphere_state.hh.

Referenced by AtmosphereState(), and update_wind().

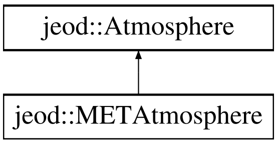
The documentation for this class was generated from the following files:

- atmosphere_state.hh
- atmosphere_state.cc

8.4 jeod::METAtmosphere Class Reference

```
#include <MET_atmosphere.hh>
```

Inheritance diagram for jeod::METAtmosphere:



Public Types

- enum AtmosMETGeoIndexType { ATMOS_MET_GI_AP = 0, ATMOS_MET_GI_KP = 1 }

Public Member Functions

- METAtmosphere (const double &trunc_julian_time_in)
- ∼METAtmosphere () override=default
- METAtmosphere & operator= (const METAtmosphere &)=delete
- METAtmosphere (const METAtmosphere &)=delete
- void update_atmosphere (const PlanetFixedPosition ∗pfix_pos, AtmosphereState ∗state) override

  A pure virtual function for updating the atmosphere, and inserting.
- void update_atmosphere (const PlanetFixedPosition ∗pfix_pos, METAtmosphereStateVars ∗state)

  Front-end to the computation of the METAtmosphere at the current time Inserts the results into the METAtmosphereStateVars pointed to by ext_state.

Data Fields

- AtmosMETGeoIndexType geo_index_type {ATMOS_MET_GI_AP}
- double geo_index {}
- double F10 {}
- double F10B {}
- METAtmosphereChemical species

Private Member Functions

- void update_atmosphere (const PlanetFixedPosition ∗pfix_pos)

  Calculates the METAtmosphere, at the current time.
- void modify_densities ()
- void compute_solar_angles ()
- void compute_exospheric_temperature ()
- void jacchia ()
- void compute_seasonal_latitude_variation ()
- void compute_seasonal_lat_variation_He ()
- void atmos_MET_FAIR5 ()
- double compute_mol_wt (double altitude)
- double apply_gauss_quadrature (int altitude_index_start, double ceiling)

Private Attributes

- double altitude_km {}
- double latitude {}
- double longitude {}
- double barometric_equation_ceiling {105.0}
- const double & trunc_julian_time
- double tjt_year_start {11544.0}
- double fraction_of_year {}
- int day_of_year {1}
- int max_days_this_year {366}
- int year {2000}
- double solar_declination_angle {}
- double solar_hour_angle {}
- METAtmosphereStateVars state
- METAtmosphereThermal thermal
- const double R_gas_constant {8.31432}
- const double days_per_year {365.2422}
- const double Avogadro {6.02257E23}
- const double two_pi {6.28318531}
- const double three_pi_two {4.71238898}
- const double deg_to_rad {0.017453293}
- const int days_per_century {36525}
- const int minutes_per_day {1440}
- const double mol_weight_barometric_ceiling {27.72594278125}
- const double base_fairing_height {440.0}
- const double fairing_k

Static Private Attributes

- static const int num_mol_wt_coeffs = 7
- static const double mol_wt_coeffs [num_mol_wt_coeffs]
- static const int num_integ_divisions = 8
- static const double gauss_altitudes [num_integ_divisions+1] = {90.0, 105.0, 125.0, 160.0, 200.0, 300.↩ 0, 500.0, 1500.0, 2500.0}
- static const int gauss_n [num_integ_divisions] = {4, 5, 6, 6, 6, 6, 6, 6}

Friends

- class InputProcessor
- void init_attrjeod__METAtmosphere ()

8.4.1 Detailed Description

Definition at line 179 of file MET_atmosphere.hh.

8.4.2 Member Enumeration Documentation

8.4.2.1 AtmosMETGeoIndexType

```
enum jeod::METAtmosphere::AtmosMETGeoIndexType
```

Enumerator

ATMOS_MET_GI_AP	
ATMOS_MET_GI_KP	

Definition at line 182 of file MET_atmosphere.hh.

8.4.3 Constructor & Destructor Documentation

8.4.3.1 METAtmosphere() [1/2]

```
jeod::METAtmosphere::METAtmosphere (
             const double & trunc_julian_time_in )  [explicit]
```

Definition at line 84 of file MET_atmosphere.cc.

8.4.3.2 ∼METAtmosphere()

```
jeod::METAtmosphere::∼METAtmosphere ( )  [override], [default]
```

8.4.3.3 METAtmosphere() [2/2]

```
jeod::METAtmosphere::METAtmosphere (
            const METAtmosphere &  )  [delete]
```

8.4.4 Member Function Documentation

8.4.4.1 apply_gauss_quadrature()

```
double jeod::METAtmosphere::apply_gauss_quadrature (
            int altitude_index_start,
            double ceiling )  [private]
```

Definition at line 1149 of file MET_atmosphere.cc.

References barometric_equation_ceiling, compute_mol_wt(), jeod::METAtmosphereThermal::compute_↩ temperature(), gauss_altitudes, gauss_n, and thermal.

Referenced by jacchia().

8.4.4.2 atmos_MET_FAIR5()

```
void jeod::METAtmosphere::atmos_MET_FAIR5 ( )  [private]
```

Definition at line 1018 of file MET_atmosphere.cc.

References altitude_km, base_fairing_height, compute_seasonal_lat_variation_He(), jeod::AtmosphereState↩ ::density, fairing_k, jeod::METAtmosphereChemical::num_density, species, and state.

Referenced by modify_densities().

8.4.4.3 compute_exospheric_temperature()

```
void jeod::METAtmosphere::compute_exospheric_temperature ( )  [private]
```

Definition at line 541 of file MET_atmosphere.cc.

References ATMOS_MET_GI_KP, jeod::METAtmosphereStateVars::exo_temp, F10, F10B, fraction_of_year, geo_index, geo_index_type, latitude, jeod::AtmosphereMessages::numerical_warning, solar_declination_angle, solar_hour_angle, state, and two_pi.

Referenced by update_atmosphere().

8.4.4.4 compute_mol_wt()

```
double jeod::METAtmosphere::compute_mol_wt (
             double altitude )  [private]
```

Definition at line 1069 of file MET_atmosphere.cc.

References barometric_equation_ceiling, mol_weight_barometric_ceiling, and mol_wt_coeffs.

Referenced by apply_gauss_quadrature(), and jacchia().

8.4.4.5 compute_seasonal_lat_variation_He()

```
void jeod::METAtmosphere::compute_seasonal_lat_variation_He ( )  [private]
```

Definition at line 960 of file MET_atmosphere.cc.

References jeod::AtmosphereState::density, latitude, jeod::METAtmosphereChemical::num_density, solar_declination_angle, species, and state.

Referenced by atmos_MET_FAIR5(), and modify_densities().

8.4.4.6 compute_seasonal_latitude_variation()

```
void jeod::METAtmosphere::compute_seasonal_latitude_variation ( )  [private]
```

Definition at line 904 of file MET_atmosphere.cc.

References altitude_km, jeod::AtmosphereState::density, fraction_of_year, latitude, and state.

Referenced by modify_densities().

8.4.4.7 compute_solar_angles()

```
void jeod::METAtmosphere::compute_solar_angles ( )  [private]
```

Definition at line 347 of file MET_atmosphere.cc.

References day_of_year, days_per_century, days_per_year, deg_to_rad, fraction_of_year, longitude, max_days_this_year, minutes_per_day, solar_declination_angle, solar_hour_angle, three_pi_two, tjt_year_start, trunc_julian_time, two_pi, and year.

Referenced by update_atmosphere().

8.4.4.8 jacchia()

```
void jeod::METAtmosphere::jacchia ( )  [private]
```

Definition at line 695 of file MET_atmosphere.cc.

References altitude_km, apply_gauss_quadrature(), Avogadro, barometric_equation_ceiling, compute_mol↩_wt(), jeod::METAtmosphereThermal::compute_temperature(), jeod::AtmosphereState::density, jeod::MET↩AtmosphereChemical::frac, jeod::METAtmosphereStateVars::mol_weight, jeod::METAtmosphereChemical↩::mol_weight, mol_weight_barometric_ceiling, jeod::METAtmosphereChemical::nominal_mol_weight, jeod::M↩ETAtmosphereChemical::num_density, R_gas_constant, species, state, jeod::METAtmosphereThermal::T_out, jeod::AtmosphereState::temperature, thermal, and jeod::METAtmosphereThermal::update().

