

AtmosphereModel

5.1

Generated by Doxygen 1.8.5

Mon Jul 31 2023 11:39:30

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	11
6.2	Environment	12
6.2.1	Detailed Description	12
6.3	Atmosphere	13
6.3.1	Detailed Description	14
6.3.2	Macro Definition Documentation	14
6.3.2.1	_USE_MATH_DEFINES_	14
6.3.2.2	PATH	14
6.4	BaseAtmosphere	15
6.4.1	Detailed Description	15
7	Namespace Documentation	17
7.1	jeod Namespace Reference	17
7.1.1	Detailed Description	17
8	Data Structure Documentation	19
8.1	jeod::Atmosphere Class Reference	19

8.1.1	Detailed Description	20
8.1.2	Constructor & Destructor Documentation	20
8.1.2.1	Atmosphere	20
8.1.2.2	~Atmosphere	20
8.1.2.3	Atmosphere	20
8.1.3	Member Function Documentation	20
8.1.3.1	operator=	20
8.1.3.2	update_atmosphere	20
8.1.4	Friends And Related Function Documentation	20
8.1.4.1	init_attrjeod__Atmosphere	20
8.1.4.2	InputProcessor	20
8.1.5	Field Documentation	20
8.1.5.1	active	20
8.2	jeod::AtmosphereMessages Class Reference	21
8.2.1	Detailed Description	21
8.2.2	Constructor & Destructor Documentation	21
8.2.2.1	AtmosphereMessages	21
8.2.2.2	AtmosphereMessages	21
8.2.3	Member Function Documentation	21
8.2.3.1	operator=	21
8.2.4	Friends And Related Function Documentation	21
8.2.4.1	init_attrjeod__AtmosphereMessages	21
8.2.4.2	InputProcessor	21
8.2.5	Field Documentation	22
8.2.5.1	framework_error	22
8.2.5.2	framework_warning	22
8.2.5.3	initialization_error	22
8.2.5.4	numerical_warning	22
8.3	jeod::AtmosphereState Class Reference	23
8.3.1	Detailed Description	24
8.3.2	Constructor & Destructor Documentation	24
8.3.2.1	AtmosphereState	24
8.3.2.2	AtmosphereState	24
8.3.2.3	~AtmosphereState	24
8.3.2.4	AtmosphereState	24
8.3.3	Member Function Documentation	24
8.3.3.1	operator=	24
8.3.3.2	update_state	25
8.3.3.3	update_state	25
8.3.3.4	update_wind	25

8.3.4	Friends And Related Function Documentation	25
8.3.4.1	init_attrjeod__AtmosphereState	25
8.3.4.2	InputProcessor	25
8.3.5	Field Documentation	25
8.3.5.1	active	25
8.3.5.2	atmos	26
8.3.5.3	density	26
8.3.5.4	pfix_pos	26
8.3.5.5	pressure	26
8.3.5.6	temperature	26
8.3.5.7	wind	26
8.4	jeod::METAtmosphere Class Reference	27
8.4.1	Detailed Description	28
8.4.2	Member Enumeration Documentation	29
8.4.2.1	AtmosMETGeoIndexType	29
8.4.3	Constructor & Destructor Documentation	29
8.4.3.1	METAtmosphere	29
8.4.3.2	~METAtmosphere	29
8.4.3.3	METAtmosphere	29
8.4.4	Member Function Documentation	29
8.4.4.1	apply_gauss_quadrature	29
8.4.4.2	atmos_MET_FAIR5	29
8.4.4.3	compute_exospheric_temperature	29
8.4.4.4	compute_mol_wt	29
8.4.4.5	compute_seasonal_lat_variation_He	30
8.4.4.6	compute_seasonal_latitude_variation	30
8.4.4.7	compute_solar_angles	30
8.4.4.8	jacchia	30
8.4.4.9	modify_densities	30
8.4.4.10	operator=	30
8.4.4.11	update_atmosphere	30
8.4.4.12	update_atmosphere	31
8.4.4.13	update_atmosphere	31
8.4.4.14	update_time	31
8.4.5	Friends And Related Function Documentation	31
8.4.5.1	init_attrjeod__METAtmosphere	31
8.4.5.2	InputProcessor	31
8.4.6	Field Documentation	32
8.4.6.1	altitude_km	32
8.4.6.2	Avogadro	32

8.4.6.3	barometric_equation_ceiling	32
8.4.6.4	base_fairing_height	32
8.4.6.5	day_of_year	32
8.4.6.6	days_per_century	32
8.4.6.7	days_per_year	33
8.4.6.8	deg_to_rad	33
8.4.6.9	F10	33
8.4.6.10	F10B	33
8.4.6.11	fairing_k	33
8.4.6.12	fraction_of_year	33
8.4.6.13	gauss_altitudes	34
8.4.6.14	gauss_n	34
8.4.6.15	geo_index	34
8.4.6.16	geo_index_type	34
8.4.6.17	latitude	34
8.4.6.18	longitude	35
8.4.6.19	max_days_this_year	35
8.4.6.20	minutes_per_day	35
8.4.6.21	mol_weight_barometric_ceiling	35
8.4.6.22	mol_wt_coeffs	35
8.4.6.23	num_integ_divisions	35
8.4.6.24	num_mol_wt_coeffs	36
8.4.6.25	R_gas_constant	36
8.4.6.26	solar_declination_angle	36
8.4.6.27	solar_hour_angle	36
8.4.6.28	species	36
8.4.6.29	state	36
8.4.6.30	thermal	36
8.4.6.31	three_pi_two	37
8.4.6.32	tjt_year_start	37
8.4.6.33	trunc_julian_time	37
8.4.6.34	two_pi	37
8.4.6.35	year	37
8.5	jeod::METAtmosphere_solar_max_default_data Class Reference	37
8.5.1	Detailed Description	38
8.5.2	Member Function Documentation	38
8.5.2.1	initialize	38
8.6	jeod::METAtmosphere_solar_mean_default_data Class Reference	38
8.6.1	Detailed Description	38
8.6.2	Member Function Documentation	38

8.6.2.1	initialize	38
8.7	jeod::METAtmosphere_solar_min_default_data Class Reference	38
8.7.1	Detailed Description	39
8.7.2	Member Function Documentation	39
8.7.2.1	initialize	39
8.8	jeod::METAtmosphereChemical Class Reference	39
8.8.1	Detailed Description	40
8.8.2	Constructor & Destructor Documentation	40
8.8.2.1	METAtmosphereChemical	40
8.8.2.2	~METAtmosphereChemical	40
8.8.2.3	METAtmosphereChemical	40
8.8.3	Member Function Documentation	40
8.8.3.1	operator=	40
8.8.4	Friends And Related Function Documentation	40
8.8.4.1	init_attrjeod__METAtmosphereChemical	40
8.8.4.2	InputProcessor	40
8.8.5	Field Documentation	40
8.8.5.1	frac	40
8.8.5.2	mol_weight	40
8.8.5.3	nominal_mol_weight	40
8.8.5.4	num_density	41
8.8.5.5	num_species	41
8.9	jeod::METAtmosphereState Class Reference	41
8.9.1	Detailed Description	42
8.9.2	Constructor & Destructor Documentation	42
8.9.2.1	METAtmosphereState	42
8.9.2.2	METAtmosphereState	42
8.9.2.3	~METAtmosphereState	42
8.9.2.4	METAtmosphereState	42
8.9.3	Member Function Documentation	42
8.9.3.1	operator=	42
8.9.3.2	update_state	42
8.9.3.3	update_state	42
8.9.4	Friends And Related Function Documentation	43
8.9.4.1	init_attrjeod__METAtmosphereState	43
8.9.4.2	InputProcessor	43
8.9.5	Field Documentation	43
8.9.5.1	met_atmos	43
8.10	jeod::METAtmosphereStateVars Class Reference	43
8.10.1	Detailed Description	44

