

# AtmosphereModel

5.0

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>11</b>
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

<b>7</b>	<b>Namespace Documentation</b>	<b>17</b>
7.1	jeod Namespace Reference . . . . .	17
7.1.1	Detailed Description . . . . .	17
<b>8</b>	<b>Data Structure Documentation</b>	<b>19</b>
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() [1/2] . . . . .	20
8.1.2.2	~Atmosphere() . . . . .	20
8.1.2.3	Atmosphere() [2/2] . . . . .	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 . . . . .	21
8.1.4.1	init_attrjeod__Atmosphere . . . . .	21
8.1.4.2	InputProcessor . . . . .	21
8.1.5	Field Documentation . . . . .	21
8.1.5.1	active . . . . .	21
8.2	jeod::AtmosphereMessages Class Reference . . . . .	22
8.2.1	Detailed Description . . . . .	22
8.2.2	Constructor & Destructor Documentation . . . . .	22
8.2.2.1	AtmosphereMessages() [1/2] . . . . .	22
8.2.2.2	AtmosphereMessages() [2/2] . . . . .	23
8.2.3	Member Function Documentation . . . . .	23
8.2.3.1	operator=() . . . . .	23
8.2.4	Friends And Related Function Documentation . . . . .	23
8.2.4.1	init_attrjeod__AtmosphereMessages . . . . .	23
8.2.4.2	InputProcessor . . . . .	23
8.2.5	Field Documentation . . . . .	23
8.2.5.1	framework_error . . . . .	23

8.2.5.2	<a href="#">framework_warning</a>	24
8.2.5.3	<a href="#">initialization_error</a>	24
8.2.5.4	<a href="#">numerical_warning</a>	24
8.3	<a href="#">jeod::AtmosphereState Class Reference</a>	25
8.3.1	<a href="#">Detailed Description</a>	26
8.3.2	<a href="#">Constructor &amp; Destructor Documentation</a>	26
8.3.2.1	<a href="#">AtmosphereState() [1/3]</a>	26
8.3.2.2	<a href="#">AtmosphereState() [2/3]</a>	26
8.3.2.3	<a href="#">~AtmosphereState()</a>	26
8.3.2.4	<a href="#">AtmosphereState() [3/3]</a>	26
8.3.3	<a href="#">Member Function Documentation</a>	27
8.3.3.1	<a href="#">operator=()</a>	27
8.3.3.2	<a href="#">update_state() [1/2]</a>	27
8.3.3.3	<a href="#">update_state() [2/2]</a>	28
8.3.3.4	<a href="#">update_wind()</a>	28
8.3.4	<a href="#">Friends And Related Function Documentation</a>	29
8.3.4.1	<a href="#">init_attrjeod__AtmosphereState</a>	29
8.3.4.2	<a href="#">InputProcessor</a>	29
8.3.5	<a href="#">Field Documentation</a>	29
8.3.5.1	<a href="#">active</a>	29
8.3.5.2	<a href="#">atmos</a>	29
8.3.5.3	<a href="#">density</a>	30
8.3.5.4	<a href="#">pfix_pos</a>	30
8.3.5.5	<a href="#">pressure</a>	30
8.3.5.6	<a href="#">temperature</a>	30
8.3.5.7	<a href="#">wind</a>	31
8.4	<a href="#">jeod::METAtmosphere Class Reference</a>	31
8.4.1	<a href="#">Detailed Description</a>	33
8.4.2	<a href="#">Member Enumeration Documentation</a>	33
8.4.2.1	<a href="#">AtmosMETGeoIndexType</a>	33

8.4.3	Constructor & Destructor Documentation . . . . .	33
8.4.3.1	METAtmosphere() [1/2] . . . . .	33
8.4.3.2	~METAtmosphere() . . . . .	33
8.4.3.3	METAtmosphere() [2/2] . . . . .	34
8.4.4	Member Function Documentation . . . . .	34
8.4.4.1	apply_gauss_quadrature() . . . . .	34
8.4.4.2	atmos_MET_FAIR5() . . . . .	34
8.4.4.3	compute_exospheric_temperature() . . . . .	34
8.4.4.4	compute_mol_wt() . . . . .	35
8.4.4.5	compute_seasonal_lat_variation_He() . . . . .	35
8.4.4.6	compute_seasonal_latitude_variation() . . . . .	35
8.4.4.7	compute_solar_angles() . . . . .	35
8.4.4.8	jacchia() . . . . .	36
8.4.4.9	modify_densities() . . . . .	36
8.4.4.10	operator=() . . . . .	36
8.4.4.11	update_atmosphere() [1/3] . . . . .	36
8.4.4.12	update_atmosphere() [2/3] . . . . .	37
8.4.4.13	update_atmosphere() [3/3] . . . . .	37
8.4.4.14	update_time() . . . . .	38
8.4.5	Friends And Related Function Documentation . . . . .	38
8.4.5.1	init_attrjeod__METAtmosphere . . . . .	38
8.4.5.2	InputProcessor . . . . .	38
8.4.6	Field Documentation . . . . .	38
8.4.6.1	altitude_km . . . . .	38
8.4.6.2	Avogadro . . . . .	39
8.4.6.3	barometric_equation_ceiling . . . . .	39
8.4.6.4	base_fairing_height . . . . .	39
8.4.6.5	day_of_year . . . . .	39
8.4.6.6	days_per_century . . . . .	40
8.4.6.7	days_per_year . . . . .	40

8.4.6.8	<a href="#">deg_to_rad</a>	40
8.4.6.9	<a href="#">F10</a>	40
8.4.6.10	<a href="#">F10B</a>	41
8.4.6.11	<a href="#">fairing_k</a>	41
8.4.6.12	<a href="#">fraction_of_year</a>	41
8.4.6.13	<a href="#">gauss_altitudes</a>	41
8.4.6.14	<a href="#">gauss_n</a>	42
8.4.6.15	<a href="#">geo_index</a>	42
8.4.6.16	<a href="#">geo_index_type</a>	42
8.4.6.17	<a href="#">latitude</a>	42
8.4.6.18	<a href="#">longitude</a>	43
8.4.6.19	<a href="#">max_days_this_year</a>	43
8.4.6.20	<a href="#">minutes_per_day</a>	43
8.4.6.21	<a href="#">mol_weight_barometric_ceiling</a>	43
8.4.6.22	<a href="#">mol_wt_coeffs</a>	44
8.4.6.23	<a href="#">num_integ_divisions</a>	44
8.4.6.24	<a href="#">num_mol_wt_coeffs</a>	44
8.4.6.25	<a href="#">R_gas_constant</a>	44
8.4.6.26	<a href="#">solar_declination_angle</a>	45
8.4.6.27	<a href="#">solar_hour_angle</a>	45
8.4.6.28	<a href="#">species</a>	45
8.4.6.29	<a href="#">state</a>	45
8.4.6.30	<a href="#">thermal</a>	46
8.4.6.31	<a href="#">three_pi_two</a>	46
8.4.6.32	<a href="#">tjt_year_start</a>	46
8.4.6.33	<a href="#">trunc_julian_time</a>	46
8.4.6.34	<a href="#">two_pi</a>	47
8.4.6.35	<a href="#">year</a>	47
8.5	<a href="#">jeod::METAtmosphere_solar_max_default_data Class Reference</a>	47
8.5.1	<a href="#">Detailed Description</a>	47

8.5.2	Member Function Documentation	47
8.5.2.1	initialize()	48
8.6	jeod::METAtmosphere_solar_mean_default_data Class Reference	48
8.6.1	Detailed Description	48
8.6.2	Member Function Documentation	48
8.6.2.1	initialize()	48
8.7	jeod::METAtmosphere_solar_min_default_data Class Reference	49
8.7.1	Detailed Description	49
8.7.2	Member Function Documentation	49
8.7.2.1	initialize()	49
8.8	jeod::METAtmosphereChemical Class Reference	49
8.8.1	Detailed Description	50
8.8.2	Constructor & Destructor Documentation	50
8.8.2.1	METAtmosphereChemical() [1/2]	50
8.8.2.2	~METAtmosphereChemical()	50
8.8.2.3	METAtmosphereChemical() [2/2]	51
8.8.3	Member Function Documentation	51
8.8.3.1	operator=()	51
8.8.4	Friends And Related Function Documentation	51
8.8.4.1	init_attrjeod__METAtmosphereChemical	51
8.8.4.2	InputProcessor	51
8.8.5	Field Documentation	51
8.8.5.1	frac	51
8.8.5.2	mol_weight	52
8.8.5.3	nominal_mol_weight	52
8.8.5.4	num_density	52
8.8.5.5	num_species	52
8.9	jeod::METAtmosphereState Class Reference	53
8.9.1	Detailed Description	53
8.9.2	Constructor & Destructor Documentation	54



8.9.2.1	METAtmosphereState() [1/3]	54
8.9.2.2	METAtmosphereState() [2/3]	54
8.9.2.3	~METAtmosphereState()	54
8.9.2.4	METAtmosphereState() [3/3]	54
8.9.3	Member Function Documentation	54
8.9.3.1	operator=()	54
8.9.3.2	update_state() [1/2]	54
8.9.3.3	update_state() [2/2]	55
8.9.4	Friends And Related Function Documentation	55
8.9.4.1	init_attrjeod__METAtmosphereState	55
8.9.4.2	InputProcessor	55
8.9.5	Field Documentation	55
8.9.5.1	met_atmos	56
8.10	jeod::METAtmosphereStateVars Class Reference	56
8.10.1	Detailed Description	57
8.10.2	Constructor & Destructor Documentation	57
8.10.2.1	METAtmosphereStateVars() [1/3]	57
8.10.2.2	METAtmosphereStateVars() [2/3]	57
8.10.2.3	~METAtmosphereStateVars()	57
8.10.2.4	METAtmosphereStateVars() [3/3]	57
8.10.3	Member Function Documentation	58
8.10.3.1	operator=()	58
8.10.4	Friends And Related Function Documentation	58
8.10.4.1	init_attrjeod__METAtmosphereStateVars	58
8.10.4.2	InputProcessor	58
8.10.5	Field Documentation	59
8.10.5.1	A	59
8.10.5.2	exo_temp	59
8.10.5.3	He	59
8.10.5.4	Hyd	59

