

AerodynamicsModel

5.0

Generated by Doxygen 1.8.5

Wed Jun 1 2022 12:05:34

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	9
5.1	File List	9
6	Module Documentation	11
6.1	Models	11
6.1.1	Detailed Description	11
6.2	Interactions	12
6.2.1	Detailed Description	12
6.3	Aerodynamics	13
6.3.1	Detailed Description	14
6.3.2	Macro Definition Documentation	14
6.3.2.1	PATH	14
7	Namespace Documentation	15
7.1	jeod Namespace Reference	15
7.1.1	Detailed Description	15
8	Data Structure Documentation	17
8.1	jeod::AeroDragEnum Class Reference	17
8.1.1	Detailed Description	17
8.1.2	Member Enumeration Documentation	17
8.1.2.1	CoefCalcMethod	17

8.2	jeod::AeroDragParameters Class Reference	18
8.2.1	Detailed Description	18
8.2.2	Field Documentation	18
8.2.2.1	dynamic_pressure	18
8.2.2.2	gas_const	18
8.2.2.3	temp_free_stream	18
8.3	jeod::AerodynamicDrag Class Reference	19
8.3.1	Detailed Description	20
8.3.2	Constructor & Destructor Documentation	20
8.3.2.1	AerodynamicDrag	20
8.3.2.2	~AerodynamicDrag	20
8.3.2.3	AerodynamicDrag	20
8.3.3	Member Function Documentation	20
8.3.3.1	aero_drag	20
8.3.3.2	clear_aero_surface	20
8.3.3.3	operator=	21
8.3.3.4	set_aero_surface	21
8.3.4	Field Documentation	22
8.3.4.1	active	22
8.3.4.2	aero_force	22
8.3.4.3	aero_surface_ptr	22
8.3.4.4	aero_torque	22
8.3.4.5	ballistic_drag	22
8.3.4.6	constant_density	22
8.3.4.7	default_behavior	23
8.3.4.8	density	23
8.3.4.9	param	23
8.3.4.10	use_default_behavior	23
8.4	jeod::AerodynamicDrag_aero_model_default_data Class Reference	23
8.4.1	Detailed Description	24
8.4.2	Member Function Documentation	24
8.4.2.1	initialize	24
8.5	jeod::AerodynamicsMessages Class Reference	24
8.5.1	Detailed Description	24
8.5.2	Constructor & Destructor Documentation	25
8.5.2.1	AerodynamicsMessages	25
8.5.2.2	AerodynamicsMessages	25
8.5.3	Member Function Documentation	25
8.5.3.1	operator=	25
8.5.4	Friends And Related Function Documentation	25

8.5.4.1	init_attrjeod__AerodynamicsMessages	25
8.5.4.2	InputProcessor	25
8.5.5	Field Documentation	25
8.5.5.1	initialization_error	25
8.5.5.2	pre_initialization_error	25
8.5.5.3	runtime_error	25
8.5.5.4	runtime_warns	26
8.6	jeod::AeroFacet Class Reference	26
8.6.1	Detailed Description	27
8.6.2	Constructor & Destructor Documentation	27
8.6.2.1	AeroFacet	27
8.6.2.2	~AeroFacet	27
8.6.2.3	AeroFacet	27
8.6.3	Member Function Documentation	27
8.6.3.1	aerodrag_force	27
8.6.3.2	operator=	27
8.6.4	Friends And Related Function Documentation	27
8.6.4.1	init_attrjeod__AeroFacet	27
8.6.4.2	InputProcessor	28
8.7	jeod::AeroParams Class Reference	28
8.7.1	Detailed Description	28
8.7.2	Constructor & Destructor Documentation	28
8.7.2.1	AeroParams	28
8.7.2.2	~AeroParams	29
8.7.2.3	AeroParams	29
8.7.3	Member Function Documentation	29
8.7.3.1	operator=	29
8.7.4	Friends And Related Function Documentation	29
8.7.4.1	init_attrjeod__AeroParams	29
8.7.4.2	InputProcessor	29
8.8	jeod::AeroSurface Class Reference	29
8.8.1	Detailed Description	30
8.8.2	Constructor & Destructor Documentation	30
8.8.2.1	AeroSurface	30
8.8.2.2	~AeroSurface	30
8.8.2.3	AeroSurface	30
8.8.3	Member Function Documentation	30
8.8.3.1	allocate_array	30
8.8.3.2	allocate_interaction_facet	31
8.8.3.3	operator=	31

8.8.4	Friends And Related Function Documentation	31
8.8.4.1	init_attrjeod__AeroSurface	31
8.8.4.2	InputProcessor	31
8.8.5	Field Documentation	31
8.8.5.1	aero_facets	31
8.8.5.2	facets_size	31
8.9	jeod::AeroSurfaceFactory Class Reference	32
8.9.1	Detailed Description	32
8.9.2	Constructor & Destructor Documentation	33
8.9.2.1	AeroSurfaceFactory	33
8.9.2.2	~AeroSurfaceFactory	33
8.9.2.3	AeroSurfaceFactory	33
8.9.3	Member Function Documentation	33
8.9.3.1	add_facet_params	33
8.9.3.2	operator=	33
8.9.4	Friends And Related Function Documentation	33
8.9.4.1	init_attrjeod__AeroSurfaceFactory	33
8.9.4.2	InputProcessor	33
8.9.5	Field Documentation	33
8.9.5.1	flat_plate_aero_factory	33
8.9.5.2	flat_plate_thermal_aero_factory	34
8.10	jeod::DefaultAero Class Reference	34
8.10.1	Detailed Description	35
8.10.2	Member Enumeration Documentation	35
8.10.2.1	DragOption	35
8.10.3	Constructor & Destructor Documentation	35
8.10.3.1	DefaultAero	35
8.10.3.2	~DefaultAero	35
8.10.3.3	DefaultAero	35
8.10.4	Member Function Documentation	35
8.10.4.1	aerodrag_force	35
8.10.4.2	operator=	36
8.10.5	Friends And Related Function Documentation	36
8.10.5.1	init_attrjeod__DefaultAero	36
8.10.5.2	InputProcessor	36
8.10.6	Field Documentation	36
8.10.6.1	area	36
8.10.6.2	BC	36
8.10.6.3	Cd	36
8.10.6.4	drag	37

8.10.6.5 option	37
8.11 jeod::FlatPlateAeroFacet Class Reference	37
8.11.1 Detailed Description	38
8.11.2 Constructor & Destructor Documentation	38
8.11.2.1 FlatPlateAeroFacet	38
8.11.2.2 ~FlatPlateAeroFacet	39
8.11.2.3 FlatPlateAeroFacet	39
8.11.3 Member Function Documentation	39
8.11.3.1 aerodrag_force	39
8.11.3.2 operator=	39
8.11.4 Friends And Related Function Documentation	39
8.11.4.1 init_attrjeod__FlatPlateAeroFacet	39
8.11.4.2 InputProcessor	39
8.11.5 Field Documentation	39
8.11.5.1 area	39
8.11.5.2 calculate_drag_coef	40
8.11.5.3 center_pressure	40
8.11.5.4 coef_method	40
8.11.5.5 drag_coef_diff	40
8.11.5.6 drag_coef_norm	40
8.11.5.7 drag_coef_spec	40
8.11.5.8 drag_coef_tang	40
8.11.5.9 epsilon	41
8.11.5.10 force_n	41
8.11.5.11 force_t	41
8.11.5.12 normal	41
8.11.5.13 temp_reflect	41
8.11.5.14 temperature	41
8.12 jeod::FlatPlateAeroFactory Class Reference	42
8.12.1 Detailed Description	42
8.12.2 Constructor & Destructor Documentation	43
8.12.2.1 FlatPlateAeroFactory	43
8.12.2.2 ~FlatPlateAeroFactory	43
8.12.2.3 FlatPlateAeroFactory	43
8.12.3 Member Function Documentation	43
8.12.3.1 create_facet	43
8.12.3.2 is_correct_factory	43
8.12.3.3 operator=	44
8.12.4 Friends And Related Function Documentation	44
8.12.4.1 init_attrjeod__FlatPlateAeroFactory	44

8.12.4.2	InputProcessor	44
8.13	jeod::FlatPlateAeroParams Class Reference	44
8.13.1	Detailed Description	45
8.13.2	Constructor & Destructor Documentation	45
8.13.2.1	FlatPlateAeroParams	45
8.13.2.2	~FlatPlateAeroParams	45
8.13.2.3	FlatPlateAeroParams	45
8.13.3	Member Function Documentation	45
8.13.3.1	operator=	45
8.13.4	Friends And Related Function Documentation	45
8.13.4.1	init_attrjeod__FlatPlateAeroParams	45
8.13.4.2	InputProcessor	46
8.13.5	Field Documentation	46
8.13.5.1	calculate_drag_coef	46
8.13.5.2	coef_method	46
8.13.5.3	drag_coef_diff	46
8.13.5.4	drag_coef_norm	46
8.13.5.5	drag_coef_spec	46
8.13.5.6	drag_coef_tang	46
8.13.5.7	epsilon	47
8.13.5.8	temp_reflect	47
8.14	jeod::FlatPlateThermalAeroFactory Class Reference	47
8.14.1	Detailed Description	48
8.14.2	Constructor & Destructor Documentation	48
8.14.2.1	FlatPlateThermalAeroFactory	48
8.14.2.2	~FlatPlateThermalAeroFactory	48
8.14.2.3	FlatPlateThermalAeroFactory	48
8.14.3	Member Function Documentation	48
8.14.3.1	is_correct_factory	48
8.14.3.2	operator=	48
8.14.4	Friends And Related Function Documentation	48
8.14.4.1	init_attrjeod__FlatPlateThermalAeroFactory	48
8.14.4.2	InputProcessor	48
9	File Documentation	51
9.1	aero_drag.cc File Reference	51
9.1.1	Detailed Description	51
9.2	aero_drag.hh File Reference	51
9.2.1	Detailed Description	52
9.3	aero_facet.cc File Reference	52

9.3.1 Detailed Description	52
9.4 aero_facet.hh File Reference	52
9.4.1 Detailed Description	53
9.5 aero_model.cc File Reference	53
9.5.1 Macro Definition Documentation	53
9.5.1.1 JEOD_FRIEND_CLASS	53
9.6 aero_model.hh File Reference	53
9.7 aero_params.cc File Reference	53
9.7.1 Detailed Description	54
9.8 aero_params.hh File Reference	54
9.8.1 Detailed Description	54
9.9 aero_surface.cc File Reference	54
9.9.1 Detailed Description	55
9.10 aero_surface.hh File Reference	55
9.10.1 Detailed Description	55
9.11 aero_surface_factory.cc File Reference	55
9.11.1 Detailed Description	56
9.12 aero_surface_factory.hh File Reference	56
9.12.1 Detailed Description	56
9.13 aerodynamics_messages.cc File Reference	56
9.13.1 Detailed Description	56
9.14 aerodynamics_messages.hh File Reference	57
9.14.1 Detailed Description	57
9.15 class_declarations.hh File Reference	57
9.15.1 Detailed Description	57
9.16 default_aero.cc File Reference	57
9.16.1 Detailed Description	58
9.17 default_aero.hh File Reference	58
9.17.1 Detailed Description	58
9.18 flat_plate_aero_facet.cc File Reference	58
9.18.1 Detailed Description	59
9.19 flat_plate_aero_facet.hh File Reference	59
9.19.1 Detailed Description	59
9.20 flat_plate_aero_factory.cc File Reference	59
9.20.1 Detailed Description	60
9.21 flat_plate_aero_factory.hh File Reference	60
9.21.1 Detailed Description	60
9.22 flat_plate_aero_params.cc File Reference	60
9.22.1 Detailed Description	60
9.23 flat_plate_aero_params.hh File Reference	61

9.23.1 Detailed Description	61
9.24 flat_plate_thermal_aero_factory.cc File Reference	61
9.24.1 Detailed Description	61
9.25 flat_plate_thermal_aero_factory.hh File Reference	62
9.25.1 Detailed Description	62
 Index	 63

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models	11
Interactions	12
Aerodynamics	13