8.10.2	Constructor & Destructor Documentation	44
8.10.2.1	METAtmosphereStateVars	44
8.10.2.2	METAtmosphereStateVars	44
8.10.2.3	~METAtmosphereStateVars	44
8.10.2.4	METAtmosphereStateVars	44
8.10.3	Member Function Documentation	45
8.10.3.1	operator=	45
8.10.4	Friends And Related Function Documentation	45
8.10.4.1	init_attrjeod__METAtmosphereStateVars	45
8.10.4.2	InputProcessor	45
8.10.5	Field Documentation	45
8.10.5.1	A	45
8.10.5.2	exo_temp	45
8.10.5.3	He	45
8.10.5.4	Hyd	45
8.10.5.5	log10_dens	46
8.10.5.6	mol_weight	46
8.10.5.7	N2	46
8.10.5.8	Ox	46
8.10.5.9	Ox2	46
8.11	jeod::METAtmosphereThermal Class Reference	46
8.11.1	Detailed Description	47
8.11.2	Constructor & Destructor Documentation	47
8.11.2.1	METAtmosphereThermal	47
8.11.2.2	~METAtmosphereThermal	47
8.11.2.3	METAtmosphereThermal	48
8.11.3	Member Function Documentation	48
8.11.3.1	compute_temperature	48
8.11.3.2	generate_base_temperature	48
8.11.3.3	operator=	48
8.11.3.4	update	48
8.11.4	Friends And Related Function Documentation	48
8.11.4.1	init_attrjeod__METAtmosphereThermal	48
8.11.4.2	InputProcessor	48
8.11.5	Field Documentation	48
8.11.5.1	altitude_km	48
8.11.5.2	k_1	48
8.11.5.3	k_3	49
8.11.5.4	k_4	49
8.11.5.5	T_125	49

8.11.5.6	T_90	49
8.11.5.7	T_exosphere	49
8.11.5.8	T_out	49
8.12	jeod::WindVelocity::OmegaTableEntry Struct Reference	49
8.12.1	Detailed Description	50
8.12.2	Field Documentation	50
8.12.2.1	altitude	50
8.12.2.2	scale_factor	50
8.13	jeod::WindVelocity Class Reference	50
8.13.1	Detailed Description	51
8.13.2	Constructor & Destructor Documentation	51
8.13.2.1	WindVelocity	51
8.13.2.2	~WindVelocity	52
8.13.2.3	WindVelocity	52
8.13.3	Member Function Documentation	52
8.13.3.1	get_num_layers	52
8.13.3.2	get_omega_scale_table	52
8.13.3.3	operator=	52
8.13.3.4	set_omega_scale_table	52
8.13.3.5	set_omega_scale_table	52
8.13.3.6	update_wind	52
8.13.4	Friends And Related Function Documentation	53
8.13.4.1	init_attrjeod__WindVelocity	53
8.13.4.2	InputProcessor	53
8.13.5	Field Documentation	53
8.13.5.1	active	53
8.13.5.2	array_index	53
8.13.5.3	first_pass	53
8.13.5.4	increasing_altitude	53
8.13.5.5	num_layers	53
8.13.5.6	omega	54
8.13.5.7	omega_scale_table	54
8.14	jeod::WindVelocity_wind_velocity_default_data Class Reference	54
8.14.1	Detailed Description	54
8.14.2	Constructor & Destructor Documentation	55
8.14.2.1	WindVelocity_wind_velocity_default_data	55
8.14.3	Member Function Documentation	55
8.14.3.1	initialize	55
8.14.3.2	initialize	55
8.14.4	Field Documentation	55

8.14.4.1	num_layers	55
8.14.4.2	omega	55
8.14.4.3	omega_scale_alt	55
8.14.4.4	omega_scale_fac	55
8.15	jeod::WindVelocityBase Class Reference	56
8.15.1	Detailed Description	56
8.15.2	Constructor & Destructor Documentation	56
8.15.2.1	WindVelocityBase	56
8.15.2.2	~WindVelocityBase	56
8.15.2.3	WindVelocityBase	56
8.15.3	Member Function Documentation	56
8.15.3.1	operator=	56
8.15.3.2	update_wind	57
8.15.4	Friends And Related Function Documentation	58
8.15.4.1	init_attrjeod__WindVelocityBase	58
8.15.4.2	InputProcessor	58
9	File Documentation	59
9.1	atmosphere.hh File Reference	59
9.1.1	Detailed Description	59
9.2	atmosphere_messages.cc File Reference	59
9.2.1	Detailed Description	60
9.3	atmosphere_messages.hh File Reference	60
9.3.1	Detailed Description	60
9.4	atmosphere_state.cc File Reference	60
9.4.1	Detailed Description	60
9.5	atmosphere_state.hh File Reference	61
9.6	class_declarations.hh File Reference	61
9.6.1	Detailed Description	61
9.7	class_declarations.hh File Reference	61
9.7.1	Detailed Description	61
9.8	data_met_wind_velocity.cc File Reference	62
9.8.1	Macro Definition Documentation	62
9.8.1.1	JEOD_FRIEND_CLASS	62
9.9	MET_atmosphere.cc File Reference	62
9.9.1	Detailed Description	62
9.10	MET_atmosphere.hh File Reference	63
9.10.1	Detailed Description	63
9.11	MET_atmosphere_state.cc File Reference	63
9.12	MET_atmosphere_state.hh File Reference	63

9.12.1 Detailed Description	64
9.13 MET_atmosphere_state_vars.cc File Reference	64
9.13.1 Detailed Description	64
9.14 MET_atmosphere_state_vars.hh File Reference	64
9.14.1 Detailed Description	65
9.15 met_data_wind_velocity.hh File Reference	65
9.16 solar_max.cc File Reference	65
9.16.1 Macro Definition Documentation	65
9.16.1.1 JEOD_FRIEND_CLASS	65
9.17 solar_max.hh File Reference	66
9.18 solar_mean.cc File Reference	66
9.18.1 Macro Definition Documentation	66
9.18.1.1 JEOD_FRIEND_CLASS	66
9.19 solar_mean.hh File Reference	66
9.20 solar_min.cc File Reference	66
9.20.1 Macro Definition Documentation	67
9.20.1.1 JEOD_FRIEND_CLASS	67
9.21 solar_min.hh File Reference	67
9.22 wind_velocity.cc File Reference	67
9.22.1 Detailed Description	67
9.23 wind_velocity.hh File Reference	68
9.23.1 Detailed Description	68
9.24 wind_velocity_base.cc File Reference	68
9.24.1 Detailed Description	68
9.25 wind_velocity_base.hh File Reference	68
9.25.1 Detailed Description	69

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models	11
Environment	12
Atmosphere	13
BaseAtmosphere	15

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod	Namespace jeod	17
----------------------	--------------------------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

jeod::Atmosphere	19
jeod::METAtmosphere	27
jeod::AtmosphereMessages	21
jeod::AtmosphereState	23
jeod::METAtmosphereStateVars	43
jeod::METAtmosphereState	41
jeod::METAtmosphere_solar_max_default_data	37
jeod::METAtmosphere_solar_mean_default_data	38
jeod::METAtmosphere_solar_min_default_data	38
jeod::METAtmosphereChemical	39
jeod::METAtmosphereThermal	46
jeod::WindVelocity::OmegaTableEntry	49
jeod::WindVelocity	50
jeod::WindVelocity_wind_velocity_default_data	54
jeod::WindVelocityBase	56