8.10.5.5	<a href="#">log10_dens</a>	60
8.10.5.6	<a href="#">mol_weight</a>	60
8.10.5.7	<a href="#">N2</a>	60
8.10.5.8	<a href="#">Ox</a>	60
8.10.5.9	<a href="#">Ox2</a>	61
8.11	<a href="#">jeod::METAtmosphereThermal Class Reference</a>	61
8.11.1	<a href="#">Detailed Description</a>	62
8.11.2	<a href="#">Constructor &amp; Destructor Documentation</a>	62
8.11.2.1	<a href="#">METAtmosphereThermal() [1/2]</a>	62
8.11.2.2	<a href="#">~METAtmosphereThermal()</a>	62
8.11.2.3	<a href="#">METAtmosphereThermal() [2/2]</a>	62
8.11.3	<a href="#">Member Function Documentation</a>	62
8.11.3.1	<a href="#">compute_temperature()</a>	63
8.11.3.2	<a href="#">generate_base_temperature()</a>	63
8.11.3.3	<a href="#">operator=()</a>	63
8.11.3.4	<a href="#">update()</a>	63
8.11.4	<a href="#">Friends And Related Function Documentation</a>	63
8.11.4.1	<a href="#">init_attrjeod__METAtmosphereThermal</a>	63
8.11.4.2	<a href="#">InputProcessor</a>	64
8.11.5	<a href="#">Field Documentation</a>	64
8.11.5.1	<a href="#">altitude_km</a>	64
8.11.5.2	<a href="#">k_1</a>	64
8.11.5.3	<a href="#">k_3</a>	64
8.11.5.4	<a href="#">k_4</a>	65
8.11.5.5	<a href="#">T_125</a>	65
8.11.5.6	<a href="#">T_90</a>	65
8.11.5.7	<a href="#">T_exosphere</a>	65
8.11.5.8	<a href="#">T_out</a>	66
8.12	<a href="#">jeod::WindVelocity::OmegaTableEntry Struct Reference</a>	66
8.12.1	<a href="#">Detailed Description</a>	66

8.12.2	Field Documentation	66
8.12.2.1	altitude	66
8.12.2.2	scale_factor	67
8.13	jeod::WindVelocity Class Reference	67
8.13.1	Detailed Description	68
8.13.2	Constructor & Destructor Documentation	68
8.13.2.1	WindVelocity() [1/2]	68
8.13.2.2	~WindVelocity()	69
8.13.2.3	WindVelocity() [2/2]	69
8.13.3	Member Function Documentation	69
8.13.3.1	get_num_layers()	69
8.13.3.2	get_omega_scale_table()	69
8.13.3.3	operator=()	69
8.13.3.4	set_omega_scale_table() [1/2]	70
8.13.3.5	set_omega_scale_table() [2/2]	70
8.13.3.6	update_wind()	70
8.13.4	Friends And Related Function Documentation	71
8.13.4.1	init_attrjeod__WindVelocity	71
8.13.4.2	InputProcessor	71
8.13.5	Field Documentation	71
8.13.5.1	active	71
8.13.5.2	array_index	71
8.13.5.3	first_pass	72
8.13.5.4	increasing_altitude	72
8.13.5.5	num_layers	72
8.13.5.6	omega	72
8.13.5.7	omega_scale_table	73
8.14	jeod::WindVelocity_wind_velocity_default_data Class Reference	73
8.14.1	Detailed Description	73
8.14.2	Constructor & Destructor Documentation	73

8.14.2.1	<a href="#">WindVelocity_wind_velocity_default_data()</a>	74
8.14.3	<a href="#">Member Function Documentation</a>	74
8.14.3.1	<a href="#">initialize()</a> [1/2]	74
8.14.3.2	<a href="#">initialize()</a> [2/2]	74
8.14.4	<a href="#">Field Documentation</a>	74
8.14.4.1	<a href="#">num_layers</a>	74
8.14.4.2	<a href="#">omega</a>	75
8.14.4.3	<a href="#">omega_scale_alt</a>	75
8.14.4.4	<a href="#">omega_scale_fac</a>	75
8.15	<a href="#">jeod::WindVelocityBase Class Reference</a>	75
8.15.1	<a href="#">Detailed Description</a>	76
8.15.2	<a href="#">Constructor &amp; Destructor Documentation</a>	76
8.15.2.1	<a href="#">WindVelocityBase()</a> [1/2]	76
8.15.2.2	<a href="#">~WindVelocityBase()</a>	76
8.15.2.3	<a href="#">WindVelocityBase()</a> [2/2]	76
8.15.3	<a href="#">Member Function Documentation</a>	77
8.15.3.1	<a href="#">operator=()</a>	77
8.15.3.2	<a href="#">update_wind()</a>	77
8.15.4	<a href="#">Friends And Related Function Documentation</a>	77
8.15.4.1	<a href="#">init_attrjeod__WindVelocityBase</a>	77
8.15.4.2	<a href="#">InputProcessor</a>	77

<b>9 File Documentation</b>	<b>79</b>
9.1 atmosphere.hh File Reference	79
9.1.1 Detailed Description	79
9.2 atmosphere_messages.cc File Reference	79
9.2.1 Detailed Description	80
9.3 atmosphere_messages.hh File Reference	80
9.3.1 Detailed Description	80
9.4 atmosphere_state.cc File Reference	80
9.4.1 Detailed Description	81
9.5 atmosphere_state.hh File Reference	81
9.6 class_declarations.hh File Reference	81
9.6.1 Detailed Description	81
9.7 class_declarations.hh File Reference	82
9.7.1 Detailed Description	82
9.8 data_met_wind_velocity.cc File Reference	82
9.8.1 Macro Definition Documentation	82
9.8.1.1 JEOD_FRIEND_CLASS	82
9.9 MET_atmosphere.cc File Reference	83
9.9.1 Detailed Description	83
9.10 MET_atmosphere.hh File Reference	83
9.10.1 Detailed Description	84
9.11 MET_atmosphere_state.cc File Reference	84
9.12 MET_atmosphere_state.hh File Reference	84
9.12.1 Detailed Description	85
9.13 MET_atmosphere_state_vars.cc File Reference	85
9.13.1 Detailed Description	85
9.14 MET_atmosphere_state_vars.hh File Reference	85
9.14.1 Detailed Description	85
9.15 met_data_wind_velocity.hh File Reference	86
9.16 solar_max.cc File Reference	86

9.16.1 Macro Definition Documentation . . . . .	86
9.16.1.1 JEOD_FRIEND_CLASS . . . . .	86
9.17 solar_max.hh File Reference . . . . .	86
9.18 solar_mean.cc File Reference . . . . .	87
9.18.1 Macro Definition Documentation . . . . .	87
9.18.1.1 JEOD_FRIEND_CLASS . . . . .	87
9.19 solar_mean.hh File Reference . . . . .	87
9.20 solar_min.cc File Reference . . . . .	88
9.20.1 Macro Definition Documentation . . . . .	88
9.20.1.1 JEOD_FRIEND_CLASS . . . . .	88
9.21 solar_min.hh File Reference . . . . .	88
9.22 wind_velocity.cc File Reference . . . . .	88
9.22.1 Detailed Description . . . . .	89
9.23 wind_velocity.hh File Reference . . . . .	89
9.23.1 Detailed Description . . . . .	89
9.24 wind_velocity_base.cc File Reference . . . . .	89
9.24.1 Detailed Description . . . . .	90
9.25 wind_velocity_base.hh File Reference . . . . .	90
9.25.1 Detailed Description . . . . .	90
<b>Index</b>	<b>91</b>

# 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:

<a href="#">jeod</a>	Namespace jeod . . . . .	<a href="#">17</a>
----------------------	--------------------------	--------------------



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

jeod::Atmosphere . . . . .	19
jeod::METAtmosphere . . . . .	31
jeod::AtmosphereMessages . . . . .	22
jeod::AtmosphereState . . . . .	25
jeod::METAtmosphereStateVars . . . . .	56
jeod::METAtmosphereState . . . . .	53
jeod::METAtmosphere_solar_max_default_data . . . . .	47
jeod::METAtmosphere_solar_mean_default_data . . . . .	48
jeod::METAtmosphere_solar_min_default_data . . . . .	49
jeod::METAtmosphereChemical . . . . .	49
jeod::METAtmosphereThermal . . . . .	61
jeod::WindVelocity::OmegaTableEntry . . . . .	66
jeod::WindVelocity . . . . .	67
jeod::WindVelocity_wind_velocity_default_data . . . . .	73
jeod::WindVelocityBase . . . . .	75