Referenced by update_atmosphere().

8.4.4.9 modify_densities()

```
void jeod::METAtmosphere::modify_densities ( )  [private]
```

Definition at line 308 of file MET_atmosphere.cc.

References altitude_km, atmos_MET_FAIR5(), base_fairing_height, compute_seasonal_lat_variation_He(), and compute_seasonal_latitude_variation().

Referenced by update_atmosphere().

8.4.4.10 operator=()

```
METAtmosphere& jeod::METAtmosphere::operator= (
             const METAtmosphere &  )  [delete]
```

8.4.4.11 update_atmosphere() [1/3]

```
void jeod::METAtmosphere::update_atmosphere (
             const PlanetFixedPosition * position,
             AtmosphereState * state )  [override], [virtual]
```

A pure virtual function for updating the atmosphere, and inserting.

Parameters

in	*position*	planet fixed position
out	*state*	The AtmosphereState

Implements jeod::Atmosphere.

Definition at line 205 of file MET_atmosphere.cc.

References jeod::AtmosphereMessages::framework_error, and state.

Referenced by update_atmosphere(), and jeod::METAtmosphereState::update_state().

8.4.4.12 update_atmosphere() [2/3]

```
void jeod::METAtmosphere::update_atmosphere (
            const PlanetFixedPosition * pfix_pos,
            METAtmosphereStateVars * ext_state )
```

Front-end to the computation of the METAtmosphere at the current time Inserts the results into the METAtmosphereStateVars pointed to by ext_state.

This function is for a METAtmosphereStateVars.

Parameters

in	*pfix_pos*	Geodetic altitude, latitude and longitude.
out	*ext_state*	Where the state results will be sent.

Definition at line 240 of file MET_atmosphere.cc.

References jeod::AtmosphereMessages::framework_error, state, and update_atmosphere().

8.4.4.13 update_atmosphere() [3/3]

```
void jeod::METAtmosphere::update_atmosphere (
            const PlanetFixedPosition * pfix_pos )  [private]
```

Calculates the METAtmosphere, at the current time.

Parameters

in	*pfix_pos*	Geodetic altitude, latitude and longitude.

Definition at line 262 of file MET_atmosphere.cc.

References jeod::METAtmosphereStateVars::A, altitude_km, compute_exospheric_temperature(), compute←_solar_angles(), jeod::AtmosphereState::density, jeod::AtmosphereMessages::framework_error, jeod::MET←AtmosphereStateVars::He, jeod::METAtmosphereStateVars::Hyd, jacchia(), latitude, jeod::METAtmosphere←StateVars::log10_dens, longitude, modify_densities(), jeod::METAtmosphereStateVars::mol_weight, jeod::ME←TAtmosphereStateVars::N2, jeod::METAtmosphereChemical::num_density, jeod::METAtmosphereStateVars::Ox, jeod::METAtmosphereStateVars::Ox2, jeod::AtmosphereState::pressure, R_gas_constant, species, state, and jeod::AtmosphereState::temperature.

8.4.5 Friends And Related Function Documentation

8.4.5.1 init_attrjeod__METAtmosphere

```
void init_attrjeod__METAtmosphere ( )  [friend]
```

8.4.5.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 181 of file MET_atmosphere.hh.

8.4.6 Field Documentation

8.4.6.1 altitude_km

```
double jeod::METAtmosphere::altitude_km {}  [private]
```

trick_units(km) Copy of vehicle altitude

Definition at line 202 of file MET_atmosphere.hh.

Referenced by atmos_MET_FAIR5(), compute_seasonal_latitude_variation(), jacchia(), modify_densities(), and update_atmosphere().

8.4.6.2 Avogadro

```
const double jeod::METAtmosphere::Avogadro {6.02257E23}  [private]
```

trick_units(–) Avogadros number

Definition at line 243 of file MET_atmosphere.hh.

Referenced by jacchia().

8.4.6.3 barometric_equation_ceiling

```
double jeod::METAtmosphere::barometric_equation_ceiling {105.0}  [private]
```

trick_units(km) the ceiling for integration using the barometric equation. Above this value, the integration switches to the diffusion equation. Value is 105km in the 1970 paper and 100km in the 1971 paper.

Definition at line 206 of file MET_atmosphere.hh.

Referenced by apply_gauss_quadrature(), compute_mol_wt(), and jacchia().

8.4.6.4 base_fairing_height

```
const double jeod::METAtmosphere::base_fairing_height {440.0}  [private]
```

trick_units(km) Altitude at which to start fairing between the lower altitude which has no seasonal-latitude Helium density variation, and the upper atmosphere – starting at 500km – which does.

Definition at line 258 of file MET_atmosphere.hh.

Referenced by atmos_MET_FAIR5(), and modify_densities().

8.4.6.5 day_of_year

```
int jeod::METAtmosphere::day_of_year {1}  [private]
```

trick_units(count) day number since start of year.

Definition at line 221 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.6 days_per_century

```
const int jeod::METAtmosphere::days_per_century {36525}  [private]
```

trick_units(count) days per century

Definition at line 250 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.7 days_per_year

```
const double jeod::METAtmosphere::days_per_year {365.2422}  [private]
```

trick_units(day) days per year

Definition at line 242 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.8 deg_to_rad

```
const double jeod::METAtmosphere::deg_to_rad {0.017453293}  [private]
```

trick_units(degree/rad) degree-to-radian conversion

Definition at line 249 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.9 F10

```
double jeod::METAtmosphere::F10 {}
```

trick_units(–) Solar radio noise flux.

Definition at line 195 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), jeod::METAtmosphere_solar_min_default_data::initialize(), jeod::METAtmosphere_solar_mean_default_data::initialize(), and jeod::METAtmosphere_solar_max_default_data::initialize().

8.4.6.10 F10B

```
double jeod::METAtmosphere::F10B {}
```

trick_units(–) 90 day average of solar radio noise flux.

Definition at line 197 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), jeod::METAtmosphere_solar_min_default_data::initialize(), jeod::METAtmosphere_solar_mean_default_data::initialize(), and jeod::METAtmosphere_solar_max_default_data::initialize().

8.4.6.11 fairing_k

```
const double jeod::METAtmosphere::fairing_k  [private]
```

trick_units(rad/km) Factor which, when multiplied by the altitude delta above the base-fairing-height provides an angle. The square of the cosine of that angle indicates how much of the seasonal-variation in Helium density to apply. density = corrected-density ∗ (non-corrected-density / corrected-density) ^ (cos^2 (fairing_k ∗ delta-altitude)) At base-fairing-height, none gets applied. By 500km, it all gets applied.

Definition at line 262 of file MET_atmosphere.hh.

Referenced by atmos_MET_FAIR5().

8.4.6.12 fraction_of_year

```
double jeod::METAtmosphere::fraction_of_year {}  [private]
```

trick_units(–) fraction of this year that has passed.

Definition at line 219 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), compute_seasonal_latitude_variation(), and compute_solar↩ _angles().

8.4.6.13 gauss_altitudes

```
const double jeod::METAtmosphere::gauss_altitudes = {90.0, 105.0, 125.0, 160.0, 200.0, 300.0,
500.0, 1500.0, 2500.0}  [static], [private]
```

trick_units(–) The boundaries of the cells that are used to break down the integration over the atmosphere into more manaegable pieces. NOTE - gauss_altitudes[1] must mark the upper limit of the altitude over which the barometric equation is valid, this is either 100km or 105km, depending on which paper is used; gauss-altitude[6] must be equal to 500km.

Definition at line 283 of file MET_atmosphere.hh.

Referenced by apply_gauss_quadrature().

8.4.6.14 gauss_n

```
const int jeod::METAtmosphere::gauss_n = {4, 5, 6, 6, 6, 6, 6, 6}  [static], [private]
```

trick_units(–) The number of data-points to be used for the gauss-quadrature integration for each interval defined in the gauss_altitudes array. AKA the order of the gauss-quadrature.

