AtmosphereModel

5.0

Generated by Doxygen 1.8.5

Wed Jun 1 2022 12:05:25

Contents

1 Module Index **1**

1.1 Modules 1

2 Namespace Index **3**

2.1 Namespace List 3

3 Hierarchical Index **5**

3.1 Class Hierarchy 5

4 Data Structure Index **7**

4.1 Data Structures 7

5 File Index **9**

5.1 File List 9

6 Module Documentation **11**

6.1 Models 11

6.1.1 Detailed Description 12

6.1.2 Function Documentation 12

6.1.2.1 Atmosphere 12

6.1.2.2 Atmosphere 12

6.1.2.3 operator= 12

6.1.2.4 operator= 12

6.1.2.5 update_atmosphere 12

6.1.2.6 update_wind 12

6.1.2.7 WindVelocity 13

6.1.2.8 WindVelocity 13

6.1.2.9 ∼Atmosphere 13

6.1.2.10 ∼WindVelocity 13

6.1.3 Variable Documentation 13

6.1.3.1 active 13

6.1.3.2 active 13

6.1.3.3 array_index 13

6.1.3.4 num_layers 14

6.1.3.5 omega 14

6.1.3.6 omega_scale_alt 14

6.1.3.7 omega_scale_fac 14

6.1.4 Friends 14

6.1.4.1 init_attrjeod__Atmosphere 14

6.1.4.2 init_attrjeod__WindVelocity 14

6.1.4.3 InputProcessor 14

6.1.4.4 InputProcessor 14

6.2 Environment 15

6.2.1 Detailed Description 15

6.3 Atmosphere 16

6.3.1 Detailed Description 17

6.3.2 Macro Definition Documentation 17

6.3.2.1 _USE_MATH_DEFINES_ 17

6.3.2.2 PATH 17

6.4 BaseAtmosphere 18

6.4.1 Detailed Description 18

7 Namespace Documentation 19

7.1 jeod Namespace Reference 19

7.1.1 Detailed Description 19

8 Data Structure Documentation 21

8.1 jeod::Atmosphere Class Reference 21

8.1.1 Detailed Description 22

8.2 jeod::AtmosphereMessages Class Reference 22

8.2.1 Detailed Description 22

8.2.2 Constructor & Destructor Documentation 22

8.2.2.1 AtmosphereMessages 22

8.2.2.2 AtmosphereMessages 22

8.2.3 Member Function Documentation 22

8.2.3.1 operator= 23

8.2.4 Friends And Related Function Documentation 23

8.2.4.1 init_attrjeod__AtmosphereMessages 23

8.2.4.2 InputProcessor 23

8.2.5 Field Documentation 23

8.2.5.1 framework_error 23

8.2.5.2 framework_warning 23

8.2.5.3 initialization_error 23

8.2.5.4 numerical_warning 23

8.3 jeod::AtmosphereState Class Reference 24
8.3.1 Detailed Description 25
8.3.2 Constructor & Destructor Documentation 25
8.3.2.1 AtmosphereState 25
8.3.2.2 AtmosphereState 25
8.3.2.3 ∼AtmosphereState 25
8.3.2.4 AtmosphereState 25
8.3.3 Member Function Documentation 26
8.3.3.1 operator= 26
8.3.3.2 update_state 26
8.3.3.3 update_state 26
8.3.3.4 update_wind 26
8.3.4 Friends And Related Function Documentation 27
8.3.4.1 init_attrjeod__AtmosphereState 27
8.3.4.2 InputProcessor 27
8.3.5 Field Documentation 27
8.3.5.1 active 27
8.3.5.2 atmos 27
8.3.5.3 density 27
8.3.5.4 pfix_pos 27
8.3.5.5 pressure 27
8.3.5.6 temperature 28
8.3.5.7 wind 28
8.4 jeod::METAtmosphere Class Reference 28
8.4.1 Detailed Description 30
8.4.2 Member Enumeration Documentation 30
8.4.2.1 AtmosMETGeoIndexType 30
8.4.3 Constructor & Destructor Documentation 30
8.4.3.1 METAtmosphere 30
8.4.3.2 ∼METAtmosphere 30
8.4.3.3 METAtmosphere 30
8.4.4 Member Function Documentation 30
8.4.4.1 apply_gauss_quadrature 30
8.4.4.2 atmos_MET_FAIR5 31
8.4.4.3 compute_exospheric_temperature 31
8.4.4.4 compute_mol_wt 31
8.4.4.5 compute_seasonal_lat_variation_He 31
8.4.4.6 compute_seasonal_latitude_variation 31
8.4.4.7 compute_solar_angles 31
8.4.4.8 jacchia 31

8.4.4.9 modify_densities 32

8.4.4.10 operator= 32

8.4.4.11 update_atmosphere 32

8.4.4.12 update_atmosphere 32

8.4.4.13 update_atmosphere 32

8.4.4.14 update_time 33

8.4.5 Friends And Related Function Documentation 33

8.4.5.1 init_attrjeod__METAtmosphere 33

8.4.5.2 InputProcessor 33

8.4.6 Field Documentation 33

8.4.6.1 altitude_km 33

8.4.6.2 Avogadro 33

8.4.6.3 barometric_equation_ceiling 33

8.4.6.4 base_fairing_height 34

8.4.6.5 day_of_year 34

8.4.6.6 days_per_century 34

8.4.6.7 days_per_year 34

8.4.6.8 deg_to_rad 34

8.4.6.9 F10 34

8.4.6.10 F10B 35

8.4.6.11 fairing_k 35

8.4.6.12 fraction_of_year 35

8.4.6.13 gauss_altitudes 35

8.4.6.14 gauss_n 35

8.4.6.15 geo_index 36

8.4.6.16 geo_index_type 36

8.4.6.17 latitude 36

8.4.6.18 longitude 36

8.4.6.19 max_days_this_year 36

8.4.6.20 minutes_per_day 36

8.4.6.21 mol_weight_barometric_ceiling 36

8.4.6.22 mol_wt_coeffs 37

8.4.6.23 num_integ_divisions 37

8.4.6.24 num_mol_wt_coeffs 37

8.4.6.25 R_gas_constant 37

8.4.6.26 solar_declination_angle 37

8.4.6.27 solar_hour_angle 37

8.4.6.28 species 38

8.4.6.29 state 38

8.4.6.30 thermal 38

8.4.6.31 three_pi_two 38
8.4.6.32 tjt_year_start 38
8.4.6.33 trunc_julian_time 38
8.4.6.34 two_pi 39
8.4.6.35 year 39
8.5 jeod::METAtmosphere_solar_max_default_data Class Reference 39
8.5.1 Detailed Description 39
8.5.2 Member Function Documentation 39
8.5.2.1 initialize 39
8.6 jeod::METAtmosphere_solar_mean_default_data Class Reference 39
8.6.1 Detailed Description 40
8.6.2 Member Function Documentation 40
8.6.2.1 initialize 40
8.7 jeod::METAtmosphere_solar_min_default_data Class Reference 40
8.7.1 Detailed Description 40
8.7.2 Member Function Documentation 40
8.7.2.1 initialize 40
8.8 jeod::METAtmosphereChemical Class Reference 40
8.8.1 Detailed Description 41
8.8.2 Constructor & Destructor Documentation 41
8.8.2.1 METAtmosphereChemical 41
8.8.2.2 ∼METAtmosphereChemical 41
8.8.2.3 METAtmosphereChemical 41
8.8.3 Member Function Documentation 41
8.8.3.1 operator= 41
8.8.4 Friends And Related Function Documentation 41
8.8.4.1 init_attrjeod__METAtmosphereChemical 41
8.8.4.2 InputProcessor 42
8.8.5 Field Documentation 42
8.8.5.1 frac 42
8.8.5.2 mol_weight 42
8.8.5.3 nominal_mol_weight 42
8.8.5.4 num_density 42
8.8.5.5 num_species 42
8.9 jeod::METAtmosphereState Class Reference 42
8.9.1 Detailed Description 43
8.9.2 Constructor & Destructor Documentation 43
8.9.2.1 METAtmosphereState 43
8.9.2.2 METAtmosphereState 43
8.9.2.3 ∼METAtmosphereState 44

8.9.2.4 METAtmosphereState . . . 44
8.9.3 Member Function Documentation . . . 44
8.9.3.1 operator= . . . 44
8.9.3.2 update_state . . . 44
8.9.3.3 update_state . . . 44
8.9.4 Friends And Related Function Documentation . . . 44
8.9.4.1 init_attrjeod__METAtmosphereState . . . 44
8.9.4.2 InputProcessor . . . 44
8.9.5 Field Documentation . . . 44
8.9.5.1 met_atmos . . . 44
8.10 jeod::METAtmosphereStateVars Class Reference . . . 45
8.10.1 Detailed Description . . . 45
8.10.2 Constructor & Destructor Documentation . . . 46
8.10.2.1 METAtmosphereStateVars . . . 46
8.10.2.2 METAtmosphereStateVars . . . 46
8.10.2.3 ∼METAtmosphereStateVars . . . 46
8.10.2.4 METAtmosphereStateVars . . . 46
8.10.3 Member Function Documentation . . . 46
8.10.3.1 operator= . . . 46
8.10.4 Friends And Related Function Documentation . . . 46
8.10.4.1 init_attrjeod__METAtmosphereStateVars . . . 46
8.10.4.2 InputProcessor . . . 46
8.10.5 Field Documentation . . . 47
8.10.5.1 A . . . 47
8.10.5.2 exo_temp . . . 47
8.10.5.3 He . . . 47
8.10.5.4 Hyd . . . 47
8.10.5.5 log10_dens . . . 47
8.10.5.6 mol_weight . . . 47
8.10.5.7 N2 . . . 47
8.10.5.8 Ox . . . 48
8.10.5.9 Ox2 . . . 48
8.11 jeod::METAtmosphereThermal Class Reference . . . 48
8.11.1 Detailed Description . . . 49
8.11.2 Constructor & Destructor Documentation . . . 49
8.11.2.1 METAtmosphereThermal . . . 49
8.11.2.2 ∼METAtmosphereThermal . . . 49
8.11.2.3 METAtmosphereThermal . . . 49
8.11.3 Member Function Documentation . . . 49
8.11.3.1 compute_temperature . . . 49