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::Atmosphere</a>	
A generic base class for atmospheres . . . . .	19
<a href="#">jeod::AtmosphereMessages</a>	
Describes messages used in the <a href="#">Atmosphere</a> model . . . . .	22
<a href="#">jeod::AtmosphereState</a>	
A generic base class for atmosphere state, containing common atmosphere state parameters, i.e . . . . .	25
<a href="#">jeod::METAtmosphere</a>	31
<a href="#">jeod::METAtmosphere_solar_max_default_data</a>	47
<a href="#">jeod::METAtmosphere_solar_mean_default_data</a>	48
<a href="#">jeod::METAtmosphere_solar_min_default_data</a>	49
<a href="#">jeod::METAtmosphereChemical</a>	
The chemical composition of the MET <a href="#">Atmosphere</a> . . . . .	49
<a href="#">jeod::METAtmosphereState</a>	
The MET specific implementation of <a href="#">AtmosphereState</a> . . . . .	53
<a href="#">jeod::METAtmosphereStateVars</a>	
The data variables component of the MET specific implementation of <a href="#">AtmosphereState</a> . . . .	56
<a href="#">jeod::METAtmosphereThermal</a>	
The Thermal aspect of the computation . . . . .	61
<a href="#">jeod::WindVelocity::OmegaTableEntry</a>	
An entry in an omega scale table . . . . .	66
<a href="#">jeod::WindVelocity</a>	
A generic wind velocity implementation . . . . .	67
<a href="#">jeod::WindVelocity_wind_velocity_default_data</a>	73
<a href="#">jeod::WindVelocityBase</a>	
The generic base class for wind velocity classes . . . . .	75



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">atmosphere.hh</a>	General base class for atmosphere models . . . . .	79
<a href="#">atmosphere_messages.cc</a>	Implement atmosphere_messages . . . . .	79
<a href="#">atmosphere_messages.hh</a>	Implement atmosphere_messages . . . . .	80
<a href="#">atmosphere_state.cc</a>	Implementation of the base atmosphere-state model . . . . .	80
<a href="#">atmosphere_state.hh</a>	. . . . .	81
<a href="#">base_atmos/include/class_declarations.hh</a>	Forward declarations of classes defined for JEOD 2.0 Atmosphere . . . . .	81
<a href="#">MET/include/class_declarations.hh</a>	Forward declarations of classes defined for JEOD 2.0 Atmosphere . . . . .	82
<a href="#">data_met_wind_velocity.cc</a>	. . . . .	82
<a href="#">MET_atmosphere.cc</a>	Implementation of MET atmosphere model . . . . .	83
<a href="#">MET_atmosphere.hh</a>	Implement the MET atmosphere using the atmosphere framework . . . . .	83
<a href="#">MET_atmosphere_state.cc</a>	. . . . .	84
<a href="#">MET_atmosphere_state.hh</a>	Implement the MET atmosphere state using the atmosphere framework . . . . .	84
<a href="#">MET_atmosphere_state_vars.cc</a>	Implementation of MET atmosphere model . . . . .	85
<a href="#">MET_atmosphere_state_vars.hh</a>	Implement the MET atmosphere state variables using the atmosphere framework . . . . .	85
<a href="#">met_data_wind_velocity.hh</a>	. . . . .	86
<a href="#">solar_max.cc</a>	. . . . .	86
<a href="#">solar_max.hh</a>	. . . . .	86
<a href="#">solar_mean.cc</a>	. . . . .	87
<a href="#">solar_mean.hh</a>	. . . . .	87
<a href="#">solar_min.cc</a>	. . . . .	88
<a href="#">solar_min.hh</a>	. . . . .	88
<a href="#">wind_velocity.cc</a>	General base class for wind velocity models . . . . .	88
<a href="#">wind_velocity.hh</a>	A wind velocity model based on winds caused by rotation of the planet . . . . .	89

<a href="#">wind_velocity_base.cc</a>	
General base class for wind velocity models . . . . .	<a href="#">89</a>
<a href="#">wind_velocity_base.hh</a>	
General base class for wind velocity models . . . . .	<a href="#">90</a>



## Chapter 6

# Module Documentation

### 6.1 Models

#### Modules

- [Environment](#)

#### 6.1.1 Detailed Description

## 6.2 Environment

### Modules

- [Atmosphere](#)

### 6.2.1 Detailed Description

## 6.3 Atmosphere

### Modules

- [BaseAtmosphere](#)

### Files

- file [atmosphere\\_messages.hh](#)  
*Implement atmosphere\_messages.*
- file [atmosphere.hh](#)  
*General base class for atmosphere models.*
- file [base\\_atmos/include/class\\_declarations.hh](#)  
*Forward declarations of classes defined for JEOD 2.0 Atmosphere.*
- file [wind\\_velocity\\_base.hh](#)  
*General base class for wind velocity models.*
- file [atmosphere\\_messages.cc](#)  
*Implement atmosphere\_messages.*
- file [atmosphere\\_state.cc](#)  
*Implementation of the base atmosphere-state model.*
- file [wind\\_velocity.cc](#)  
*General base class for wind velocity models.*
- file [wind\\_velocity\\_base.cc](#)  
*General base class for wind velocity models.*
- file [MET/include/class\\_declarations.hh](#)  
*Forward declarations of classes defined for JEOD 2.0 Atmosphere.*
- file [MET\\_atmosphere.hh](#)  
*Implement the MET atmosphere using the atmosphere framework.*
- file [MET\\_atmosphere\\_state.hh](#)  
*Implement the MET atmosphere state using the atmosphere framework.*
- file [MET\\_atmosphere\\_state\\_vars.hh](#)  
*Implement the MET atmosphere state variables using the atmosphere framework.*
- file [MET\\_atmosphere.cc](#)  
*Implementation of MET atmosphere model.*
- file [MET\\_atmosphere.cc](#)  
*Implementation of MET atmosphere model.*
- file [MET\\_atmosphere\\_state\\_vars.cc](#)  
*Implementation of MET atmosphere model.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### Macros

- `#define PATH "environment/atmosphere/base_atmos"`
- `#define \_USE\_MATH\_DEFINES`

### 6.3.1 Detailed Description

### 6.3.2 Macro Definition Documentation

#### 6.3.2.1 `_USE_MATH_DEFINES_`

```
#define _USE_MATH_DEFINES_
```

Definition at line 39 of file MET\_atmosphere.cc.

#### 6.3.2.2 `PATH`

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

Definition at line 28 of file atmosphere\_messages.cc.

## 6.4 BaseAtmosphere

### Files

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

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 6.4.1 Detailed Description



## Chapter 7

# Namespace Documentation

### 7.1 jeod Namespace Reference

Namespace jeod.

#### Data Structures

- class [Atmosphere](#)  
*A generic base class for atmospheres.*
- class [AtmosphereMessages](#)  
*Describes messages used in the [Atmosphere](#) model.*
- class [AtmosphereState](#)  
*A generic base class for atmosphere state, containing common atmosphere state parameters, i.e.*
- class [METAtmosphere](#)
- class [METAtmosphere\\_solar\\_max\\_default\\_data](#)
- class [METAtmosphere\\_solar\\_mean\\_default\\_data](#)
- class [METAtmosphere\\_solar\\_min\\_default\\_data](#)
- class [METAtmosphereChemical](#)  
*The chemical composition of the MET [Atmosphere](#).*
- class [METAtmosphereState](#)  
*The MET specific implementation of [AtmosphereState](#).*
- class [METAtmosphereStateVars](#)  
*The data variables component of the MET specific implementation of [AtmosphereState](#).*
- class [METAtmosphereThermal](#)  
*The Thermal aspect of the computation.*
- class [WindVelocity](#)  
*A generic wind velocity implementation.*
- class [WindVelocity\\_wind\\_velocity\\_default\\_data](#)
- class [WindVelocityBase](#)  
*The generic base class for wind velocity classes.*

#### 7.1.1 Detailed Description

Namespace jeod.





## Chapter 8

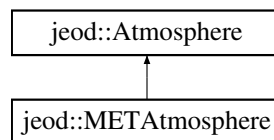
# Data Structure Documentation

### 8.1 jeod::Atmosphere Class Reference

A generic base class for atmospheres.

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

Inheritance diagram for jeod::Atmosphere:



#### Public Member Functions

- [Atmosphere](#) ()
- 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 Atmosphere() [1/2]

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

Definition at line 90 of file atmosphere.hh.

#### 8.1.2.2 ~Atmosphere()

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

Definition at line 95 of file atmosphere.hh.

#### 8.1.2.3 Atmosphere() [2/2]

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

### 8.1.3 Member Function Documentation

#### 8.1.3.1 operator=()

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

#### 8.1.3.2 update\_atmosphere()

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

A pure virtual function for updating the atmosphere, and inserting

## Parameters

in	<i>position</i>	planet fixed position
out	<i>state</i>	The <a href="#">AtmosphereState</a>

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

Referenced by [jeod::AtmosphereState::update\\_state\(\)](#).

## 8.1.4 Friends And Related Function Documentation

### 8.1.4.1 init\_attrjeod\_\_Atmosphere

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

### 8.1.4.2 InputProcessor

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

Definition at line 80 of file atmosphere.hh.

## 8.1.5 Field Documentation

### 8.1.5.1 active

```
bool jeod::Atmosphere::active
```

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 [AtmosphereMessages](#)() [1/2]

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

### 8.2.2.2 AtmosphereMessages() [2/2]

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

## 8.2.3 Member Function Documentation

### 8.2.3.1 operator=()

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

## 8.2.4 Friends And Related Function Documentation

### 8.2.4.1 init\_attrjeod\_\_AtmosphereMessages

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

### 8.2.4.2 InputProcessor

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

Definition at line 77 of file atmosphere\_messages.hh.

## 8.2.5 Field Documentation

### 8.2.5.1 framework\_error

```
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 framework\_warning

```
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 initialization\_error

```
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 numerical\_warning

```
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:

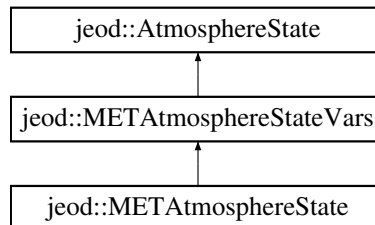
- [atmosphere\\_messages.hh](#)
- [atmosphere\\_messages.cc](#)

## 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 `AtmosphereState()` [1/3]

```
jeod::AtmosphereState::AtmosphereState ( )
```

Definition at line 38 of file `atmosphere_state.cc`.

References `wind`.