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	19
jeod::AtmosphereMessages	
Describes messages used in the Atmosphere model	21
jeod::AtmosphereState	
A generic base class for atmosphere state, containing common atmosphere state parameters, i.e	23
jeod::METAtmosphere	27
jeod::METAtmosphere_solar_max_default_data	37
jeod::METAtmosphere_solar_mean_default_data	38
jeod::METAtmosphere_solar_min_default_data	38
jeod::METAtmosphereChemical	
The chemical composition of the MET Atmosphere	39
jeod::METAtmosphereState	
The MET specific implementation of AtmosphereState	41
jeod::METAtmosphereStateVars	
The data variables component of the MET specific implementation of AtmosphereState	43
jeod::METAtmosphereThermal	
The Thermal aspect of the computation	46
jeod::WindVelocity::OmegaTableEntry	
An entry in an omega scale table	49
jeod::WindVelocity	
A generic wind velocity implementation	50
jeod::WindVelocity_wind_velocity_default_data	54
jeod::WindVelocityBase	
The generic base class for wind velocity classes	56

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	59
atmosphere_messages.cc	Implement atmosphere_messages	59
atmosphere_messages.hh	Implement atmosphere_messages	60
atmosphere_state.cc	Implementation of the base atmosphere-state model	60
atmosphere_state.hh	61
base_atmos/include/class_declarations.hh	Forward declarations of classes defined for JEOD 2.0 Atmosphere	61
MET/include/class_declarations.hh	Forward declarations of classes defined for JEOD 2.0 Atmosphere	61
data_met_wind_velocity.cc	62
MET_atmosphere.cc	Implementation of MET atmosphere model	62
MET_atmosphere.hh	Implement the MET atmosphere using the atmosphere framework	63
MET_atmosphere_state.cc	63
MET_atmosphere_state.hh	Implement the MET atmosphere state using the atmosphere framework	63
MET_atmosphere_state_vars.cc	Implementation of MET atmosphere model	64
MET_atmosphere_state_vars.hh	Implement the MET atmosphere state variables using the atmosphere framework	64
met_data_wind_velocity.hh	65
solar_max.cc	65
solar_max.hh	66
solar_mean.cc	66
solar_mean.hh	66
solar_min.cc	66
solar_min.hh	67
wind_velocity.cc	General base class for wind velocity models	67
wind_velocity.hh	A wind velocity model based on winds caused by rotation of the planet	68
wind_velocity_base.cc	General base class for wind velocity models	68

wind_velocity_base.hh	
General base class for wind velocity models	68

Chapter 6

Module Documentation

6.1 Models

Modules

- [Environment](#)

6.1.1 Detailed Description

6.2 Environment

Modules

- [Atmosphere](#)

6.2.1 Detailed Description

6.3 Atmosphere

Modules

- [BaseAtmosphere](#)

Files

- file [atmosphere_messages.hh](#)
Implement atmosphere_messages.
- file [atmosphere.hh](#)
General base class for atmosphere models.
- file [base_atmos/include/class_declarations.hh](#)
Forward declarations of classes defined for JEOD 2.0 Atmosphere.
- file [wind_velocity_base.hh](#)
General base class for wind velocity models.
- file [atmosphere_messages.cc](#)
Implement atmosphere_messages.
- file [atmosphere_state.cc](#)
Implementation of the base atmosphere-state model.
- file [wind_velocity.cc](#)
General base class for wind velocity models.
- file [wind_velocity_base.cc](#)
General base class for wind velocity models.
- file [MET/include/class_declarations.hh](#)
Forward declarations of classes defined for JEOD 2.0 Atmosphere.
- file [MET_atmosphere.hh](#)
Implement the MET atmosphere using the atmosphere framework.
- file [MET_atmosphere_state.hh](#)
Implement the MET atmosphere state using the atmosphere framework.
- file [MET_atmosphere_state_vars.hh](#)
Implement the MET atmosphere state variables using the atmosphere framework.
- file [MET_atmosphere.cc](#)
Implementation of MET atmosphere model.
- file [MET_atmosphere.cc](#)
Implementation of MET atmosphere model.
- file [MET_atmosphere_state_vars.cc](#)
Implementation of MET atmosphere model.

Namespaces

- [jeod](#)
Namespace jeod.

Macros

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

6.3.1 Detailed Description

6.3.2 Macro Definition Documentation

6.3.2.1 `#define _USE_MATH_DEFINES_`

Definition at line 39 of file MET_atmosphere.cc.

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

Definition at line 28 of file atmosphere_messages.cc.

6.4 BaseAtmosphere

Files

- file [atmosphere.hh](#)
General base class for atmosphere models.
- file [wind_velocity.hh](#)
A wind velocity model based on winds caused by rotation of the planet.

Namespaces

- [jeod](#)
Namespace jeod.

6.4.1 Detailed Description

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class [Atmosphere](#)
A generic base class for atmospheres.
- class [AtmosphereMessages](#)
Describes messages used in the [Atmosphere](#) model.
- class [AtmosphereState](#)
A generic base class for atmosphere state, containing common atmosphere state parameters, i.e.
- class [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

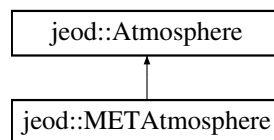
Data Structure Documentation

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 78 of file atmosphere.hh.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 `jeod::Atmosphere::Atmosphere ()` `[inline]`

Definition at line 90 of file atmosphere.hh.

8.1.2.2 `virtual jeod::Atmosphere::~~Atmosphere ()` `[inline]`, `[virtual]`

Definition at line 95 of file atmosphere.hh.

8.1.2.3 `jeod::Atmosphere::Atmosphere (const Atmosphere & rhs)` `[private]`

8.1.3 Member Function Documentation

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

8.1.3.2 `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	<i>position</i>	planet fixed position
out	<i>state</i>	The AtmosphereState

Implemented in [jeod::METAtmosphere](#).

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

8.1.4 Friends And Related Function Documentation

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

8.1.4.2 `friend class InputProcessor` `[friend]`

Definition at line 80 of file atmosphere.hh.

8.1.5 Field Documentation

8.1.5.1 `bool jeod::Atmosphere::active`

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

`trick_units(-)` activity-control flag.

Definition at line 87 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 75 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 77 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 93 of file atmosphere_messages.hh.

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

8.2.5.2 `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 100 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 88 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 105 of file atmosphere_messages.hh.

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

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

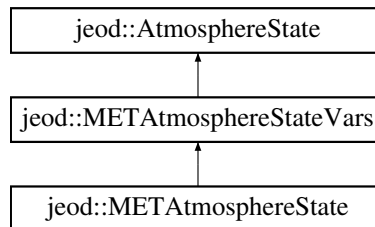
- [atmosphere_messages.hh](#)
- [atmosphere_messages.cc](#)

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 85 of file atmosphere_state.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 jeod::AtmosphereState::AtmosphereState ()

Definition at line 38 of file atmosphere_state.cc.
 References wind.

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

Definition at line 50 of file atmosphere_state.cc.
 References wind.

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

Definition at line 67 of file atmosphere_state.cc.

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

Copy Constructor.

Parameters

<i>in</i>	<i>rhs</i>	The AtmosphereState to copy from
-----------	------------	--

Definition at line 77 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

<i>in</i>	<i>rhs</i>	The AtmosphereState to copy
-----------	------------	---

Definition at line 100 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	<i>atmos_model_</i>	Atmosphere model.
in	<i>pfix_pos_</i>	Planetary fixed position.

Definition at line 125 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 be used as the [Atmosphere](#) pointer but only the values associated with [AtmosphereState](#) will be copied back out.

Reimplemented in [jeod::METAtmosphereState](#).