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod	Namespace jeod	15
----------------------	--------------------------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

jeod::AeroDragEnum	17
jeod::AeroDragParameters	18
jeod::AerodynamicDrag	19
jeod::AerodynamicDrag_aero_model_default_data	23
jeod::AerodynamicsMessages	24
jeod::DefaultAero	34
FacetParams	
jeod::AeroParams	28
jeod::FlatPlateAeroParams	44
InteractionFacet	
jeod::AeroFacet	26
jeod::FlatPlateAeroFacet	37
InteractionFacetFactory	
jeod::FlatPlateAeroFactory	42
jeod::FlatPlateThermalAeroFactory	47
InteractionSurface	
jeod::AeroSurface	29
InteractionSurfaceFactory	
jeod::AeroSurfaceFactory	32

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

jeod::AeroDragEnum	Contains enumerations associated with aerodynamic drag	17
jeod::AeroDragParameters	Contains parameters associated with aerodynamic drag	18
jeod::AerodynamicDrag	The main class for calculating aerodynamic drag	19
jeod::AerodynamicDrag_aero_model_default_data	23
jeod::AerodynamicsMessages	Messages associated with use of the aerodynamics model	24
jeod::AeroFacet	An aerodynamic interaction specific facet for use in the surface model	26
jeod::AeroParams	A base class for all aerodynamic parameters used in the surface model	28
jeod::AeroSurface	The aerodynamic specific interaction surface, for use with the surface model	29
jeod::AeroSurfaceFactory	The surface factory that creates an aerodynamic specific surface from a general surface	32
jeod::DefaultAero	The simple, default, aerodynamic drag model, including coefficient of drag, ballistic coefficient, etc	34
jeod::FlatPlateAeroFacet	The aerodynamic specific version of a flat plate	37
jeod::FlatPlateAeroFactory	Creates a FlatPlateAeroFacet from a FlatPlate	42
jeod::FlatPlateAeroParams	Used in the FlatPlateAeroFactory to create a FlatPlateAeroFacet	44
jeod::FlatPlateThermalAeroFactory	Creates a FlatPlateAeroFacet from a FlatPlate	47

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

aero_drag.cc	Orbital aerodynamic force and torque computation, and related classes	51
aero_drag.hh	Orbital aerodynamics paramter definitions and the main class for calculating aerodynamic drag	51
aero_facet.cc	Individual facets for use with aero environment interaction models	52
aero_facet.hh	Individual facets for use with aero environment interaction models	52
aero_model.cc	53
aero_model.hh	53
aero_params.cc	A virtual base class for aero facet parameters, used to create interaction facets for aero in the InteractionSurfaceFactorys	53
aero_params.hh	A virtual base class for aero facet parameters, used to create interaction facets for aero in the InteractionSurfaceFactorys	54
aero_surface.cc	Vehicle surface model for the aerodyanmic interaction models	54
aero_surface.hh	Vehicle surface model for aerodynamics	55
aero_surface_factory.cc	Factory that creates an aerodynamics surface, from a surface model	55
aero_surface_factory.hh	Factory that creates an aerodynamic interaction surface from a surface model	56
aerodynamics_messages.cc	Implement aerodynamics_messages	56
aerodynamics_messages.hh	Aerodynamics message for message handling	57
class_declarations.hh	Forward declaration of classes defined in the aerodynamics package	57
default_aero.cc	Implement a virtual base class for aerodynamic drag calculations	57
default_aero.hh	An implementation of ballistic coefficent and coefficient of drag for use in the AerodynamicDrag object	58
flat_plate_aero_facet.cc	Individual facets for use with aero environment interaction models	58

flat_plate_aero_facet.hh	
The aerodynamic specific implementation of flat plate	59
flat_plate_aero_factory.cc	
Factory that creates a FlatPlateAeroFacet from a FlatPlate, using a FlatPlateAeroParams object	59
flat_plate_aero_factory.hh	
Creates a flat plate aero facet from a basic flat plate facet	60
flat_plate_aero_params.cc	
Flat plates aerodynamic parameters for use in the surface model	60
flat_plate_aero_params.hh	
The set of parameters used to create an aerodynamic interaction specific flat plate facet from a general flat plate facet	61
flat_plate_thermal_aero_factory.cc	
Factory that creates a FlatPlateAeroFacet from a FlatPlateThermal, using a FlatPlateAeroParams object	61
flat_plate_thermal_aero_factory.hh	
Creates a flat plate aero facet from a flat plate thermal facet	62

Chapter 6

Module Documentation

6.1 Models

Modules

- [Interactions](#)

6.1.1 Detailed Description

6.2 Interactions

Modules

- [Aerodynamics](#)

6.2.1 Detailed Description

6.3 Aerodynamics

Files

- file [aero_drag.hh](#)
Orbital aerodynamics parameter definitions and the main class for calculating aerodynamic drag.
- file [aero_facet.hh](#)
Individual facets for use with aero environment interaction models.
- file [aero_params.hh](#)
A virtual base class for aero facet parameters, used to create interaction facets for aero in the InteractionSurface-Factorys.
- file [aero_surface.hh](#)
Vehicle surface model for aerodynamics.
- file [aero_surface_factory.hh](#)
Factory that creates an aerodynamic interaction surface from a surface model.
- file [aerodynamics_messages.hh](#)
Aerodynamics message for message handling.
- file [class_declarations.hh](#)
Forward declaration of classes defined in the aerodynamics package.
- file [default_aero.hh](#)
An implementation of ballistic coefficient and coefficient of drag for use in the AerodynamicDrag object.
- file [flat_plate_aero_facet.hh](#)
The aerodynamic specific implementation of flat plate.
- file [flat_plate_aero_factory.hh](#)
Creates a flat plate aero facet from a basic flat plate facet.
- file [flat_plate_aero_params.hh](#)
The set of parameters used to create an aerodynamic interaction specific flat plate facet from a general flat plate facet.
- file [flat_plate_thermal_aero_factory.hh](#)
Creates a flat plate aero facet from a flat plate thermal facet.
- file [aero_drag.cc](#)
Orbital aerodynamic force and torque computation, and related classes.
- file [aero_facet.cc](#)
Individual facets for use with aero environment interaction models.
- file [aero_params.cc](#)
A virtual base class for aero facet parameters, used to create interaction facets for aero in the InteractionSurface-Factorys.
- file [aero_surface.cc](#)
Vehicle surface model for the aerodynamic interaction models.
- file [aero_surface_factory.cc](#)
Factory that creates an aerodynamics surface, from a surface model.
- file [aerodynamics_messages.cc](#)
Implement aerodynamics_messages.
- file [default_aero.cc](#)
Implement a virtual base class for aerodynamic drag calculations.
- file [flat_plate_aero_facet.cc](#)
Individual facets for use with aero environment interaction models.
- file [flat_plate_aero_factory.cc](#)
Factory that creates a FlatPlateAeroFacet from a FlatPlate, using a FlatPlateAeroParams object.
- file [flat_plate_aero_params.cc](#)
Flat plates aerodynamic parameters for use in the surface model.
- file [flat_plate_thermal_aero_factory.cc](#)
Factory that creates a FlatPlateAeroFacet from a FlatPlateThermal, using a FlatPlateAeroParams object.

Namespaces

- [jeod](#)

Namespace jeod.

Macros

- `#define PATH "interactions/aerodynamics/"`

6.3.1 Detailed Description

6.3.2 Macro Definition Documentation

6.3.2.1 `#define PATH "interactions/aerodynamics/"`

Definition at line 39 of file aerodynamics_messages.cc.

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class [AerodynamicDrag_aero_model_default_data](#)
- class [AeroDragEnum](#)
Contains enumerations associated with aerodynamic drag.
- class [AeroDragParameters](#)
Contains parameters associated with aerodynamic drag.
- class [AerodynamicDrag](#)
The main class for calculating aerodynamic drag.
- class [AeroFacet](#)
An aerodynamic interaction specific facet for use in the surface model.
- class [AeroParams](#)
A base class for all aerodynamic parameters used in the surface model.
- class [AeroSurface](#)
The aerodynamic specific interaction surface, for use with the surface model.
- class [AeroSurfaceFactory](#)
The surface factory that creates an aerodynamic specific surface from a general surface.
- class [AerodynamicsMessages](#)
Messages associated with use of the aerodynamics model.
- class [DefaultAero](#)
The simple, default, aerodynamic drag model, including coefficient of drag, ballistic coefficient, etc.
- class [FlatPlateAeroFacet](#)
The aerodynamic specific version of a flat plate.
- class [FlatPlateAeroFactory](#)
Creates a [FlatPlateAeroFacet](#) from a FlatPlate.
- class [FlatPlateAeroParams](#)
used in the [FlatPlateAeroFactory](#) to create a [FlatPlateAeroFacet](#).
- class [FlatPlateThermalAeroFactory](#)
Creates a [FlatPlateAeroFacet](#) from a FlatPlate.

7.1.1 Detailed Description

Namespace jeod.

Chapter 8

Data Structure Documentation

8.1 jeod::AeroDragEnum Class Reference

Contains enumerations associated with aerodynamic drag.

```
#include <aero_drag.hh>
```

Public Types

- enum [CoefCalcMethod](#) { [Specular](#) = 0, [Diffuse](#), [Mixed](#), [Calc_coef](#) }
Dictates how the coefficients of drag will be calculated when using a flat plate model.

8.1.1 Detailed Description

Contains enumerations associated with aerodynamic drag.

Definition at line 61 of file [aero_drag.hh](#).

8.1.2 Member Enumeration Documentation

8.1.2.1 enum jeod::AeroDragEnum::CoefCalcMethod

Dictates how the coefficients of drag will be calculated when using a flat plate model.

Enumerator

Specular

Diffuse

Mixed

Calc_coef

Definition at line 69 of file [aero_drag.hh](#).

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

- [aero_drag.hh](#)

8.2 jeod::AeroDragParameters Class Reference

Contains parameters associated with aerodynamic drag.

```
#include <aero_drag.hh>
```

Data Fields

- double [dynamic_pressure](#)
*dynamic pressure, $0.5 * \text{density} * \text{velocity}^2$*
- double [gas_const](#)
gas constant, ala $PV = mRT$; $R = 287$ for air.
- double [temp_free_stream](#)
temperature of the incident stream of free molecular flow.

8.2.1 Detailed Description

Contains parameters associated with aerodynamic drag.

Definition at line 90 of file `aero_drag.hh`.

8.2.2 Field Documentation

8.2.2.1 double jeod::AeroDragParameters::dynamic_pressure

dynamic pressure, $0.5 * \text{density} * \text{velocity}^2$

trick_units(N/m2)

Definition at line 95 of file `aero_drag.hh`.

Referenced by `jeod::AerodynamicDrag::aero_drag()`, `jeod::FlatPlateAeroFacet::aerodrag_force()`, and `jeod::DefaultAero::aerodrag_force()`.

8.2.2.2 double jeod::AeroDragParameters::gas_const

gas constant, ala $PV = mRT$; $R = 287$ for air.

trick_units(N*m/kg/K)

Definition at line 100 of file `aero_drag.hh`.

Referenced by `jeod::FlatPlateAeroFacet::aerodrag_force()`, and `jeod::AerodynamicDrag_aero_model_default_data::initialize()`.

8.2.2.3 double jeod::AeroDragParameters::temp_free_stream

temperature of the incident stream of free molecular flow.

trick_units(K)

Definition at line 105 of file `aero_drag.hh`.

Referenced by `jeod::FlatPlateAeroFacet::aerodrag_force()`, and `jeod::AerodynamicDrag_aero_model_default_data::initialize()`.

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

- [aero_drag.hh](#)

8.3 jeod::AerodynamicDrag Class Reference

The main class for calculating aerodynamic drag.

```
#include <aero_drag.hh>
```

Public Member Functions

- [AerodynamicDrag](#) ()
default constructor
- virtual [~AerodynamicDrag](#) ()
Destructor.
- void [aero_drag](#) (double inertial_velocity[3], AtmosphereState *atmos_ptr, double T_inertial_struct[3][3], double mass, double center_grav[3])
Calculates the total aerodynamic drag force and torque, from the information given.
- void [set_aero_surface](#) ([AeroSurface](#) &to_set)
Set the surface this AeroDrag object will calculate drag for.
- void [clear_aero_surface](#) ()
Remove any [AeroSurface](#) being used for calculation.

Data Fields

- bool [active](#)
On = aerodynamics enabled.
- bool [constant_density](#)
Use constant density for aero drag?
- double [density](#)
Density of the last time [AerodynamicDrag](#) was used.
- double [aero_force](#) [3]
Total Force due to aero drag, resulting from all plates combined.
- double [aero_torque](#) [3]
Total torque due to aero drag, resulting from all plates combined.
- [AeroDragParameters](#) param
parameters shared with plate model
- bool [use_default_behavior](#)
Use the default behavior?
- [AeroSurface](#) * [aero_surface_ptr](#)
Pointer to the current aero surface.
- [DefaultAero](#) * [default_behavior](#)
Pointer to an object that defines the default aero behavior.
- [DefaultAero](#) [ballistic_drag](#)
Spherical, ballistic drag.