#### 8.3.2.2 `AtmosphereState()` [2/3]

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

Definition at line 50 of file `atmosphere_state.cc`.

References `wind`.

#### 8.3.2.3 `~AtmosphereState()`

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

Definition at line 67 of file `atmosphere_state.cc`.

#### 8.3.2.4 `AtmosphereState()` [3/3]

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

Copy Constructor.



## Parameters

in	<i>rhs</i>	The <a href="#">AtmosphereState</a> 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 operator=()

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

[AtmosphereState](#) Operator =.

## Returns

The newly copied [AtmosphereState](#)

## Parameters

in	<i>rhs</i>	The <a href="#">AtmosphereState</a> 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 update\_state() [1/2]

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

Updates the invoking atmosphere state, using the atmosphere model pointed to by [atmos\\_model](#), and calculated at the planet fixed position pointed to by [pfix\\_pos](#).

Note that any type inheriting from [Atmosphere](#) can be sent in for [atmos\\_model](#).

## Parameters

in	<i>atmos_↔ model_</i>	<a href="#">Atmosphere</a> 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 update\_state() [2/2]

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

Updates the invoking atmosphere state, using the atmosphere model pointed to by *atmos*, and calculated at the planet fixed position pointed to by *pfix\_pos*.

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

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

Definition at line 145 of file atmosphere\_state.cc.

References active, *atmos*, *pfix\_pos*, and jeod::Atmosphere::update\_atmosphere().

## 8.3.3.4 update\_wind()

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

Updates the wind portion of the invoking atmosphere state, using the wind model pointed to by *wind\_vel*, calculated at the inertial position given by *inrtl\_pos* and the altitude given.

## Parameters

in	<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 init\_attrjeod\_\_AtmosphereState

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

#### 8.3.4.2 InputProcessor

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

Definition at line 87 of file atmosphere\_state.hh.

### 8.3.5 Field Documentation

#### 8.3.5.1 active

```
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 atmos

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

Definition at line 103 of file atmosphere\_state.hh.

Referenced by AtmosphereState(), and update\_state().

### 8.3.5.3 density

```
double jeod::AtmosphereState::density
```

trick\_units(kg/m3) total density at altitude

Definition at line 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 pfix\_pos

```
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 pressure

```
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 temperature

```
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 wind

```
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().

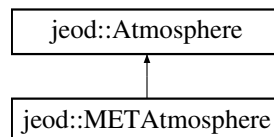
The documentation for this class was generated from the following files:

- [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](#) ()
- virtual [~METAtmosphere](#) ()
- virtual void [update\\_atmosphere](#) (const PlanetFixedPosition \*pfix\_pos, [AtmosphereState](#) \*state)

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

- void [update\\_atmosphere](#) (const PlanetFixedPosition \*pfix\_pos, [METAtmosphereStateVars](#) \*state)

*Front-end to the computation of the [METAtmosphere](#) at the current time Inserts the results into the [METAtmosphereStateVars](#) pointed to by ext\_state.*

- void [update\\_time](#) (const TimeUTC &time\_utc)

## Data Fields

- [AtmosMETGeoIndexType](#) [geo\\_index\\_type](#)
- double [geo\\_index](#)
- double [F10](#)
- double [F10B](#)
- [METAtmosphereChemical](#) [species](#)

## Private Member Functions

- void `update_atmosphere` (const PlanetFixedPosition \*pfix\_pos)  
*Calculates the `METAtmosphere`, at the current time.*
- void `modify_densities` ()
- void `compute_solar_angles` ()
- void `compute_exospheric_temperature` ()
- void `jacchia` ()
- void `compute_seasonal_latitude_variation` ()
- void `compute_seasonal_lat_variation_He` ()
- void `atmos_MET_FAIR5` ()
- double `compute_mol_wt` (double altitude)
- double `apply_gauss_quadrature` (int altitude\_index\_start, double ceiling)
- `METAtmosphere` & `operator=` (const `METAtmosphere` &rhs)
- `METAtmosphere` (const `METAtmosphere` &rhs)

## Private Attributes

- double `altitude_km`
- double `latitude`
- double `longitude`
- double `barometric_equation_ceiling`
- double `trunc_julian_time`
- double `tjt_year_start`
- double `fraction_of_year`
- int `day_of_year`
- int `max_days_this_year`
- int `year`
- double `solar_declination_angle`
- double `solar_hour_angle`
- `METAtmosphereStateVars` state
- `METAtmosphereThermal` thermal
- const double `R_gas_constant`
- const double `days_per_year`
- const double `Avogadro`
- const double `two_pi`
- const double `three_pi_two`
- const double `deg_to_rad`
- const int `days_per_century`
- const int `minutes_per_day`
- const double `mol_weight_barometric_ceiling`
- const double `base_fairing_height`
- const double `fairing_k`

## Static Private Attributes

- static const int `num_mol_wt_coeffs` = 7
- static const double `mol_wt_coeffs` [`num_mol_wt_coeffs`]
- static const int `num_integ_divisions` = 8
- static const double `gauss_altitudes` [`num_integ_divisions`+1]
- static const int `gauss_n` [`num_integ_divisions`] = { 4, 5, 6, 6, 6, 6, 6 }

## 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 AtmosMETGeoIndexType

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 METAtmosphere() [1/2]

[jeod::METAtmosphere::METAtmosphere](#) ( )

Definition at line 120 of file MET\_atmosphere.cc.

#### 8.4.3.2 ~METAtmosphere()

[virtual jeod::METAtmosphere::~~METAtmosphere](#) ( ) [\[inline\]](#), [\[virtual\]](#)

Definition at line 297 of file MET\_atmosphere.hh.

#### 8.4.3.3 METAtmosphere() [2/2]

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

### 8.4.4 Member Function Documentation

#### 8.4.4.1 apply\_gauss\_quadrature()

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

Definition at line 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 atmos\_MET\_FAIR5()

```
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 compute\_exospheric\_temperature()

```
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 compute\_mol\_wt()

```
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 compute\_seasonal\_lat\_variation\_He()

```
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 compute\_seasonal\_latitude\_variation()

```
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 compute\_solar\_angles()

```
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 jacchia()

```
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 modify\_densities()

```
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 operator=()

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

#### 8.4.4.11 update\_atmosphere() [1/3]

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

A pure virtual function for updating the atmosphere, and inserting

##### Parameters

in	<i>position</i>	planet fixed position
out	<i>state</i>	The <a href="#">AtmosphereState</a>

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 update\_atmosphere() [2/3]

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

Front-end to the computation of the [METAtmosphere](#) at the current time Inserts the results into the [METAtmosphereStateVars](#) pointed to by *ext\_state*.

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

##### Parameters

in	<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 update\_atmosphere() [3/3]

```
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 update\_time()

```
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 init\_attrjeod\_\_METAtmosphere

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

#### 8.4.5.2 InputProcessor

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

Definition at line 177 of file MET\_atmosphere.hh.

### 8.4.6 Field Documentation

#### 8.4.6.1 altitude\_km

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

trick\_units(km) Copy of vehicle altitude

Definition at line 202 of file MET\_atmosphere.hh.

Referenced by atmos\_MET\_FAIR5(), compute\_seasonal\_latitude\_variation(), jacchia(), modify\_densities(), and update\_atmosphere().

#### 8.4.6.2 Avogadro

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

trick\_units(–) Avogadros number

Definition at line 241 of file MET\_atmosphere.hh.

Referenced by jacchia().

#### 8.4.6.3 barometric\_equation\_ceiling

```
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 base\_fairing\_height

```
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 day\_of\_year

```
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 days\_per\_century

```
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 days\_per\_year

```
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 deg\_to\_rad

```
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 F10

```
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 F10B

```
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 fairing\_k

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

trick\_units(rad/km) Factor which, when multiplied by the altitude delta above the base-fairing-height provides an angle. The square of the cosine of that angle indicates how much of the seasonal-variation in Helium density to apply.  $\text{density} = \text{corrected-density} * (\text{non-corrected-density} / \text{corrected-density})^{\cos^2(\text{fairing\_k} * \text{delta-altitude})}$  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 fraction\_of\_year

```
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 gauss\_altitudes

```
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 gauss\_n

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

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

Definition at line 285 of file MET\_atmosphere.hh.

Referenced by apply\_gauss\_quadrature().

#### 8.4.6.15 geo\_index

```
double jeod::METAtmosphere::geo_index
```

trick\_units(–) Geomagnetic variations index (Ap or Kp).

Definition at line 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 geo\_index\_type

```
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 latitude

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

trick\_units(rad) Copy of vehicle latitude

Definition at line 203 of file MET\_atmosphere.hh.

Referenced by compute\_exospheric\_temperature(), compute\_seasonal\_lat\_variation\_He(), compute\_seasonal\_latitude\_variation(), and update\_atmosphere().



#### 8.4.6.18 longitude

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

trick\_units(rad) Copy of vehicle longitude

Definition at line 204 of file MET\_atmosphere.hh.

Referenced by compute\_solar\_angles(), and update\_atmosphere().

#### 8.4.6.19 max\_days\_this\_year

```
int jeod::METAtmosphere::max_days_this_year [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 minutes\_per\_day

```
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 mol\_weight\_barometric\_ceiling

```
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 mol\_wt\_coeffs

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

##### Initial value:

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

trick\_units(-) polynomial coefficients for computing the molecular weights in the region where the barometric equation is used.

Definition at line 270 of file MET\_atmosphere.hh.

Referenced by compute\_mol\_wt().

#### 8.4.6.23 num\_integ\_divisions

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

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

Definition at line 275 of file MET\_atmosphere.hh.

#### 8.4.6.24 num\_mol\_wt\_coeffs

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

trick\_units(count) the number of polynomial coefficients.

Definition at line 268 of file MET\_atmosphere.hh.

#### 8.4.6.25 R\_gas\_constant

```
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 solar\_declination\_angle

```
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 solar\_hour\_angle

```
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 species

```
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 state

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

trick\_units(-) A scratch set of state variables, used for populating state variables internally before being copied onto the real state.

Definition at line 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 thermal

```
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 three\_pi\_two

```
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 tjt\_year\_start

```
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 trunc\_julian\_time

```
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 two\_pi

```
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 year

```
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 initialize()

```
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\\_GI\\_AP](#), [jeod::METAtmosphere::F10](#), [jeod::METAtmosphere::F10B](#), [jeod::METAtmosphere::geo\\_index](#), and [jeod::METAtmosphere::geo\\_index\\_type](#).