Definition at line 145 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	<i>wind_vel</i>	Wind velocity model.
in	<i>inrtl_pos</i>	Current inertial position. Units: M
in	<i>altitude</i>	Geodetic (elliptic) altitude. Units: M

Definition at line 164 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 87 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 91 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 103 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 95 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 104 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 97 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 93 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 99 of file atmosphere_state.hh.

Referenced by `AtmosphereState()`, and `update_wind()`.

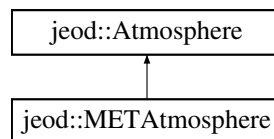
The documentation for this class was generated from the following files:

- [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](#) ()
- [~METAtmosphere](#) () override
- void [update_atmosphere](#) (const PlanetFixedPosition *pfix_pos, [AtmosphereState](#) *state) override
A pure virtual function for updating the atmosphere, and inserting.
- void [update_atmosphere](#) (const PlanetFixedPosition *pfix_pos, [METAtmosphereStateVars](#) *state)
Front-end to the computation of the [METAtmosphere](#) at the current time Inserts the results into the [METAtmosphere-StateVars](#) 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 }

Friends

- class [InputProcessor](#)
- void [init_attrjeod__METAtmosphere](#) ()

8.4.1 Detailed Description

Definition at line 175 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 180 of file MET_atmosphere.hh.

8.4.3 Constructor & Destructor Documentation

8.4.3.1 jeod::METAtmosphere::METAtmosphere ()

Definition at line 120 of file MET_atmosphere.cc.

8.4.3.2 jeod::METAtmosphere::~~METAtmosphere () [inline],[override]

Definition at line 297 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 1234 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 1101 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 623 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 1157 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 1041 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 984 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 431 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 767 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 394 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) [override],[virtual]`

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

Parameters

in	<i>position</i>	planet fixed position
out	<i>state</i>	The AtmosphereState

Implements [jeod::Atmosphere](#).

Definition at line 288 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 [METAtmosphere-StateVars](#) pointed to by *ext_state*.

This function is for a [METAtmosphereStateVars](#).

Parameters

in	<i>pfix_pos</i>	Geodetic altitude, latitude and longitude.
out	<i>ext_state</i>	Where the state results will be sent.

Definition at line 324 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	<i>pfix_pos</i>	Geodetic altitude, latitude and longitude.
----	-----------------	--

Definition at line 347 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 310 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 177 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 202 of file MET_atmosphere.hh.

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

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

trick_units(-) Avogadros number

Definition at line 241 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 206 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 252 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 218 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 246 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 240 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/rad)

degree-to-radian conversion

Definition at line 244 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 192 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 194 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(rad/km)

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

Definition at line 256 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 215 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 manageable 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 278 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 }` [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 285 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 190 of file MET_atmosphere.hh.

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

8.4.6.16 `AtmosMETGeoIndexType jeod::METAtmosphere::geo_index_type`

Definition at line 187 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(rad) Copy of vehicle latitude

Definition at line 203 of file MET_atmosphere.hh.

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

8.4.6.18 double jeod::METAtmosphere::longitude [private]

trick_units(rad) Copy of vehicle longitude

Definition at line 204 of file MET_atmosphere.hh.

Referenced by compute_solar_angles(), and update_atmosphere().

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

trick_units(count)

number of days this year (365 or 366)

Definition at line 221 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 247 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 250 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 270 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 275 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 268 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 239 of file MET_atmosphere.hh.

Referenced by `jacchia()`, and `update_atmosphere()`.

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

`trick_units(rad)` declination angle

Definition at line 226 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(rad)` solar hour angle

Definition at line 228 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 197 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 230 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 234 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 243 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 212 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 211 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 242 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 224 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 54 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 37 of file `solar_max.cc`.

References `jeod::METAtmosphere::ATMOS_MET_GL_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 54 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 37 of file `solar_mean.cc`.

References `jeod::METAtmosphere::ATMOS_MET_GL_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 54 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 37 of file `solar_min.cc`.

References `jeod::METAtmosphere::ATMOS_MET_GL_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 86 of file MET_atmosphere.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 `jeod::METAtmosphereChemical::METAtmosphereChemical ()`

Definition at line 78 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 106 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 87 of file MET_atmosphere.hh.

8.8.5 Field Documentation

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

Definition at line 96 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 99 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 102 of file MET_atmosphere.hh.

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

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

Definition at line 93 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 90 of file MET_atmosphere.hh.

Referenced by METAtmosphereChemical().

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

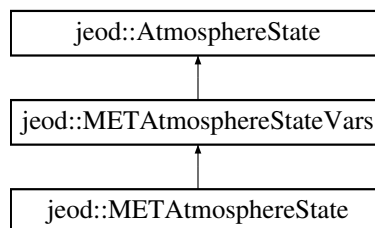
- [MET_atmosphere.hh](#)
- [MET_atmosphere.cc](#)

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)
- [~METAtmosphereState](#) () override
- void [update_state](#) ([METAtmosphere](#) *atmos_model, const PlanetFixedPosition *pfix_pos)
Updates the [METAtmosphereState](#) from the [METAtmosphere](#) pointed to by atmos_model.
- void [update_state](#) () override
Updates the [METAtmosphereState](#) from the [METAtmosphere](#) pointed to by class member atmos_model using class member pointer pfix_pos.

Private 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 83 of file MET_atmosphere_state.hh.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 `jeod::METAtmosphereState::METAtmosphereState ()`

Definition at line 55 of file MET_atmosphere_state.cc.

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

Definition at line 61 of file MET_atmosphere_state.cc.

8.9.2.3 `jeod::METAtmosphereState::~~METAtmosphereState ()` `[inline]`, `[override]`

Definition at line 102 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

<code>in</code>	<code><i>atmos_model_</i></code>	METAtmosphere Model.
<code>in</code>	<code><i>pfix_pos_</i></code>	Current vehicle position.

Definition at line 79 of file MET_atmosphere_state.cc.

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

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

Updates the [METAtmosphereState](#) from the [METAtmosphere](#) pointed to by class member `atmos_model` using class member pointer `pfix_pos`.

This is a specific function for the case of an [METAtmosphere](#) state updating an [METAtmosphere](#) when constructed with the pointers set.

Reimplemented from [jeod::AtmosphereState](#).

Definition at line 97 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 86 of file MET_atmosphere_state.hh.

8.9.5 Field Documentation

8.9.5.1 `METAtmosphere* jeod::METAtmosphereState::met_atmos` [\[private\]](#)

Definition at line 89 of file MET_atmosphere_state.hh.

Referenced by [update_state\(\)](#).

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

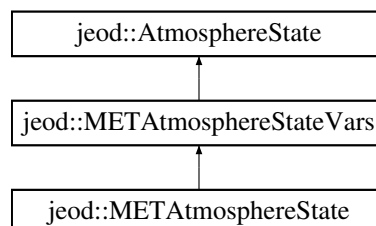
- [MET_atmosphere_state.hh](#)
- [MET_atmosphere_state.cc](#)

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)
- [~METAtmosphereStateVars](#) () override
- [METAtmosphereStateVars](#) (const [METAtmosphereStateVars](#) &rhs)
Copy Constructor.
- [METAtmosphereStateVars](#) & operator= (const [METAtmosphereStateVars](#) &rhs)
METAtmosphereStateVars operator =.

Data Fields

- double [exo_temp](#)
- double [log10_dens](#)
- double [mol_weight](#)
- double [N2](#)
- double [Ox2](#)
- double [Ox](#)
- double [A](#)
- double [He](#)
- double [Hyd](#)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__METAtmosphereStateVars](#) ()

Additional Inherited Members

8.10.1 Detailed Description

The data variables component of the MET specific implementation of [AtmosphereState](#).