8.11.3.2 generate_base_temperature 49
8.11.3.3 operator= 49
8.11.3.4 update 49
8.11.4 Friends And Related Function Documentation 49
8.11.4.1 init_attrjeod__METAtmosphereThermal 49
8.11.4.2 InputProcessor 49
8.11.5 Field Documentation 50
8.11.5.1 altitude_km 50
8.11.5.2 k_1 50
8.11.5.3 k_3 50
8.11.5.4 k_4 50
8.11.5.5 T_125 50
8.11.5.6 T_90 50
8.11.5.7 T_exosphere 51
8.11.5.8 T_out 51
8.12 jeod::WindVelocity Class Reference 51
8.12.1 Detailed Description 52
8.13 jeod::WindVelocity_wind_velocity_default_data Class Reference 52
8.13.1 Detailed Description 52
8.13.2 Constructor & Destructor Documentation 52
8.13.2.1 WindVelocity_wind_velocity_default_data 52
8.13.3 Member Function Documentation 53
8.13.3.1 initialize 53
8.13.3.2 initialize 53
8.13.4 Field Documentation 53
8.13.4.1 num_layers 53
8.13.4.2 omega 53
8.13.4.3 omega_scale_alt 53
8.13.4.4 omega_scale_fac 53
8.14 jeod::WindVelocityBase Class Reference 53
8.14.1 Detailed Description 54
8.14.2 Constructor & Destructor Documentation 54
8.14.2.1 WindVelocityBase 54
8.14.2.2 ∼WindVelocityBase 54
8.14.2.3 WindVelocityBase 54
8.14.3 Member Function Documentation 54
8.14.3.1 operator= 54
8.14.3.2 update_wind 54
8.14.4 Friends And Related Function Documentation 55
8.14.4.1 init_attrjeod__WindVelocityBase 55

8.14.4.2 InputProcessor 55

9 File Documentation 57

9.1 atmosphere.hh File Reference 57

9.1.1 Detailed Description 57

9.2 atmosphere_messages.cc File Reference 57

9.2.1 Detailed Description 58

9.3 atmosphere_messages.hh File Reference 58

9.3.1 Detailed Description 58

9.4 atmosphere_state.cc File Reference 58

9.4.1 Detailed Description 58

9.5 atmosphere_state.hh File Reference 59

9.6 class_declarations.hh File Reference 59

9.6.1 Detailed Description 59

9.7 class_declarations.hh File Reference 59

9.7.1 Detailed Description 59

9.8 data_met_wind_velocity.cc File Reference 60

9.8.1 Macro Definition Documentation 60

9.8.1.1 JEOD_FRIEND_CLASS 60

9.9 MET_atmosphere.cc File Reference 60

9.9.1 Detailed Description 60

9.10 MET_atmosphere.hh File Reference 61

9.10.1 Detailed Description 61

9.11 MET_atmosphere_state.cc File Reference 61

9.12 MET_atmosphere_state.hh File Reference 61

9.12.1 Detailed Description 62

9.13 MET_atmosphere_state_vars.cc File Reference 62

9.13.1 Detailed Description 62

9.14 MET_atmosphere_state_vars.hh File Reference 62

9.14.1 Detailed Description 63

9.15 met_data_wind_velocity.hh File Reference 63

9.16 solar_max.cc File Reference 63

9.16.1 Macro Definition Documentation 63

9.16.1.1 JEOD_FRIEND_CLASS 63

9.17 solar_max.hh File Reference 64

9.18 solar_mean.cc File Reference 64

9.18.1 Macro Definition Documentation 64

9.18.1.1 JEOD_FRIEND_CLASS 64

9.19 solar_mean.hh File Reference 64

9.20 solar_min.cc File Reference 64

9.20.1 Macro Definition Documentation 65
9.20.1.1 JEOD_FRIEND_CLASS 65
9.21 solar_min.hh File Reference 65
9.22 wind_velocity.cc File Reference 65
9.22.1 Detailed Description 65
9.23 wind_velocity.hh File Reference 66
9.24 wind_velocity_base.cc File Reference 66
9.24.1 Detailed Description 66
9.25 wind_velocity_base.hh File Reference 66
9.25.1 Detailed Description 67

Index 68

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models 11
  Environment 15
    Atmosphere 16
      BaseAtmosphere 18

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod

Namespace jeod 19

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

- jeod::Atmosphere 21
  - jeod::METAtmosphere 28
- jeod::AtmosphereMessages 22
- jeod::AtmosphereState 24
  - jeod::METAtmosphereStateVars 45
    - jeod::METAtmosphereState 42
- jeod::METAtmosphere_solar_max_default_data 39
- jeod::METAtmosphere_solar_mean_default_data 39
- jeod::METAtmosphere_solar_min_default_data 40
- jeod::METAtmosphereChemical 40
- jeod::METAtmosphereThermal 48
- jeod::WindVelocity 51
- jeod::WindVelocity_wind_velocity_default_data 52
- jeod::WindVelocityBase 53

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

jeod::Atmosphere
A generic base class for atmospheres 21
jeod::AtmosphereMessages
Describes messages used in the Atmosphere model 22
jeod::AtmosphereState
A generic base class for atmosphere state, containing common atmosphere state parameters, i.e 24
jeod::METAtmosphere 28
jeod::METAtmosphere_solar_max_default_data 39
jeod::METAtmosphere_solar_mean_default_data 39
jeod::METAtmosphere_solar_min_default_data 40
jeod::METAtmosphereChemical
The chemical composition of the MET Atmosphere 40
jeod::METAtmosphereState
The MET specific implementation of AtmosphereState 42
jeod::METAtmosphereStateVars
The data variables component of the MET specific implementation of AtmosphereState 45
jeod::METAtmosphereThermal
The Thermal aspect of the computation 48
jeod::WindVelocity
A generic wind velocity implementation 51
jeod::WindVelocity_wind_velocity_default_data 52
jeod::WindVelocityBase
The generic base class for wind velocity classes 53

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

atmosphere.hh
General base class for atmosphere models . . . 57
atmosphere_messages.cc
Implement atmosphere_messages . . . 57
atmosphere_messages.hh
Implement atmosphere_messages . . . 58
atmosphere_state.cc
Implementation of the base atmosphere-state model . . . 58
atmosphere_state.hh . . . 59
base_atmos/include/class_declarations.hh
Forward declarations of classes defined for JEOD 2.0 Atmosphere . . . 59
MET/include/class_declarations.hh
Forward declarations of classes defined for JEOD 2.0 Atmosphere . . . 59
data_met_wind_velocity.cc . . . 60
MET_atmosphere.cc
Implementation of MET atmosphere model . . . 60
MET_atmosphere.hh
Implement the MET atmosphere using the atmosphere framework . . . 61
MET_atmosphere_state.cc . . . 61
MET_atmosphere_state.hh
Implement the MET atmosphere state using the atmosphere framework . . . 61
MET_atmosphere_state_vars.cc
Implementation of MET atmosphere model . . . 62
MET_atmosphere_state_vars.hh
Implement the MET atmosphere state variables using the atmosphere framework . . . 62
met_data_wind_velocity.hh . . . 63
solar_max.cc . . . 63
solar_max.hh . . . 64
solar_mean.cc . . . 64
solar_mean.hh . . . 64
solar_min.cc . . . 64
solar_min.hh . . . 65
wind_velocity.cc
General base class for wind velocity models . . . 65
wind_velocity.hh . . . 66
wind_velocity_base.cc
General base class for wind velocity models . . . 66

wind_velocity_base.hh
General base class for wind velocity models 66

Chapter 6

Module Documentation

6.1 Models

Modules

- Environment

Data Structures

- class jeod::Atmosphere

  A generic base class for atmospheres.
- class jeod::WindVelocity

  A generic wind velocity implementation.