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

- [solar\\_max.hh](#)
- [solar\\_max.cc](#)

## 8.6 jeod::METAtmosphere\_solar\_mean\_default\_data Class Reference

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

### Public Member Functions

- void [initialize](#) ([METAtmosphere](#) \*)

### 8.6.1 Detailed Description

Definition at line 54 of file solar\_mean.hh.

### 8.6.2 Member Function Documentation

#### 8.6.2.1 initialize()

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

Definition at line 37 of file solar\_mean.cc.

References [jeod::METAtmosphere::ATMOS\\_MET\\_GI\\_AP](#), [jeod::METAtmosphere::F10](#), [jeod::METAtmosphere::F10B](#), [jeod::METAtmosphere::geo\\_index](#), and [jeod::METAtmosphere::geo\\_index\\_type](#).

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

- [solar\\_mean.hh](#)
- [solar\\_mean.cc](#)

## 8.7 jeod::METAtmosphere\_solar\_min\_default\_data Class Reference

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

### Public Member Functions

- void [initialize](#) ([METAtmosphere](#) \*)

### 8.7.1 Detailed Description

Definition at line 54 of file [solar\\_min.hh](#).

### 8.7.2 Member Function Documentation

#### 8.7.2.1 initialize()

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

Definition at line 37 of file [solar\\_min.cc](#).

References [jeod::METAtmosphere::ATMOS\\_MET\\_GI\\_AP](#), [jeod::METAtmosphere::F10](#), [jeod::METAtmosphere::F10B](#), [jeod::METAtmosphere::geo\\_index](#), and [jeod::METAtmosphere::geo\\_index\\_type](#).

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

- [solar\\_min.hh](#)
- [solar\\_min.cc](#)

## 8.8 jeod::METAtmosphereChemical Class Reference

The chemical composition of the MET [Atmosphere](#).

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

### Public Member Functions

- [METAtmosphereChemical](#) ()
- virtual [~METAtmosphereChemical](#) ()

## Data Fields

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

## Static Public Attributes

- static const int `num_species` = 6

## Private Member Functions

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

## Friends

- class `InputProcessor`
- void `init_attrjeod__METAtmosphereChemical` ()

### 8.8.1 Detailed Description

The chemical composition of the MET `Atmosphere`.

Definition at line 86 of file `MET_atmosphere.hh`.

### 8.8.2 Constructor & Destructor Documentation

#### 8.8.2.1 `METAtmosphereChemical()` [1/2]

```
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 `~METAtmosphereChemical()`

```
virtual jeod::METAtmosphereChemical::~~METAtmosphereChemical ( ) [inline], [virtual]
```

Definition at line 106 of file `MET_atmosphere.hh`.



### 8.8.2.3 METAtmosphereChemical() [2/2]

```
jeod::METAtmosphereChemical::METAtmosphereChemical (
    const METAtmosphereChemical & rhs ) [private]
```

## 8.8.3 Member Function Documentation

### 8.8.3.1 operator=()

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

## 8.8.4 Friends And Related Function Documentation

### 8.8.4.1 init\_attrjeod\_\_METAtmosphereChemical

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

### 8.8.4.2 InputProcessor

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

Definition at line 87 of file MET\_atmosphere.hh.

## 8.8.5 Field Documentation

### 8.8.5.1 frac

```
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 mol\_weight

```
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 nominal\_mol\_weight

```
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 num\_density

```
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 num\_species

```
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:

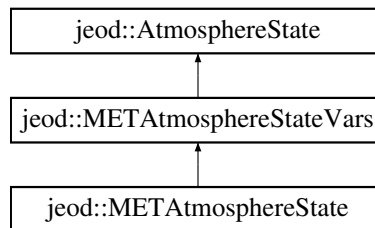
- [MET\\_atmosphere.hh](#)
- [MET\\_atmosphere.cc](#)

## 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)
- virtual [~METAtmosphereState](#) ()
- void [update\\_state](#) ([METAtmosphere](#) \*atmos\_model, const PlanetFixedPosition \*pfix\_pos)  
*Updates the [METAtmosphereState](#) from the [METAtmosphere](#) pointed to by atmos\_model\_.*
- virtual void [update\\_state](#) ()  
*Updates the [METAtmosphereState](#) from the [METAtmosphere](#) pointed to by class member atmos\_model using class member pointer pfix\_pos.*

### Private Member Functions

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

### Private Attributes

- [METAtmosphere](#) \* met\_atmos

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_METAtmosphereState](#) ()

### Additional Inherited Members

#### 8.9.1 Detailed Description

The MET specific implementation of [AtmosphereState](#).

Definition at line 83 of file MET\_atmosphere\_state.hh.

## 8.9.2 Constructor & Destructor Documentation

### 8.9.2.1 METAtmosphereState() [1/3]

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

Definition at line 55 of file MET\_atmosphere\_state.cc.

### 8.9.2.2 METAtmosphereState() [2/3]

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

Definition at line 61 of file MET\_atmosphere\_state.cc.

### 8.9.2.3 ~METAtmosphereState()

```
virtual jeod::METAtmosphereState::~~METAtmosphereState ( ) [inline], [virtual]
```

Definition at line 102 of file MET\_atmosphere\_state.hh.

### 8.9.2.4 METAtmosphereState() [3/3]

```
jeod::METAtmosphereState::METAtmosphereState (
    const METAtmosphereState & rhs ) [private]
```

## 8.9.3 Member Function Documentation

### 8.9.3.1 operator=()

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

### 8.9.3.2 update\_state() [1/2]

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

Updates the [METAtmosphereState](#) from the [METAtmosphere](#) pointed to by `atmos_model_`.

This is a specific function for the case of an [METAtmosphere](#) state updating an [METAtmosphere](#)

## Parameters

in	<i>atmos_↔ model_</i>	<a href="#">METAtmosphere</a> Model.
in	<i>prefix_pos_</i>	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 update\_state() [2/2]

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

Updates the [METAtmosphereState](#) from the [METAtmosphere](#) pointed to by class member `atmos_model` using class member pointer `prefix_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::prefix\\_pos](#), and [jeod::MET↔Atmosphere::update\\_atmosphere\(\)](#).

## 8.9.4 Friends And Related Function Documentation

### 8.9.4.1 init\_attrjeod\_\_METAtmosphereState

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

### 8.9.4.2 InputProcessor

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

Definition at line 86 of file MET\_atmosphere\_state.hh.

## 8.9.5 Field Documentation

### 8.9.5.1 met\_atmos

`METAtmosphere*` jeod::METAtmosphereState::met\_atmos [private]

Definition at line 89 of file MET\_atmosphere\_state.hh.

Referenced by update\_state().

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

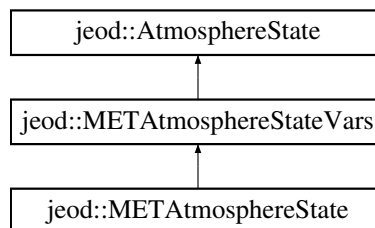
- [MET\\_atmosphere\\_state.hh](#)
- [MET\\_atmosphere\\_state.cc](#)

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

### Data Fields

- double [exo\\_temp](#)
- double [log10\\_dens](#)
- double [mol\\_weight](#)
- double [N2](#)
- double [Ox2](#)
- double [Ox](#)
- double [A](#)
- double [He](#)
- double [Hyd](#)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_METAtmosphereStateVars](#) ()

## Additional Inherited Members

### 8.10.1 Detailed Description

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

Definition at line 82 of file MET\_atmosphere\_state\_vars.hh.

### 8.10.2 Constructor & Destructor Documentation

#### 8.10.2.1 METAtmosphereStateVars() [1/3]

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

Definition at line 48 of file MET\_atmosphere\_state\_vars.cc.