Definition at line 82 of file MET_atmosphere_state_vars.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 `jeod::METAtmosphereStateVars::METAtmosphereStateVars ()`

Definition at line 48 of file MET_atmosphere_state_vars.cc.

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

Definition at line 62 of file MET_atmosphere_state_vars.cc.

8.10.2.3 `jeod::METAtmosphereStateVars::~~METAtmosphereStateVars ()` `[override]`

Definition at line 83 of file MET_atmosphere_state_vars.cc.

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

Copy Constructor.

Parameters

<code>in</code>	<code>rhs</code>	The METAtmosphereStateVars to copy
-----------------	------------------	--

Definition at line 91 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

<i>in</i>	<i>rhs</i>	The METAtmosphereStateVars to copy from
-----------	------------	---

Definition at line 115 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 84 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 93 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 87 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 94 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 95 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 88 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 89 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 90 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 92 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 91 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 121 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 103 of file MET_atmosphere.cc.

8.11.2.2 virtual jeod::METAtmosphereThermal::~~METAtmosphereThermal () `[inline], [virtual]`

Definition at line 130 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 225 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 172 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 122 of file MET_atmosphere.hh.

8.11.5 Field Documentation

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

Definition at line 159 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 130 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 142 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 146 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 153 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 150 of file MET_atmosphere.hh.

Referenced by `compute_temperature()`.

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

Definition at line 156 of file MET_atmosphere.hh.

Referenced by `compute_temperature()`, and `update()`.

8.11.5.8 `double jeod::METAtmosphereThermal::T_out`

Definition at line 124 of file MET_atmosphere.hh.

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

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

- [MET_atmosphere.hh](#)
- [MET_atmosphere.cc](#)

8.12 `jeod::WindVelocity::OmegaTableEntry` Struct Reference

An entry in an omega scale table.

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

Data Fields

- double [altitude](#)
Altitude at which omega is multiplied by the corresponding factor.
- double [scale_factor](#)
Factor by which omega is multiplied depending on altitude.

8.12.1 Detailed Description

An entry in an omega scale table.

Definition at line 112 of file `wind_velocity.hh`.

8.12.2 Field Documentation

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

Altitude at which omega is multiplied by the corresponding factor.

`trick_units(m)`

Definition at line 117 of file `wind_velocity.hh`.

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

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

Factor by which omega is multiplied depending on altitude.

`trick_units(-)`

Definition at line 122 of file `wind_velocity.hh`.

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

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

- [wind_velocity.hh](#)

8.13 jeod::WindVelocity Class Reference

A generic wind velocity implementation.

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

Data Structures

- struct [OmegaTableEntry](#)
An entry in an omega scale table.

Public Member Functions

- [WindVelocity](#) ()
Default 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.
- unsigned int [get_num_layers](#) ()
- void [set_omega_scale_table](#) (double altitude, double factor)
- void [set_omega_scale_table](#) (unsigned int [num_layers](#), const double *altitude, const double *factor)
- [OmegaTableEntry](#) * [get_omega_scale_table](#) ()

Data Fields

- bool [active](#)
trick_units(-)
- double [omega](#)
The rotational velocity of the planet.

Protected Attributes

- unsigned int [num_layers](#)
Number of altitude layers.
- [OmegaTableEntry](#) * [omega_scale_table](#)
Table of factors to scale omega based on altitude.

Private Member Functions

- [WindVelocity](#) (const [WindVelocity](#) &rhs)
- [WindVelocity](#) & [operator=](#) (const [WindVelocity](#) &rhs)

Private Attributes

- unsigned int [array_index](#)
last known index into the arrays
- bool [first_pass](#)
Altitude direction check flag.
- bool [increasing_altitude](#)
Altitude increasing or decreasing flag.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__WindVelocity](#) ()

8.13.1 Detailed Description

A generic wind velocity implementation.

Definition at line 76 of file `wind_velocity.hh`.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 jeod::WindVelocity::WindVelocity (void)

Default Constructor.

Definition at line 43 of file `wind_velocity.cc`.

8.13.2.2 jeod::WindVelocity::~~WindVelocity (void) [virtual]

Destructor.

Definition at line 60 of file wind_velocity.cc.

References omega_scale_table.

8.13.2.3 jeod::WindVelocity::WindVelocity (const WindVelocity & rhs) [private]

8.13.3 Member Function Documentation

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

Definition at line 197 of file wind_velocity.cc.

References num_layers.

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

Definition at line 233 of file wind_velocity.cc.

References omega_scale_table.

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

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

Definition at line 202 of file wind_velocity.cc.

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

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

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

Definition at line 213 of file wind_velocity.cc.

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

8.13.3.6 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	<i>inertial_pos</i>	The inertial position of the vehicle Units: M
in	<i>altitude</i>	The altitude of the vehicle Units: M

out	<i>wind_inertial</i>	The wind, in the inertial frame, applied to the vehicle Units: M/s
-----	----------------------	---

Definition at line 76 of file wind_velocity.cc.

References `active`, `jeod::WindVelocity::OmegaTableEntry::altitude`, `array_index`, `first_pass`, `jeod::AtmosphereMessages::framework_error`, `increasing_altitude`, `num_layers`, `omega`, `omega_scale_table`, and `jeod::WindVelocity::OmegaTableEntry::scale_factor`.

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

8.13.4 Friends And Related Function Documentation

8.13.4.1 `void init_attrjeod_WindVelocity () [friend]`

8.13.4.2 `friend class InputProcessor [friend]`

Definition at line 78 of file wind_velocity.hh.

8.13.5 Field Documentation

8.13.5.1 `bool jeod::WindVelocity::active`

`trick_units(-)`

Definition at line 100 of file wind_velocity.hh.

Referenced by `update_wind()`.

8.13.5.2 `unsigned int jeod::WindVelocity::array_index [private]`

last known index into the arrays

Definition at line 142 of file wind_velocity.hh.

Referenced by `update_wind()`.

8.13.5.3 `bool jeod::WindVelocity::first_pass [private]`

Altitude direction check flag.

`trick_units(-)`

Definition at line 147 of file wind_velocity.hh.

Referenced by `update_wind()`.

8.13.5.4 `bool jeod::WindVelocity::increasing_altitude [private]`

Altitude increasing or decreasing flag.

`trick_units(-)`

Definition at line 152 of file wind_velocity.hh.

Referenced by `update_wind()`.

8.13.5.5 `unsigned int jeod::WindVelocity::num_layers [protected]`

Number of altitude layers.

trick_units(count)

Definition at line 131 of file wind_velocity.hh.

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

8.13.5.6 double jeod::WindVelocity::omega

The rotational velocity of the planet.

trick_units(rad/s)

Definition at line 105 of file wind_velocity.hh.

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

8.13.5.7 OmegaTableEntry* jeod::WindVelocity::omega_scale_table [protected]

Table of factors to scale omega based on altitude.

Definition at line 136 of file wind_velocity.hh.

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

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

- [wind_velocity.hh](#)
- [wind_velocity.cc](#)

8.14 jeod::WindVelocity_wind_velocity_default_data Class Reference

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

Public Member Functions

- [WindVelocity_wind_velocity_default_data](#) ()
- 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

8.14.1 Detailed Description

Definition at line 56 of file met_data_wind_velocity.hh.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 jeod::WindVelocity_wind_velocity_default_data::WindVelocity_wind_velocity_default_data ()

Definition at line 38 of file data_met_wind_velocity.cc.

References num_layers, omega_scale_alt, and omega_scale_fac.

8.14.3 Member Function Documentation

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

Definition at line 60 of file data_met_wind_velocity.cc.

References initialize().

Referenced by initialize().

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

Definition at line 72 of file data_met_wind_velocity.cc.