Private Member Functions

- [AerodynamicDrag](#) (const [AerodynamicDrag](#) &)
- [AerodynamicDrag](#) & operator= (const [AerodynamicDrag](#) &)

8.3.1 Detailed Description

The main class for calculating aerodynamic drag.

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

8.3.2 Constructor & Destructor Documentation

8.3.2.1 `jeod::AerodynamicDrag::AerodynamicDrag (void)`

default constructor

Definition at line 82 of file `aero_drag.cc`.

References `aero_force`, `aero_torque`, `ballistic_drag`, and `default_behavior`.

8.3.2.2 `jeod::AerodynamicDrag::~AerodynamicDrag (void) [virtual]`

Destructor.

Definition at line 100 of file `aero_drag.cc`.

8.3.2.3 `jeod::AerodynamicDrag::AerodynamicDrag (const AerodynamicDrag &) [private]`

8.3.3 Member Function Documentation

8.3.3.1 `void jeod::AerodynamicDrag::aero_drag (double inertial_velocity[3], AtmosphereState * atmos_ptr, double T_inertial_struct[3][3], double mass, double center_grav[3])`

Calculates the total aerodynamic drag force and torque, from the information given.

Parameters

in	<i>inertial_velocity</i>	vehicle velocity in inertial RF Units: M/s
in	<i>atmos_ptr</i>	Pointer to the AtmosphereState used for density and wind information
in	<i>T_inertial_struct</i>	Transformation matrix from the inertial frame to the structural
in	<i>mass</i>	kg Mass of the vehicle
in	<i>center_grav</i>	position of the center of gravity, in the structural frame Units: M

Definition at line 116 of file `aero_drag.cc`.

References `active`, `jeod::AeroSurface::aero_facets`, `aero_force`, `aero_surface_ptr`, `aero_torque`, `jeod::AeroFacet::aerodrag_force()`, `jeod::DefaultAero::aerodrag_force()`, `constant_density`, `default_behavior`, `density`, `jeod::AeroDragParameters::dynamic_pressure`, `jeod::AeroSurface::facets_size`, `param`, `jeod::AerodynamicsMessages::runtime_error`, and `use_default_behavior`.

8.3.3.2 `void jeod::AerodynamicDrag::clear_aero_surface ()`

Remove any [AeroSurface](#) being used for calculation.

Note: The variable "use_default_behavior" must be set to true if there is no set aero surface

Definition at line 221 of file `aero_drag.cc`.

References `aero_surface_ptr`.

8.3.3.3 **AerodynamicDrag&** jeod::AerodynamicDrag::operator= (const **AerodynamicDrag** &) [private]

8.3.3.4 void jeod::AerodynamicDrag::set_aero_surface (**AeroSurface** & *to_set*)

Set the surface this AeroDrag object will calculate drag for.

Parameters

<code>in</code>	<code>to_set</code>	The AeroSurface to be used
-----------------	---------------------	--

Definition at line 209 of file `aero_drag.cc`.

References `aero_surface_ptr`.

8.3.4 Field Documentation**8.3.4.1 `bool jeod::AerodynamicDrag::active`**

On = aerodynamics enabled.

`trick_units(-)`

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

Referenced by `aero_drag()`, and `jeod::AerodynamicDrag_aero_model_default_data::initialize()`.

8.3.4.2 `double jeod::AerodynamicDrag::aero_force[3]`

Total Force due to aero drag, resulting from all plates combined.

`trick_units(N)`

Definition at line 134 of file `aero_drag.hh`.

Referenced by `aero_drag()`, and `AerodynamicDrag()`.

8.3.4.3 `AeroSurface* jeod::AerodynamicDrag::aero_surface_ptr`

Pointer to the current aero surface.

`trick_units(-)`

Definition at line 154 of file `aero_drag.hh`.

Referenced by `aero_drag()`, `clear_aero_surface()`, and `set_aero_surface()`.

8.3.4.4 `double jeod::AerodynamicDrag::aero_torque[3]`

Total torque due to aero drag, resulting from all plates combined.

`trick_units(N*m)`

Definition at line 139 of file `aero_drag.hh`.

Referenced by `aero_drag()`, and `AerodynamicDrag()`.

8.3.4.5 `DefaultAero jeod::AerodynamicDrag::ballistic_drag`

Spherical, ballistic drag.

The default, default behavior. Can be overridden by resetting the "default_behavior" pointer `trick_units(-)`

Definition at line 168 of file `aero_drag.hh`.

Referenced by `AerodynamicDrag()`.

8.3.4.6 `bool jeod::AerodynamicDrag::constant_density`

Use constant density for aero drag?

trick_units(-)

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

Referenced by `aero_drag()`.

8.3.4.7 DefaultAero* jeod::AerodynamicDrag::default_behavior

Pointer to an object that defines the default aero behavior.

This is used if the [AeroSurface](#) pointer in `aero_drag` is set to NULL. Defaults to ballistic drag, but can be overridden.

Definition at line 161 of file `aero_drag.hh`.

Referenced by `aero_drag()`, and `AerodynamicDrag()`.

8.3.4.8 double jeod::AerodynamicDrag::density

Density of the last time [AerodynamicDrag](#) was used.

If `constant_density` is set true, then this is the density that will be used.

Definition at line 129 of file `aero_drag.hh`.

Referenced by `aero_drag()`.

8.3.4.9 AeroDragParameters jeod::AerodynamicDrag::param

parameters shared with plate model

trick_units(-)

Definition at line 144 of file `aero_drag.hh`.

Referenced by `aero_drag()`, and `jeod::AerodynamicDrag_aero_model_default_data::initialize()`.

8.3.4.10 bool jeod::AerodynamicDrag::use_default_behavior

Use the default behavior?

trick_units(-)

Definition at line 149 of file `aero_drag.hh`.

Referenced by `aero_drag()`.

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

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

8.4 jeod::AerodynamicDrag_aero_model_default_data Class Reference

```
#include <aero_model.hh>
```

Public Member Functions

- void [initialize](#) ([AerodynamicDrag](#) *)

8.4.1 Detailed Description

Definition at line 18 of file `aero_model.hh`.

8.4.2 Member Function Documentation

8.4.2.1 `void jeod::AerodynamicDrag_aero_model_default_data::initialize (AerodynamicDrag * AerodynamicDrag_ptr)`

Definition at line 33 of file `aero_model.cc`.

References `jeod::AerodynamicDrag::active`, `jeod::AeroDragParameters::gas_const`, `jeod::AerodynamicDrag::param`, and `jeod::AeroDragParameters::temp_free_stream`.

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

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

8.5 jeod::AerodynamicsMessages Class Reference

Messages associated with use of the aerodynamics model.

```
#include <aerodynamics_messages.hh>
```

Static Public Attributes

- static char const * [initialization_error](#)
Associated with errors during initialization of the drag model.
- static char const * [runtime_error](#)
Associated with errors during the runtime of the drag model.
- static char const * [pre_initialization_error](#)
Associated with errors during the setup of the system, before runtime.
- static char const * [runtime_warns](#)
Associated with warnings given at runtime.

Private Member Functions

- [AerodynamicsMessages](#) (void)
- [AerodynamicsMessages](#) (const [AerodynamicsMessages](#) &rhs)
- [AerodynamicsMessages](#) & operator= (const [AerodynamicsMessages](#) &rhs)

Friends

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

8.5.1 Detailed Description

Messages associated with use of the aerodynamics model.

Definition at line 49 of file `aerodynamics_messages.hh`.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 `jeod::AerodynamicsMessages::AerodynamicsMessages (void) [private]`

8.5.2.2 `jeod::AerodynamicsMessages::AerodynamicsMessages (const AerodynamicsMessages & rhs) [private]`

8.5.3 Member Function Documentation

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

8.5.4 Friends And Related Function Documentation

8.5.4.1 `void init_attrjeod__AerodynamicsMessages () [friend]`

8.5.4.2 `friend class InputProcessor [friend]`

Definition at line 51 of file aerodynamics_messages.hh.

8.5.5 Field Documentation

8.5.5.1 `char const * jeod::AerodynamicsMessages::initialization_error [static]`

Initial value:

```
=
    "interactions/aerodynamics/" "initialization_error"
```

Associated with errors during initialization of the drag model.

trick_units(-)

Definition at line 62 of file aerodynamics_messages.hh.

Referenced by `jeod::AeroSurface::allocate_array()`, `jeod::AeroSurface::allocate_interaction_facet()`, and `jeod::FlatPlateAeroFactory::create_facet()`.

8.5.5.2 `char const * jeod::AerodynamicsMessages::pre_initialization_error [static]`

Initial value:

```
=
    "interactions/aerodynamics/" "pre_initialization_error"
```

Associated with errors during the setup of the system, before runtime.

trick_units(-)

Definition at line 70 of file aerodynamics_messages.hh.

Referenced by `jeod::AeroSurfaceFactory::add_facet_params()`.

8.5.5.3 `char const * jeod::AerodynamicsMessages::runtime_error [static]`

Initial value:

```
=
    "interactions/aerodynamics/" "runtime_error"
```

Associated with errors during the runtime of the drag model.

trick_units(-)

Definition at line 66 of file aerodynamics_messages.hh.

Referenced by jeod::AerodynamicDrag::aero_drag(), jeod::FlatPlateAeroFacet::aerodrag_force(), and jeod::DefaultAero::aerodrag_force().

8.5.5.4 `char const * jeod::AerodynamicsMessages::runtime_warns` [static]

Initial value:

```
=
    "interactions/aerodynamics/" "runtime_warns"
```

Associated with warnings given at runtime.

trick_units(-)

Definition at line 77 of file aerodynamics_messages.hh.

Referenced by jeod::FlatPlateAeroFacet::aerodrag_force(), and jeod::DefaultAero::aerodrag_force().

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

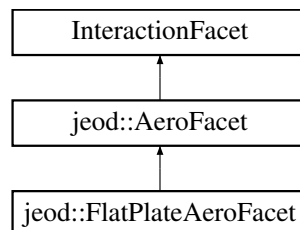
- [aerodynamics_messages.hh](#)
- [aerodynamics_messages.cc](#)

8.6 jeod::AeroFacet Class Reference

An aerodynamic interaction specific facet for use in the surface model.

```
#include <aero_facet.hh>
```

Inheritance diagram for jeod::AeroFacet:



Public Member Functions

- [AeroFacet](#) ()
Default constructor.
- virtual [~AeroFacet](#) ()
Destructor.
- virtual void [aerodrag_force](#) (const double velocity_mag, const double rel_vel_hat[3], [AeroDragParameters](#) *aero_drag_param_ptr, double center_grav[3])=0
A pure virtual function defining the interface for all aerodynamic interaction facets.

Private Member Functions

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

Friends

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

8.6.1 Detailed Description

An aerodynamic interaction specific facet for use in the surface model.

Definition at line 45 of file `aero_facet.hh`.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 `jeod::AeroFacet::AeroFacet (void)`

Default constructor.

Definition at line 45 of file `aero_facet.cc`.

8.6.2.2 `jeod::AeroFacet::~~AeroFacet (void)` `[virtual]`

Destructor.

Definition at line 57 of file `aero_facet.cc`.

8.6.2.3 `jeod::AeroFacet::AeroFacet (const AeroFacet & rhs)` `[private]`

8.6.3 Member Function Documentation

8.6.3.1 `virtual void jeod::AeroFacet::aerodrag_force (const double velocity_mag, const double rel_vel_hat[3], AeroDragParameters * aero_drag_param_ptr, double center_grav[3])` `[pure virtual]`

A pure virtual function defining the interface for all aerodynamic interaction facets.

All aerodynamic interaction facets inherited from [AeroFacet](#) must implement this function

Parameters

in	<i>velocity_mag</i>	The magnitude of the relative inertial velocity, including wind, of the vehicle Units: m/s
in	<i>rel_vel_hat</i>	The Unit vector of the relative inertial velocity
in	<i>aero_drag_param_ptr</i>	The parameters used to calculate aerodynamic drag
in	<i>center_grav</i>	The position of the center of gravity of the vehicle, in the structural frame Units: m

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

Referenced by `jeod::AerodynamicDrag::aero_drag()`.