#### 8.10.2.2 METAtmosphereStateVars() [2/3]

```
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 ~METAtmosphereStateVars()

```
jeod::METAtmosphereStateVars::~~METAtmosphereStateVars ( ) [virtual]
```

Definition at line 83 of file MET\_atmosphere\_state\_vars.cc.

#### 8.10.2.4 METAtmosphereStateVars() [3/3]

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

Copy Constructor.

**Parameters**

<i>in</i>	<i>rhs</i>	The <a href="#">METAtmosphereStateVars</a> 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 operator=()**

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

[METAtmosphereStateVars](#) operator =.

**Returns**

The newly copied into [METAtmosphereStateVars](#)

**Parameters**

<i>in</i>	<i>rhs</i>	The <a href="#">METAtmosphereStateVars</a> 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 init\_attrjeod\_\_METAtmosphereStateVars**

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

**8.10.4.2 InputProcessor**

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

Definition at line 84 of file MET\_atmosphere\_state\_vars.hh.



### 8.10.5 Field Documentation

#### 8.10.5.1 A

`double jeod::METAtmosphereStateVars::A`

trick\_units(–) A number density

Definition at line 93 of file MET\_atmosphere\_state\_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update\_atmosphere().

#### 8.10.5.2 exo\_temp

`double jeod::METAtmosphereStateVars::exo_temp`

trick\_units(K) Exospheric temperature

Definition at line 87 of file MET\_atmosphere\_state\_vars.hh.

Referenced by jeod::METAtmosphere::compute\_exospheric\_temperature(), METAtmosphereStateVars(), and operator=().

#### 8.10.5.3 He

`double jeod::METAtmosphereStateVars::He`

trick\_units(–) He number density

Definition at line 94 of file MET\_atmosphere\_state\_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update\_atmosphere().

#### 8.10.5.4 Hyd

`double jeod::METAtmosphereStateVars::Hyd`

trick\_units(–) H number density

Definition at line 95 of file MET\_atmosphere\_state\_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update\_atmosphere().

#### 8.10.5.5 log10\_dens

```
double jeod::METAtmosphereStateVars::log10_dens
```

trick\_units(-) Log10( total density )

Definition at line 88 of file MET\_atmosphere\_state\_vars.hh.

Referenced by METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update\_atmosphere().

#### 8.10.5.6 mol\_weight

```
double jeod::METAtmosphereStateVars::mol_weight
```

trick\_units(-) Average molecular weight

Definition at line 89 of file MET\_atmosphere\_state\_vars.hh.

Referenced by jeod::METAtmosphere::jacchia(), METAtmosphereStateVars(), operator=(), and jeod::METAtmosphere::update\_atmosphere().

#### 8.10.5.7 N2

```
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 Ox

```
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 O<sub>2</sub>

```
double jeod::METAtmosphereStateVars::O2
```

trick\_units(-) O<sub>2</sub> 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 METAtmosphereThermal() [1/2]

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

Definition at line 103 of file MET\_atmosphere.cc.

#### 8.11.2.2 ~METAtmosphereThermal()

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

Definition at line 130 of file MET\_atmosphere.hh.

#### 8.11.2.3 METAtmosphereThermal() [2/2]

```
jeod::METAtmosphereThermal::METAtmosphereThermal (
    const METAtmosphereThermal & rhs ) [private]
```

### 8.11.3 Member Function Documentation

### 8.11.3.1 compute\_temperature()

```
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 generate\_base\_temperature()

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

### 8.11.3.3 operator=()

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

### 8.11.3.4 update()

```
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 init\_attrjeod\_\_METAtmosphereThermal

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

#### 8.11.4.2 InputProcessor

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

Definition at line 122 of file MET\_atmosphere.hh.

### 8.11.5 Field Documentation

#### 8.11.5.1 altitude\_km

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

Definition at line 159 of file MET\_atmosphere.hh.

Referenced by update().

#### 8.11.5.2 k\_1

```
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 k\_3

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

trick\_units(1/m<sup>3</sup>) 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 k\_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 T\_125

```
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 T\_90

```
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 T\_exosphere

```
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 T\_out

```
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 altitude

```
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 scale\_factor

```
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](#), double \*altitude, 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 [WindVelocity\(\)](#) [1/2]

```
jeod::WindVelocity::WindVelocity (
    void )
```

Default Constructor.

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

#### 8.13.2.2 ~WindVelocity()

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

Destructor.

Definition at line 60 of file wind\_velocity.cc.

References `omega_scale_table`.

#### 8.13.2.3 WindVelocity() [2/2]

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

### 8.13.3 Member Function Documentation

#### 8.13.3.1 get\_num\_layers()

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

Definition at line 196 of file wind\_velocity.cc.

References `num_layers`.

#### 8.13.3.2 get\_omega\_scale\_table()

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

Definition at line 232 of file wind\_velocity.cc.

References `omega_scale_table`.

#### 8.13.3.3 operator=()

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

#### 8.13.3.4 `set_omega_scale_table()` [1/2]

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

Definition at line 201 of file `wind_velocity.cc`.

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

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

#### 8.13.3.5 `set_omega_scale_table()` [2/2]

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

Definition at line 212 of file `wind_velocity.cc`.

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

#### 8.13.3.6 `update_wind()`

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

Updates the wind velocity from the parameters given.

##### Parameters

in	<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 init\_attrjeod\_\_WindVelocity

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

### 8.13.4.2 InputProcessor

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

Definition at line 78 of file wind\_velocity.hh.

## 8.13.5 Field Documentation

### 8.13.5.1 active

```
bool jeod::WindVelocity::active
```

trick\_units(-)

Definition at line 100 of file wind\_velocity.hh.

Referenced by update\_wind().

### 8.13.5.2 array\_index

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

last known index into the arrays

Definition at line 142 of file wind\_velocity.hh.

Referenced by update\_wind().

#### 8.13.5.3 first\_pass

```
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 increasing\_altitude

```
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 num\_layers

```
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 omega

```
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 omega\_scale\_table

```
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 WindVelocity\_wind\_velocity\_default\_data()

```
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 initialize() [1/2]

```
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 initialize() [2/2]

```
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 num\_layers

```
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 omega

```
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 omega\_scale\_alt

```
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 omega\_scale\_fac

```
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 [WindVelocityBase\(\)](#) [1/2]

```
jeod::WindVelocityBase::WindVelocityBase (
    void )
```

Default Constructor.

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

#### 8.15.2.2 [~WindVelocityBase\(\)](#)

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

Destructor.

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

#### 8.15.2.3 [WindVelocityBase\(\)](#) [2/2]

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

### 8.15.3 Member Function Documentation

#### 8.15.3.1 operator=()

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

#### 8.15.3.2 update\_wind()

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

Virtual function to define the interface for inheriting functions.

##### Parameters

in	<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 init\_attrjeod\_\_WindVelocityBase

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

#### 8.15.4.2 InputProcessor

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

Definition at line 79 of file wind\_velocity\_base.hh.

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

- [wind\\_velocity\\_base.hh](#)
- [wind\\_velocity\\_base.cc](#)



## Chapter 9

# File Documentation

### 9.1 atmosphere.hh File Reference

General base class for atmosphere models.

```
#include "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.

### 9.2 atmosphere\_messages.cc File Reference

Implement atmosphere\_messages.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

- `#define` [PATH](#) "environment/atmosphere/base\_atmos"

### 9.2.1 Detailed Description

Implement atmosphere\_messages.

## 9.3 atmosphere\_messages.hh File Reference

Implement atmosphere\_messages.

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

## Data Structures

- class [jeod::AtmosphereMessages](#)

*Describes messages used in the [Atmosphere](#) model.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.3.1 Detailed Description

Implement atmosphere\_messages.

## 9.4 atmosphere\_state.cc File Reference

Implementation of the base atmosphere-state model.

```
#include <cstdlib>
#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.

## 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.

## 9.7 class\_declarations.hh File Reference

Forward declarations of classes defined for JEOD 2.0 Atmosphere.

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.7.1 Detailed Description

Forward declarations of classes defined for JEOD 2.0 Atmosphere.

## 9.8 data\_met\_wind\_velocity.cc File Reference

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

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### Macros

- `#define JEOD\_FRIEND\_CLASS WindVelocity_wind_velocity_default_data`

### 9.8.1 Macro Definition Documentation

#### 9.8.1.1 JEOD\_FRIEND\_CLASS

```
#define JEOD_FRIEND_CLASS WindVelocity_wind_velocity_default_data
```

Definition at line 21 of file data\_met\_wind\_velocity.cc.



## 9.9 MET\_atmosphere.cc File Reference

Implementation of MET atmosphere model.

```
#include <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.

## 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.

### 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.

## 9.13 MET\_atmosphere\_state\_vars.cc File Reference

Implementation of MET atmosphere model.

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

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.13.1 Detailed Description

Implementation of MET atmosphere model.

## 9.14 MET\_atmosphere\_state\_vars.hh File Reference

Implement the MET atmosphere state variables using the atmosphere framework.

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

### Data Structures

- class [jeod::METAtmosphereStateVars](#)  
*The data variables component of the MET specific implementation of [AtmosphereState](#).*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.14.1 Detailed Description

Implement the MET atmosphere state variables using the atmosphere framework.

## 9.15 met\_data\_wind\_velocity.hh File Reference

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

### Data Structures

- class [jeod::WindVelocity\\_wind\\_velocity\\_default\\_data](#)

### Namespaces

- [jeod](#)  
*Namespace jeod.*

## 9.16 solar\_max.cc File Reference

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

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### Macros

- `#define` [JEOD\\_FRIEND\\_CLASS](#) METAtmosphere\_solar\_max\_default\_data

### 9.16.1 Macro Definition Documentation

#### 9.16.1.1 JEOD\_FRIEND\_CLASS

```
#define JEOD_FRIEND_CLASS METAtmosphere_solar_max_default_data
```

Definition at line 23 of file solar\_max.cc.

## 9.17 solar\_max.hh File Reference

### Data Structures

- class [jeod::METAtmosphere\\_solar\\_max\\_default\\_data](#)

## Namespaces

- [jeod](#)

*Namespace jeod.*

## 9.18 solar\_mean.cc File Reference

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

- #define [JEOD\\_FRIEND\\_CLASS](#) METAtmosphere\_solar\_mean\_default\_data

### 9.18.1 Macro Definition Documentation

#### 9.18.1.1 JEOD\_FRIEND\_CLASS

```
#define JEOD_FRIEND_CLASS METAtmosphere_solar_mean_default_data
```

Definition at line 23 of file solar\_mean.cc.

## 9.19 solar\_mean.hh File Reference

## Data Structures

- class [jeod::METAtmosphere\\_solar\\_mean\\_default\\_data](#)

## Namespaces

- [jeod](#)

*Namespace jeod.*

## 9.20 solar\_min.cc File Reference

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

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### Macros

- #define [JEOD\\_FRIEND\\_CLASS](#) METAtmosphere\_solar\_min\_default\_data

### 9.20.1 Macro Definition Documentation

#### 9.20.1.1 JEOD\_FRIEND\_CLASS

```
#define JEOD_FRIEND_CLASS METAtmosphere_solar_min_default_data
```

Definition at line 23 of file solar\_min.cc.

## 9.21 solar\_min.hh File Reference

### Data Structures

- class [jeod::METAtmosphere\\_solar\\_min\\_default\\_data](#)

### Namespaces

- [jeod](#)  
*Namespace jeod.*

## 9.22 wind\_velocity.cc File Reference

General base class for wind velocity models.