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

8.14.4 Field Documentation

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

Definition at line 58 of file met_data_wind_velocity.hh.

Referenced by initialize(), and WindVelocity_wind_velocity_default_data().

8.14.4.2 double jeod::WindVelocity_wind_velocity_default_data::omega

Definition at line 64 of file met_data_wind_velocity.hh.

Referenced by initialize().

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

Definition at line 62 of file met_data_wind_velocity.hh.

Referenced by initialize(), and WindVelocity_wind_velocity_default_data().

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

Definition at line 60 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](#)

8.15 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](#) ()

8.15.1 Detailed Description

The generic base class for wind velocity classes.

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

Definition at line 77 of file `wind_velocity_base.hh`.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 jeod::WindVelocityBase::WindVelocityBase (void)

Default Constructor.

Definition at line 34 of file `wind_velocity_base.cc`.

8.15.2.2 jeod::WindVelocityBase::~~WindVelocityBase (void) [virtual]

Destructor.

Definition at line 46 of file `wind_velocity_base.cc`.

8.15.2.3 jeod::WindVelocityBase::WindVelocityBase (const WindVelocityBase & rhs) [private]

8.15.3 Member Function Documentation

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

8.15.3.2 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	<i>position</i>	The position of the vehicle, however the specific implementation defines it
in	<i>altitude</i>	The altitude of the vehicle, however the specific implementation defines it
out	<i>wind_inertial</i>	The wind applied to the craft, in the inertial frame

Definition at line 62 of file `wind_velocity_base.cc`.

References `jeod::AtmosphereMessages::framework_warning`.

8.15.4 Friends And Related Function Documentation

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

8.15.4.2 `friend class InputProcessor` [*friend*]

Definition at line 79 of file `wind_velocity_base.hh`.

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

- [wind_velocity_base.hh](#)
- [wind_velocity_base.cc](#)

Chapter 9

File Documentation

9.1 atmosphere.hh File Reference

General base class for atmosphere models.

```
#include "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 <cstdint>
#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 <cstdint>
#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 21 of file data_met_wind_velocity.cc.

9.9 MET_atmosphere.cc File Reference

Implementation of MET atmosphere model.

```
#include <cstdint>
#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 <cstdint>
#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 23 of file solar_max.cc.

9.17 solar_max.hh File Reference

Data Structures

- class [jeod::METAtmosphere_solar_max_default_data](#)

Namespaces

- [jeod](#)
Namespace jeod.

9.18 solar_mean.cc File Reference

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

Namespaces

- [jeod](#)
Namespace jeod.

Macros

- #define [JEOD_FRIEND_CLASS](#) METAtmosphere_solar_mean_default_data

9.18.1 Macro Definition Documentation

9.18.1.1 #define JEOD_FRIEND_CLASS METAtmosphere_solar_mean_default_data

Definition at line 23 of file solar_mean.cc.

9.19 solar_mean.hh File Reference

Data Structures

- class [jeod::METAtmosphere_solar_mean_default_data](#)

Namespaces

- [jeod](#)
Namespace jeod.

9.20 solar_min.cc File Reference

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

Namespaces

- [jeod](#)

Namespace jeod.

Macros

- `#define` [JEOD_FRIEND_CLASS](#) METAtmosphere_solar_min_default_data

9.20.1 Macro Definition Documentation

9.20.1.1 `#define` JEOD_FRIEND_CLASS METAtmosphere_solar_min_default_data

Definition at line 23 of file solar_min.cc.

9.21 solar_min.hh File Reference

Data Structures

- class [jeod::METAtmosphere_solar_min_default_data](#)

Namespaces

- [jeod](#)

Namespace jeod.