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

8.6.4 Friends And Related Function Documentation

8.6.4.1 `void init_attrjeod__AeroFacet ()` `[friend]`

8.6.4.2 friend class InputProcessor [friend]

Definition at line 47 of file aero_facet.hh.

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

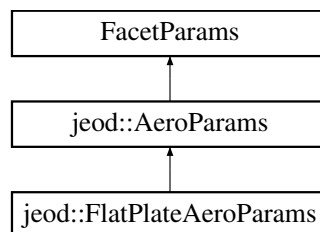
- [aero_facet.hh](#)
- [aero_facet.cc](#)

8.7 jeod::AeroParams Class Reference

A base class for all aerodynamic parameters used in the surface model.

```
#include <aero_params.hh>
```

Inheritance diagram for jeod::AeroParams:



Public Member Functions

- [AeroParams](#) ()
Default Constructor.
- virtual [~AeroParams](#) ()
Destructor.

Private Member Functions

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

Friends

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

8.7.1 Detailed Description

A base class for all aerodynamic parameters used in the surface model.

Definition at line 44 of file aero_params.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 jeod::AeroParams::AeroParams (void)

Default Constructor.

Definition at line 41 of file aero_params.cc.

8.7.2.2 `jeod::AeroParams::~~AeroParams (void) [virtual]`

Destructor.

Definition at line 53 of file aero_params.cc.

8.7.2.3 `jeod::AeroParams::AeroParams (const AeroParams & rhs) [private]`

8.7.3 Member Function Documentation

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

8.7.4 Friends And Related Function Documentation

8.7.4.1 `void init_attrjeod__AeroParams () [friend]`

8.7.4.2 `friend class InputProcessor [friend]`

Definition at line 46 of file aero_params.hh.

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

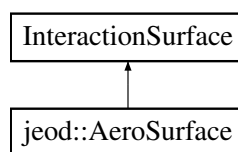
- [aero_params.hh](#)
- [aero_params.cc](#)

8.8 jeod::AeroSurface Class Reference

The aerodynamic specific interaction surface, for use with the surface model.

```
#include <aero_surface.hh>
```

Inheritance diagram for jeod::AeroSurface:



Public Member Functions

- [AeroSurface \(\)](#)
Default Constructor.
- `virtual ~AeroSurface ()`
Destructor.
- `virtual void allocate_array (unsigned int size)`
Allocates an array of [AeroFacet](#) pointers, of the size indicated by the input variable.
- `virtual void allocate_interaction_facet (Facet *facet, InteractionFacetFactory *factory, FacetParams *params, unsigned int index)`
Allocates a particular interaction facet, from an inputted general facet, using the inputted parameters and interaction facet factory.

Data Fields

- [AeroFacet](#) ** [aero_facets](#)
An array of pointers to aerodynamic interaction facets.
- unsigned int [facets_size](#)
Size of the [aero_facets](#) array.

Private Member Functions

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

Friends

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

8.8.1 Detailed Description

The aerodynamic specific interaction surface, for use with the surface model.

Definition at line 50 of file [aero_surface.hh](#).

8.8.2 Constructor & Destructor Documentation

8.8.2.1 `jeod::AeroSurface::AeroSurface (void)`

Default Constructor.

Definition at line 59 of file [aero_surface.cc](#).

8.8.2.2 `jeod::AeroSurface::~~AeroSurface (void)` `[virtual]`

Destructor.

Definition at line 73 of file [aero_surface.cc](#).

References [aero_facets](#), and [facets_size](#).

8.8.2.3 `jeod::AeroSurface::AeroSurface (const AeroSurface & rhs)` `[private]`

8.8.3 Member Function Documentation

8.8.3.1 `void jeod::AeroSurface::allocate_array (unsigned int size)` `[virtual]`

Allocates an array of [AeroFacet](#) pointers, of the size indicated by the input variable.

Parameters

<code>in</code>	<code>size</code>	The size of the needed array Units: cnt:
-----------------	-------------------	---

Definition at line 98 of file [aero_surface.cc](#).

References [aero_facets](#), [facets_size](#), and [jeod::AerodynamicsMessages::initialization_error](#).

8.8.3.2 `void jeod::AeroSurface::allocate_interaction_facet (Facet * facet, InteractionFacetFactory * factory, FacetParams * params, unsigned int index) [virtual]`

Allocates a particular interaction facet, from an inputted general facet, using the inputted parameters and interaction facet factory.

This facet is then placed at the index given. If the correct InteractionFacetFactory and Facet Params are not given for the aerodynamic interaction or for the type of facet given, a fail message will be sent

Parameters

in	<i>facet</i>	The basic facet used to create the interaction facet
in	<i>factory</i>	The factory used to create the interaction facet
in	<i>params</i>	The aero params used to create the interaction facet
in	<i>index</i>	Where the new interaction facet will be placed in the aero_facets array Units: cnt

Definition at line 149 of file `aero_surface.cc`.

References `aero_facets`, `facets_size`, and `jeod::AerodynamicsMessages::initialization_error`.

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

8.8.4 Friends And Related Function Documentation

8.8.4.1 `void init_attrjeod__AeroSurface () [friend]`

8.8.4.2 `friend class InputProcessor [friend]`

Definition at line 52 of file `aero_surface.hh`.

8.8.5 Field Documentation

8.8.5.1 `AeroFacet** jeod::AeroSurface::aero_facets`

An array of pointers to aerodynamic interaction facets.

AeroFacets is a pure virtual, so these will all be pointed to inheriting classes through polymorphism

Definition at line 67 of file `aero_surface.hh`.

Referenced by `jeod::AerodynamicDrag::aero_drag()`, `allocate_array()`, `allocate_interaction_facet()`, and `~AeroSurface()`.

8.8.5.2 `unsigned int jeod::AeroSurface::facets_size`

Size of the `aero_facets` array.

`trick_units(count)`

Definition at line 72 of file `aero_surface.hh`.

Referenced by `jeod::AerodynamicDrag::aero_drag()`, `allocate_array()`, `allocate_interaction_facet()`, and `~AeroSurface()`.

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

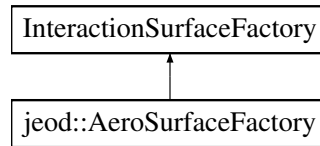
- [aero_surface.hh](#)
- [aero_surface.cc](#)

8.9 jeod::AeroSurfaceFactory Class Reference

The surface factory that creates an aerodynamic specific surface from a general surface.

```
#include <aero_surface_factory.hh>
```

Inheritance diagram for jeod::AeroSurfaceFactory:



Public Member Functions

- [AeroSurfaceFactory](#) ()
Default Constructor.
- virtual [~AeroSurfaceFactory](#) ()
Destructor.
- virtual void [add_facet_params](#) (FacetParams *to_add)
Add a named set of facet params to the surface factory.

Protected Attributes

- [FlatPlateAeroFactory](#) flat_plate_aero_factory
A factory that can create a flat plate aero facet from a flat plate.
- [FlatPlateThermalAeroFactory](#) flat_plate_thermal_aero_factory
A factory that can create a flat plate aero facet from a flat plate.

Private Member Functions

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

Friends

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

8.9.1 Detailed Description

The surface factory that creates an aerodynamic specific surface from a general surface.

Used with the surface model.

Definition at line 51 of file `aero_surface_factory.hh`.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 jeod::AeroSurfaceFactory::AeroSurfaceFactory (void)

Default Constructor.

Definition at line 49 of file `aero_surface_factory.cc`.

References `flat_plate_aero_factory`, and `flat_plate_thermal_aero_factory`.

8.9.2.2 jeod::AeroSurfaceFactory::~AeroSurfaceFactory (void) [virtual]

Destructor.

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

8.9.2.3 jeod::AeroSurfaceFactory::AeroSurfaceFactory (const AeroSurfaceFactory & rhs) [private]

8.9.3 Member Function Documentation

8.9.3.1 void jeod::AeroSurfaceFactory::add_facet_params (FacetParams * to_add) [virtual]

Add a named set of facet params to the surface factory.

Intended to be used when an aerodynamic specific surface is created, to convert a basic facet to an aerodynamic interaction facet. This MUST be a parameter inheriting from `AeroParam`, or the function will fail and send a failure message

Parameters

<i>in</i>	<i>to_add</i>	The facet parameters to add
-----------	---------------	-----------------------------

Definition at line 80 of file `aero_surface_factory.cc`.

References `jeod::AerodynamicsMessages::pre_initialization_error`.

8.9.3.2 AeroSurfaceFactory& jeod::AeroSurfaceFactory::operator= (const AeroSurfaceFactory & rhs) [private]

8.9.4 Friends And Related Function Documentation

8.9.4.1 void init_attrjeod__AeroSurfaceFactory () [friend]

8.9.4.2 friend class InputProcessor [friend]

Definition at line 53 of file `aero_surface_factory.hh`.

8.9.5 Field Documentation

8.9.5.1 FlatPlateAeroFactory jeod::AeroSurfaceFactory::flat_plate_aero_factory [protected]

A factory that can create a flat plate aero facet from a flat plate.

`trick_units(-)`

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

Referenced by `AeroSurfaceFactory()`.

8.9.5.2 FlatPlateThermalAeroFactory jeod::AeroSurfaceFactory::flat_plate_thermal_aero_factory [protected]

A factory that can create a flat plate aero facet from a flat plate.

trick_units(-)

Definition at line 81 of file aero_surface_factory.hh.

Referenced by AeroSurfaceFactory().

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

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

8.10 jeod::DefaultAero Class Reference

The simple, default, aerodynamic drag model, including coefficient of drag, ballistic coefficient, etc.

```
#include <default_aero.hh>
```

Public Types

- enum [DragOption](#) { [DRAG_OPT_CD](#) = 0, [DRAG_OPT_BC](#) = 1, [DRAG_OPT_CONST](#) = 2 }
- Specifies how drag is to be computed.*

Public Member Functions

- [DefaultAero](#) ()
Defaul Constructor.
- virtual [~DefaultAero](#) ()
DefaultConstructor.
- virtual void [aerodrag_force](#) (const double velocity_mag, const double rel_vel_hat[3], [AeroDragParameters](#) *aero_drag_param_ptr, double mass, double force[3], double torque[3])
The implementation for this aerodynamic drags force and torque calculations.

Data Fields

- double [Cd](#)
Coefficient of drag.
- double [BC](#)
Ballistic Coefficient.
- double [area](#)
Vehicle aerodynamic area.
- double [drag](#)
Drag calculated during use.
- [DragOption](#) option
The type of simple drag to use.

Private Member Functions

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

Friends

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

8.10.1 Detailed Description

The simple, default, aerodynamic drag model, including coefficient of drag, ballistic coefficient, etc.

This can be overridden with a user defined model in the [AerodynamicDrag](#) class.

Definition at line 51 of file `default_aero.hh`.

8.10.2 Member Enumeration Documentation

8.10.2.1 enum `jeod::DefaultAero::DragOption`

Specifies how drag is to be computed.

Enumerator

DRAG_OPT_CD Use Coefficient of drag for drag computations.

DRAG_OPT_BC Use Ballistic Coefficient for drag computations.

DRAG_OPT_CONST Use specified constant drag.

Definition at line 60 of file `default_aero.hh`.

8.10.3 Constructor & Destructor Documentation

8.10.3.1 `jeod::DefaultAero::DefaultAero (void)`

Default Constructor.

Definition at line 52 of file `default_aero.cc`.

8.10.3.2 `jeod::DefaultAero::~DefaultAero (void) [virtual]`

DefaultConstructor.

Definition at line 68 of file `default_aero.cc`.

8.10.3.3 `jeod::DefaultAero::DefaultAero (const DefaultAero & rhs) [private]`

8.10.4 Member Function Documentation

8.10.4.1 `void jeod::DefaultAero::aerodrag_force (const double velocity_mag, const double rel_vel_hat[3], AeroDragParameters * aero_drag_param_ptr, double mass, double force[3], double torque[3]) [virtual]`

The implementation for this aerodynamic drags force and torque calculations.

Can be overridden by an inheriting class to create extensibility

Parameters

in	<i>velocity_mag</i>	The magnitude of the relative velocity of the vehicle; not used here but some child classes need it Units: M/s
in	<i>rel_vel_hat</i>	The unit vector of the relative velocity of the vehicle, in the structural frame
in	<i>aero_drag_param_ptr</i>	The aerodynamic drag parameters used to calculate drag
in	<i>mass</i>	The current mass of the vehicle Units: kg
out	<i>force</i>	The aerodynamic force, in the structural frame Units: N
out	<i>torque</i>	The aerodynamic torque, in the structural frame Units: N*M

Definition at line 89 of file default_aero.cc.