```
#include <cstdint>  
#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.

## 9.23 wind\_velocity.hh File Reference

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

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

## Data Structures

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

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.23.1 Detailed Description

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

## 9.24 wind\_velocity\_base.cc File Reference

General base class for wind velocity models.

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

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.24.1 Detailed Description

General base class for wind velocity models.

## 9.25 wind\_velocity\_base.hh File Reference

General base class for wind velocity models.

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

### Data Structures

- class [jeod::WindVelocityBase](#)  
*The generic base class for wind velocity classes.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.25.1 Detailed Description

General base class for wind velocity models.



# Index

`_USE_MATH_DEFINES_`  
    Atmosphere, 14  
~Atmosphere  
    jeod::Atmosphere, 20  
~AtmosphereState  
    jeod::AtmosphereState, 26  
~METAtmosphere  
    jeod::METAtmosphere, 33  
~METAtmosphereChemical  
    jeod::METAtmosphereChemical, 50  
~METAtmosphereState  
    jeod::METAtmosphereState, 54  
~METAtmosphereStateVars  
    jeod::METAtmosphereStateVars, 57  
~METAtmosphereThermal  
    jeod::METAtmosphereThermal, 62  
~WindVelocity  
    jeod::WindVelocity, 68  
~WindVelocityBase  
    jeod::WindVelocityBase, 76

A  
    jeod::METAtmosphereStateVars, 59

active  
    jeod::Atmosphere, 21  
    jeod::AtmosphereState, 29  
    jeod::WindVelocity, 71

altitude  
    jeod::WindVelocity::OmegaTableEntry, 66

altitude\_km  
    jeod::METAtmosphere, 38  
    jeod::METAtmosphereThermal, 64

apply\_gauss\_quadrature  
    jeod::METAtmosphere, 34

array\_index  
    jeod::WindVelocity, 71

atmos  
    jeod::AtmosphereState, 29

atmos\_MET\_FAIR5  
    jeod::METAtmosphere, 34

AtmosMETGeoIndexType  
    jeod::METAtmosphere, 33

Atmosphere, 13  
    \_USE\_MATH\_DEFINES\_, 14  
    jeod::Atmosphere, 20  
    PATH, 14

atmosphere.hh, 79

atmosphere\_messages.cc, 79

atmosphere\_messages.hh, 80

atmosphere\_state.cc, 80

atmosphere\_state.hh, 81

AtmosphereMessages  
    jeod::AtmosphereMessages, 22

AtmosphereState  
    jeod::AtmosphereState, 26

Avogadro  
    jeod::METAtmosphere, 38

barometric\_equation\_ceiling  
    jeod::METAtmosphere, 39

base\_fairing\_height  
    jeod::METAtmosphere, 39

BaseAtmosphere, 15

class\_declarations.hh, 81, 82

compute\_exospheric\_temperature  
    jeod::METAtmosphere, 34

compute\_mol\_wt  
    jeod::METAtmosphere, 34

compute\_seasonal\_lat\_variation\_He  
    jeod::METAtmosphere, 35

compute\_seasonal\_latitude\_variation  
    jeod::METAtmosphere, 35

compute\_solar\_angles  
    jeod::METAtmosphere, 35

compute\_temperature  
    jeod::METAtmosphereThermal, 62

data\_met\_wind\_velocity.cc, 82  
    JEOD\_FRIEND\_CLASS, 82

day\_of\_year  
    jeod::METAtmosphere, 39

days\_per\_century  
    jeod::METAtmosphere, 39

days\_per\_year  
    jeod::METAtmosphere, 40

deg\_to\_rad  
    jeod::METAtmosphere, 40

density  
    jeod::AtmosphereState, 29

Environment, 12

exo\_temp  
    jeod::METAtmosphereStateVars, 59

F10  
    jeod::METAtmosphere, 40

F10B  
    jeod::METAtmosphere, 40

fairing\_k  
    jeod::METAtmosphere, 41

- first\_pass
  - jeod::WindVelocity, 71
- frac
  - jeod::METAtmosphereChemical, 51
- fraction\_of\_year
  - jeod::METAtmosphere, 41
- framework\_error
  - jeod::AtmosphereMessages, 23
- framework\_warning
  - jeod::AtmosphereMessages, 23
- gauss\_altitudes
  - jeod::METAtmosphere, 41
- gauss\_n
  - jeod::METAtmosphere, 41
- generate\_base\_temperature
  - jeod::METAtmosphereThermal, 63
- geo\_index
  - jeod::METAtmosphere, 42
- geo\_index\_type
  - jeod::METAtmosphere, 42
- get\_num\_layers
  - jeod::WindVelocity, 69
- get\_omega\_scale\_table
  - jeod::WindVelocity, 69
- He
  - jeod::METAtmosphereStateVars, 59
- Hyd
  - jeod::METAtmosphereStateVars, 59
- increasing\_altitude
  - jeod::WindVelocity, 72
- init\_attrjeod\_\_Atmosphere
  - jeod::Atmosphere, 21
- init\_attrjeod\_\_AtmosphereMessages
  - jeod::AtmosphereMessages, 23
- init\_attrjeod\_\_AtmosphereState
  - jeod::AtmosphereState, 29
- init\_attrjeod\_\_METAtmosphere
  - jeod::METAtmosphere, 38
- init\_attrjeod\_\_METAtmosphereChemical
  - jeod::METAtmosphereChemical, 51
- init\_attrjeod\_\_METAtmosphereState
  - jeod::METAtmosphereState, 55
- init\_attrjeod\_\_METAtmosphereStateVars
  - jeod::METAtmosphereStateVars, 58
- init\_attrjeod\_\_METAtmosphereThermal
  - jeod::METAtmosphereThermal, 63
- init\_attrjeod\_\_WindVelocity
  - jeod::WindVelocity, 71
- init\_attrjeod\_\_WindVelocityBase
  - jeod::WindVelocityBase, 77
- initialization\_error
  - jeod::AtmosphereMessages, 24
- initialize
  - jeod::METAtmosphere\_solar\_max\_default\_data, 47
- jeod::METAtmosphere\_solar\_mean\_default\_data, 48
- jeod::METAtmosphere\_solar\_min\_default\_data, 49
- jeod::WindVelocity\_wind\_velocity\_default\_data, 74
- InputProcessor
  - jeod::Atmosphere, 21
  - jeod::AtmosphereMessages, 23
  - jeod::AtmosphereState, 29
  - jeod::METAtmosphere, 38
  - jeod::METAtmosphereChemical, 51
  - jeod::METAtmosphereState, 55
  - jeod::METAtmosphereStateVars, 58
  - jeod::METAtmosphereThermal, 63
  - jeod::WindVelocity, 71
  - jeod::WindVelocityBase, 77
- JEOD\_FRIEND\_CLASS
  - data\_met\_wind\_velocity.cc, 82
  - solar\_max.cc, 86
  - solar\_mean.cc, 87
  - solar\_min.cc, 88
- jacchia
  - jeod::METAtmosphere, 35
- jeod, 17
- jeod::Atmosphere, 19
  - ~Atmosphere, 20
  - active, 21
  - Atmosphere, 20
  - init\_attrjeod\_\_Atmosphere, 21
  - InputProcessor, 21
  - operator=, 20
  - update\_atmosphere, 20
- jeod::AtmosphereMessages, 22
  - AtmosphereMessages, 22
  - framework\_error, 23
  - framework\_warning, 23
  - init\_attrjeod\_\_AtmosphereMessages, 23
  - initialization\_error, 24
  - InputProcessor, 23
  - numerical\_warning, 24
  - operator=, 23
- jeod::AtmosphereState, 25
  - ~AtmosphereState, 26
  - active, 29
  - atmos, 29
  - AtmosphereState, 26
  - density, 29
  - init\_attrjeod\_\_AtmosphereState, 29
  - InputProcessor, 29
  - operator=, 27
  - pfix\_pos, 30
  - pressure, 30
  - temperature, 30
  - update\_state, 27, 28
  - update\_wind, 28
  - wind, 30
- jeod::METAtmosphere, 31
  - ~METAtmosphere, 33
  - altitude\_km, 38