Functions

- jeod::Atmosphere::Atmosphere ()
- virtual jeod::Atmosphere::∼Atmosphere ()
- virtual void jeod::Atmosphere::update_atmosphere (const PlanetFixedPosition ∗position, AtmosphereState ∗state)=0

  A pure virtual function for updating the atmosphere, and inserting.
- Atmosphere & jeod::Atmosphere::operator= (const Atmosphere &rhs)
- jeod::Atmosphere::Atmosphere (const Atmosphere &rhs)
- jeod::WindVelocity::WindVelocity ()

  Default Constructor.
- virtual jeod::WindVelocity::∼WindVelocity ()

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

  Updates the wind velocity from the parameters given.
- jeod::WindVelocity::WindVelocity (const WindVelocity &rhs)
- WindVelocity & jeod::WindVelocity::operator= (const WindVelocity &rhs)

Variables

- bool jeod::Atmosphere::active

  If true the atmosphere state will calculate, if false it will not.
- bool jeod::WindVelocity::active

trick_units(–)

- unsigned int jeod::WindVelocity::num_layers

  Number of altitude layers.

- double ∗ jeod::WindVelocity::omega_scale_fac

  Factor by which omega is multiplied depending on altitude.

- double ∗ jeod::WindVelocity::omega_scale_alt

  Altitudes at which omega is multiplied by the corresponding factor.

- double jeod::WindVelocity::omega

  The rotational velocity of the planet.

- unsigned int jeod::WindVelocity::array_index

  last known index into the arrays

Friends

- class jeod::Atmosphere::InputProcessor
- class jeod::WindVelocity::InputProcessor
- void jeod::Atmosphere::init_attrjeod__Atmosphere ()
- void jeod::WindVelocity::init_attrjeod__WindVelocity ()

6.1.1 Detailed Description

6.1.2 Function Documentation

6.1.2.1 jeod::Atmosphere::Atmosphere ( ) [inline]

Definition at line 59 of file atmosphere.hh.

6.1.2.2 jeod::Atmosphere::Atmosphere (const **Atmosphere** & *rhs*) [private]

6.1.2.3 Atmosphere& jeod::Atmosphere::operator= (const **Atmosphere** & *rhs*) [private]

6.1.2.4 WindVelocity& jeod::WindVelocity::operator= (const **WindVelocity** & *rhs*) [private]

6.1.2.5 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().

6.1.2.6 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 82 of file wind_velocity.cc.

References jeod::WindVelocity::active, jeod::WindVelocity::array_index, jeod::AtmosphereMessages::framework_error, jeod::WindVelocity::num_layers, jeod::WindVelocity::omega, jeod::WindVelocity::omega_scale_alt, and jeod::WindVelocity::omega_scale_fac.

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

6.1.2.7 jeod::WindVelocity::WindVelocity (void)

Default Constructor.

Definition at line 47 of file wind_velocity.cc.

6.1.2.8 jeod::WindVelocity::WindVelocity (const WindVelocity & *rhs*) [private]

6.1.2.9 virtual jeod::Atmosphere::∼Atmosphere ( ) [inline], [virtual]

Definition at line 64 of file atmosphere.hh.

6.1.2.10 jeod::WindVelocity::∼WindVelocity (void) [virtual]

Destructor.

Definition at line 63 of file wind_velocity.cc.

References jeod::WindVelocity::omega_scale_alt, and jeod::WindVelocity::omega_scale_fac.

6.1.3 Variable Documentation

6.1.3.1 bool jeod::Atmosphere::active

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

trick_units(–) activity-control flag.

Definition at line 56 of file atmosphere.hh.

6.1.3.2 bool jeod::WindVelocity::active

trick_units(–)

Definition at line 62 of file wind_velocity.hh.

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

6.1.3.3 unsigned int jeod::WindVelocity::array_index [private]

last known index into the arrays

Definition at line 88 of file wind_velocity.hh.

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

6.1.3.4 unsigned int jeod::WindVelocity::num_layers

Number of altitude layers.

trick_units(count)

Definition at line 67 of file wind_velocity.hh.

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

6.1.3.5 double jeod::WindVelocity::omega

The rotational velocity of the planet.

trick_units(radian/s)

Definition at line 82 of file wind_velocity.hh.

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

6.1.3.6 double∗ jeod::WindVelocity::omega_scale_alt

Altitudes at which omega is multiplied by the corresponding factor.

trick_units(m)

Definition at line 77 of file wind_velocity.hh.

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

6.1.3.7 double∗ jeod::WindVelocity::omega_scale_fac

Factor by which omega is multiplied depending on altitude.

trick_units(–)

Definition at line 72 of file wind_velocity.hh.

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

6.1.4 Friends

6.1.4.1 void init_attrjeod__Atmosphere ( ) [friend]

6.1.4.2 void init_attrjeod__WindVelocity ( ) [friend]

6.1.4.3 friend class InputProcessor [friend]

Definition at line 45 of file wind_velocity.hh.

6.1.4.4 friend class InputProcessor [friend]

Definition at line 49 of file atmosphere.hh.

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 #define _USE_MATH_DEFINES_

Definition at line 49 of file MET_atmosphere.cc.

6.3.2.2 #define PATH "environment/atmosphere/base_atmos"

Definition at line 30 of file atmosphere_messages.cc.

6.4 BaseAtmosphere

Files

- file atmosphere.hh

    General base class for atmosphere models.

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 WindVelocity

  A generic wind velocity implementation.
- class WindVelocityBase

  The generic base class for wind velocity classes.
- class WindVelocity_wind_velocity_default_data
- 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 METAtmosphereThermal

  The Thermal aspect of the computation.
- class METAtmosphere
- class METAtmosphereState

  The MET specific implementation of AtmosphereState.
- class METAtmosphereStateVars

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

7.1.1 Detailed Description

Namespace jeod.

Chapter 8

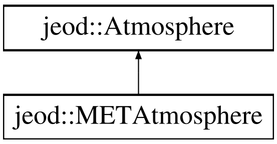
Data Structure Documentation

8.1 jeod::Atmosphere Class Reference

A generic base class for atmospheres.

`#include <atmosphere.hh>`

Inheritance diagram for jeod::Atmosphere:



Public Member Functions

- Atmosphere ()
- virtual ∼Atmosphere ()
- virtual void update_atmosphere (const PlanetFixedPosition ∗position, AtmosphereState ∗state)=0

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

Data Fields

- bool active

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

Private Member Functions

- Atmosphere & operator= (const Atmosphere &rhs)
- Atmosphere (const Atmosphere &rhs)

Friends

- class InputProcessor
- void init_attrjeod__Atmosphere ()

8.1.1 Detailed Description

A generic base class for atmospheres.

Definition at line 47 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>`

Static Public Attributes

- static char const ∗ initialization_error

  Indicates an error during initialization.
- static char const ∗ framework_error

  Indicates an error during use of the generic framework.
- static char const ∗ framework_warning

  Indicates a warning associated with the generic framework.
- static char const ∗ numerical_warning

  Indicates a warning associated with numerical values.

Private Member Functions

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

Friends

- class InputProcessor
- void init_attrjeod__AtmosphereMessages ()

8.2.1 Detailed Description

Describes messages used in the Atmosphere model.

Definition at line 42 of file atmosphere_messages.hh.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 jeod::AtmosphereMessages::AtmosphereMessages (void) `[private]`

8.2.2.2 jeod::AtmosphereMessages::AtmosphereMessages (const AtmosphereMessages & *rhs*) `[private]`

8.2.3 Member Function Documentation

8.2.3.1 AtmosphereMessages& jeod::AtmosphereMessages::operator= (const AtmosphereMessages & *rhs*) [private]

8.2.4 Friends And Related Function Documentation

8.2.4.1 void init_attrjeod__AtmosphereMessages ( ) [friend]

8.2.4.2 friend class InputProcessor [friend]

Definition at line 44 of file atmosphere_messages.hh.

8.2.5 Field Documentation

8.2.5.1 char const ∗ jeod::AtmosphereMessages::framework_error [static]

Initial value:

```
=
    "environment/atmosphere/base_atmos"  "framework_error"
```

Indicates an error during use of the generic framework.

trick_units(–)

Definition at line 60 of file atmosphere_messages.hh.

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

8.2.5.2 char const ∗ jeod::AtmosphereMessages::framework_warning [static]

Initial value:

```
=
    "environment/atmosphere/base_atmos"  "framework_warning"
```

Indicates a warning associated with the generic framework.

trick_units(–)

Definition at line 67 of file atmosphere_messages.hh.

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

8.2.5.3 char const ∗ jeod::AtmosphereMessages::initialization_error [static]

Initial value:

```
=
    "environment/atmosphere/base_atmos"  "initialization_error"
```

Indicates an error during initialization.

trick_units(–)

Definition at line 55 of file atmosphere_messages.hh.

8.2.5.4 char const ∗ jeod::AtmosphereMessages::numerical_warning [static]

Initial value:

```
=
    "environment/atmosphere/base_atmos"  "numerical_warning"
```

Indicates a warning associated with numerical values.

trick_units(–)

Definition at line 72 of file atmosphere_messages.hh.

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

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

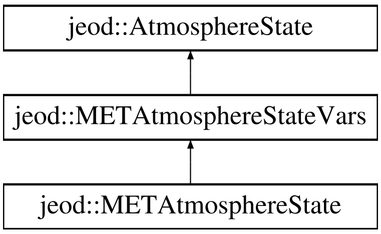
- atmosphere_messages.hh
- atmosphere_messages.cc

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 ()
- AtmosphereState (Atmosphere &atmos, const PlanetFixedPosition &pfix_pos)
- virtual ∼AtmosphereState ()
- 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
- 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 55 of file atmosphere_state.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 jeod::AtmosphereState::AtmosphereState ( )

Definition at line 40 of file atmosphere_state.cc.