Definition at line 290 of file MET_atmosphere.hh.

Referenced by apply_gauss_quadrature().

8.4.6.15 geo_index

```
double jeod::METAtmosphere::geo_index {}
```

trick_units(–) Geomagnetic variations index (Ap or Kp).

Definition at line 193 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), jeod::METAtmosphere_solar_min_default_data::initialize(), jeod::METAtmosphere_solar_mean_default_data::initialize(), and jeod::METAtmosphere_solar_max_default_data::initialize().

8.4.6.16 geo_index_type

```
AtmosMETGeoIndexType jeod::METAtmosphere::geo_index_type {ATMOS_MET_GI_AP}
```

Definition at line 190 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), jeod::METAtmosphere_solar_min_default_data::initialize(), jeod::METAtmosphere_solar_mean_default_data::initialize(), and jeod::METAtmosphere_solar_max_default_data::initialize().

8.4.6.17 latitude

```
double jeod::METAtmosphere::latitude {}  [private]
```

trick_units(rad) Copy of vehicle latitude

Definition at line 203 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), compute_seasonal_lat_variation_He(), compute_seasonal_latitude_variation(), and update_atmosphere().

8.4.6.18 longitude

```
double jeod::METAtmosphere::longitude {}  [private]
```

trick_units(rad) Copy of vehicle longitude

Definition at line 204 of file MET_atmosphere.hh.

Referenced by compute_solar_angles(), and update_atmosphere().

8.4.6.19 max_days_this_year

```
int jeod::METAtmosphere::max_days_this_year {366}  [private]
```

trick_units(count) number of days this year (365 or 366)

Definition at line 223 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.20 minutes_per_day

```
const int jeod::METAtmosphere::minutes_per_day {1440}  [private]
```

trick_units(count) minutes per day

Definition at line 251 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.21 mol_weight_barometric_ceiling

```
const double jeod::METAtmosphere::mol_weight_barometric_ceiling {27.72594278125}  [private]
```

trick_units(g/mol) mean molar mass at barometric-ceiling and higher.

Definition at line 254 of file MET_atmosphere.hh.

Referenced by compute_mol_wt(), and jacchia().

8.4.6.22 mol_wt_coeffs

```
const double jeod::METAtmosphere::mol_wt_coeffs  [static], [private]
```

Initial value:

```
=
    {28.15204, -0.085586, 1.284E-4, -1.0056E-5, -1.021E-5, 1.5044E-6, 9.9826E-8}
```

trick_units(–) polynomial coefficients for computing the molecular weights in the region where the barometric equation is used.

Definition at line 275 of file MET_atmosphere.hh.

Referenced by compute_mol_wt().

8.4.6.23 num_integ_divisions

```
const int jeod::METAtmosphere::num_integ_divisions = 8  [static], [private]
```

trick_units(count) the number of altitude bins used for dividing the atmosphere into manageable pieces.

Definition at line 280 of file MET_atmosphere.hh.

8.4.6.24 num_mol_wt_coeffs

```
const int jeod::METAtmosphere::num_mol_wt_coeffs = 7  [static], [private]
```

trick_units(count) the number of polynomial coefficients.

Definition at line 274 of file MET_atmosphere.hh.

8.4.6.25 R_gas_constant

```
const double jeod::METAtmosphere::R_gas_constant {8.31432}  [private]
```

trick_units(J/(mol∗K)) R

Definition at line 238 of file MET_atmosphere.hh.

Referenced by jacchia(), and update_atmosphere().

8.4.6.26 solar_declination_angle

```
double jeod::METAtmosphere::solar_declination_angle {}  [private]
```

trick_units(rad) declination angle

Definition at line 227 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), compute_seasonal_lat_variation_He(), and compute_solar_↩ angles().

8.4.6.27 solar_hour_angle

```
double jeod::METAtmosphere::solar_hour_angle {}  [private]
```

trick_units(rad) solar hour angle

Definition at line 229 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), and compute_solar_angles().

8.4.6.28 species

```
METAtmosphereChemical jeod::METAtmosphere::species
```

trick_units(–) The chemical composition of the atmosphere.

Definition at line 199 of file MET_atmosphere.hh.

Referenced by atmos_MET_FAIR5(), compute_seasonal_lat_variation_He(), jacchia(), and update_atmosphere().

8.4.6.29 state

```
METAtmosphereStateVars jeod::METAtmosphere::state [private]
```

trick_units(–) A scratch set of state variables, used for populating state variables internally before being copied onto the real state.

Definition at line 231 of file MET_atmosphere.hh.

Referenced by atmos_MET_FAIR5(), compute_exospheric_temperature(), compute_seasonal_lat_variation_He(), compute_seasonal_latitude_variation(), jacchia(), and update_atmosphere().

8.4.6.30 thermal

```
METAtmosphereThermal jeod::METAtmosphere::thermal [private]
```

trick_units(–) Thermal aspect of the model

Definition at line 235 of file MET_atmosphere.hh.

Referenced by apply_gauss_quadrature(), and jacchia().

8.4.6.31 three_pi_two

```
const double jeod::METAtmosphere::three_pi_two {4.71238898} [private]
```

trick_units(–) 1.5 pi

Definition at line 248 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.32 tjt_year_start

```
double jeod::METAtmosphere::tjt_year_start {11544.0}  [private]
```

trick_units(day) value of trunc_julian_time at the start of the current year.

Definition at line 215 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.33 trunc_julian_time

```
const double& jeod::METAtmosphere::trunc_julian_time  [private]
```

trick_units(day) Current time

Definition at line 214 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.34 two_pi

```
const double jeod::METAtmosphere::two_pi {6.28318531}  [private]
```

trick_units(–) 2 pi

Definition at line 247 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), and compute_solar_angles().

8.4.6.35 year

```
int jeod::METAtmosphere::year {2000}  [private]
```

trick_units(count) current year identifier

Definition at line 225 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

The documentation for this class was generated from the following files:

- MET_atmosphere.hh
- MET_atmosphere.cc

8.5 jeod::METAtmosphere_solar_max_default_data Class Reference

```
#include <solar_max.hh>
```

Public Member Functions

- void initialize (METAtmosphere ∗)

8.5.1 Detailed Description

Definition at line 55 of file solar_max.hh.

8.5.2 Member Function Documentation

8.5.2.1 initialize()

```
void jeod::METAtmosphere_solar_max_default_data::initialize (
            METAtmosphere ∗ METAtmosphere_ptr )
```

Definition at line 35 of file solar_max.cc.

References jeod::METAtmosphere::ATMOS_MET_GI_AP, jeod::METAtmosphere::F10, jeod::METAtmosphere::↩ F10B, jeod::METAtmosphere::geo_index, and jeod::METAtmosphere::geo_index_type.

The documentation for this class was generated from the following files:

- solar_max.hh
- solar_max.cc

8.6 jeod::METAtmosphere_solar_mean_default_data Class Reference

```
#include <solar_mean.hh>
```

Public Member Functions

- void initialize (METAtmosphere ∗)

8.6.1 Detailed Description

Definition at line 55 of file solar_mean.hh.

8.6.2 Member Function Documentation

8.6.2.1 initialize()

```
void jeod::METAtmosphere_solar_mean_default_data::initialize (
            METAtmosphere * METAtmosphere_ptr )
```

Definition at line 35 of file solar_mean.cc.

References jeod::METAtmosphere::ATMOS_MET_GI_AP, jeod::METAtmosphere::F10, jeod::METAtmosphere::F10B, jeod::METAtmosphere::geo_index, and jeod::METAtmosphere::geo_index_type.

The documentation for this class was generated from the following files:

- solar_mean.hh
- solar_mean.cc

8.7 jeod::METAtmosphere_solar_min_default_data Class Reference

```
#include <solar_min.hh>
```

Public Member Functions

- void initialize (METAtmosphere ∗)

8.7.1 Detailed Description

Definition at line 55 of file solar_min.hh.

8.7.2 Member Function Documentation

8.7.2.1 initialize()

```
void jeod::METAtmosphere_solar_min_default_data::initialize (
            METAtmosphere * METAtmosphere_ptr )
```

Definition at line 35 of file solar_min.cc.