References `area`, `BC`, `Cd`, `drag`, `DRAG_OPT_BC`, `DRAG_OPT_CD`, `DRAG_OPT_CONST`, `jeod::AeroDragParameters::dynamic_pressure`, `option`, `jeod::AerodynamicsMessages::runtime_error`, and `jeod::AerodynamicsMessages::runtime_warns`.

Referenced by `jeod::AerodynamicDrag::aero_drag()`.

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

8.10.5 Friends And Related Function Documentation

8.10.5.1 `void init_attrjeod__DefaultAero ()` `[friend]`

8.10.5.2 `friend class InputProcessor` `[friend]`

Definition at line 53 of file default_aero.hh.

8.10.6 Field Documentation

8.10.6.1 `double jeod::DefaultAero::area`

Vehicle aerodynamic area.

`trick_units(m2)`

Definition at line 93 of file default_aero.hh.

Referenced by `aerodrag_force()`.

8.10.6.2 `double jeod::DefaultAero::BC`

Ballistic Coefficient.

`trick_units(kg/m2)`

Definition at line 89 of file default_aero.hh.

Referenced by `aerodrag_force()`.

8.10.6.3 `double jeod::DefaultAero::Cd`

Coefficient of drag.

`trick_units(-)`

Definition at line 85 of file default_aero.hh.

Referenced by aerodrag_force().

8.10.6.4 double jeod::DefaultAero::drag

Drag calculated during use.

Can be set by user and will then never be changed with a DRAG_OPT_CONSTtrick_units(N)

Definition at line 99 of file default_aero.hh.

Referenced by aerodrag_force().

8.10.6.5 DragOption jeod::DefaultAero::option

The type of simple drag to use.

trick_units(-)

Definition at line 104 of file default_aero.hh.

Referenced by aerodrag_force().

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

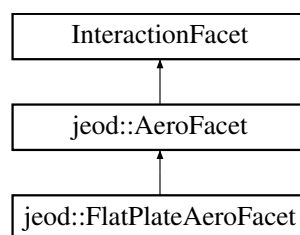
- [default_aero.hh](#)
- [default_aero.cc](#)

8.11 jeod::FlatPlateAeroFacet Class Reference

The aerodynamic specific version of a flat plate.

```
#include <flat_plate_aero_facet.hh>
```

Inheritance diagram for jeod::FlatPlateAeroFacet:



Public Member Functions

- [FlatPlateAeroFacet](#) ()
Default Constructor.
- virtual [~FlatPlateAeroFacet](#) ()
Destructor.
- virtual void [aerodrag_force](#) (const double velocity_mag, const double rel_vel_hat[3], [AeroDragParameters](#) *aero_drag_param_ptr, double center_grav[3])
The [FlatPlateAeroFacet](#) specific implementation of aerodynamic drag force, based on the given parameters.

Data Fields

- double * [center_pressure](#)
Flat plate center of pressure (in structural frame).
- double * [normal](#)
Unit vector normal to the plate surface, pointing outward (structural frame).
- double [force_n](#)
Magnitude of the force normal to the plate.
- double [force_t](#)
Magnitude of the force tangential to the plate, or parallel to the velocity vector, depending on application.
- [AeroDragEnum::CoefCalcMethod](#) [coef_method](#)
Enum indicating which method of coef calculation to use: specular, diffuse, calculated, mixed.
- bool [calculate_drag_coef](#)
whether to calculate the drag coefficient
- double [epsilon](#)
fraction of molecules that "bounce"
- double [temp_reflect](#)
temperature of reflected molecules
- double [drag_coef_norm](#)
The coefficient for calculating drag normal to the plate.
- double [drag_coef_tang](#)
The coefficient for calculating drag tangential to the plate.
- double [drag_coef_spec](#)
The coefficient for calculating drag resulting only from molecules bouncing off the surface.
- double [drag_coef_diff](#)
The coefficient for calculating drag resulting only from molecules sticking to the surface.
- double [temperature](#)
Temperature of the plate.
- double [area](#)
area of the plate

Private Member Functions

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

Friends

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

8.11.1 Detailed Description

The aerodynamic specific version of a flat plate.

Definition at line 48 of file `flat_plate_aero_facet.hh`.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 `jeod::FlatPlateAeroFacet::FlatPlateAeroFacet (void)`

Default Constructor.

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

8.11.2.2 `jeod::FlatPlateAeroFacet::~~FlatPlateAeroFacet (void) [virtual]`

Destructor.

Definition at line 71 of file `flat_plate_aero_facet.cc`.

8.11.2.3 `jeod::FlatPlateAeroFacet::FlatPlateAeroFacet (const FlatPlateAeroFacet & rhs) [private]`

8.11.3 Member Function Documentation

8.11.3.1 `void jeod::FlatPlateAeroFacet::aerodrag_force (const double rel_vel_mag, const double rel_vel_struct_hat[3], AeroDragParameters * aero_drag_param_ptr, double center_grav[3]) [virtual]`

The [FlatPlateAeroFacet](#) specific implementation of aerodynamic drag force, based on the given parameters.

Parameters

in	<i>rel_vel_mag</i>	The magnitude of the relative velocity Units: M/s
in	<i>rel_vel_struct_hat</i>	The unit vector of the total relative velocity, in the structural frame
in	<i>aero_drag_param_ptr</i>	The aerodynamic drag parameters used for drag calculation
in	<i>center_grav</i>	The center of gravity of the vehicle, in the structural frame Units: M

Implements [jeod::AeroFacet](#).

Definition at line 88 of file `flat_plate_aero_facet.cc`.

References `area`, `jeod::AeroDragEnum::Calc_coef`, `calculate_drag_coef`, `center_pressure`, `coef_method`, `jeod::AeroDragEnum::Diffuse`, `drag_coef_diff`, `drag_coef_norm`, `drag_coef_spec`, `drag_coef_tang`, `jeod::AeroDragParameters::dynamic_pressure`, `epsilon`, `force_n`, `force_t`, `jeod::AeroDragParameters::gas_const`, `jeod::AeroDragEnum::Mixed`, `normal`, `jeod::AerodynamicsMessages::runtime_error`, `jeod::AerodynamicsMessages::runtime_warns`, `jeod::AeroDragEnum::Specular`, `jeod::AeroDragParameters::temp_free_stream`, and `temperature`.

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

8.11.4 Friends And Related Function Documentation

8.11.4.1 `void init_attrjeod_FlatPlateAeroFacet () [friend]`

8.11.4.2 `friend class InputProcessor [friend]`

Definition at line 50 of file `flat_plate_aero_facet.hh`.

8.11.5 Field Documentation

8.11.5.1 `double jeod::FlatPlateAeroFacet::area`

area of the plate

`trick_units(m2)`

Definition at line 133 of file `flat_plate_aero_facet.hh`.

Referenced by `aerodrag_force()`, and `jeod::FlatPlateAeroFactory::create_facet()`.

8.11.5.2 `bool jeod::FlatPlateAeroFacet::calculate_drag_coef`

whether to calculate the drag coefficient

trick_units(-)

Definition at line 99 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force(), and jeod::FlatPlateAeroFactory::create_facet().

8.11.5.3 `double* jeod::FlatPlateAeroFacet::center_pressure`

Flat plate center of pressure (in structural frame).

Once the aero surface is initialized, it points to the position found in FlatPlatetrick_units(m)

Definition at line 72 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force(), and jeod::FlatPlateAeroFactory::create_facet().

8.11.5.4 `AeroDragEnum::CoefCalcMethod jeod::FlatPlateAeroFacet::coef_method`

Enum indicating which method of coef calculation to use: specular, diffuse, calculated, mixed.

trick_units(-)

Definition at line 95 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force(), and jeod::FlatPlateAeroFactory::create_facet().

8.11.5.5 `double jeod::FlatPlateAeroFacet::drag_coef_diff`

The coefficient for calculating drag resulting only from molecules sticking to the surface.

trick_units(-)

Definition at line 125 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force(), and jeod::FlatPlateAeroFactory::create_facet().

8.11.5.6 `double jeod::FlatPlateAeroFacet::drag_coef_norm`

The coefficient for calculating drag normal to the plate.

trick_units(-)

Definition at line 111 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force(), and jeod::FlatPlateAeroFactory::create_facet().

8.11.5.7 `double jeod::FlatPlateAeroFacet::drag_coef_spec`

The coefficient for calculating drag resulting only from molecules bouncing off the surface.

trick_units(-)

Definition at line 120 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force(), and jeod::FlatPlateAeroFactory::create_facet().

8.11.5.8 `double jeod::FlatPlateAeroFacet::drag_coef_tang`

The coefficient for calculating drag tangential to the plate.

trick_units(-)

Definition at line 115 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force(), and jeod::FlatPlateAeroFactory::create_facet().

8.11.5.9 double jeod::FlatPlateAeroFacet::epsilon

fraction of molecules that "bounce"

trick_units(-)

Definition at line 103 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force(), and jeod::FlatPlateAeroFactory::create_facet().

8.11.5.10 double jeod::FlatPlateAeroFacet::force_n

Magnitude of the force normal to the plate.

trick_units(N)

Definition at line 84 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force().

8.11.5.11 double jeod::FlatPlateAeroFacet::force_t

Magnitude of the force tangential to the plate, or parallel to the velocity vector, depending on application.

trick_units(N)

Definition at line 89 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force().

8.11.5.12 double* jeod::FlatPlateAeroFacet::normal

Unit vector normal to the plate surface, pointing outward (structural frame).

Once the aero surface is initialized, it points to the normal found in FlatPlatetrick_units(-)

Definition at line 79 of file flat_plate_aero_facet.hh.

Referenced by aerodrag_force(), and jeod::FlatPlateAeroFactory::create_facet().

8.11.5.13 double jeod::FlatPlateAeroFacet::temp_reflect

temperature of reflected molecules

trick_units(K)

Definition at line 107 of file flat_plate_aero_facet.hh.

Referenced by jeod::FlatPlateAeroFactory::create_facet().

8.11.5.14 double jeod::FlatPlateAeroFacet::temperature

Temperature of the plate.

trick_units(K)

Definition at line 129 of file flat_plate_aero_facet.hh.

Referenced by `aerodrag_force()`, and `jeod::FlatPlateAeroFactory::create_facet()`.

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

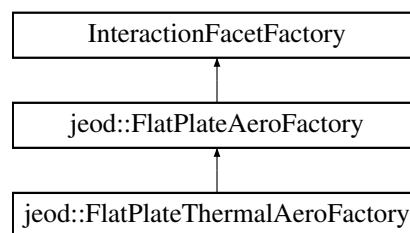
- [flat_plate_aero_facet.hh](#)
- [flat_plate_aero_facet.cc](#)

8.12 jeod::FlatPlateAeroFactory Class Reference

Creates a [FlatPlateAeroFacet](#) from a FlatPlate.

```
#include <flat_plate_aero_factory.hh>
```

Inheritance diagram for `jeod::FlatPlateAeroFactory`:



Public Member Functions

- [FlatPlateAeroFactory](#) ()
Default Constructor.
- [~FlatPlateAeroFactory](#) ()
Destructor.
- virtual `InteractionFacet * create_facet` (Facet *facet, FacetParams *params)
Create a [FlatPlateAeroFacet](#) from a flat plate facet and a FlatPlateAeroParams object.
- virtual bool `is_correct_factory` (Facet *facet)
[FlatPlateAeroFactory](#) specific implementation of this function.

Private Member Functions

- [FlatPlateAeroFactory](#) & `operator=` (const [FlatPlateAeroFactory](#) &rhs)
- [FlatPlateAeroFactory](#) (const [FlatPlateAeroFactory](#) &rhs)

Friends

- class [InputProcessor](#)
- void `init_attrjeod__FlatPlateAeroFactory` ()

8.12.1 Detailed Description

Creates a [FlatPlateAeroFacet](#) from a FlatPlate.

Definition at line 48 of file `flat_plate_aero_factory.hh`.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 jeod::FlatPlateAeroFactory::FlatPlateAeroFactory (void)

Default Constructor.

Definition at line 62 of file flat_plate_aero_factory.cc.

8.12.2.2 jeod::FlatPlateAeroFactory::~~FlatPlateAeroFactory (void)

Destructor.