References wind.

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

Definition at line 52 of file atmosphere_state.cc.

References wind.

8.3.2.3 jeod::AtmosphereState::∼AtmosphereState ( ) [virtual]

Definition at line 69 of file atmosphere_state.cc.

8.3.2.4 jeod::AtmosphereState::AtmosphereState (const **AtmosphereState** & *rhs*)

Copy Constructor.

Parameters

in	*rhs*	The AtmosphereState to copy from

Definition at line 79 of file atmosphere_state.cc.

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

8.3.3 Member Function Documentation

8.3.3.1 AtmosphereState & jeod::AtmosphereState::operator= (const AtmosphereState & *rhs*)

AtmosphereState Operator =.

Returns

The newly copied AtmosphereState

Parameters

in	*rhs*	The AtmosphereState to copy

Definition at line 102 of file atmosphere_state.cc.

References density, pressure, and temperature.

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

8.3.3.2 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 127 of file atmosphere_state.cc.

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

8.3.3.3 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 147 of file atmosphere_state.cc.

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

8.3.3.4 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 166 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 void init_attrjeod__AtmosphereState ( ) [friend]

8.3.4.2 friend class InputProcessor [friend]

Definition at line 57 of file atmosphere_state.hh.

8.3.5 Field Documentation

8.3.5.1 bool jeod::AtmosphereState::active

```
trick_units(--)
```

Activation flag for computing state.

Definition at line 61 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 Atmosphere* jeod::AtmosphereState::atmos [protected]

Definition at line 73 of file atmosphere_state.hh.

Referenced by AtmosphereState(), and update_state().

8.3.5.3 double jeod::AtmosphereState::density

```
trick_units(kg/m3)
```

total density at altitude

Definition at line 65 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 const PlanetFixedPosition* jeod::AtmosphereState::pfix_pos [protected]

Definition at line 74 of file atmosphere_state.hh.

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

8.3.5.5 double jeod::AtmosphereState::pressure

```
trick_units(N/m2)
```

Total pressure

Definition at line 67 of file atmosphere_state.hh.

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

8.3.5.6 double jeod::AtmosphereState::temperature

```
                  trick_units(K)
```

Temperature at altitude

Definition at line 63 of file atmosphere_state.hh.

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

8.3.5.7 double jeod::AtmosphereState::wind[3]

```
               trick_units(m/s)
```

Wind velocity

Definition at line 69 of file atmosphere_state.hh.

Referenced by AtmosphereState(), and update_wind().

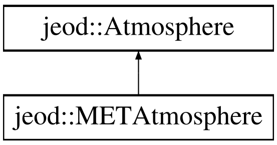
The documentation for this class was generated from the following files:

- 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 ()
- virtual ∼METAtmosphere ()
- virtual void update_atmosphere (const PlanetFixedPosition ∗pfix_pos, AtmosphereState ∗state)

  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.
- void update_time (const TimeUTC &time_utc)

Data Fields

- AtmosMETGeoIndexType geo_index_type
- 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)
- METAtmosphere & operator= (const METAtmosphere &rhs)
- METAtmosphere (const METAtmosphere &rhs)

Private Attributes

- double altitude_km
- double latitude
- double longitude
- double barometric_equation_ceiling
- double trunc_julian_time
- double tjt_year_start
- double fraction_of_year
- int day_of_year
- int max_days_this_year
- int year
- double solar_declination_angle
- double solar_hour_angle
- METAtmosphereStateVars state
- METAtmosphereThermal thermal
- const double R_gas_constant
- const double days_per_year
- const double Avogadro
- const double two_pi
- const double three_pi_two
- const double deg_to_rad
- const int days_per_century
- const int minutes_per_day
- const double mol_weight_barometric_ceiling
- const double base_fairing_height
- 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]
- 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 144 of file MET_atmosphere.hh.

8.4.2 Member Enumeration Documentation

8.4.2.1 enum **jeod::METAtmosphere::AtmosMETGeoIndexType**

Enumerator

ATMOS_MET_GI_AP

ATMOS_MET_GI_KP

Definition at line 149 of file MET_atmosphere.hh.

8.4.3 Constructor & Destructor Documentation

8.4.3.1 jeod::METAtmosphere::METAtmosphere ( )

Definition at line 130 of file MET_atmosphere.cc.

8.4.3.2 virtual jeod::METAtmosphere::∼METAtmosphere ( ) `[inline]`, `[virtual]`

Definition at line 266 of file MET_atmosphere.hh.

8.4.3.3 jeod::METAtmosphere::METAtmosphere (const **METAtmosphere** & *rhs*) `[private]`

8.4.4 Member Function Documentation

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

Definition at line 1282 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 void jeod::METAtmosphere::atmos_MET_FAIR5 ( ) [private]

Definition at line 1143 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 void jeod::METAtmosphere::compute_exospheric_temperature ( ) [private]

Definition at line 651 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 double jeod::METAtmosphere::compute_mol_wt (double *altitude*) [private]

Definition at line 1202 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 void jeod::METAtmosphere::compute_seasonal_lat_variation_He ( ) [private]

Definition at line 1078 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 void jeod::METAtmosphere::compute_seasonal_latitude_variation ( ) [private]

Definition at line 1018 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 void jeod::METAtmosphere::compute_solar_angles ( ) [private]

Definition at line 455 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 void jeod::METAtmosphere::jacchia ( ) [private]

Definition at line 798 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::METAtmosphereChemical::frac, jeod::METAtmosphereStateVars::mol_weight, jeod::METAtmosphereChemical::mol_weight, mol_weight_barometric_ceiling, jeod::METAtmosphereChemical::nominal_mol_weight, jeod::METAtmosphereChemical::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 void jeod::METAtmosphere::modify_densities ( ) [private]

Definition at line 411 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 METAtmosphere& jeod::METAtmosphere::operator= (const METAtmosphere & *rhs*) [private]

8.4.4.11 void jeod::METAtmosphere::update_atmosphere (const PlanetFixedPosition ∗ *position,* AtmosphereState ∗ *state*) [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 305 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 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 341 of file MET_atmosphere.cc.

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

8.4.4.13 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 364 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::METAtmosphereStateVars::He, jeod::METAtmosphereStateVars::Hyd, jacchia(), latitude, jeod::METAtmosphereStateVars::log10_dens, longitude, modify_densities(), jeod::METAtmosphereStateVars::mol_weight, jeod::METAtmosphereStateVars::N2, jeod::METAtmosphereChemical::num_density, jeod::METAtmosphereStateVars::Ox, jeod::METAtmosphereStateVars::Ox2, jeod::AtmosphereState::pressure, R_gas_constant, species, state, and jeod::AtmosphereState::temperature.

8.4.4.14 void jeod::METAtmosphere::update_time (const TimeUTC & *time_utc*) [inline]

Definition at line 279 of file MET_atmosphere.hh.

References trunc_julian_time.

8.4.5 Friends And Related Function Documentation

8.4.5.1 void init_attrjeod__METAtmosphere ( ) [friend]

8.4.5.2 friend class InputProcessor [friend]

Definition at line 146 of file MET_atmosphere.hh.

8.4.6 Field Documentation

8.4.6.1 double jeod::METAtmosphere::altitude_km [private]

trick_units(km) Copy of vehicle altitude

Definition at line 171 of file MET_atmosphere.hh.

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

8.4.6.2 const double jeod::METAtmosphere::Avogadro [private]

trick_units(–) Avogadros number

Definition at line 210 of file MET_atmosphere.hh.

Referenced by jacchia().

8.4.6.3 double jeod::METAtmosphere::barometric_equation_ceiling [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 175 of file MET_atmosphere.hh.

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

8.4.6.4 const double jeod::METAtmosphere::base_fairing_height [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 221 of file MET_atmosphere.hh.

Referenced by atmos_MET_FAIR5(), and modify_densities().

8.4.6.5 int jeod::METAtmosphere::day_of_year [private]

```
trick_units(count)
```

day number since start of year.

Definition at line 187 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.6 const int jeod::METAtmosphere::days_per_century [private]

trick_units(count) days per century

Definition at line 215 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.7 const double jeod::METAtmosphere::days_per_year [private]

trick_units(day) days per year

Definition at line 209 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.8 const double jeod::METAtmosphere::deg_to_rad [private]

```
trick_units(degree/radian)
```

degree-to-radian conversion

Definition at line 213 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.9 double jeod::METAtmosphere::F10

```
trick_units(--)
```

Solar radio noise flux.

Definition at line 161 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 double jeod::METAtmosphere::F10B

```
trick_units(--)
```

90 day average of solar radio noise flux.

Definition at line 163 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 const double jeod::METAtmosphere::fairing_k [private]

```
trick_units(radian/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 225 of file MET_atmosphere.hh.

Referenced by atmos_MET_FAIR5().

8.4.6.12 double jeod::METAtmosphere::fraction_of_year [private]

```
trick_units(--)
```

fraction of this year that has passed.

Definition at line 184 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), compute_seasonal_latitude_variation(), and compute_solar_angles().

8.4.6.13 const double jeod::METAtmosphere::gauss_altitudes [static], [private]

Initial value:

```
= {
    90.0, 105.0, 125.0, 160.0, 200.0, 300.0, 500.0, 1500.0, 2500.0 }
```

```
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 247 of file MET_atmosphere.hh.