References jeod::METAtmosphere::ATMOS_MET_GI_AP, jeod::METAtmosphere::F10, jeod::METAtmosphere::F10B, jeod::METAtmosphere::geo_index, and jeod::METAtmosphere::geo_index_type.

The documentation for this class was generated from the following files:

- solar_min.hh
- solar_min.cc

8.8 jeod::METAtmosphereChemical Class Reference

The chemical composition of the MET Atmosphere.

```
#include <MET_atmosphere.hh>
```

Public Member Functions

- METAtmosphereChemical ()=default
- virtual ∼METAtmosphereChemical ()=default
- METAtmosphereChemical & operator= (const METAtmosphereChemical &)=delete
- METAtmosphereChemical (const METAtmosphereChemical &)=delete

Data Fields

- double num_density [num_species] {}
- double frac [num_species]
- double mol_weight [num_species]
- const double nominal_mol_weight {28.96}

Static Public Attributes

- static const int num_species = 6

Friends

- class InputProcessor
- void init_attrjeod__METAtmosphereChemical ()

8.8.1 Detailed Description

The chemical composition of the MET Atmosphere.

Definition at line 87 of file MET_atmosphere.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 METAtmosphereChemical() [1/2]

```
jeod::METAtmosphereChemical::METAtmosphereChemical ( )  [default]
```

8.8.2.2 ∼METAtmosphereChemical()

```
virtual jeod::METAtmosphereChemical::∼METAtmosphereChemical ( )  [virtual], [default]
```

8.8.2.3 METAtmosphereChemical() [2/2]

```
jeod::METAtmosphereChemical::METAtmosphereChemical (
            const METAtmosphereChemical &  )  [delete]
```

8.8.3 Member Function Documentation

8.8.3.1 operator=()

```
METAtmosphereChemical& jeod::METAtmosphereChemical::operator= (
            const METAtmosphereChemical &  )  [delete]
```

8.8.4 Friends And Related Function Documentation

8.8.4.1 init_attrjeod__METAtmosphereChemical

```
void init_attrjeod__METAtmosphereChemical ( )  [friend]
```

8.8.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 89 of file MET_atmosphere.hh.

8.8.5 Field Documentation

8.8.5.1 frac

```
double jeod::METAtmosphereChemical::frac[num_species]
```

Initial value:

```
{
        0.78110,
        0.20955,
        0.0,
        0.0093432,
        1.289E-05,


        0.0
    }
```

Definition at line 95 of file MET_atmosphere.hh.

Referenced by jeod::METAtmosphere::jacchia().

8.8.5.2 mol_weight

```
double jeod::METAtmosphereChemical::mol_weight[num_species]
```

Initial value:

```
{
        28.0134,
        31.9988,
        15.9994,
        39.948,
        4.0026,
        1.00797
    }
```

Definition at line 106 of file MET_atmosphere.hh.

Referenced by jeod::METAtmosphere::jacchia().

8.8.5.3 nominal_mol_weight

```
const double jeod::METAtmosphereChemical::nominal_mol_weight {28.96}
```

Definition at line 115 of file MET_atmosphere.hh.

Referenced by jeod::METAtmosphere::jacchia().

8.8.5.4 num_density

```
double jeod::METAtmosphereChemical::num_density[num_species] {}
```

Definition at line 92 of file MET_atmosphere.hh.

Referenced by jeod::METAtmosphere::atmos_MET_FAIR5(), jeod::METAtmosphere::compute_seasonal_lat_variation_He(), jeod::METAtmosphere::jacchia(), and jeod::METAtmosphere::update_atmosphere().

8.8.5.5 num_species

```
const int jeod::METAtmosphereChemical::num_species = 6  [static]
```

Definition at line 90 of file MET_atmosphere.hh.

The documentation for this class was generated from the following file:

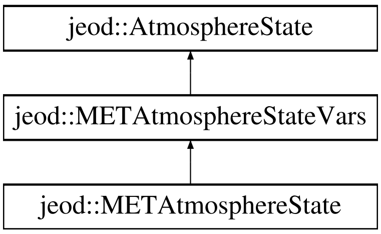
- MET_atmosphere.hh

8.9 jeod::METAtmosphereState Class Reference

The MET specific implementation of AtmosphereState.

```
#include <MET_atmosphere_state.hh>
```

Inheritance diagram for jeod::METAtmosphereState:



Public Member Functions

- METAtmosphereState (METAtmosphere &atmos_model, const PlanetFixedPosition &pfix_pos)
- METAtmosphereState ()=default
- ∼METAtmosphereState () override=default
- METAtmosphereState & operator= (const METAtmosphereState &)=delete
- METAtmosphereState (const METAtmosphereState &)=delete
- void update_state (METAtmosphere ∗atmos_model, const PlanetFixedPosition ∗pfix_pos)

  Updates the METAtmosphereState from the METAtmosphere pointed to by atmos_model_.
- void update_state () override

  Updates the METAtmosphereState from the METAtmosphere pointed to by class member atmos_model using class member pointer pfix_pos.

Private Attributes

- METAtmosphere ∗ met_atmos {}

Friends

- class InputProcessor
- void init_attrjeod__METAtmosphereState ()

Additional Inherited Members

8.9.1 Detailed Description

The MET specific implementation of AtmosphereState.

Definition at line 84 of file MET_atmosphere_state.hh.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 METAtmosphereState() [1/3]

```
jeod::METAtmosphereState::METAtmosphereState (
            METAtmosphere & atmos_model,
            const PlanetFixedPosition & pfix_pos )
```

Definition at line 51 of file MET_atmosphere_state.cc.

8.9.2.2 METAtmosphereState() [2/3]

```
jeod::METAtmosphereState::METAtmosphereState ( )  [default]
```

8.9.2.3 ∼METAtmosphereState()

```
jeod::METAtmosphereState::∼METAtmosphereState ( )  [override], [default]
```

8.9.2.4 METAtmosphereState() [3/3]

```
jeod::METAtmosphereState::METAtmosphereState (
            const METAtmosphereState &  )  [delete]
```

8.9.3 Member Function Documentation

8.9.3.1 operator=()

```
METAtmosphereState& jeod::METAtmosphereState::operator= (
            const METAtmosphereState &  )  [delete]
```

8.9.3.2 update_state() [1/2]

```
void jeod::METAtmosphereState::update_state (
            METAtmosphere * atmos_model_,
            const PlanetFixedPosition * pfix_pos_ )
```

Updates the METAtmosphereState from the METAtmosphere pointed to by atmos_model_.

This is a specific function for the case of an METAtmosphere state updating an METAtmosphere

Parameters

in	*atmos_model_*	METAtmosphere Model.
in	*pfix_pos_*	Current vehicle position.

Definition at line 65 of file MET_atmosphere_state.cc.

References jeod::AtmosphereState::active, and jeod::METAtmosphere::update_atmosphere().

8.9.3.3 update_state() [2/2]

```
void jeod::METAtmosphereState::update_state ( )  [override], [virtual]
```

Updates the METAtmosphereState from the METAtmosphere pointed to by class member atmos_model using class member pointer pfix_pos.

This is a specific function for the case of an METAtmosphere state updating an METAtmosphere when constructed with the pointers set.

Reimplemented from jeod::AtmosphereState.

Definition at line 80 of file MET_atmosphere_state.cc.

References jeod::AtmosphereState::active, met_atmos, jeod::AtmosphereState::pfix_pos, and jeod::METAtmosphere::update_atmosphere().

8.9.4 Friends And Related Function Documentation

8.9.4.1 init_attrjeod__METAtmosphereState

```
void init_attrjeod__METAtmosphereState ( )  [friend]
```

8.9.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 86 of file MET_atmosphere_state.hh.

8.9.5 Field Documentation

8.9.5.1 met_atmos

```
METAtmosphere* jeod::METAtmosphereState::met_atmos {}  [private]
```

Definition at line 87 of file MET_atmosphere_state.hh.

Referenced by update_state().