Definition at line 73 of file flat_plate_aero_factory.cc.

8.12.2.3 jeod::FlatPlateAeroFactory::FlatPlateAeroFactory (const FlatPlateAeroFactory & rhs) [private]

8.12.3 Member Function Documentation

8.12.3.1 InteractionFacet * jeod::FlatPlateAeroFactory::create_facet (Facet * facet, FacetParams * params) [virtual]

Create a [FlatPlateAeroFacet](#) from a flat plate facet and a FlatPlateAeroParams object.

Returns

The new [FlatPlateAeroFacet](#). Note that this is allocated and YOU are responsible for destroying it at the end!

Parameters

in	<i>facet</i>	The FlatPlate. This MUST be a flat plate or the algorithm will send a failure message
in	<i>params</i>	FlatPlateAeroParams . This MUST be of the type FlatPlateAeroParams , or the algorithm will send a failure message Units: The

Definition at line 90 of file flat_plate_aero_factory.cc.

References [jeod::FlatPlateAeroFacet::area](#), [jeod::FlatPlateAeroParams::calculate_drag_coef](#), [jeod::FlatPlateAeroFacet::calculate_drag_coef](#), [jeod::FlatPlateAeroFacet::center_pressure](#), [jeod::FlatPlateAeroParams::coef_method](#), [jeod::FlatPlateAeroFacet::coef_method](#), [jeod::FlatPlateAeroParams::drag_coef_diff](#), [jeod::FlatPlateAeroFacet::drag_coef_diff](#), [jeod::FlatPlateAeroParams::drag_coef_norm](#), [jeod::FlatPlateAeroFacet::drag_coef_norm](#), [jeod::FlatPlateAeroParams::drag_coef_spec](#), [jeod::FlatPlateAeroFacet::drag_coef_spec](#), [jeod::FlatPlateAeroParams::drag_coef_tang](#), [jeod::FlatPlateAeroFacet::drag_coef_tang](#), [jeod::FlatPlateAeroParams::epsilon](#), [jeod::FlatPlateAeroFacet::epsilon](#), [jeod::AerodynamicsMessages::initialization_error](#), [jeod::FlatPlateAeroFacet::normal](#), [jeod::FlatPlateAeroParams::temp_reflect](#), [jeod::FlatPlateAeroFacet::temp_reflect](#), and [jeod::FlatPlateAeroFacet::temperature](#).

8.12.3.2 bool jeod::FlatPlateAeroFactory::is_correct_factory (Facet * facet) [virtual]

[FlatPlateAeroFactory](#) specific implementation of this function.

If the Facet is of type FlatPlate, returns true. False otherwise

Returns

true if facet is a FlatPlate, false otherwise

Parameters

<code>in</code>	<code>facet</code>	The facet to check
-----------------	--------------------	--------------------

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

Definition at line 162 of file `flat_plate_aero_factory.cc`.

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

8.12.4 Friends And Related Function Documentation

8.12.4.1 `void init_attrjeod__FlatPlateAeroFactory ()` `[friend]`

8.12.4.2 `friend class InputProcessor` `[friend]`

Definition at line 50 of file `flat_plate_aero_factory.hh`.

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

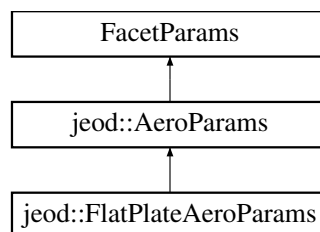
- [flat_plate_aero_factory.hh](#)
- [flat_plate_aero_factory.cc](#)

8.13 jeod::FlatPlateAeroParams Class Reference

used in the [FlatPlateAeroFactory](#) to create a [FlatPlateAeroFacet](#).

```
#include <flat_plate_aero_params.hh>
```

Inheritance diagram for `jeod::FlatPlateAeroParams`:



Public Member Functions

- [FlatPlateAeroParams](#) ()
Default Constructor.
- virtual [~FlatPlateAeroParams](#) ()
Destructor.

Data Fields

- [AeroDragEnum::CoefCalcMethod](#) `coef_method`
Enum indicating which method of coef calculation to use: specular, diffuse, calculated, mixed.
- bool [calculate_drag_coef](#)
Whether to calculate the drag coefficient.
- double [epsilon](#)

- *Fraction of molecules that "bounce".*
- double [temp_reflect](#)
- *Temperature of reflected molecules.*
- double [drag_coef_norm](#)
- *The coefficient for calculating drag normal to the plate.*
- double [drag_coef_tang](#)
- *The coefficient for calculating drag tangential to the plate.*
- double [drag_coef_spec](#)
- *The coefficient for calculating drag resulting only from molecules bouncing off the surface.*
- double [drag_coef_diff](#)
- *The coefficient for calculating drag resulting only from molecules sticking to the surface.*

Private Member Functions

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

Friends

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

8.13.1 Detailed Description

used in the [FlatPlateAeroFactory](#) to create a [FlatPlateAeroFacet](#).

Definition at line 44 of file `flat_plate_aero_params.hh`.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 jeod::FlatPlateAeroParams::FlatPlateAeroParams (void)

Default Constructor.

Definition at line 41 of file `flat_plate_aero_params.cc`.

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

Destructor.

Definition at line 60 of file `flat_plate_aero_params.cc`.

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

8.13.3 Member Function Documentation

8.13.3.1 FlatPlateAeroParams& jeod::FlatPlateAeroParams::operator= (const FlatPlateAeroParams & rhs) [private]

8.13.4 Friends And Related Function Documentation

8.13.4.1 void init_attrjeod__FlatPlateAeroParams () [friend]

8.13.4.2 friend class InputProcessor [friend]

Definition at line 46 of file flat_plate_aero_params.hh.

8.13.5 Field Documentation

8.13.5.1 bool jeod::FlatPlateAeroParams::calculate_drag_coef

Whether to calculate the drag coefficient.

trick_units(–)

Definition at line 64 of file flat_plate_aero_params.hh.

Referenced by jeod::FlatPlateAeroFactory::create_facet().

8.13.5.2 AeroDragEnum::CoefCalcMethod jeod::FlatPlateAeroParams::coef_method

Enum indicating which method of coef calculation to use: specular, diffuse, calculated, mixed.

trick_units(–)

Definition at line 60 of file flat_plate_aero_params.hh.

Referenced by jeod::FlatPlateAeroFactory::create_facet().

8.13.5.3 double jeod::FlatPlateAeroParams::drag_coef_diff

The coefficient for calculating drag resulting only from molecules sticking to the surface.

trick_units(–)

Definition at line 90 of file flat_plate_aero_params.hh.

Referenced by jeod::FlatPlateAeroFactory::create_facet().

8.13.5.4 double jeod::FlatPlateAeroParams::drag_coef_norm

The coefficient for calculating drag normal to the plate.

trick_units(–)

Definition at line 76 of file flat_plate_aero_params.hh.

Referenced by jeod::FlatPlateAeroFactory::create_facet().

8.13.5.5 double jeod::FlatPlateAeroParams::drag_coef_spec

The coefficient for calculating drag resulting only from molecules bouncing off the surface.

trick_units(–)

Definition at line 85 of file flat_plate_aero_params.hh.

Referenced by jeod::FlatPlateAeroFactory::create_facet().

8.13.5.6 double jeod::FlatPlateAeroParams::drag_coef_tang

The coefficient for calculating drag tangential to the plate.

trick_units(–)

Definition at line 80 of file flat_plate_aero_params.hh.

Referenced by jeod::FlatPlateAeroFactory::create_facet().

8.13.5.7 double jeod::FlatPlateAeroParams::epsilon

Fraction of molecules that "bounce".

trick_units(—)

Definition at line 68 of file flat_plate_aero_params.hh.

Referenced by jeod::FlatPlateAeroFactory::create_facet().

8.13.5.8 double jeod::FlatPlateAeroParams::temp_reflect

Temperature of reflected molecules.

trick_units(K)

Definition at line 72 of file flat_plate_aero_params.hh.

Referenced by jeod::FlatPlateAeroFactory::create_facet().

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

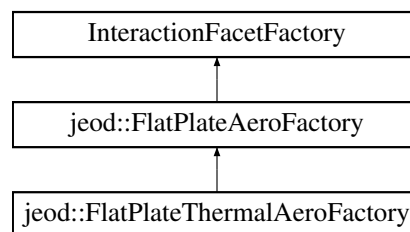
- [flat_plate_aero_params.hh](#)
- [flat_plate_aero_params.cc](#)

8.14 jeod::FlatPlateThermalAeroFactory Class Reference

Creates a [FlatPlateAeroFacet](#) from a FlatPlate.

```
#include <flat_plate_thermal_aero_factory.hh>
```

Inheritance diagram for jeod::FlatPlateThermalAeroFactory:



Public Member Functions

- [FlatPlateThermalAeroFactory](#) ()
Default Constructor.
- [~FlatPlateThermalAeroFactory](#) ()
Destructor.
- virtual bool [is_correct_factory](#) (Facet *facet)
FlatPlateThermalAeroFactory specific implementation of this function.

Private Member Functions

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

Friends

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

8.14.1 Detailed Description

Creates a [FlatPlateAeroFacet](#) from a FlatPlate.

Definition at line 45 of file flat_plate_thermal_aero_factory.hh.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 `jeod::FlatPlateThermalAeroFactory::FlatPlateThermalAeroFactory (void)`

Default Constructor.

Definition at line 51 of file flat_plate_thermal_aero_factory.cc.

8.14.2.2 `jeod::FlatPlateThermalAeroFactory::~FlatPlateThermalAeroFactory (void)`

Destructor.

Definition at line 64 of file flat_plate_thermal_aero_factory.cc.

8.14.2.3 `jeod::FlatPlateThermalAeroFactory::FlatPlateThermalAeroFactory (const FlatPlateThermalAeroFactory & rhs)` [private]

8.14.3 Member Function Documentation

8.14.3.1 `bool jeod::FlatPlateThermalAeroFactory::is_correct_factory (Facet * facet)` [virtual]

[FlatPlateThermalAeroFactory](#) specific implementation of this function.

If the Facet is of type FlatPlate, returns true. False otherwise

Returns

true if facet is a FlatPlate, false otherwise

Parameters

<code>in</code>	<code>facet</code>	The facet to check
-----------------	--------------------	--------------------

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

Definition at line 80 of file flat_plate_thermal_aero_factory.cc.

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

8.14.4 Friends And Related Function Documentation

8.14.4.1 `void init_attrjeod__FlatPlateThermalAeroFactory ()` [friend]

8.14.4.2 `friend class InputProcessor` [friend]

Definition at line 47 of file flat_plate_thermal_aero_factory.hh.

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

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

Chapter 9

File Documentation

9.1 `aero_drag.cc` File Reference

Orbital aerodynamic force and torque computation, and related classes.

```
#include <cstdlib>
#include <cmath>
#include "utils/math/include/vector3.hh"
#include "environment/atmosphere/base_atmos/include/atmosphere.hh"
#include "environment/atmosphere/base_atmos/include/atmosphere_state.hh"
#include "../include/aero_drag.hh"
#include "../include/aerodynamics_messages.hh"
#include "../include/aero_surface.hh"
#include "../include/aero_facet.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.1.1 Detailed Description

Orbital aerodynamic force and torque computation, and related classes.

Definition in file [aero_drag.cc](#).

9.2 `aero_drag.hh` File Reference

Orbital aerodynamics paramter definitions and the main class for calculating aerodynamic drag.

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

Data Structures

- class [jeod::AeroDragEnum](#)

Contains enumerations associated with aerodynamic drag.

- class [jeod::AeroDragParameters](#)

Contains parameters associated with aerodynamic drag.

- class [jeod::AerodynamicDrag](#)

The main class for calculating aerodynamic drag.

Namespaces

- [jeod](#)

Namespace jeod.

9.2.1 Detailed Description

Orbital aerodynamics paramter definitions and the main class for calculating aerodynamic drag.

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

9.3 aero_facet.cc File Reference

Individual facets for use with aero environment interaction models.

```
#include "utils/math/include/vector3.hh"
#include "../include/aero_facet.hh"
#include "../include/aero_drag.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.3.1 Detailed Description

Individual facets for use with aero environment interaction models.

Definition in file [aero_facet.cc](#).

9.4 aero_facet.hh File Reference

Individual facets for use with aero environment interaction models.

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

Data Structures

- class [jeod::AeroFacet](#)

An aerodynamic interaction specific facet for use in the surface model.