- apply\_gauss\_quadrature, 34
- atmos\_MET\_FAIR5, 34
- AtmosMETGeoIndexType, 33
- Avogadro, 38
- barometric\_equation\_ceiling, 39
- base\_fairing\_height, 39
- compute\_exospheric\_temperature, 34
- compute\_mol\_wt, 34
- compute\_seasonal\_lat\_variation\_He, 35
- compute\_seasonal\_latitude\_variation, 35
- compute\_solar\_angles, 35
- day\_of\_year, 39
- days\_per\_century, 39
- days\_per\_year, 40
- deg\_to\_rad, 40
- F10, 40
- F10B, 40
- fairing\_k, 41
- fraction\_of\_year, 41
- gauss\_altitudes, 41
- gauss\_n, 41
- geo\_index, 42
- geo\_index\_type, 42
- init\_attrjeod\_\_METAtmosphere, 38
- InputProcessor, 38
- jacchia, 35
- latitude, 42
- longitude, 42
- METAtmosphere, 33
- max\_days\_this\_year, 43
- minutes\_per\_day, 43
- modify\_densities, 36
- mol\_weight\_barometric\_ceiling, 43
- mol\_wt\_coeffs, 43
- num\_integ\_divisions, 44
- num\_mol\_wt\_coeffs, 44
- operator=, 36
- R\_gas\_constant, 44
- solar\_declination\_angle, 44
- solar\_hour\_angle, 45
- species, 45
- state, 45
- thermal, 45
- three\_pi\_two, 46
- tjt\_year\_start, 46
- trunc\_julian\_time, 46
- two\_pi, 46
- update\_atmosphere, 36, 37
- update\_time, 38
- year, 47
- jeod::METAtmosphere\_solar\_max\_default\_data, 47
- initialize, 47
- jeod::METAtmosphere\_solar\_mean\_default\_data, 48
- initialize, 48
- jeod::METAtmosphere\_solar\_min\_default\_data, 49
- initialize, 49
- jeod::METAtmosphereChemical, 49
- ~METAtmosphereChemical, 50
- frac, 51
- init\_attrjeod\_\_METAtmosphereChemical, 51
- InputProcessor, 51
- METAtmosphereChemical, 50
- mol\_weight, 51
- nominal\_mol\_weight, 52
- num\_density, 52
- num\_species, 52
- operator=, 51
- jeod::METAtmosphereState, 53
- ~METAtmosphereState, 54
- init\_attrjeod\_\_METAtmosphereState, 55
- InputProcessor, 55
- METAtmosphereState, 54
- met\_atmos, 55
- operator=, 54
- update\_state, 54, 55
- jeod::METAtmosphereStateVars, 56
- ~METAtmosphereStateVars, 57
- A, 59
- exo\_temp, 59
- He, 59
- Hyd, 59
- init\_attrjeod\_\_METAtmosphereStateVars, 58
- InputProcessor, 58
- log10\_dens, 59
- METAtmosphereStateVars, 57
- mol\_weight, 60
- N2, 60
- operator=, 58
- Ox, 60
- Ox2, 60
- jeod::METAtmosphereThermal, 61
- ~METAtmosphereThermal, 62
- altitude\_km, 64
- compute\_temperature, 62
- generate\_base\_temperature, 63
- init\_attrjeod\_\_METAtmosphereThermal, 63
- InputProcessor, 63
- k\_1, 64
- k\_3, 64
- k\_4, 64
- METAtmosphereThermal, 62
- operator=, 63
- T\_125, 65
- T\_90, 65
- T\_exosphere, 65
- T\_out, 65
- update, 63
- jeod::WindVelocity, 67
- ~WindVelocity, 68
- active, 71
- array\_index, 71
- first\_pass, 71
- get\_num\_layers, 69
- get\_omega\_scale\_table, 69
- increasing\_altitude, 72
- init\_attrjeod\_\_WindVelocity, 71

- InputProcessor, 71
- num\_layers, 72
- omega, 72
- omega\_scale\_table, 72
- operator=, 69
- set\_omega\_scale\_table, 69, 70
- update\_wind, 70
- WindVelocity, 68, 69
- jeod::WindVelocity::OmegaTableEntry, 66
  - altitude, 66
  - scale\_factor, 66
- jeod::WindVelocity\_wind\_velocity\_default\_data, 73
  - initialize, 74
  - num\_layers, 74
  - omega, 74
  - omega\_scale\_alt, 75
  - omega\_scale\_fac, 75
  - WindVelocity\_wind\_velocity\_default\_data, 73
- jeod::WindVelocityBase, 75
  - ~WindVelocityBase, 76
  - init\_attrjeod\_\_WindVelocityBase, 77
  - InputProcessor, 77
  - operator=, 77
  - update\_wind, 77
  - WindVelocityBase, 76
- k\_1
  - jeod::METAtmosphereThermal, 64
- k\_3
  - jeod::METAtmosphereThermal, 64
- k\_4
  - jeod::METAtmosphereThermal, 64
- latitude
  - jeod::METAtmosphere, 42
- log10\_dens
  - jeod::METAtmosphereStateVars, 59
- longitude
  - jeod::METAtmosphere, 42
- MET\_atmosphere.cc, 83
- MET\_atmosphere.hh, 83
- MET\_atmosphere\_state.cc, 84
- MET\_atmosphere\_state.hh, 84
- MET\_atmosphere\_state\_vars.cc, 85
- MET\_atmosphere\_state\_vars.hh, 85
- METAtmosphere
  - jeod::METAtmosphere, 33
- METAtmosphereChemical
  - jeod::METAtmosphereChemical, 50
- METAtmosphereState
  - jeod::METAtmosphereState, 54
- METAtmosphereStateVars
  - jeod::METAtmosphereStateVars, 57
- METAtmosphereThermal
  - jeod::METAtmosphereThermal, 62
- max\_days\_this\_year
  - jeod::METAtmosphere, 43
- met\_atmos
  - jeod::METAtmosphereState, 55
- met\_data\_wind\_velocity.hh, 86
- minutes\_per\_day
  - jeod::METAtmosphere, 43
- Models, 11
- modify\_densities
  - jeod::METAtmosphere, 36
- mol\_weight
  - jeod::METAtmosphereChemical, 51
  - jeod::METAtmosphereStateVars, 60
- mol\_weight\_barometric\_ceiling
  - jeod::METAtmosphere, 43
- mol\_wt\_coeffs
  - jeod::METAtmosphere, 43
- N2
  - jeod::METAtmosphereStateVars, 60
- nominal\_mol\_weight
  - jeod::METAtmosphereChemical, 52
- num\_density
  - jeod::METAtmosphereChemical, 52
- num\_integ\_divisions
  - jeod::METAtmosphere, 44
- num\_layers
  - jeod::WindVelocity, 72
  - jeod::WindVelocity\_wind\_velocity\_default\_data, 74
- num\_mol\_wt\_coeffs
  - jeod::METAtmosphere, 44
- num\_species
  - jeod::METAtmosphereChemical, 52
- numerical\_warning
  - jeod::AtmosphereMessages, 24
- omega
  - jeod::WindVelocity, 72
  - jeod::WindVelocity\_wind\_velocity\_default\_data, 74
- omega\_scale\_alt
  - jeod::WindVelocity\_wind\_velocity\_default\_data, 75
- omega\_scale\_fac
  - jeod::WindVelocity\_wind\_velocity\_default\_data, 75
- omega\_scale\_table
  - jeod::WindVelocity, 72
- operator=
  - jeod::Atmosphere, 20
  - jeod::AtmosphereMessages, 23
  - jeod::AtmosphereState, 27
  - jeod::METAtmosphere, 36
  - jeod::METAtmosphereChemical, 51
  - jeod::METAtmosphereState, 54
  - jeod::METAtmosphereStateVars, 58
  - jeod::METAtmosphereThermal, 63
  - jeod::WindVelocity, 69
  - jeod::WindVelocityBase, 77
- Ox
  - jeod::METAtmosphereStateVars, 60
- Ox2
  - jeod::METAtmosphereStateVars, 60
- PATH

- Atmosphere, [14](#)
- pfix\_pos
  - jeod::AtmosphereState, [30](#)
- pressure
  - jeod::AtmosphereState, [30](#)
- R\_gas\_constant
  - jeod::METAtmosphere, [44](#)
- scale\_factor
  - jeod::WindVelocity::OmegaTableEntry, [66](#)
- set\_omega\_scale\_table
  - jeod::WindVelocity, [69](#), [70](#)
- solar\_declination\_angle
  - jeod::METAtmosphere, [44](#)
- solar\_hour\_angle
  - jeod::METAtmosphere, [45](#)
- solar\_max.cc, [86](#)
  - JEOD\_FRIEND\_CLASS, [86](#)
- solar\_max.hh, [86](#)
- solar\_mean.cc, [87](#)
  - JEOD\_FRIEND\_CLASS, [87](#)
- solar\_mean.hh, [87](#)
- solar\_min.cc, [88](#)
  - JEOD\_FRIEND\_CLASS, [88](#)
- solar\_min.hh, [88](#)
- species
  - jeod::METAtmosphere, [45](#)
- state
  - jeod::METAtmosphere, [45](#)
- T\_125
  - jeod::METAtmosphereThermal, [65](#)
- T\_90
  - jeod::METAtmosphereThermal, [65](#)
- T\_exosphere
  - jeod::METAtmosphereThermal, [65](#)
- T\_out
  - jeod::METAtmosphereThermal, [65](#)
- temperature
  - jeod::AtmosphereState, [30](#)
- thermal
  - jeod::METAtmosphere, [45](#)
- three\_pi\_two
  - jeod::METAtmosphere, [46](#)
- tjt\_year\_start
  - jeod::METAtmosphere, [46](#)
- trunc\_julian\_time
  - jeod::METAtmosphere, [46](#)
- two\_pi
  - jeod::METAtmosphere, [46](#)
- update
  - jeod::METAtmosphereThermal, [63](#)
- update\_atmosphere
  - jeod::Atmosphere, [20](#)
  - jeod::METAtmosphere, [36](#), [37](#)
- update\_state
  - jeod::AtmosphereState, [27](#), [28](#)
- jeod::METAtmosphereState, [54](#), [55](#)
- update\_time
  - jeod::METAtmosphere, [38](#)
- update\_wind
  - jeod::AtmosphereState, [28](#)
  - jeod::WindVelocity, [70](#)
  - jeod::WindVelocityBase, [77](#)
- wind
  - jeod::AtmosphereState, [30](#)
- wind\_velocity.cc, [88](#)
- wind\_velocity.hh, [89](#)
- wind\_velocity\_base.cc, [89](#)
- wind\_velocity\_base.hh, [90](#)
- WindVelocity
  - jeod::WindVelocity, [68](#), [69](#)
- WindVelocity\_wind\_velocity\_default\_data
  - jeod::WindVelocity\_wind\_velocity\_default\_data, [73](#)
- WindVelocityBase
  - jeod::WindVelocityBase, [76](#)
- year
  - jeod::METAtmosphere, [47](#)