The documentation for this class was generated from the following files:

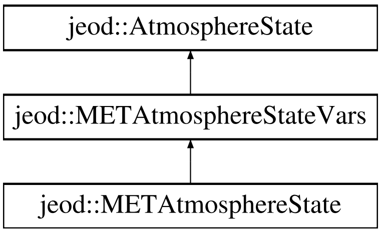
- MET_atmosphere_state.hh
- MET_atmosphere_state.cc

8.10 jeod::METAtmosphereStateVars Class Reference

The data variables component of the MET specific implementation of AtmosphereState.

```
#include <MET_atmosphere_state_vars.hh>
```

Inheritance diagram for jeod::METAtmosphereStateVars:



Public Member Functions

- METAtmosphereStateVars ()=default
- METAtmosphereStateVars (Atmosphere &atmos_model, const PlanetFixedPosition &pfix_pos)
- ∼METAtmosphereStateVars () override=default
- METAtmosphereStateVars (const METAtmosphereStateVars &rhs)

  Copy Constructor.
- METAtmosphereStateVars & operator= (const METAtmosphereStateVars &rhs)

  METAtmosphereStateVars operator =.

Data Fields

- double exo_temp {}
- double log10_dens {}
- double mol_weight {}
- double N2 {}
- double Ox2 {}
- double Ox {}
- double A {}
- double He {}
- double Hyd {}

Friends

- class InputProcessor
- void init_attrjeod__METAtmosphereStateVars ()

Additional Inherited Members

8.10.1 Detailed Description

The data variables component of the MET specific implementation of AtmosphereState.

Definition at line 82 of file MET_atmosphere_state_vars.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 METAtmosphereStateVars() [1/3]

```
jeod::METAtmosphereStateVars::METAtmosphereStateVars ( )  [default]
```

8.10.2.2 METAtmosphereStateVars() [2/3]

```
jeod::METAtmosphereStateVars::METAtmosphereStateVars (
            Atmosphere & atmos_model,
            const PlanetFixedPosition & pfix_pos )
```

Definition at line 49 of file MET_atmosphere_state_vars.cc.

8.10.2.3 ∼METAtmosphereStateVars()

```
jeod::METAtmosphereStateVars::∼METAtmosphereStateVars ( )  [override], [default]
```

8.10.2.4 METAtmosphereStateVars() [3/3]

```
jeod::METAtmosphereStateVars::METAtmosphereStateVars (
            const METAtmosphereStateVars & rhs )
```

Copy Constructor.

Parameters

in	*rhs*	The METAtmosphereStateVars to copy

Definition at line 59 of file MET_atmosphere_state_vars.cc.

References A, jeod::AtmosphereState::active, exo_temp, He, Hyd, log10_dens, mol_weight, N2, Ox, and Ox2.

8.10.3 Member Function Documentation

8.10.3.1 operator=()

```
METAtmosphereStateVars & jeod::METAtmosphereStateVars::operator= (
            const METAtmosphereStateVars & rhs )
```

METAtmosphereStateVars operator =.

Returns

The newly copied into METAtmosphereStateVars

Parameters

in	*rhs*	The METAtmosphereStateVars to copy from

Definition at line 80 of file MET_atmosphere_state_vars.cc.

References A, jeod::AtmosphereState::active, exo_temp, He, Hyd, log10_dens, mol_weight, N2, jeod::AtmosphereState::operator=(), Ox, and Ox2.

8.10.4 Friends And Related Function Documentation

8.10.4.1 init_attrjeod__METAtmosphereStateVars

```
void init_attrjeod__METAtmosphereStateVars ( )  [friend]
```

8.10.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 84 of file MET_atmosphere_state_vars.hh.

8.10.5 Field Documentation

8.10.5.1 A

```
double jeod::METAtmosphereStateVars::A {}
```

trick_units(–) A number density

Definition at line 91 of file MET_atmosphere_state_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update_atmosphere().

8.10.5.2 exo_temp

```
double jeod::METAtmosphereStateVars::exo_temp {}
```

trick_units(K) Exospheric temperature

Definition at line 85 of file MET_atmosphere_state_vars.hh.

Referenced by jeod::METAtmosphere::compute_exospheric_temperature(), METAtmosphereStateVars(), and operator=().

8.10.5.3 He

```
double jeod::METAtmosphereStateVars::He {}
```

trick_units(–) He number density

Definition at line 92 of file MET_atmosphere_state_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update_atmosphere().

8.10.5.4 Hyd

```
double jeod::METAtmosphereStateVars::Hyd {}
```

trick_units(–) H number density

Definition at line 93 of file MET_atmosphere_state_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update_atmosphere().

8.10.5.5 log10_dens

```
double jeod::METAtmosphereStateVars::log10_dens {}
```

trick_units(–) Log10(total density)

Definition at line 86 of file MET_atmosphere_state_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update_atmosphere().

8.10.5.6 mol_weight

```
double jeod::METAtmosphereStateVars::mol_weight {}
```

trick_units(–) Average molecular weight

Definition at line 87 of file MET_atmosphere_state_vars.hh.

Referenced by jeod::METAtmosphere::jacchia(), METAtmosphereStateVars(), operator=(), and jeod::MET↩ Atmosphere::update_atmosphere().

8.10.5.7 N2

```
double jeod::METAtmosphereStateVars::N2 {}
```

trick_units(–) N2 number density

Definition at line 88 of file MET_atmosphere_state_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update_atmosphere().

8.10.5.8 Ox

```
double jeod::METAtmosphereStateVars::Ox {}
```

trick_units(–) O number density

Definition at line 90 of file MET_atmosphere_state_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update_atmosphere().

8.10.5.9 Ox2

```
double jeod::METAtmosphereStateVars::Ox2 {}
```

trick_units(–) O2 number density

Definition at line 89 of file MET_atmosphere_state_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update_atmosphere().

The documentation for this class was generated from the following files:

- MET_atmosphere_state_vars.hh
- MET_atmosphere_state_vars.cc

8.11 jeod::METAtmosphereThermal Class Reference

The Thermal aspect of the computation.

```
#include <MET_atmosphere.hh>
```

Public Member Functions

- void update ()
- double compute_temperature (double altitude_km)
- METAtmosphereThermal (const double &T_exosphere, const double &altitude_km)
- virtual ∼METAtmosphereThermal ()=default
- METAtmosphereThermal & operator= (const METAtmosphereThermal &)=delete
- METAtmosphereThermal (const METAtmosphereThermal &)=delete

Data Fields

- double T_out {}

Private Member Functions

- void generate_base_temperature ()

Private Attributes

- const double k_1 {0.054285714}

  Temperature coefficients.
- const double k_3 {-3.96501457725948E-5}
- const double k_4 {-5.3311120366514E-7}
- const double T_90 {183.0}
- double T_125 {}
- const double & T_exosphere
- const double & altitude_km

Friends

- class InputProcessor
- void init_attrjeod__METAtmosphereThermal ()

8.11.1 Detailed Description

The Thermal aspect of the computation.

Definition at line 131 of file MET_atmosphere.hh.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 METAtmosphereThermal() [1/2]

```
jeod::METAtmosphereThermal::METAtmosphereThermal (
            const double & T_exosphere,
            const double & altitude_km )
```

Definition at line 77 of file MET_atmosphere.cc.

8.11.2.2 ∼METAtmosphereThermal()

```
virtual jeod::METAtmosphereThermal::∼METAtmosphereThermal ( )  [virtual], [default]
```

8.11.2.3 METAtmosphereThermal() [2/2]

```
jeod::METAtmosphereThermal::METAtmosphereThermal (
            const METAtmosphereThermal &  )  [delete]
```

8.11.3 Member Function Documentation

8.11.3.1 compute_temperature()

```
double jeod::METAtmosphereThermal::compute_temperature (
            double altitude_km )
```

Definition at line 147 of file MET_atmosphere.cc.

References k_1, k_3, k_4, T_125, T_90, and T_exosphere.

Referenced by jeod::METAtmosphere::apply_gauss_quadrature(), jeod::METAtmosphere::jacchia(), and update().

8.11.3.2 generate_base_temperature()

```
void jeod::METAtmosphereThermal::generate_base_temperature ( )  [private]
```

8.11.3.3 operator=()

```
METAtmosphereThermal& jeod::METAtmosphereThermal::operator= (
             const METAtmosphereThermal &  )  [delete]
```

8.11.3.4 update()

```
void jeod::METAtmosphereThermal::update ( )
```

Definition at line 99 of file MET_atmosphere.cc.