Referenced by apply_gauss_quadrature().

8.4.6.14 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 254 of file MET_atmosphere.hh.

Referenced by apply_gauss_quadrature().

8.4.6.15 double jeod::METAtmosphere::geo_index

```
trick_units(--)
```

Geomagnetic variations index (Ap or Kp).

Definition at line 159 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 AtmosMETGeoIndexType jeod::METAtmosphere::geo_index_type

Definition at line 156 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 double jeod::METAtmosphere::latitude [private]

trick_units(radian) Copy of vehicle latitude

Definition at line 172 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 double jeod::METAtmosphere::longitude [private]

trick_units(radian) Copy of vehicle longitude

Definition at line 173 of file MET_atmosphere.hh.

Referenced by compute_solar_angles(), and update_atmosphere().

8.4.6.19 int jeod::METAtmosphere::max_days_this_year [private]

```
trick_units(count)
```

number of days this year (365 or 366)

Definition at line 190 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.20 const int jeod::METAtmosphere::minutes_per_day [private]

trick_units(count) minutes per day

Definition at line 216 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.21 const double jeod::METAtmosphere::mol_weight_barometric_ceiling [private]

```
trick_units(g/mol)
```

mean molar mass at barometric-ceiling and higher.

Definition at line 219 of file MET_atmosphere.hh.

Referenced by compute_mol_wt(), and jacchia().

8.4.6.22 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 239 of file MET_atmosphere.hh.

Referenced by compute_mol_wt().

8.4.6.23 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 244 of file MET_atmosphere.hh.

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

```
trick_units(count)
```

the number of polynomial coefficients.

Definition at line 237 of file MET_atmosphere.hh.

8.4.6.25 const double jeod::METAtmosphere::R_gas_constant [private]

trick_units(J/(mol∗K)) R

Definition at line 208 of file MET_atmosphere.hh.

Referenced by jacchia(), and update_atmosphere().

8.4.6.26 double jeod::METAtmosphere::solar_declination_angle [private]

trick_units(radian) declination angle

Definition at line 195 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), compute_seasonal_lat_variation_He(), and compute_solar_angles().

8.4.6.27 double jeod::METAtmosphere::solar_hour_angle [private]

trick_units(radian) solar hour angle

Definition at line 197 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), and compute_solar_angles().

8.4.6.28 METAtmosphereChemical jeod::METAtmosphere::species

```
trick_units(--)
```

The chemical composition of the atmosphere.

Definition at line 166 of file MET_atmosphere.hh.

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

8.4.6.29 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 199 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 METAtmosphereThermal jeod::METAtmosphere::thermal [private]

```
trick_units(--)
```

Thermal aspect of the model

Definition at line 203 of file MET_atmosphere.hh.

Referenced by apply_gauss_quadrature(), and jacchia().

8.4.6.31 const double jeod::METAtmosphere::three_pi_two [private]

trick_units(–) 1.5 pi

Definition at line 212 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.32 double jeod::METAtmosphere::tjt_year_start [private]

```
trick_units(day)
```

value of trunc_julian_time at the start of the current year.

Definition at line 181 of file MET_atmosphere.hh.

Referenced by compute_solar_angles().

8.4.6.33 double jeod::METAtmosphere::trunc_julian_time [private]

trick_units(day) Current time

Definition at line 180 of file MET_atmosphere.hh.

Referenced by compute_solar_angles(), and update_time().

8.4.6.34 const double jeod::METAtmosphere::two_pi [private]

trick_units(–) 2 pi

Definition at line 211 of file MET_atmosphere.hh.

Referenced by compute_exospheric_temperature(), and compute_solar_angles().

8.4.6.35 int jeod::METAtmosphere::year [private]

trick_units(count) current year identifier

Definition at line 193 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 18 of file solar_max.hh.

8.5.2 Member Function Documentation

8.5.2.1 void jeod::METAtmosphere_solar_max_default_data::initialize (**METAtmosphere** ∗ *METAtmosphere_ptr*)

Definition at line 38 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 18 of file solar_mean.hh.

8.6.2 Member Function Documentation

8.6.2.1 void jeod::METAtmosphere_solar_mean_default_data::initialize (**METAtmosphere** ∗ *METAtmosphere_ptr*)

Definition at line 38 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 18 of file solar_min.hh.

8.7.2 Member Function Documentation

8.7.2.1 void jeod::METAtmosphere_solar_min_default_data::initialize (**METAtmosphere** ∗ *METAtmosphere_ptr*)

Definition at line 38 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 ()
- virtual ∼METAtmosphereChemical ()

Data Fields

- double num_density [num_species]
- double frac [num_species]
- double mol_weight [num_species]
- const double nominal_mol_weight

Static Public Attributes

- static const int num_species = 6

Private Member Functions

- METAtmosphereChemical & operator= (const METAtmosphereChemical &rhs)
- METAtmosphereChemical (const METAtmosphereChemical &rhs)

Friends

- class InputProcessor
- void init_attrjeod__METAtmosphereChemical ()

8.8.1 Detailed Description

The chemical composition of the MET Atmosphere.

Definition at line 55 of file MET_atmosphere.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 jeod::METAtmosphereChemical::METAtmosphereChemical ( )

Definition at line 88 of file MET_atmosphere.cc.

References frac, mol_weight, num_density, and num_species.

8.8.2.2 virtual jeod::METAtmosphereChemical::∼METAtmosphereChemical ( ) [inline], [virtual]

Definition at line 75 of file MET_atmosphere.hh.

8.8.2.3 jeod::METAtmosphereChemical::METAtmosphereChemical (const **METAtmosphereChemical** & *rhs*) [private]

8.8.3 Member Function Documentation

8.8.3.1 **METAtmosphereChemical**& jeod::METAtmosphereChemical::operator= (const **METAtmosphereChemical** & *rhs*) [private]

8.8.4 Friends And Related Function Documentation

8.8.4.1 void init_attrjeod__METAtmosphereChemical ( ) [friend]

8.8.4.2 friend class InputProcessor [friend]

Definition at line 56 of file MET_atmosphere.hh.

8.8.5 Field Documentation

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

Definition at line 65 of file MET_atmosphere.hh.

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

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

Definition at line 68 of file MET_atmosphere.hh.

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

8.8.5.3 const double jeod::METAtmosphereChemical::nominal_mol_weight

Definition at line 71 of file MET_atmosphere.hh.

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

8.8.5.4 double jeod::METAtmosphereChemical::num_density[num_species]

Definition at line 62 of file MET_atmosphere.hh.

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

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

Definition at line 59 of file MET_atmosphere.hh.

Referenced by METAtmosphereChemical().

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

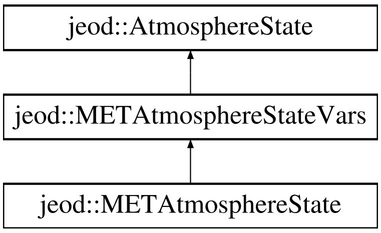
- MET_atmosphere.hh
- MET_atmosphere.cc

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 ()
- METAtmosphereState (METAtmosphere &atmos_model, const PlanetFixedPosition &pfix_pos)
- virtual ∼METAtmosphereState ()
- void update_state (METAtmosphere ∗atmos_model, const PlanetFixedPosition ∗pfix_pos)

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

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

Private Member Functions

- METAtmosphereState & operator= (const METAtmosphereState &rhs)
- METAtmosphereState (const METAtmosphereState &rhs)

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 55 of file MET_atmosphere_state.hh.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 jeod::METAtmosphereState::METAtmosphereState ( )

Definition at line 62 of file MET_atmosphere_state.cc.

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

Definition at line 68 of file MET_atmosphere_state.cc.

8.9.2.3 virtual jeod::METAtmosphereState::∼METAtmosphereState ( ) [inline], [virtual]

Definition at line 74 of file MET_atmosphere_state.hh.

8.9.2.4 jeod::METAtmosphereState::METAtmosphereState (const **METAtmosphereState** & *rhs*) [private]

8.9.3 Member Function Documentation

8.9.3.1 **METAtmosphereState**& jeod::METAtmosphereState::operator= (const **METAtmosphereState** & *rhs*) [private]

8.9.3.2 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 86 of file MET_atmosphere_state.cc.

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

8.9.3.3 void jeod::METAtmosphereState::update_state ( ) [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 104 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 void init_attrjeod__METAtmosphereState ( ) [friend]

8.9.4.2 friend class InputProcessor [friend]

Definition at line 58 of file MET_atmosphere_state.hh.

8.9.5 Field Documentation

8.9.5.1 **METAtmosphere**∗ jeod::METAtmosphereState::met_atmos [private]

Definition at line 61 of file MET_atmosphere_state.hh.

Referenced by update_state().