Namespaces

- [jeod](#)
Namespace jeod.

9.4.1 Detailed Description

Individual facets for use with aero environment interaction models.

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

9.5 `aero_model.cc` File Reference

```
#include "interactions/aerodynamics/include/aero_drag.hh"
#include "../include/aero_model.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

Macros

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

9.5.1 Macro Definition Documentation

9.5.1.1 `#define` JEOD_FRIEND_CLASS AerodynamicDrag_aero_model_default_data

Definition at line 19 of file `aero_model.cc`.

9.6 `aero_model.hh` File Reference

Data Structures

- class [jeod::AerodynamicDrag_aero_model_default_data](#)

Namespaces

- [jeod](#)
Namespace jeod.

9.7 `aero_params.cc` File Reference

A virtual base class for aero facet parameters, used to create interaction facets for aero in the InteractionSurface-Factorys.

```
#include "../include/aero_params.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.7.1 Detailed Description

A virtual base class for aero facet parameters, used to create interaction facets for aero in the InteractionSurface-Factorys.

Definition in file [aero_params.cc](#).

9.8 aero_params.hh File Reference

A virtual base class for aero facet parameters, used to create interaction facets for aero in the InteractionSurface-Factorys.

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

Data Structures

- class [jeod::AeroParams](#)

A base class for all aerodynamic parameters used in the surface model.

Namespaces

- [jeod](#)

Namespace jeod.

9.8.1 Detailed Description

A virtual base class for aero facet parameters, used to create interaction facets for aero in the InteractionSurface-Factorys.

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

9.9 aero_surface.cc File Reference

Vehicle surface model for the aerodynamic interaction models.

```
#include <cstdlib>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/surface_model/include/interaction_facet_factory.hh"
#include "utils/surface_model/include/facet.hh"
#include "../include/aero_surface.hh"
#include "../include/aero_facet.hh"
#include "../include/aerodynamics_messages.hh"
```


Namespaces

- [jeod](#)
Namespace jeod.

9.9.1 Detailed Description

Vehicle surface model for the aerodynamic interaction models.

Definition in file [aero_surface.cc](#).

9.10 `aero_surface.hh` File Reference

Vehicle surface model for aerodynamics.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/surface_model/include/interaction_surface.hh"
#include "aero_facet.hh"
#include "utils/surface_model/include/facet.hh"
```

Data Structures

- class [jeod::AeroSurface](#)
The aerodynamic specific interaction surface, for use with the surface model.

Namespaces

- [jeod](#)
Namespace jeod.

9.10.1 Detailed Description

Vehicle surface model for aerodynamics.

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

9.11 `aero_surface_factory.cc` File Reference

Factory that creates an aerodynamics surface, from a surface model.

```
#include <cstdint>
#include "utils/surface_model/include/facet_params.hh"
#include "../include/aero_surface_factory.hh"
#include "../include/aerodynamics_messages.hh"
#include "../include/aero_params.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.11.1 Detailed Description

Factory that creates an aerodynamics surface, from a surface model.

Definition in file [aero_surface_factory.cc](#).

9.12 aero_surface_factory.hh File Reference

Factory that creates an aerodynamic interaction surface from a surface model.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/surface_model/include/interaction_surface_factory.hh"
#include "flat_plate_aero_factory.hh"
#include "flat_plate_thermal_aero_factory.hh"
```

Data Structures

- class [jeod::AeroSurfaceFactory](#)

The surface factory that creates an aerodynamic specific surface from a general surface.

Namespaces

- [jeod](#)

Namespace jeod.

9.12.1 Detailed Description

Factory that creates an aerodynamic interaction surface from a surface model.

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

9.13 aerodynamics_messages.cc File Reference

Implement aerodynamics_messages.

```
#include "../include/aerodynamics_messages.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

Macros

- #define [PATH](#) "interactions/aerodynamics/"

9.13.1 Detailed Description

Implement aerodynamics_messages.

Definition in file [aerodynamics_messages.cc](#).

9.14 aerodynamics_messages.hh File Reference

Aerodynamics message for message handling.

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

Data Structures

- class [jeod::AerodynamicsMessages](#)
Messages associated with use of the aerodynamics model.

Namespaces

- [jeod](#)
Namespace jeod.

9.14.1 Detailed Description

Aerodynamics message for message handling.

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

9.15 class_declarations.hh File Reference

Forward declaration of classes defined in the aerodynamics package.

Namespaces

- [jeod](#)
Namespace jeod.

9.15.1 Detailed Description

Forward declaration of classes defined in the aerodynamics package.

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

9.16 default_aero.cc File Reference

Implement a virtual base class for aerodynamic drag calculations.

```
#include "utils/math/include/vector3.hh"
#include "../include/default_aero.hh"
#include "../include/aerodynamics_messages.hh"
#include "../include/aero_drag.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.16.1 Detailed Description

Implement a virtual base class for aerodynamic drag calculations. Also implement a specific version, in the base class, that contains ballistic coefficient and coefficient of drag options

Definition in file [default_aero.cc](#).

9.17 default_aero.hh File Reference

An implementation of ballistic coefficient and coefficient of drag for use in the AerodynamicDrag object.

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

Data Structures

- class [jeod::DefaultAero](#)

The simple, default, aerodynamic drag model, including coefficient of drag, ballistic coefficient, etc.

Namespaces

- [jeod](#)

Namespace jeod.

9.17.1 Detailed Description

An implementation of ballistic coefficient and coefficient of drag for use in the AerodynamicDrag object. This class can be inherited from and overridden for use with the AerodynamicDrag object.

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

9.18 flat_plate_aero_facet.cc File Reference

Individual facets for use with aero environment interaction models.

```
#include <cstdint>
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/surface_model/include/facet.hh"
#include "../include/flat_plate_aero_facet.hh"
#include "../include/aerodynamics_messages.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.18.1 Detailed Description

Individual facets for use with aero environment interaction models.

Definition in file [flat_plate_aero_facet.cc](#).

9.19 flat_plate_aero_facet.hh File Reference

The aerodynamic specific implementation of flat plate.

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

Data Structures

- class [jeod::FlatPlateAeroFacet](#)
The aerodynamic specific version of a flat plate.

Namespaces

- [jeod](#)
Namespace jeod.

9.19.1 Detailed Description

The aerodynamic specific implementation of flat plate.

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

9.20 flat_plate_aero_factory.cc File Reference

Factory that creates a FlatPlateAeroFacet from a FlatPlate, using a FlatPlateAeroParams object.

```
#include <cstdlib>
#include <typeinfo>
#include "utils/math/include/vector3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/surface_model/include/flat_plate.hh"
#include "utils/surface_model/include/facet.hh"
#include "utils/surface_model/include/interaction_facet.hh"
#include "../include/flat_plate_aero_factory.hh"
#include "../include/flat_plate_aero_params.hh"
#include "../include/flat_plate_aero_facet.hh"
#include "../include/aerodynamics_messages.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.20.1 Detailed Description

Factory that creates a FlatPlateAeroFacet from a FlatPlate, using a FlatPlateAeroParams object.

Definition in file [flat_plate_aero_factory.cc](#).

9.21 flat_plate_aero_factory.hh File Reference

Creates a flat plate aero facet from a basic flat plate facet.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/surface_model/include/interaction_facet_factory.hh"
#include "flat_plate_aero_facet.hh"
#include "utils/surface_model/include/facet.hh"
```

Data Structures

- class [jeod::FlatPlateAeroFactory](#)
Creates a [FlatPlateAeroFacet](#) from a [FlatPlate](#).

Namespaces

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

9.21.1 Detailed Description

Creates a flat plate aero facet from a basic flat plate facet.

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

9.22 flat_plate_aero_params.cc File Reference

Flat plates aerodynamic parameters for use in the surface model.

```
#include "../include/flat_plate_aero_params.hh"
```

Namespaces

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

9.22.1 Detailed Description

Flat plates aerodynamic parameters for use in the surface model.

Definition in file [flat_plate_aero_params.cc](#).

9.23 flat_plate_aero_params.hh File Reference

The set of parameters used to create an aerodynamic interaction specific flat plate facet from a general flat plate facet.

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

Data Structures

- class [jeod::FlatPlateAeroParams](#)
used in the [FlatPlateAeroFactory](#) to create a [FlatPlateAeroFacet](#).

Namespaces

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

9.23.1 Detailed Description

The set of parameters used to create an aerodynamic interaction specific flat plate facet from a general flat plate facet.

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

9.24 flat_plate_thermal_aero_factory.cc File Reference

Factory that creates a FlatPlateAeroFacet from a FlatPlateThermal, using a FlatPlateAeroParams object.

```
#include <typeinfo>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/surface_model/include/flat_plate_thermal.hh"
#include "../include/flat_plate_thermal_aero_factory.hh"
#include "../include/flat_plate_aero_params.hh"
#include "../include/flat_plate_aero_facet.hh"
#include "../include/aerodynamics_messages.hh"
```

Namespaces

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

9.24.1 Detailed Description

Factory that creates a FlatPlateAeroFacet from a FlatPlateThermal, using a FlatPlateAeroParams object.

Definition in file [flat_plate_thermal_aero_factory.cc](#).