References altitude_km, compute_temperature(), T_125, T_exosphere, and T_out.

Referenced by jeod::METAtmosphere::jacchia().

8.11.4 Friends And Related Function Documentation

8.11.4.1 init_attrjeod__METAtmosphereThermal

```
void init_attrjeod__METAtmosphereThermal ( )  [friend]
```

8.11.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 133 of file MET_atmosphere.hh.

8.11.5 Field Documentation

8.11.5.1 altitude_km

```
const double& jeod::METAtmosphereThermal::altitude_km  [private]
```

Definition at line 169 of file MET_atmosphere.hh.

Referenced by update().

8.11.5.2 k_1

```
const double jeod::METAtmosphereThermal::k_1 {0.054285714}  [private]
```

Temperature coefficients.

trick_units(1/m) parameter used to obtain the first coefficient of the temperature polynomial, which is also the temperature gradient at 125km.

Definition at line 147 of file MET_atmosphere.hh.

Referenced by compute_temperature().

8.11.5.3 k_3

```
const double jeod::METAtmosphereThermal::k_3 {-3.96501457725948E-5}  [private]
```

trick_units(1/m3) parameter used to obtain the 3rd coefficient of the temperature polynomial.

Definition at line 152 of file MET_atmosphere.hh.

Referenced by compute_temperature().

8.11.5.4 k_4

```
const double jeod::METAtmosphereThermal::k_4 {-5.3311120366514E-7}  [private]
```

trick_units(1/m4) parameter used to obtain the 4th coefficient of the temperature polynomial.

Definition at line 156 of file MET_atmosphere.hh.

Referenced by compute_temperature().

8.11.5.5 T_125

```
double jeod::METAtmosphereThermal::T_125 {}  [private]
```

trick_units(K) Temperature at 125km reference point.

Definition at line 163 of file MET_atmosphere.hh.

Referenced by compute_temperature(), and update().

8.11.5.6 T_90

```
const double jeod::METAtmosphereThermal::T_90 {183.0}  [private]
```

trick_units(K) Temperature at 90km reference point.

Definition at line 160 of file MET_atmosphere.hh.

Referenced by compute_temperature().

8.11.5.7 T_exosphere

```
const double& jeod::METAtmosphereThermal::T_exosphere  [private]
```

Definition at line 166 of file MET_atmosphere.hh.

Referenced by compute_temperature(), and update().

8.11.5.8 T_out

```
double jeod::METAtmosphereThermal::T_out {}
```

Definition at line 134 of file MET_atmosphere.hh.

Referenced by jeod::METAtmosphere::jacchia(), and update().

The documentation for this class was generated from the following files:

- MET_atmosphere.hh
- MET_atmosphere.cc

8.12 jeod::WindVelocity::OmegaTableEntry Struct Reference

An entry in an omega scale table.

```
#include <wind_velocity.hh>
```

Data Fields

- double altitude

  Altitude at which omega is multiplied by the corresponding factor.
- double scale_factor

  Factor by which omega is multiplied depending on altitude.

8.12.1 Detailed Description

An entry in an omega scale table.

Definition at line 107 of file wind_velocity.hh.

8.12.2 Field Documentation

8.12.2.1 altitude

```
double jeod::WindVelocity::OmegaTableEntry::altitude
```

Altitude at which omega is multiplied by the corresponding factor.

trick_units(m)

Definition at line 112 of file wind_velocity.hh.

Referenced by jeod::WindVelocity::set_omega_scale_table(), and jeod::WindVelocity::update_wind().

8.12.2.2 scale_factor

```
double jeod::WindVelocity::OmegaTableEntry::scale_factor
```

Factor by which omega is multiplied depending on altitude.

trick_units(–)

Definition at line 117 of file wind_velocity.hh.

Referenced by jeod::WindVelocity::set_omega_scale_table(), and jeod::WindVelocity::update_wind().

The documentation for this struct was generated from the following file:

- wind_velocity.hh

8.13 jeod::WindVelocity Class Reference

A generic wind velocity implementation.

```
#include <wind_velocity.hh>
```

Data Structures

- struct OmegaTableEntry

  An entry in an omega scale table.

Public Member Functions

- WindVelocity ()=default
- virtual ∼WindVelocity ()

  Destructor.
- WindVelocity (const WindVelocity &)=delete
- WindVelocity & operator= (const WindVelocity &)=delete
- virtual void update_wind (double inertial_pos[3], double altitude, double wind_inertial[3])

  Updates the wind velocity from the parameters given.
- unsigned int get_num_layers ()
- void set_omega_scale_table (double altitude, double factor)
- void set_omega_scale_table (unsigned int num_layers, const double ∗altitude, const double ∗factor)
- OmegaTableEntry ∗ get_omega_scale_table ()

Data Fields

- bool active {true}

  trick_units(–)
- double omega {}

  The rotational velocity of the planet.

Protected Attributes

- unsigned int num_layers {}

  Number of altitude layers.
- OmegaTableEntry ∗ omega_scale_table {}

  Table of factors to scale omega based on altitude.

Private Attributes

- unsigned int array_index {}

  last known index into the arrays
- bool first_pass {true}

  Altitude direction check flag.
- bool increasing_altitude {true}

  Altitude increasing or decreasing flag.

Friends

- class InputProcessor
- void init_attrjeod__WindVelocity ()

8.13.1 Detailed Description

A generic wind velocity implementation.

Definition at line 76 of file wind_velocity.hh.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 WindVelocity() [1/2]

```
jeod::WindVelocity::WindVelocity ( )  [default]
```

8.13.2.2 ∼WindVelocity()

```
jeod::WindVelocity::∼WindVelocity ( )  [virtual]
```

Destructor.

Definition at line 42 of file wind_velocity.cc.

References omega_scale_table.

8.13.2.3 WindVelocity() [2/2]

```
jeod::WindVelocity::WindVelocity (
            const WindVelocity &  )  [delete]
```

8.13.3 Member Function Documentation

8.13.3.1 get_num_layers()

```
unsigned int jeod::WindVelocity::get_num_layers ( )
```

Definition at line 212 of file wind_velocity.cc.

References num_layers.

8.13.3.2 get_omega_scale_table()

```
WindVelocity::OmegaTableEntry * jeod::WindVelocity::get_omega_scale_table ( )
```

Definition at line 248 of file wind_velocity.cc.

References omega_scale_table.

8.13.3.3 operator=()

```
WindVelocity& jeod::WindVelocity::operator= (
             const WindVelocity & ) [delete]
```

8.13.3.4 set_omega_scale_table() [1/2]

```
void jeod::WindVelocity::set_omega_scale_table (
             double altitude,
             double factor )
```

Definition at line 217 of file wind_velocity.cc.

References jeod::WindVelocity::OmegaTableEntry::altitude, num_layers, omega_scale_table, and jeod::WindVelocity::OmegaTableEntry::scale_factor.

Referenced by jeod::WindVelocity_wind_velocity_default_data::initialize().

8.13.3.5 set_omega_scale_table() [2/2]

```
void jeod::WindVelocity::set_omega_scale_table (
             unsigned int num_layers,
             const double * altitude,
             const double * factor )
```

Definition at line 226 of file wind_velocity.cc.

References jeod::WindVelocity::OmegaTableEntry::altitude, jeod::AtmosphereMessages::framework_error, num_layers, omega_scale_table, and jeod::WindVelocity::OmegaTableEntry::scale_factor.

8.13.3.6 update_wind()

```
void jeod::WindVelocity::update_wind (
             double inertial_pos[3],
             double altitude,
             double wind_inertial[3] ) [virtual]
```

Updates the wind velocity from the parameters given.

Parameters

in	*inertial_pos*	The inertial position of the vehicle Units: M
in	*altitude*	The altitude of the vehicle Units: M
out	*wind_inertial*	The wind, in the inertial frame, applied to the vehicle Units: M/s

Definition at line 53 of file wind_velocity.cc.