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

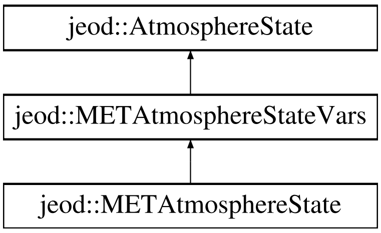
- MET_atmosphere_state.hh
- MET_atmosphere_state.cc

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 ()
- METAtmosphereStateVars (Atmosphere &atmos_model, const PlanetFixedPosition &pfix_pos)
- virtual ∼METAtmosphereStateVars ()
- 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 49 of file MET_atmosphere_state_vars.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 jeod::METAtmosphereStateVars::METAtmosphereStateVars ( )

Definition at line 55 of file MET_atmosphere_state_vars.cc.

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

Definition at line 69 of file MET_atmosphere_state_vars.cc.

8.10.2.3 jeod::METAtmosphereStateVars::∼METAtmosphereStateVars ( ) [virtual]

Definition at line 90 of file MET_atmosphere_state_vars.cc.

8.10.2.4 jeod::METAtmosphereStateVars::METAtmosphereStateVars (const **METAtmosphereStateVars** & *rhs*)

Copy Constructor.

Parameters

in	*rhs*	The METAtmosphereStateVars to copy

Definition at line 98 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 **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 122 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 void init_attrjeod__METAtmosphereStateVars ( ) [friend]

8.10.4.2 friend class InputProcessor [friend]

Definition at line 51 of file MET_atmosphere_state_vars.hh.

8.10.5 Field Documentation

8.10.5.1 double jeod::METAtmosphereStateVars::A

trick_units(–) A number density

Definition at line 60 of file MET_atmosphere_state_vars.hh.

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

8.10.5.2 double jeod::METAtmosphereStateVars::exo_temp

trick_units(K) Exospheric temperature

Definition at line 54 of file MET_atmosphere_state_vars.hh.

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

8.10.5.3 double jeod::METAtmosphereStateVars::He

trick_units(–) He number density

Definition at line 61 of file MET_atmosphere_state_vars.hh.

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

8.10.5.4 double jeod::METAtmosphereStateVars::Hyd

trick_units(–) H number density

Definition at line 62 of file MET_atmosphere_state_vars.hh.

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

8.10.5.5 double jeod::METAtmosphereStateVars::log10_dens

trick_units(–) Log10(total density)

Definition at line 55 of file MET_atmosphere_state_vars.hh.

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

8.10.5.6 double jeod::METAtmosphereStateVars::mol_weight

trick_units(–) Average molecular weight

Definition at line 56 of file MET_atmosphere_state_vars.hh.

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

8.10.5.7 double jeod::METAtmosphereStateVars::N2

trick_units(–) N2 number density

Definition at line 57 of file MET_atmosphere_state_vars.hh.

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

8.10.5.8 double jeod::METAtmosphereStateVars::Ox

trick_units(--) O number density

Definition at line 59 of file MET_atmosphere_state_vars.hh.

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

8.10.5.9 double jeod::METAtmosphereStateVars::Ox2

trick_units(--) O2 number density

Definition at line 58 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 ()

Data Fields

- double T_out

Private Member Functions

- void generate_base_temperature ()
- METAtmosphereThermal & operator= (const METAtmosphereThermal &rhs)
- METAtmosphereThermal (const METAtmosphereThermal &rhs)

Private Attributes

- const double k_1

  Temperature coefficients.
- const double k_3
- const double k_4
- const double T_90
- 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 90 of file MET_atmosphere.hh.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 jeod::METAtmosphereThermal::METAtmosphereThermal (const double & *T_exosphere,* const double & *altitude_km*)

Definition at line 113 of file MET_atmosphere.cc.

8.11.2.2 virtual jeod::METAtmosphereThermal::∼METAtmosphereThermal ( ) [inline], [virtual]

Definition at line 99 of file MET_atmosphere.hh.

8.11.2.3 jeod::METAtmosphereThermal::METAtmosphereThermal (const METAtmosphereThermal & *rhs*) [private]

8.11.3 Member Function Documentation

8.11.3.1 double jeod::METAtmosphereThermal::compute_temperature (double *altitude_km*)

Definition at line 242 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 void jeod::METAtmosphereThermal::generate_base_temperature ( ) [private]

8.11.3.3 METAtmosphereThermal& jeod::METAtmosphereThermal::operator= (const METAtmosphereThermal & *rhs*) [private]

8.11.3.4 void jeod::METAtmosphereThermal::update ( )

Definition at line 182 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 void init_attrjeod__METAtmosphereThermal ( ) [friend]

8.11.4.2 friend class InputProcessor [friend]

Definition at line 91 of file MET_atmosphere.hh.

8.11.5 Field Documentation

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

Definition at line 128 of file MET_atmosphere.hh.

Referenced by update().

8.11.5.2 const double jeod::METAtmosphereThermal::k_1 [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 99 of file MET_atmosphere.hh.

Referenced by compute_temperature().

8.11.5.3 const double jeod::METAtmosphereThermal::k_3 [private]

```
trick_units(1/m3)
```

parameter used to obtain the 3rd coefficient of the temperature polynomial.

Definition at line 111 of file MET_atmosphere.hh.

Referenced by compute_temperature().

8.11.5.4 const double jeod::METAtmosphereThermal::k_4 [private]

```
trick_units(1/m4)
```

parameter used to obtain the 4th coefficient of the temperature polynomial.

Definition at line 115 of file MET_atmosphere.hh.

Referenced by compute_temperature().

8.11.5.5 double jeod::METAtmosphereThermal::T_125 [private]

trick_units(K) Temperature at 125km reference point.

Definition at line 122 of file MET_atmosphere.hh.

Referenced by compute_temperature(), and update().

8.11.5.6 const double jeod::METAtmosphereThermal::T_90 [private]

```
     trick_units(K)
```

Temperature at 90km reference point.

Definition at line 119 of file MET_atmosphere.hh.

Referenced by compute_temperature().

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

Definition at line 125 of file MET_atmosphere.hh.

Referenced by compute_temperature(), and update().

8.11.5.8 double jeod::METAtmosphereThermal::T_out

Definition at line 93 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 Class Reference

A generic wind velocity implementation.

`#include <wind_velocity.hh>`

Public Member Functions

- WindVelocity ()

  Default Constructor.
- virtual ∼WindVelocity ()

  Destructor.
- virtual void update_wind (double inertial_pos[3], double altitude, double wind_inertial[3])

  Updates the wind velocity from the parameters given.

Data Fields

- bool active

  trick_units(–)
- unsigned int num_layers

  Number of altitude layers.
- double ∗ omega_scale_fac

  Factor by which omega is multiplied depending on altitude.
- double ∗ omega_scale_alt

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

  The rotational velocity of the planet.

Private Member Functions

- WindVelocity (const WindVelocity &rhs)
- WindVelocity & operator= (const WindVelocity &rhs)

Private Attributes

- unsigned int array_index

  last known index into the arrays

Friends

- class InputProcessor
- void init_attrjeod__WindVelocity ()

8.12.1 Detailed Description

A generic wind velocity implementation.

Definition at line 43 of file wind_velocity.hh.

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

- wind_velocity.hh
- wind_velocity.cc

8.13 jeod::WindVelocity_wind_velocity_default_data Class Reference

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

Public Member Functions

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

Data Fields

- double omega_scale_fac [num_layers]
- double omega_scale_alt [num_layers]
- double omega

Static Public Attributes

- static const int num_layers = 12

8.13.1 Detailed Description

Definition at line 20 of file met_data_wind_velocity.hh.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 jeod::WindVelocity_wind_velocity_default_data::WindVelocity_wind_velocity_default_data ( )

Definition at line 48 of file data_met_wind_velocity.cc.

References num_layers, omega_scale_alt, and omega_scale_fac.

8.13.3 Member Function Documentation

8.13.3.1 void jeod::WindVelocity_wind_velocity_default_data::initialize (WindVelocity ∗ *WindVelocity_ptr*)

Definition at line 70 of file data_met_wind_velocity.cc.

8.13.3.2 void jeod::WindVelocity_wind_velocity_default_data::initialize (WindVelocity & *wind_velocity*)

Definition at line 82 of file data_met_wind_velocity.cc.

References num_layers, jeod::WindVelocity::num_layers, omega, jeod::WindVelocity::omega, omega_scale_alt, jeod::WindVelocity::omega_scale_alt, omega_scale_fac, and jeod::WindVelocity::omega_scale_fac.

8.13.4 Field Documentation

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

Definition at line 22 of file met_data_wind_velocity.hh.

Referenced by initialize(), and WindVelocity_wind_velocity_default_data().

8.13.4.2 double jeod::WindVelocity_wind_velocity_default_data::omega

Definition at line 28 of file met_data_wind_velocity.hh.

Referenced by initialize().

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

Definition at line 26 of file met_data_wind_velocity.hh.

Referenced by initialize(), and WindVelocity_wind_velocity_default_data().

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

Definition at line 24 of file met_data_wind_velocity.hh.

Referenced by initialize(), and WindVelocity_wind_velocity_default_data().

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

- met_data_wind_velocity.hh
- data_met_wind_velocity.cc

8.14 jeod::WindVelocityBase Class Reference

The generic base class for wind velocity classes.

`#include <wind_velocity_base.hh>`

Public Member Functions

- WindVelocityBase ()

  Default Constructor.
- virtual ∼WindVelocityBase ()

Destructor.