9.25 flat_plate_thermal_aero_factory.hh File Reference

Creates a flat plate aero facet from a flat plate thermal facet.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/surface_model/include/facet.hh"
#include "utils/surface_model/include/interaction_facet.hh"
#include "utils/surface_model/include/interaction_facet_factory.hh"
#include "flat_plate_aero_factory.hh"
```

Data Structures

- class [jeod::FlatPlateThermalAeroFactory](#)
Creates a [FlatPlateAeroFacet](#) from a [FlatPlate](#).

Namespaces

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

9.25.1 Detailed Description

Creates a flat plate aero facet from a flat plate thermal facet.

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

Index

- ~AeroFacet
 - jeod::AeroFacet, [27](#)
- ~AeroParams
 - jeod::AeroParams, [29](#)
- ~AeroSurface
 - jeod::AeroSurface, [30](#)
- ~AeroSurfaceFactory
 - jeod::AeroSurfaceFactory, [33](#)
- ~AerodynamicDrag
 - jeod::AerodynamicDrag, [20](#)
- ~DefaultAero
 - jeod::DefaultAero, [35](#)
- ~FlatPlateAeroFacet
 - jeod::FlatPlateAeroFacet, [38](#)
- ~FlatPlateAeroFactory
 - jeod::FlatPlateAeroFactory, [43](#)
- ~FlatPlateAeroParams
 - jeod::FlatPlateAeroParams, [45](#)
- ~FlatPlateThermalAeroFactory
 - jeod::FlatPlateThermalAeroFactory, [48](#)
- active
 - jeod::AerodynamicDrag, [22](#)
- add_facet_params
 - jeod::AeroSurfaceFactory, [33](#)
- aero_drag
 - jeod::AerodynamicDrag, [20](#)
- aero_drag.cc, [51](#)
- aero_drag.hh, [51](#)
- aero_facet.cc, [52](#)
- aero_facet.hh, [52](#)
- aero_facets
 - jeod::AeroSurface, [31](#)
- aero_force
 - jeod::AerodynamicDrag, [22](#)
- aero_model.cc, [53](#)
 - JEOD_FRIEND_CLASS, [53](#)
- aero_model.hh, [53](#)
- aero_params.cc, [53](#)
- aero_params.hh, [54](#)
- aero_surface.cc, [54](#)
- aero_surface.hh, [55](#)
- aero_surface_factory.cc, [55](#)
- aero_surface_factory.hh, [56](#)
- aero_surface_ptr
 - jeod::AerodynamicDrag, [22](#)
- aero_torque
 - jeod::AerodynamicDrag, [22](#)
- AeroFacet
 - jeod::AeroFacet, [27](#)
- AeroParams
 - jeod::AeroParams, [28](#), [29](#)
- AeroSurface
 - jeod::AeroSurface, [30](#)
- AeroSurfaceFactory
 - jeod::AeroSurfaceFactory, [33](#)
- aerodrag_force
 - jeod::AeroFacet, [27](#)
 - jeod::DefaultAero, [35](#)
 - jeod::FlatPlateAeroFacet, [39](#)
- AerodynamicDrag
 - jeod::AerodynamicDrag, [20](#)
- Aerodynamics, [13](#)
 - PATH, [14](#)
- aerodynamics_messages.cc, [56](#)
- aerodynamics_messages.hh, [57](#)
- AerodynamicsMessages
 - jeod::AerodynamicsMessages, [25](#)
- allocate_array
 - jeod::AeroSurface, [30](#)
- allocate_interaction_facet
 - jeod::AeroSurface, [30](#)
- area
 - jeod::DefaultAero, [36](#)
 - jeod::FlatPlateAeroFacet, [39](#)
- BC
 - jeod::DefaultAero, [36](#)
- ballistic_drag
 - jeod::AerodynamicDrag, [22](#)
- Calc_coef
 - jeod::AeroDragEnum, [17](#)
- calculate_drag_coef
 - jeod::FlatPlateAeroFacet, [39](#)
 - jeod::FlatPlateAeroParams, [46](#)
- Cd
 - jeod::DefaultAero, [36](#)
- center_pressure
 - jeod::FlatPlateAeroFacet, [40](#)
- class_declarations.hh, [57](#)
- clear_aero_surface
 - jeod::AerodynamicDrag, [20](#)
- coef_method
 - jeod::FlatPlateAeroFacet, [40](#)
 - jeod::FlatPlateAeroParams, [46](#)
- CoefCalcMethod
 - jeod::AeroDragEnum, [17](#)
- constant_density
 - jeod::AerodynamicDrag, [22](#)

create_facet
 jeod::FlatPlateAeroFactory, 43

 DRAG_OPT_BC
 jeod::DefaultAero, 35
 DRAG_OPT_CD
 jeod::DefaultAero, 35
 DRAG_OPT_CONST
 jeod::DefaultAero, 35
 default_aero.cc, 57
 default_aero.hh, 58
 default_behavior
 jeod::AerodynamicDrag, 23
 DefaultAero
 jeod::DefaultAero, 35
 density
 jeod::AerodynamicDrag, 23
 Diffuse
 jeod::AeroDragEnum, 17
 drag
 jeod::DefaultAero, 37
 drag_coef_diff
 jeod::FlatPlateAeroFacet, 40
 jeod::FlatPlateAeroParams, 46
 drag_coef_norm
 jeod::FlatPlateAeroFacet, 40
 jeod::FlatPlateAeroParams, 46
 drag_coef_spec
 jeod::FlatPlateAeroFacet, 40
 jeod::FlatPlateAeroParams, 46
 drag_coef_tang
 jeod::FlatPlateAeroFacet, 40
 jeod::FlatPlateAeroParams, 46
 DragOption
 jeod::DefaultAero, 35
 dynamic_pressure
 jeod::AeroDragParameters, 18

 epsilon
 jeod::FlatPlateAeroFacet, 41
 jeod::FlatPlateAeroParams, 47

 facets_size
 jeod::AeroSurface, 31
 flat_plate_aero_facet.cc, 58
 flat_plate_aero_facet.hh, 59
 flat_plate_aero_factory
 jeod::AeroSurfaceFactory, 33
 flat_plate_aero_factory.cc, 59
 flat_plate_aero_factory.hh, 60
 flat_plate_aero_params.cc, 60
 flat_plate_aero_params.hh, 61
 flat_plate_thermal_aero_factory
 jeod::AeroSurfaceFactory, 33
 flat_plate_thermal_aero_factory.cc, 61
 flat_plate_thermal_aero_factory.hh, 62
 FlatPlateAeroFacet
 jeod::FlatPlateAeroFacet, 38, 39
 FlatPlateAeroFactory
 jeod::FlatPlateAeroFactory, 43
 FlatPlateAeroParams
 jeod::FlatPlateAeroParams, 45
 FlatPlateThermalAeroFactory
 jeod::FlatPlateThermalAeroFactory, 48
 force_n
 jeod::FlatPlateAeroFacet, 41
 force_t
 jeod::FlatPlateAeroFacet, 41

 gas_const
 jeod::AeroDragParameters, 18

 init_attrjeod__AeroFacet
 jeod::AeroFacet, 27
 init_attrjeod__AeroParams
 jeod::AeroParams, 29
 init_attrjeod__AeroSurface
 jeod::AeroSurface, 31
 init_attrjeod__AeroSurfaceFactory
 jeod::AeroSurfaceFactory, 33
 init_attrjeod__AerodynamicsMessages
 jeod::AerodynamicsMessages, 25
 init_attrjeod__DefaultAero
 jeod::DefaultAero, 36
 init_attrjeod__FlatPlateAeroFacet
 jeod::FlatPlateAeroFacet, 39
 init_attrjeod__FlatPlateAeroFactory
 jeod::FlatPlateAeroFactory, 44
 init_attrjeod__FlatPlateAeroParams
 jeod::FlatPlateAeroParams, 45
 init_attrjeod__FlatPlateThermalAeroFactory
 jeod::FlatPlateThermalAeroFactory, 48
 initialization_error
 jeod::AerodynamicsMessages, 25
 initialize
 jeod::AerodynamicDrag_aero_model_default_data, 24
 InputProcessor
 jeod::AerodynamicsMessages, 25
 jeod::AeroFacet, 27
 jeod::AeroParams, 29
 jeod::AeroSurface, 31
 jeod::AeroSurfaceFactory, 33
 jeod::DefaultAero, 36
 jeod::FlatPlateAeroFacet, 39
 jeod::FlatPlateAeroFactory, 44
 jeod::FlatPlateAeroParams, 45
 jeod::FlatPlateThermalAeroFactory, 48
 Interactions, 12
 is_correct_factory
 jeod::FlatPlateAeroFactory, 43
 jeod::FlatPlateThermalAeroFactory, 48

 JEOD_FRIEND_CLASS
 aero_model.cc, 53
 jeod, 15
 jeod::AeroDragEnum
 Calc_coef, 17

- Diffuse, [17](#)
- Mixed, [17](#)
- Specular, [17](#)
- jeod::DefaultAero
 - DRAG_OPT_BC, [35](#)
 - DRAG_OPT_CD, [35](#)
 - DRAG_OPT_CONST, [35](#)
- jeod::AeroDragEnum, [17](#)
 - CoefCalcMethod, [17](#)
- jeod::AeroDragParameters, [18](#)
 - dynamic_pressure, [18](#)
 - gas_const, [18](#)
 - temp_free_stream, [18](#)
- jeod::AeroFacet, [26](#)
 - ~AeroFacet, [27](#)
 - AeroFacet, [27](#)
 - aerodrag_force, [27](#)
 - init_attrjeod__AeroFacet, [27](#)
 - InputProcessor, [27](#)
 - operator=, [27](#)
- jeod::AeroParams, [28](#)
 - ~AeroParams, [29](#)
 - AeroParams, [28](#), [29](#)
 - init_attrjeod__AeroParams, [29](#)
 - InputProcessor, [29](#)
 - operator=, [29](#)
- jeod::AeroSurface, [29](#)
 - ~AeroSurface, [30](#)
 - aero_facets, [31](#)
 - AeroSurface, [30](#)
 - allocate_array, [30](#)
 - allocate_interaction_facet, [30](#)
 - facets_size, [31](#)
 - init_attrjeod__AeroSurface, [31](#)
 - InputProcessor, [31](#)
 - operator=, [31](#)
- jeod::AeroSurfaceFactory, [32](#)
 - ~AeroSurfaceFactory, [33](#)
 - add_facet_params, [33](#)
 - AeroSurfaceFactory, [33](#)
 - flat_plate_aero_factory, [33](#)
 - flat_plate_thermal_aero_factory, [33](#)
 - init_attrjeod__AeroSurfaceFactory, [33](#)
 - InputProcessor, [33](#)
 - operator=, [33](#)
- jeod::AerodynamicDrag, [19](#)
 - ~AerodynamicDrag, [20](#)
 - active, [22](#)
 - aero_drag, [20](#)
 - aero_force, [22](#)
 - aero_surface_ptr, [22](#)
 - aero_torque, [22](#)
 - AerodynamicDrag, [20](#)
 - ballistic_drag, [22](#)
 - clear_aero_surface, [20](#)
 - constant_density, [22](#)
 - default_behavior, [23](#)
 - density, [23](#)
 - operator=, [20](#)
 - param, [23](#)
 - set_aero_surface, [21](#)
 - use_default_behavior, [23](#)
- jeod::AerodynamicDrag_aero_model_default_data, [23](#)
 - initialize, [24](#)
- jeod::AerodynamicsMessages, [24](#)
 - AerodynamicsMessages, [25](#)
 - init_attrjeod__AerodynamicsMessages, [25](#)
 - initialization_error, [25](#)
 - InputProcessor, [25](#)
 - operator=, [25](#)
 - pre_initialization_error, [25](#)
 - runtime_error, [25](#)
 - runtime_warns, [26](#)
- jeod::DefaultAero, [34](#)
 - ~DefaultAero, [35](#)
 - aerodrag_force, [35](#)
 - area, [36](#)
 - BC, [36](#)
 - Cd, [36](#)
 - DefaultAero, [35](#)
 - drag, [37](#)
 - DragOption, [35](#)
 - init_attrjeod__DefaultAero, [36](#)
 - InputProcessor, [36](#)
 - operator=, [36](#)
 - option, [37](#)
- jeod::FlatPlateAeroFacet, [37](#)
 - ~FlatPlateAeroFacet, [38](#)
 - aerodrag_force, [39](#)
 - area, [39](#)
 - calculate_drag_coef, [39](#)
 - center_pressure, [40](#)
 - coef_method, [40](#)
 - drag_coef_diff, [40](#)
 - drag_coef_norm, [40](#)
 - drag_coef_spec, [40](#)
 - drag_coef_tang, [40](#)
 - epsilon, [41](#)
 - FlatPlateAeroFacet, [38](#), [39](#)
 - force_n, [41](#)
 - force_t, [41](#)
 - init_attrjeod__FlatPlateAeroFacet, [39](#)
 - InputProcessor, [39](#)
 - normal, [41](#)
 - operator=, [39](#)
 - temp_reflect, [41](#)
 - temperature, [41](#)
- jeod::FlatPlateAeroFactory, [42](#)
 - ~FlatPlateAeroFactory, [43](#)
 - create_facet, [43](#)
 - FlatPlateAeroFactory, [43](#)
 - init_attrjeod__FlatPlateAeroFactory, [44](#)
 - InputProcessor, [44](#)
 - is_correct_factory, [43](#)
 - operator=, [44](#)
- jeod::FlatPlateAeroParams, [44](#)

- ~FlatPlateAeroParams, 45
 - calculate_drag_coef, 46
 - coef_method, 46
 - drag_coef_diff, 46
 - drag_coef_norm, 46
 - drag_coef_spec, 46
 - drag_coef_tang, 46
 - epsilon, 47
 - FlatPlateAeroParams, 45
 - init_attrjeod__FlatPlateAeroParams, 45
 - InputProcessor, 45
 - operator=, 45
 - temp_reflect, 47
- jeod::FlatPlateThermalAeroFactory, 47
 - ~FlatPlateThermalAeroFactory, 48
 - FlatPlateThermalAeroFactory, 48
 - init_attrjeod__FlatPlateThermalAeroFactory, 48
 - InputProcessor, 48
 - is_correct_factory, 48
 - operator=, 48
- Mixed
 - jeod::AeroDragEnum, 17
- Models, 11
- normal
 - jeod::FlatPlateAeroFacet, 41
- operator=
 - jeod::AerodynamicDrag, 20
 - jeod::AerodynamicsMessages, 25
 - jeod::AeroFacet, 27
 - jeod::AeroParams, 29
 - jeod::AeroSurface, 31
 - jeod::AeroSurfaceFactory, 33
 - jeod::DefaultAero, 36
 - jeod::FlatPlateAeroFacet, 39
 - jeod::FlatPlateAeroFactory, 44
 - jeod::FlatPlateAeroParams, 45
 - jeod::FlatPlateThermalAeroFactory, 48
- option
 - jeod::DefaultAero, 37
- PATH
 - Aerodynamics, 14
- param
 - jeod::AerodynamicDrag, 23
- pre_initialization_error
 - jeod::AerodynamicsMessages, 25
- runtime_error
 - jeod::AerodynamicsMessages, 25
- runtime_warns
 - jeod::AerodynamicsMessages, 26
- set_aero_surface
 - jeod::AerodynamicDrag, 21
- Specular
 - jeod::AeroDragEnum, 17
- temp_free_stream
 - jeod::AeroDragParameters, 18
- temp_reflect
 - jeod::FlatPlateAeroFacet, 41
 - jeod::FlatPlateAeroParams, 47
- temperature
 - jeod::FlatPlateAeroFacet, 41
- use_default_behavior
 - jeod::AerodynamicDrag, 23