References active, jeod::WindVelocity::OmegaTableEntry::altitude, array_index, first_pass, jeod::Atmosphere↩ Messages::framework_error, increasing_altitude, num_layers, omega, omega_scale_table, and jeod::Wind↩ Velocity::OmegaTableEntry::scale_factor.

Referenced by jeod::AtmosphereState::update_wind().

8.13.4 Friends And Related Function Documentation

8.13.4.1 init_attrjeod__WindVelocity

```
void init_attrjeod__WindVelocity ( )  [friend]
```

8.13.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 78 of file wind_velocity.hh.

8.13.5 Field Documentation

8.13.5.1 active

```
bool jeod::WindVelocity::active {true}
```

trick_units(–)

Definition at line 95 of file wind_velocity.hh.

Referenced by update_wind().

8.13.5.2 array_index

```
unsigned int jeod::WindVelocity::array_index {}  [private]
```

last known index into the arrays

Definition at line 137 of file wind_velocity.hh.

Referenced by update_wind().

8.13.5.3 first_pass

```
bool jeod::WindVelocity::first_pass {true}  [private]
```

Altitude direction check flag.

trick_units(–)

Definition at line 142 of file wind_velocity.hh.

Referenced by update_wind().

8.13.5.4 increasing_altitude

```
bool jeod::WindVelocity::increasing_altitude {true}  [private]
```

Altitude increasing or decreasing flag.

trick_units(–)

Definition at line 147 of file wind_velocity.hh.

Referenced by update_wind().

8.13.5.5 num_layers

```
unsigned int jeod::WindVelocity::num_layers {}  [protected]
```

Number of altitude layers.

trick_units(count)

Definition at line 126 of file wind_velocity.hh.

Referenced by get_num_layers(), set_omega_scale_table(), and update_wind().

8.13.5.6 omega

```
double jeod::WindVelocity::omega {}
```

The rotational velocity of the planet.

trick_units(rad/s)

Definition at line 100 of file wind_velocity.hh.

Referenced by jeod::WindVelocity_wind_velocity_default_data::initialize(), and update_wind().

8.13.5.7 omega_scale_table

```
OmegaTableEntry* jeod::WindVelocity::omega_scale_table {}  [protected]
```

Table of factors to scale omega based on altitude.

Definition at line 131 of file wind_velocity.hh.

Referenced by get_omega_scale_table(), set_omega_scale_table(), update_wind(), and ∼WindVelocity().

The documentation for this class was generated from the following files:

- wind_velocity.hh
- wind_velocity.cc

8.14 jeod::WindVelocity_wind_velocity_default_data Class Reference

```
#include <met_data_wind_velocity.hh>
```

Public Member Functions

- WindVelocity_wind_velocity_default_data ()=default
- void initialize (WindVelocity ∗)
- void initialize (WindVelocity &)

Data Fields

- double omega_scale_fac [num_layers]
- double omega_scale_alt [num_layers]
- double omega {7.292115146706388e-5}

Static Public Attributes

- static const int num_layers = 12

8.14.1 Detailed Description

Definition at line 57 of file met_data_wind_velocity.hh.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 WindVelocity_wind_velocity_default_data()

```
jeod::WindVelocity_wind_velocity_default_data::WindVelocity_wind_velocity_default_data ( )
[default]
```

8.14.3 Member Function Documentation

8.14.3.1 initialize() [1/2]

```
void jeod::WindVelocity_wind_velocity_default_data::initialize (
            WindVelocity * WindVelocity_ptr )
```

Definition at line 42 of file data_met_wind_velocity.cc.

8.14.3.2 initialize() [2/2]

```
void jeod::WindVelocity_wind_velocity_default_data::initialize (
            WindVelocity & wind_velocity )
```

Definition at line 56 of file data_met_wind_velocity.cc.

References num_layers, omega, jeod::WindVelocity::omega, omega_scale_alt, omega_scale_fac, and jeod::WindVelocity::set_omega_scale_table().

8.14.4 Field Documentation

8.14.4.1 num_layers

```
const int jeod::WindVelocity_wind_velocity_default_data::num_layers = 12  [static]
```

Definition at line 60 of file met_data_wind_velocity.hh.

Referenced by initialize().

8.14.4.2 omega

```
double jeod::WindVelocity_wind_velocity_default_data::omega {7.292115146706388e-5}
```

Definition at line 79 of file met_data_wind_velocity.hh.

Referenced by initialize().

8.14.4.3 omega_scale_alt

```
double jeod::WindVelocity_wind_velocity_default_data::omega_scale_alt[num_layers]
```

Initial value:

```
{180000.0,
                                              200000.0,
                                              220000.0,
                                              240000.0,
                                              260000.0,
                                              280000.0,
                                              300000.0,
                                              320000.0,
                                              340000.0,
                                              360000.0,
                                              380000.0,
                                              400000.0}
```

Definition at line 66 of file met_data_wind_velocity.hh.

Referenced by initialize().

8.14.4.4 omega_scale_fac

```
double jeod::WindVelocity_wind_velocity_default_data::omega_scale_fac[num_layers]
```

Initial value:

```
{
        1.0, 1.1, 1.16, 1.21, 1.25, 1.3, 1.34, 1.38, 1.4, 1.405, 1.41, 1.4142136}
```

Definition at line 62 of file met_data_wind_velocity.hh.

Referenced by initialize().

The documentation for this class was generated from the following files:

- met_data_wind_velocity.hh
- data_met_wind_velocity.cc

8.15 jeod::WindVelocityBase Class Reference

The generic base class for wind velocity classes.

```
#include <wind_velocity_base.hh>
```

Public Member Functions

- WindVelocityBase ()=default
- virtual ∼WindVelocityBase ()=default
- WindVelocityBase (const WindVelocityBase &)=delete
- WindVelocityBase & operator= (const WindVelocityBase &)=delete
- virtual void update_wind (double position[3], double altitude, double wind_inertial[3])

  Virtual function to define the interface for inheriting functions.

Friends

- class InputProcessor
- void init_attrjeod__WindVelocityBase ()

8.15.1 Detailed Description

The generic base class for wind velocity classes.

This class has questionable purpose because of its extremely limited capability but is left here for backward compatibility. It should not be used.

Definition at line 77 of file wind_velocity_base.hh.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 WindVelocityBase() [1/2]

```
jeod::WindVelocityBase::WindVelocityBase ( )  [default]
```

8.15.2.2 ∼WindVelocityBase()

```
virtual jeod::WindVelocityBase::∼WindVelocityBase ( )  [virtual], [default]
```

8.15.2.3 WindVelocityBase() [2/2]

```
jeod::WindVelocityBase::WindVelocityBase (
            const WindVelocityBase &  )  [delete]
```

8.15.3 Member Function Documentation

8.15.3.1 operator=()

```
WindVelocityBase& jeod::WindVelocityBase::operator= (
            const WindVelocityBase &  )  [delete]
```

8.15.3.2 update_wind()

```
void jeod::WindVelocityBase::update_wind (
            double position[3],
            double altitude,
            double wind_inertial[3] )  [virtual]
```

Virtual function to define the interface for inheriting functions.

Parameters

in	*position*	The position of the vehicle, however the specific implementation defines it
in	*altitude*	The altitude of the vehicle, however the specific implementation defines it
out	*wind_inertial*	The wind applied to the craft, in the inertial frame

Definition at line 38 of file wind_velocity_base.cc.

References jeod::AtmosphereMessages::framework_warning.

8.15.4 Friends And Related Function Documentation

8.15.4.1 init_attrjeod__WindVelocityBase

```
void init_attrjeod__WindVelocityBase ( )  [friend]
```

8.15.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 79 of file wind_velocity_base.hh.

The documentation for this class was generated from the following files:

- wind_velocity_base.hh
- wind_velocity_base.cc

Chapter 9

File Documentation

9.1 atmosphere.hh File Reference

General base class for atmosphere models.

```
#include "environment/time/include/time_standard.hh"
#include "utils/planet_fixed/planet_fixed_posn/include/planet_fixed_posn.↩
hh"
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class jeod::Atmosphere

  A generic base class for atmospheres.

Namespaces

- jeod

  Namespace jeod.

9.1.1 Detailed Description

General base class for atmosphere models.

9.2 atmosphere_messages.cc File Reference

Implement atmosphere_messages.