- virtual void update_wind (double position[3], double altitude, double wind_inertial[3])

  Virtual function to define the interface for inheriting functions.

Private Member Functions

- WindVelocityBase (const WindVelocityBase &rhs)
- WindVelocityBase & operator= (const WindVelocityBase &rhs)

Friends

- class InputProcessor
- void init_attrjeod__WindVelocityBase ()

8.14.1 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 42 of file wind_velocity_base.hh.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 jeod::WindVelocityBase::WindVelocityBase (void)

Default Constructor.

Definition at line 35 of file wind_velocity_base.cc.

8.14.2.2 jeod::WindVelocityBase::∼WindVelocityBase (void) [virtual]

Destructor.

Definition at line 47 of file wind_velocity_base.cc.

8.14.2.3 jeod::WindVelocityBase::WindVelocityBase (const WindVelocityBase & *rhs*) [private]

8.14.3 Member Function Documentation

8.14.3.1 WindVelocityBase& jeod::WindVelocityBase::operator= (const WindVelocityBase & *rhs*) [private]

8.14.3.2 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 63 of file wind_velocity_base.cc.

References jeod::AtmosphereMessages::framework_warning.

8.14.4 Friends And Related Function Documentation

8.14.4.1 void init_attrjeod__WindVelocityBase ( ) [friend]

8.14.4.2 friend class InputProcessor [friend]

Definition at line 44 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 "utils/planet_fixed/planet_fixed_posn/include/planet_fixed_posn.hh"
#include "environment/time/include/time_standard.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.

Definition in file atmosphere.hh.

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.

Definition in file atmosphere_messages.cc.

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.

Definition in file atmosphere_messages.hh.

9.4 atmosphere_state.cc File Reference

Implementation of the base atmosphere-state model.

```
#include <cstddef>
#include "utils/math/include/vector3.hh"
#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.

Definition in file atmosphere_state.cc.

9.5 atmosphere_state.hh File Reference

```
#include "utils/planet_fixed/planet_fixed_posn/include/planet_fixed_posn.-
hh"
#include "environment/time/include/time_standard.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.

Definition in file base_atmos/include/class_declarations.hh.

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.

Definition in file MET/include/class_declarations.hh.

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 #define JEOD_FRIEND_CLASS WindVelocity_wind_velocity_default_data

Definition at line 31 of file data_met_wind_velocity.cc.

9.9 MET_atmosphere.cc File Reference

Implementation of MET atmosphere model.

```
#include <cstddef>
#include <string.h>
#include <algorithm>
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "environment/time/include/time_utc.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.

Definition in file MET_atmosphere.cc.

9.10 MET_atmosphere.hh File Reference

Implement the MET atmosphere using the atmosphere framework.

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

Definition in file MET_atmosphere.hh.

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_state_vars.hh"
#include "MET_atmosphere.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.

Definition in file MET_atmosphere_state.hh.

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.

Definition in file MET_atmosphere_state_vars.cc.

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.

Definition in file MET_atmosphere_state_vars.hh.

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 #define JEOD_FRIEND_CLASS METAtmosphere_solar_max_default_data

Definition at line 24 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 #define JEOD_FRIEND_CLASS METAtmosphere_solar_mean_default_data

Definition at line 24 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 #define JEOD_FRIEND_CLASS METAtmosphere_solar_min_default_data

Definition at line 24 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/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/wind_velocity.hh"
#include "../include/atmosphere_messages.hh"
```

Namespaces

- jeod

    Namespace jeod.

9.22.1 Detailed Description

General base class for wind velocity models.

Definition in file wind_velocity.cc.

9.23 wind_velocity.hh File Reference

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

Data Structures

- class jeod::WindVelocity

  A generic wind velocity implementation.

Namespaces

- jeod

  Namespace jeod.

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.

Definition in file wind_velocity_base.cc.

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.

Definition in file wind_velocity_base.hh.

Index

~Atmosphere
    Models, 13
~AtmosphereState
    jeod::AtmosphereState, 25
~METAtmosphere
    jeod::METAtmosphere, 30
~METAtmosphereChemical
    jeod::METAtmosphereChemical, 41
~METAtmosphereState
    jeod::METAtmosphereState, 43
~METAtmosphereStateVars
    jeod::METAtmosphereStateVars, 46
~METAtmosphereThermal
    jeod::METAtmosphereThermal, 49
~WindVelocity
    Models, 13
~WindVelocityBase
    jeod::WindVelocityBase, 54

A
    jeod::METAtmosphereStateVars, 47
ATMOS_MET_GI_AP
    jeod::METAtmosphere, 30
ATMOS_MET_GI_KP
    jeod::METAtmosphere, 30
active
    jeod::AtmosphereState, 27
    Models, 13
altitude_km
    jeod::METAtmosphere, 33
    jeod::METAtmosphereThermal, 50
apply_gauss_quadrature
    jeod::METAtmosphere, 30
array_index
    Models, 13
atmos
    jeod::AtmosphereState, 27
atmos_MET_FAIR5
    jeod::METAtmosphere, 30
AtmosMETGeoIndexType
    jeod::METAtmosphere, 30
Atmosphere, 16
    Models, 12
    PATH, 17
atmosphere.hh, 57
atmosphere_messages.cc, 57
atmosphere_messages.hh, 58
atmosphere_state.cc, 58
atmosphere_state.hh, 59
AtmosphereMessages
    jeod::AtmosphereMessages, 22
AtmosphereState
    jeod::AtmosphereState, 25
Avogadro
    jeod::METAtmosphere, 33

barometric_equation_ceiling
    jeod::METAtmosphere, 33
base_fairing_height
    jeod::METAtmosphere, 33
BaseAtmosphere, 18

class_declarations.hh, 59
compute_exospheric_temperature
    jeod::METAtmosphere, 31
compute_mol_wt
    jeod::METAtmosphere, 31
compute_seasonal_lat_variation_He
    jeod::METAtmosphere, 31
compute_seasonal_latitude_variation
    jeod::METAtmosphere, 31
compute_solar_angles
    jeod::METAtmosphere, 31
compute_temperature
    jeod::METAtmosphereThermal, 49

data_met_wind_velocity.cc, 60
day_of_year
    jeod::METAtmosphere, 34
days_per_century
    jeod::METAtmosphere, 34
days_per_year
    jeod::METAtmosphere, 34
deg_to_rad
    jeod::METAtmosphere, 34
density
    jeod::AtmosphereState, 27

Environment, 15
exo_temp
    jeod::METAtmosphereStateVars, 47

F10
    jeod::METAtmosphere, 34
F10B
    jeod::METAtmosphere, 34
fairing_k
    jeod::METAtmosphere, 35
frac
    jeod::METAtmosphereChemical, 42
fraction_of_year

jeod::METAtmosphere, 35
framework_error
jeod::AtmosphereMessages, 23
framework_warning
jeod::AtmosphereMessages, 23

gauss_altitudes
jeod::METAtmosphere, 35
gauss_n
jeod::METAtmosphere, 35
generate_base_temperature
jeod::METAtmosphereThermal, 49
geo_index
jeod::METAtmosphere, 35
geo_index_type
jeod::METAtmosphere, 36

He
jeod::METAtmosphereStateVars, 47
Hyd
jeod::METAtmosphereStateVars, 47

init_attrjeod__Atmosphere
Models, 14
init_attrjeod__AtmosphereMessages
jeod::AtmosphereMessages, 23
init_attrjeod__AtmosphereState
jeod::AtmosphereState, 27
init_attrjeod__METAtmosphere
jeod::METAtmosphere, 33
init_attrjeod__METAtmosphereChemical
jeod::METAtmosphereChemical, 41
init_attrjeod__METAtmosphereState
jeod::METAtmosphereState, 44
init_attrjeod__METAtmosphereStateVars
jeod::METAtmosphereStateVars, 46
init_attrjeod__METAtmosphereThermal
jeod::METAtmosphereThermal, 49
init_attrjeod__WindVelocity
Models, 14
init_attrjeod__WindVelocityBase
jeod::WindVelocityBase, 55
initialization_error
jeod::AtmosphereMessages, 23
initialize
jeod::METAtmosphere_solar_max_default_data, 39
jeod::METAtmosphere_solar_mean_default_data, 40
jeod::METAtmosphere_solar_min_default_data, 40
jeod::WindVelocity_wind_velocity_default_data, 53
InputProcessor
jeod::AtmosphereMessages, 23
jeod::AtmosphereState, 27
jeod::METAtmosphere, 33
jeod::METAtmosphereChemical, 41
jeod::METAtmosphereState, 44
jeod::METAtmosphereStateVars, 46
jeod::METAtmosphereThermal, 49
jeod::WindVelocityBase, 55
Models, 14