9.22 wind_velocity.cc File Reference

General base class for wind velocity models.

```
#include <cstdlib>
#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

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

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

Data Structures

- class [jeod::WindVelocity](#)
A generic wind velocity implementation.
- struct [jeod::WindVelocity::OmegaTableEntry](#)
An entry in an omega scale table.

Namespaces

- [jeod](#)
Namespace jeod.

9.23.1 Detailed Description

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

Definition in file [wind_velocity.hh](#).

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
 - jeod::Atmosphere, [20](#)
- ~AtmosphereState
 - jeod::AtmosphereState, [24](#)
- ~METAtmosphere
 - jeod::METAtmosphere, [29](#)
- ~METAtmosphereChemical
 - jeod::METAtmosphereChemical, [40](#)
- ~METAtmosphereState
 - jeod::METAtmosphereState, [42](#)
- ~METAtmosphereStateVars
 - jeod::METAtmosphereStateVars, [44](#)
- ~METAtmosphereThermal
 - jeod::METAtmosphereThermal, [47](#)
- ~WindVelocity
 - jeod::WindVelocity, [51](#)
- ~WindVelocityBase
 - jeod::WindVelocityBase, [56](#)
- A
 - jeod::METAtmosphereStateVars, [45](#)
- ATMOS_MET_GI_AP
 - jeod::METAtmosphere, [29](#)
- ATMOS_MET_GI_KP
 - jeod::METAtmosphere, [29](#)
- active
 - jeod::Atmosphere, [20](#)
 - jeod::AtmosphereState, [25](#)
 - jeod::WindVelocity, [53](#)
- altitude
 - jeod::WindVelocity::OmegaTableEntry, [50](#)
- altitude_km
 - jeod::METAtmosphere, [32](#)
 - jeod::METAtmosphereThermal, [48](#)
- apply_gauss_quadrature
 - jeod::METAtmosphere, [29](#)
- array_index
 - jeod::WindVelocity, [53](#)
- atmos
 - jeod::AtmosphereState, [26](#)
- atmos_MET_FAIR5
 - jeod::METAtmosphere, [29](#)
- AtmosMETGeoIndexType
 - jeod::METAtmosphere, [29](#)
- Atmosphere, [13](#)
 - jeod::Atmosphere, [20](#)
 - PATH, [14](#)
- atmosphere.hh, [59](#)
- atmosphere_messages.cc, [59](#)
- atmosphere_messages.hh, [60](#)
- atmosphere_state.cc, [60](#)
- atmosphere_state.hh, [61](#)
- AtmosphereMessages
 - jeod::AtmosphereMessages, [21](#)
- AtmosphereState
 - jeod::AtmosphereState, [24](#)
- Avogadro
 - jeod::METAtmosphere, [32](#)
- barometric_equation_ceiling
 - jeod::METAtmosphere, [32](#)
- base_fairing_height
 - jeod::METAtmosphere, [32](#)
- BaseAtmosphere, [15](#)
- class_declarations.hh, [61](#)
- compute_exospheric_temperature
 - jeod::METAtmosphere, [29](#)
- compute_mol_wt
 - jeod::METAtmosphere, [29](#)
- compute_seasonal_lat_variation_He
 - jeod::METAtmosphere, [30](#)
- compute_seasonal_latitude_variation
 - jeod::METAtmosphere, [30](#)
- compute_solar_angles
 - jeod::METAtmosphere, [30](#)
- compute_temperature
 - jeod::METAtmosphereThermal, [48](#)
- data_met_wind_velocity.cc, [62](#)
- day_of_year
 - jeod::METAtmosphere, [32](#)
- days_per_century
 - jeod::METAtmosphere, [32](#)
- days_per_year
 - jeod::METAtmosphere, [32](#)
- deg_to_rad
 - jeod::METAtmosphere, [33](#)
- density
 - jeod::AtmosphereState, [26](#)
- Environment, [12](#)
- exo_temp
 - jeod::METAtmosphereStateVars, [45](#)
- F10
 - jeod::METAtmosphere, [33](#)
- F10B
 - jeod::METAtmosphere, [33](#)
- fairing_k
 - jeod::METAtmosphere, [33](#)

- first_pass
 - jeod::WindVelocity, [53](#)
- frac
 - jeod::METAtmosphereChemical, [40](#)
- fraction_of_year
 - jeod::METAtmosphere, [33](#)
- framework_error
 - jeod::AtmosphereMessages, [22](#)
- framework_warning
 - jeod::AtmosphereMessages, [22](#)
- gauss_altitudes
 - jeod::METAtmosphere, [34](#)
- gauss_n
 - jeod::METAtmosphere, [34](#)
- generate_base_temperature
 - jeod::METAtmosphereThermal, [48](#)
- geo_index
 - jeod::METAtmosphere, [34](#)
- geo_index_type
 - jeod::METAtmosphere, [34](#)
- get_num_layers
 - jeod::WindVelocity, [52](#)
- get_omega_scale_table
 - jeod::WindVelocity, [52](#)
- He
 - jeod::METAtmosphereStateVars, [45](#)
- Hyd
 - jeod::METAtmosphereStateVars, [45](#)
- increasing_altitude
 - jeod::WindVelocity, [53](#)
- init_attrjeod__Atmosphere
 - jeod::Atmosphere, [20](#)
- init_attrjeod__AtmosphereMessages
 - jeod::AtmosphereMessages, [21](#)
- init_attrjeod__AtmosphereState
 - jeod::AtmosphereState, [25](#)
- init_attrjeod__METAtmosphere
 - jeod::METAtmosphere, [31](#)
- init_attrjeod__METAtmosphereChemical
 - jeod::METAtmosphereChemical, [40](#)
- init_attrjeod__METAtmosphereState
 - jeod::METAtmosphereState, [43](#)
- init_attrjeod__METAtmosphereStateVars
 - jeod::METAtmosphereStateVars, [45](#)
- init_attrjeod__METAtmosphereThermal
 - jeod::METAtmosphereThermal, [48](#)
- init_attrjeod__WindVelocity
 - jeod::WindVelocity, [53](#)
- init_attrjeod__WindVelocityBase
 - jeod::WindVelocityBase, [58](#)
- initialization_error
 - jeod::AtmosphereMessages, [22](#)
- initialize
 - jeod::METAtmosphere_solar_max_default_data, [38](#)
- jeod::METAtmosphere_solar_mean_default_data, [38](#)
- jeod::METAtmosphere_solar_min_default_data, [39](#)
- jeod::WindVelocity_wind_velocity_default_data, [55](#)
- InputProcessor
 - jeod::Atmosphere, [20](#)
 - jeod::AtmosphereMessages, [21](#)
 - jeod::AtmosphereState, [25](#)
 - jeod::METAtmosphere, [31](#)
 - jeod::METAtmosphereChemical, [40](#)
 - jeod::METAtmosphereState, [43](#)
 - jeod::METAtmosphereStateVars, [45](#)
 - jeod::METAtmosphereThermal, [48](#)
 - jeod::WindVelocity, [53](#)
 - jeod::WindVelocityBase, [58](#)
- JEOD_FRIEND_CLASS
 - solar_max.cc, [65](#)
 - solar_mean.cc, [66](#)
 - solar_min.cc, [67](#)
- jacchia
 - jeod::METAtmosphere, [30](#)
- jeod, [17](#)
- jeod::METAtmosphere
 - ATMOS_MET_GI_AP, [29](#)
 - ATMOS_MET_GI_KP, [29](#)
- jeod::Atmosphere, [19](#)
 - ~Atmosphere, [20](#)
 - active, [20](#)
 - Atmosphere, [20](#)
 - init_attrjeod__Atmosphere, [20](#)
 - InputProcessor, [20](#)
 - operator=, [20](#)
 - update_atmosphere, [20](#)
- jeod::AtmosphereMessages, [21](#)
 - AtmosphereMessages, [21](#)
 - framework_error, [22](#)
 - framework_warning, [22](#)
 - init_attrjeod__AtmosphereMessages, [21](#)
 - initialization_error, [22](#)
 - InputProcessor, [21](#)
 - numerical_warning, [22](#)
 - operator=, [21](#)
- jeod::AtmosphereState, [23](#)
 - ~AtmosphereState, [24](#)
 - active, [25](#)
 - atmos, [26](#)
 - AtmosphereState, [24](#)
 - density, [26](#)
 - init_attrjeod__AtmosphereState, [25](#)
 - InputProcessor, [25](#)
 - operator=, [24](#)
 - pfix_pos, [26](#)
 - pressure, [26](#)
 - temperature, [26](#)
 - update_state, [24, 25](#)
 - update_wind, [25](#)
 - wind, [26](#)
- jeod::METAtmosphere, [27](#)

- ~METAtmosphere, 29
- altitude_km, 32
- apply_gauss_quadrature, 29
- atmos_MET_FAIR5, 29
- AtmosMETGeoIndexType, 29
- Avogadro, 32
- barometric_equation_ceiling, 32
- base_fairing_height, 32
- compute_exospheric_temperature, 29
- compute_mol_wt, 29
- compute_seasonal_lat_variation_He, 30
- compute_seasonal_latitude_variation, 30
- compute_solar_angles, 30
- day_of_year, 32
- days_per_century, 32
- days_per_year, 32
- deg_to_rad, 33
- F10, 33
- F10B, 33
- fairing_k, 33
- fraction_of_year, 33
- gauss_altitudes, 34