```
#include "../include/atmosphere_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define PATH "environment/atmosphere/base_atmos"

9.2.1 Detailed Description

Implement atmosphere_messages.

9.3 atmosphere_messages.hh File Reference

Implement atmosphere_messages.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class jeod::AtmosphereMessages

  Describes messages used in the Atmosphere model.

Namespaces

- jeod

  Namespace jeod.

9.3.1 Detailed Description

Implement atmosphere_messages.

9.4 atmosphere_state.cc File Reference

Implementation of the base atmosphere-state model.

```
#include <cstddef>
#include "../include/atmosphere_state.hh"
#include "../include/wind_velocity.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.4.1 Detailed Description

Implementation of the base atmosphere-state model.

9.5 atmosphere_state.hh File Reference

```
#include "environment/time/include/time_standard.hh"
#include "utils/planet_fixed/planet_fixed_posn/include/planet_fixed_posn.↩
hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "atmosphere.hh"
#include "wind_velocity.hh"
```

Data Structures

- class jeod::AtmosphereState

  A generic base class for atmosphere state, containing common atmosphere state parameters, i.e.

Namespaces

- jeod

  Namespace jeod.

9.6 class_declarations.hh File Reference

Forward declarations of classes defined for JEOD 2.0 Atmosphere.

Namespaces

- jeod

  Namespace jeod.

9.6.1 Detailed Description

Forward declarations of classes defined for JEOD 2.0 Atmosphere.

9.7 class_declarations.hh File Reference

Forward declarations of classes defined for JEOD 2.0 Atmosphere.

Namespaces

- jeod

  Namespace jeod.

9.7.1 Detailed Description

Forward declarations of classes defined for JEOD 2.0 Atmosphere.

9.8 data_met_wind_velocity.cc File Reference

```
#include <cstddef>
#include "environment/atmosphere/base_atmos/include/wind_velocity.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/met_data_wind_velocity.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define JEOD_FRIEND_CLASS WindVelocity_wind_velocity_default_data

9.8.1 Macro Definition Documentation

9.8.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS WindVelocity_wind_velocity_default_data
```

Definition at line 21 of file data_met_wind_velocity.cc.

9.9 MET_atmosphere.cc File Reference

Implementation of MET atmosphere model.

```
#include <algorithm>
#include <cstddef>
#include <cstring>
#include <cmath>
#include "environment/time/include/time_utc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/MET_atmosphere.hh"
#include "environment/atmosphere/base_atmos/include/atmosphere_messages.hh"
```

Namespaces

- jeod

    Namespace jeod.

Macros

- #define _USE_MATH_DEFINES_

9.9.1 Detailed Description

Implementation of MET atmosphere model.

9.10 MET_atmosphere.hh File Reference

Implement the MET atmosphere using the atmosphere framework.

```
#include "environment/time/include/time_utc.hh"
#include "utils/math/include/gauss_quadrature.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "MET_atmosphere_state_vars.hh"
#include "environment/atmosphere/base_atmos/include/atmosphere.hh"
```

Data Structures

- class jeod::METAtmosphereChemical

    The chemical composition of the MET Atmosphere.
- class jeod::METAtmosphereThermal

    The Thermal aspect of the computation.
- class jeod::METAtmosphere

Namespaces

- jeod

  Namespace jeod.

9.10.1 Detailed Description

Implement the MET atmosphere using the atmosphere framework.

9.11 MET_atmosphere_state.cc File Reference

```
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "../include/MET_atmosphere_state.hh"
#include "environment/atmosphere/base_atmos/include/atmosphere_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.12 MET_atmosphere_state.hh File Reference

Implement the MET atmosphere state using the atmosphere framework.

```
#include "utils/planet_fixed/planet_fixed_posn/include/planet_fixed_posn.↩
hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "MET_atmosphere.hh"
#include "MET_atmosphere_state_vars.hh"
```

Data Structures

- class jeod::METAtmosphereState

  The MET specific implementation of AtmosphereState.

Namespaces

- jeod

  Namespace jeod.

9.12.1 Detailed Description

Implement the MET atmosphere state using the atmosphere framework.

9.13 MET_atmosphere_state_vars.cc File Reference

Implementation of MET atmosphere model.

```
#include "../include/MET_atmosphere_state_vars.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.13.1 Detailed Description

Implementation of MET atmosphere model.

9.14 MET_atmosphere_state_vars.hh File Reference

Implement the MET atmosphere state variables using the atmosphere framework.

```
#include "utils/planet_fixed/planet_fixed_posn/include/planet_fixed_posn.↩
hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "environment/atmosphere/base_atmos/include/atmosphere.hh"
#include "environment/atmosphere/base_atmos/include/atmosphere_state.hh"
```

Data Structures

- class jeod::METAtmosphereStateVars

  The data variables component of the MET specific implementation of AtmosphereState.

Namespaces

- jeod

  Namespace jeod.

9.14.1 Detailed Description

Implement the MET atmosphere state variables using the atmosphere framework.

9.15 met_data_wind_velocity.hh File Reference

```
#include "utils/message/include/message_handler.hh"
```

Data Structures

- class jeod::WindVelocity_wind_velocity_default_data

Namespaces

- jeod

  Namespace jeod.

9.16 solar_max.cc File Reference

```
#include "environment/atmosphere/MET/include/MET_atmosphere.hh"
#include "../include/solar_max.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define JEOD_FRIEND_CLASS METAtmosphere_solar_max_default_data

9.16.1 Macro Definition Documentation

9.16.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS METAtmosphere_solar_max_default_data
```

Definition at line 23 of file solar_max.cc.

9.17 solar_max.hh File Reference

Data Structures

- class jeod::METAtmosphere_solar_max_default_data

Namespaces

- jeod

  Namespace jeod.

9.18 solar_mean.cc File Reference

```
#include "environment/atmosphere/MET/include/MET_atmosphere.hh"
#include "../include/solar_mean.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define JEOD_FRIEND_CLASS METAtmosphere_solar_mean_default_data

9.18.1 Macro Definition Documentation

9.18.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS METAtmosphere_solar_mean_default_data
```

Definition at line 23 of file solar_mean.cc.

9.19 solar_mean.hh File Reference

Data Structures

- class jeod::METAtmosphere_solar_mean_default_data

Namespaces

- jeod

  Namespace jeod.

9.20 solar_min.cc File Reference

```
#include "environment/atmosphere/MET/include/MET_atmosphere.hh"
#include "../include/solar_min.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define JEOD_FRIEND_CLASS METAtmosphere_solar_min_default_data

9.20.1 Macro Definition Documentation

9.20.1.1 JEOD_FRIEND_CLASS

```
#define JEOD_FRIEND_CLASS METAtmosphere_solar_min_default_data
```

Definition at line 23 of file solar_min.cc.

9.21 solar_min.hh File Reference

Data Structures

- class jeod::METAtmosphere_solar_min_default_data

Namespaces

- jeod

  Namespace jeod.

9.22 wind_velocity.cc File Reference

General base class for wind velocity models.

```
#include <cstddef>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/atmosphere_messages.hh"
#include "../include/wind_velocity.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.22.1 Detailed Description

General base class for wind velocity models.

9.23 wind_velocity.hh File Reference

A wind velocity model based on winds caused by rotation of the planet.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class jeod::WindVelocity

  A generic wind velocity implementation.
- struct jeod::WindVelocity::OmegaTableEntry

  An entry in an omega scale table.

Namespaces

- jeod

  Namespace jeod.

9.23.1 Detailed Description

A wind velocity model based on winds caused by rotation of the planet.

9.24 wind_velocity_base.cc File Reference

General base class for wind velocity models.

```
#include "../include/wind_velocity_base.hh"
#include "../include/atmosphere_messages.hh"
#include "utils/message/include/message_handler.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.24.1 Detailed Description

General base class for wind velocity models.

9.25 wind_velocity_base.hh File Reference

General base class for wind velocity models.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class jeod::WindVelocityBase

  The generic base class for wind velocity classes.

Namespaces

- jeod

  Namespace jeod.

9.25.1 Detailed Description

General base class for wind velocity models.

Index