JEOD_FRIEND_CLASS
solar_max.cc, 63
solar_mean.cc, 64
solar_min.cc, 65
jacchia
jeod::METAtmosphere, 31
jeod, 19
jeod::METAtmosphere
ATMOS_MET_GI_AP, 30
ATMOS_MET_GI_KP, 30
jeod::Atmosphere, 21
jeod::AtmosphereMessages, 22
AtmosphereMessages, 22
framework_error, 23
framework_warning, 23
init_attrjeod__AtmosphereMessages, 23
initialization_error, 23
InputProcessor, 23
numerical_warning, 23
operator=, 22
jeod::AtmosphereState, 24
∼AtmosphereState, 25
active, 27
atmos, 27
AtmosphereState, 25
density, 27
init_attrjeod__AtmosphereState, 27
InputProcessor, 27
operator=, 26
pfix_pos, 27
pressure, 27
temperature, 28
update_state, 26
update_wind, 26
wind, 28
jeod::METAtmosphere, 28
∼METAtmosphere, 30
altitude_km, 33
apply_gauss_quadrature, 30
atmos_MET_FAIR5, 30
AtmosMETGeoIndexType, 30
Avogadro, 33
barometric_equation_ceiling, 33
base_fairing_height, 33
compute_exospheric_temperature, 31
compute_mol_wt, 31
compute_seasonal_lat_variation_He, 31
compute_seasonal_latitude_variation, 31
compute_solar_angles, 31
day_of_year, 34
days_per_century, 34
days_per_year, 34
deg_to_rad, 34
F10, 34
F10B, 34
fairing_k, 35

fraction_of_year, 35
gauss_altitudes, 35
gauss_n, 35
geo_index, 35
geo_index_type, 36
init_attrjeod__METAtmosphere, 33
InputProcessor, 33
jacchia, 31
latitude, 36
longitude, 36
METAtmosphere, 30
max_days_this_year, 36
minutes_per_day, 36
modify_densities, 32
mol_weight_barometric_ceiling, 36
mol_wt_coeffs, 37
num_integ_divisions, 37
num_mol_wt_coeffs, 37
operator=, 32
R_gas_constant, 37
solar_declination_angle, 37
solar_hour_angle, 37
species, 37
state, 38
thermal, 38
three_pi_two, 38
tjt_year_start, 38
trunc_julian_time, 38
two_pi, 38
update_atmosphere, 32
update_time, 33
year, 39
jeod::METAtmosphere_solar_max_default_data, 39
initialize, 39
jeod::METAtmosphere_solar_mean_default_data, 39
initialize, 40
jeod::METAtmosphere_solar_min_default_data, 40
initialize, 40
jeod::METAtmosphereChemical, 40
∼METAtmosphereChemical, 41
frac, 42
init_attrjeod__METAtmosphereChemical, 41
InputProcessor, 41
METAtmosphereChemical, 41
mol_weight, 42
nominal_mol_weight, 42
num_density, 42
num_species, 42
operator=, 41
jeod::METAtmosphereState, 42
∼METAtmosphereState, 43
init_attrjeod__METAtmosphereState, 44
InputProcessor, 44
METAtmosphereState, 43, 44
met_atmos, 44
operator=, 44
update_state, 44
jeod::METAtmosphereStateVars, 45
∼METAtmosphereStateVars, 46
A, 47
exo_temp, 47
He, 47
Hyd, 47
init_attrjeod__METAtmosphereStateVars, 46
InputProcessor, 46
log10_dens, 47
METAtmosphereStateVars, 46
mol_weight, 47
N2, 47
operator=, 46
Ox, 47
Ox2, 48
jeod::METAtmosphereThermal, 48
∼METAtmosphereThermal, 49
altitude_km, 50
compute_temperature, 49
generate_base_temperature, 49
init_attrjeod__METAtmosphereThermal, 49
InputProcessor, 49
k_1, 50
k_3, 50
k_4, 50
METAtmosphereThermal, 49
operator=, 49
T_125, 50
T_90, 50
T_exosphere, 50
T_out, 51
update, 49
jeod::WindVelocity, 51
jeod::WindVelocity_wind_velocity_default_data, 52
initialize, 53
num_layers, 53
omega, 53
omega_scale_alt, 53
omega_scale_fac, 53
jeod::WindVelocityBase, 53
∼WindVelocityBase, 54
init_attrjeod__WindVelocityBase, 55
InputProcessor, 55
operator=, 54
update_wind, 54
WindVelocityBase, 54

k_1
jeod::METAtmosphereThermal, 50
k_3
jeod::METAtmosphereThermal, 50
k_4
jeod::METAtmosphereThermal, 50

latitude
jeod::METAtmosphere, 36
log10_dens
jeod::METAtmosphereStateVars, 47
longitude
jeod::METAtmosphere, 36

MET_atmosphere.cc, 60
MET_atmosphere.hh, 61
MET_atmosphere_state.cc, 61
MET_atmosphere_state.hh, 61
MET_atmosphere_state_vars.cc, 62
MET_atmosphere_state_vars.hh, 62
METAtmosphere
 jeod::METAtmosphere, 30
METAtmosphereChemical
 jeod::METAtmosphereChemical, 41
METAtmosphereState
 jeod::METAtmosphereState, 43, 44
METAtmosphereStateVars
 jeod::METAtmosphereStateVars, 46
METAtmosphereThermal
 jeod::METAtmosphereThermal, 49
max_days_this_year
 jeod::METAtmosphere, 36
met_atmos
 jeod::METAtmosphereState, 44
met_data_wind_velocity.hh, 63
minutes_per_day
 jeod::METAtmosphere, 36
Models, 11
 ∼Atmosphere, 13
 ∼WindVelocity, 13
 active, 13
 array_index, 13
 Atmosphere, 12
 init_attrjeod__Atmosphere, 14
 init_attrjeod__WindVelocity, 14
 InputProcessor, 14
 num_layers, 14
 omega, 14
 omega_scale_alt, 14
 omega_scale_fac, 14
 operator=, 12
 update_atmosphere, 12
 update_wind, 12
 WindVelocity, 13
modify_densities
 jeod::METAtmosphere, 32
mol_weight
 jeod::METAtmosphereChemical, 42
 jeod::METAtmosphereStateVars, 47
mol_weight_barometric_ceiling
 jeod::METAtmosphere, 36
mol_wt_coeffs
 jeod::METAtmosphere, 37

N2
 jeod::METAtmosphereStateVars, 47
nominal_mol_weight
 jeod::METAtmosphereChemical, 42
num_density
 jeod::METAtmosphereChemical, 42
num_integ_divisions
 jeod::METAtmosphere, 37
num_layers
 jeod::WindVelocity_wind_velocity_default_data, 53
 Models, 14
num_mol_wt_coeffs
 jeod::METAtmosphere, 37
num_species
 jeod::METAtmosphereChemical, 42
numerical_warning
 jeod::AtmosphereMessages, 23

omega
 jeod::WindVelocity_wind_velocity_default_data, 53
 Models, 14
omega_scale_alt
 jeod::WindVelocity_wind_velocity_default_data, 53
 Models, 14
omega_scale_fac
 jeod::WindVelocity_wind_velocity_default_data, 53
 Models, 14
operator=
 jeod::AtmosphereMessages, 22
 jeod::AtmosphereState, 26
 jeod::METAtmosphere, 32
 jeod::METAtmosphereChemical, 41
 jeod::METAtmosphereState, 44
 jeod::METAtmosphereStateVars, 46
 jeod::METAtmosphereThermal, 49
 jeod::WindVelocityBase, 54
 Models, 12
Ox
 jeod::METAtmosphereStateVars, 47
Ox2
 jeod::METAtmosphereStateVars, 48

PATH
 Atmosphere, 17
pfix_pos
 jeod::AtmosphereState, 27
pressure
 jeod::AtmosphereState, 27

R_gas_constant
 jeod::METAtmosphere, 37

solar_declination_angle
 jeod::METAtmosphere, 37
solar_hour_angle
 jeod::METAtmosphere, 37
solar_max.cc, 63
 JEOD_FRIEND_CLASS, 63
solar_max.hh, 64
solar_mean.cc, 64
 JEOD_FRIEND_CLASS, 64
solar_mean.hh, 64
solar_min.cc, 64
 JEOD_FRIEND_CLASS, 65
solar_min.hh, 65
species
 jeod::METAtmosphere, 37
state

jeod::METAtmosphere, 38

T_125
jeod::METAtmosphereThermal, 50
T_90
jeod::METAtmosphereThermal, 50
T_exosphere
jeod::METAtmosphereThermal, 50
T_out
jeod::METAtmosphereThermal, 51
temperature
jeod::AtmosphereState, 28
thermal
jeod::METAtmosphere, 38
three_pi_two
jeod::METAtmosphere, 38
tjt_year_start
jeod::METAtmosphere, 38
trunc_julian_time
jeod::METAtmosphere, 38
two_pi
jeod::METAtmosphere, 38

update
jeod::METAtmosphereThermal, 49
update_atmosphere
jeod::METAtmosphere, 32
Models, 12
update_state
jeod::AtmosphereState, 26
jeod::METAtmosphereState, 44
update_time
jeod::METAtmosphere, 33
update_wind
jeod::AtmosphereState, 26
jeod::WindVelocityBase, 54
Models, 12

wind
jeod::AtmosphereState, 28
wind_velocity.cc, 65
wind_velocity.hh, 66
wind_velocity_base.cc, 66
wind_velocity_base.hh, 66
WindVelocity
Models, 13
WindVelocity_wind_velocity_default_data
jeod::WindVelocity_wind_velocity_default_data, 52
WindVelocityBase
jeod::WindVelocityBase, 54

year
jeod::METAtmosphere, 39