- gauss_n, 34
- geo_index, 34
- geo_index_type, 34
- init_attrjeod__METAtmosphere, 31
- InputProcessor, 31
- jacchia, 30
- latitude, 34
- longitude, 35
- METAtmosphere, 29
- max_days_this_year, 35
- minutes_per_day, 35
- modify_densities, 30
- mol_weight_barometric_ceiling, 35
- mol_wt_coeffs, 35
- num_integ_divisions, 35
- num_mol_wt_coeffs, 35
- operator=, 30
- R_gas_constant, 36
- solar_declination_angle, 36
- solar_hour_angle, 36
- species, 36
- state, 36
- thermal, 36
- three_pi_two, 37
- tjt_year_start, 37
- trunc_julian_time, 37
- two_pi, 37
- update_atmosphere, 30, 31
- update_time, 31
- year, 37
- jeod::METAtmosphere_solar_max_default_data, 37
 - initialize, 38
- jeod::METAtmosphere_solar_mean_default_data, 38
 - initialize, 38
- jeod::METAtmosphere_solar_min_default_data, 38
 - initialize, 39
- jeod::METAtmosphereChemical, 39
 - ~METAtmosphereChemical, 40
 - frac, 40
 - init_attrjeod__METAtmosphereChemical, 40
 - InputProcessor, 40
 - METAtmosphereChemical, 40
 - mol_weight, 40
 - nominal_mol_weight, 40
 - num_density, 40
 - num_species, 41
 - operator=, 40
- jeod::METAtmosphereState, 41
 - ~METAtmosphereState, 42
 - init_attrjeod__METAtmosphereState, 43
 - InputProcessor, 43
 - METAtmosphereState, 42
 - met_atmos, 43
 - operator=, 42
 - update_state, 42
- jeod::METAtmosphereStateVars, 43
 - ~METAtmosphereStateVars, 44
 - A, 45
 - exo_temp, 45
 - He, 45
 - Hyd, 45
 - init_attrjeod__METAtmosphereStateVars, 45
 - InputProcessor, 45
 - log10_dens, 46
 - METAtmosphereStateVars, 44
 - mol_weight, 46
 - N2, 46
 - operator=, 45
 - Ox, 46
 - Ox2, 46
- jeod::METAtmosphereThermal, 46
 - ~METAtmosphereThermal, 47
 - altitude_km, 48
 - compute_temperature, 48
 - generate_base_temperature, 48
 - init_attrjeod__METAtmosphereThermal, 48
 - InputProcessor, 48
 - k_1, 48
 - k_3, 48
 - k_4, 49
 - METAtmosphereThermal, 47
 - operator=, 48
 - T_125, 49
 - T_90, 49
 - T_exosphere, 49
 - T_out, 49
 - update, 48
- jeod::WindVelocity, 50
 - ~WindVelocity, 51
 - active, 53
 - array_index, 53
 - first_pass, 53
 - get_num_layers, 52
 - get_omega_scale_table, 52

- increasing_altitude, [53](#)
- init_attrjeod__WindVelocity, [53](#)
- InputProcessor, [53](#)
- num_layers, [53](#)
- omega, [54](#)
- omega_scale_table, [54](#)
- operator=, [52](#)
- set_omega_scale_table, [52](#)
- update_wind, [52](#)
- WindVelocity, [51](#), [52](#)
- jeod::WindVelocity::OmegaTableEntry, [49](#)
 - altitude, [50](#)
 - scale_factor, [50](#)
- jeod::WindVelocity_wind_velocity_default_data, [54](#)
 - initialize, [55](#)
 - num_layers, [55](#)
 - omega, [55](#)
 - omega_scale_alt, [55](#)
 - omega_scale_fac, [55](#)
- jeod::WindVelocityBase, [56](#)
 - ~WindVelocityBase, [56](#)
 - init_attrjeod__WindVelocityBase, [58](#)
 - InputProcessor, [58](#)
 - operator=, [56](#)
 - update_wind, [56](#)
 - WindVelocityBase, [56](#)
- k_1
 - jeod::METAtmosphereThermal, [48](#)
- k_3
 - jeod::METAtmosphereThermal, [48](#)
- k_4
 - jeod::METAtmosphereThermal, [49](#)
- latitude
 - jeod::METAtmosphere, [34](#)
- log10_dens
 - jeod::METAtmosphereStateVars, [46](#)
- longitude
 - jeod::METAtmosphere, [35](#)
- MET_atmosphere.cc, [62](#)
- MET_atmosphere.hh, [63](#)
- MET_atmosphere_state.cc, [63](#)
- MET_atmosphere_state.hh, [63](#)
- MET_atmosphere_state_vars.cc, [64](#)
- MET_atmosphere_state_vars.hh, [64](#)
- METAtmosphere
 - jeod::METAtmosphere, [29](#)
- METAtmosphereChemical
 - jeod::METAtmosphereChemical, [40](#)
- METAtmosphereState
 - jeod::METAtmosphereState, [42](#)
- METAtmosphereStateVars
 - jeod::METAtmosphereStateVars, [44](#)
- METAtmosphereThermal
 - jeod::METAtmosphereThermal, [47](#)
- max_days_this_year
 - jeod::METAtmosphere, [35](#)
- met_atmos
 - jeod::METAtmosphereState, [43](#)
- met_data_wind_velocity.hh, [65](#)
- minutes_per_day
 - jeod::METAtmosphere, [35](#)
- Models, [11](#)
- modify_densities
 - jeod::METAtmosphere, [30](#)
- mol_weight
 - jeod::METAtmosphereChemical, [40](#)
 - jeod::METAtmosphereStateVars, [46](#)
- mol_weight_barometric_ceiling
 - jeod::METAtmosphere, [35](#)
- mol_wt_coeffs
 - jeod::METAtmosphere, [35](#)
- N2
 - jeod::METAtmosphereStateVars, [46](#)
- nominal_mol_weight
 - jeod::METAtmosphereChemical, [40](#)
- num_density
 - jeod::METAtmosphereChemical, [40](#)
- num_integ_divisions
 - jeod::METAtmosphere, [35](#)
- num_layers
 - jeod::WindVelocity, [53](#)
 - jeod::WindVelocity_wind_velocity_default_data, [55](#)
- num_mol_wt_coeffs
 - jeod::METAtmosphere, [35](#)
- num_species
 - jeod::METAtmosphereChemical, [41](#)
- numerical_warning
 - jeod::AtmosphereMessages, [22](#)
- omega
 - jeod::WindVelocity, [54](#)
 - jeod::WindVelocity_wind_velocity_default_data, [55](#)
- omega_scale_alt
 - jeod::WindVelocity_wind_velocity_default_data, [55](#)
- omega_scale_fac
 - jeod::WindVelocity_wind_velocity_default_data, [55](#)
- omega_scale_table
 - jeod::WindVelocity, [54](#)
- operator=
 - jeod::Atmosphere, [20](#)
 - jeod::AtmosphereMessages, [21](#)
 - jeod::AtmosphereState, [24](#)
 - jeod::METAtmosphere, [30](#)
 - jeod::METAtmosphereChemical, [40](#)
 - jeod::METAtmosphereState, [42](#)
 - jeod::METAtmosphereStateVars, [45](#)
 - jeod::METAtmosphereThermal, [48](#)
 - jeod::WindVelocity, [52](#)
 - jeod::WindVelocityBase, [56](#)
- Ox
 - jeod::METAtmosphereStateVars, [46](#)
- Ox2
 - jeod::METAtmosphereStateVars, [46](#)

PATH
 Atmosphere, 14
 prefix_pos
 jeod::AtmosphereState, 26
 pressure
 jeod::AtmosphereState, 26

 R_gas_constant
 jeod::METAtmosphere, 36

 scale_factor
 jeod::WindVelocity::OmegaTableEntry, 50
 set_omega_scale_table
 jeod::WindVelocity, 52
 solar_declination_angle
 jeod::METAtmosphere, 36
 solar_hour_angle
 jeod::METAtmosphere, 36
 solar_max.cc, 65
 JEOD_FRIEND_CLASS, 65
 solar_max.hh, 66
 solar_mean.cc, 66
 JEOD_FRIEND_CLASS, 66
 solar_mean.hh, 66
 solar_min.cc, 66
 JEOD_FRIEND_CLASS, 67
 solar_min.hh, 67
 species
 jeod::METAtmosphere, 36
 state
 jeod::METAtmosphere, 36

 T_125
 jeod::METAtmosphereThermal, 49
 T_90
 jeod::METAtmosphereThermal, 49
 T_exosphere
 jeod::METAtmosphereThermal, 49
 T_out
 jeod::METAtmosphereThermal, 49
 temperature
 jeod::AtmosphereState, 26
 thermal
 jeod::METAtmosphere, 36
 three_pi_two
 jeod::METAtmosphere, 37
 tij_year_start
 jeod::METAtmosphere, 37
 trunc_julian_time
 jeod::METAtmosphere, 37
 two_pi
 jeod::METAtmosphere, 37

 update
 jeod::METAtmosphereThermal, 48
 update_atmosphere
 jeod::Atmosphere, 20
 jeod::METAtmosphere, 30, 31
 update_state
 jeod::AtmosphereState, 24, 25
 jeod::METAtmosphereState, 42
 update_time
 jeod::METAtmosphere, 31
 update_wind
 jeod::AtmosphereState, 25
 jeod::WindVelocity, 52
 jeod::WindVelocityBase, 56

 wind
 jeod::AtmosphereState, 26
 wind_velocity.cc, 67
 wind_velocity.hh, 68
 wind_velocity_base.cc, 68
 wind_velocity_base.hh, 68
 WindVelocity
 jeod::WindVelocity, 51, 52
 WindVelocity_wind_velocity_default_data
 jeod::WindVelocity_wind_velocity_default_data, 55
 WindVelocityBase
 jeod::WindVelocityBase, 56

 year
 jeod::METAtmosphere, 37