

SurfaceModel

5.0

Generated by Doxygen 1.8.5

Wed Jun 1 2022 12:07:09



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>11</b>
6.1	Models . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.2	Utils . . . . .	12
6.2.1	Detailed Description . . . . .	12
6.3	SurfaceModel . . . . .	13
6.3.1	Detailed Description . . . . .	14
6.3.2	Macro Definition Documentation . . . . .	14
6.3.2.1	PATH . . . . .	14
<b>7</b>	<b>Namespace Documentation</b>	<b>15</b>
7.1	jeod Namespace Reference . . . . .	15
7.1.1	Detailed Description . . . . .	15
<b>8</b>	<b>Data Structure Documentation</b>	<b>17</b>
8.1	jeod::Cylinder Class Reference . . . . .	17
8.1.1	Detailed Description . . . . .	18
8.1.2	Constructor & Destructor Documentation . . . . .	18
8.1.2.1	Cylinder . . . . .	18

8.1.2.2	<a href="#">~Cylinder</a>	18
8.1.2.3	<a href="#">Cylinder</a>	18
8.1.3	<a href="#">Member Function Documentation</a>	18
8.1.3.1	<a href="#">operator=</a>	18
8.1.4	<a href="#">Friends And Related Function Documentation</a>	18
8.1.4.1	<a href="#">init_attrjeod__Cylinder</a>	18
8.1.4.2	<a href="#">InputProcessor</a>	18
8.1.5	<a href="#">Field Documentation</a>	18
8.1.5.1	<a href="#">length</a>	18
8.2	<a href="#">jeod::Facet Class Reference</a>	19
8.2.1	<a href="#">Detailed Description</a>	20
8.2.2	<a href="#">Constructor &amp; Destructor Documentation</a>	20
8.2.2.1	<a href="#">Facet</a>	20
8.2.2.2	<a href="#">~Facet</a>	20
8.2.2.3	<a href="#">Facet</a>	20
8.2.3	<a href="#">Member Function Documentation</a>	20
8.2.3.1	<a href="#">get_mass_body_ptr</a>	20
8.2.3.2	<a href="#">get_mass_rel_struct</a>	21
8.2.3.3	<a href="#">initialize_mass_connection</a>	21
8.2.3.4	<a href="#">operator=</a>	21
8.2.3.5	<a href="#">set_name</a>	21
8.2.3.6	<a href="#">update_articulation</a>	21
8.2.3.7	<a href="#">update_articulation_internal</a>	21
8.2.4	<a href="#">Friends And Related Function Documentation</a>	21
8.2.4.1	<a href="#">init_attrjeod__Facet</a>	21
8.2.4.2	<a href="#">InputProcessor</a>	21
8.2.4.3	<a href="#">SurfaceModel</a>	21
8.2.5	<a href="#">Field Documentation</a>	21
8.2.5.1	<a href="#">area</a>	22
8.2.5.2	<a href="#">connections_initialized</a>	22
8.2.5.3	<a href="#">int_pos</a>	22
8.2.5.4	<a href="#">local_position</a>	22
8.2.5.5	<a href="#">mass_body_name</a>	22
8.2.5.6	<a href="#">mass_body_ptr</a>	22
8.2.5.7	<a href="#">mass_rel_struct</a>	23
8.2.5.8	<a href="#">name</a>	23
8.2.5.9	<a href="#">param_name</a>	23
8.2.5.10	<a href="#">position</a>	23
8.2.5.11	<a href="#">temperature</a>	23
8.3	<a href="#">jeod::FacetParams Class Reference</a>	23

8.3.1	Detailed Description	24
8.3.2	Constructor & Destructor Documentation	24
8.3.2.1	FacetParams	24
8.3.2.2	~FacetParams	24
8.3.2.3	FacetParams	24
8.3.3	Member Function Documentation	24
8.3.3.1	operator=	24
8.3.3.2	set_name	24
8.3.4	Friends And Related Function Documentation	25
8.3.4.1	init_attrjeod__FacetParams	25
8.3.4.2	InputProcessor	25
8.3.5	Field Documentation	25
8.3.5.1	name	25
8.4	jeod::FacetStateInfo Struct Reference	25
8.4.1	Detailed Description	26
8.4.2	Constructor & Destructor Documentation	26
8.4.2.1	FacetStateInfo	26
8.4.2.2	FacetStateInfo	26
8.4.2.3	FacetStateInfo	26
8.4.3	Member Function Documentation	26
8.4.3.1	operator==	26
8.4.4	Friends And Related Function Documentation	27
8.4.4.1	init_attrjeod__FacetStateInfo	27
8.4.4.2	InputProcessor	27
8.4.5	Field Documentation	27
8.4.5.1	mass_body	27
8.4.5.2	mass_state	27
8.5	jeod::FlatPlate Class Reference	27
8.5.1	Detailed Description	28
8.5.2	Constructor & Destructor Documentation	28
8.5.2.1	FlatPlate	28
8.5.2.2	~FlatPlate	28
8.5.2.3	FlatPlate	29
8.5.3	Member Function Documentation	29
8.5.3.1	operator=	29
8.5.3.2	update_articulation_internal	29
8.5.4	Friends And Related Function Documentation	29
8.5.4.1	init_attrjeod__FlatPlate	29
8.5.4.2	InputProcessor	29
8.5.5	Field Documentation	29

8.5.5.1	local_normal	29
8.5.5.2	normal	29
8.6	jeod::FlatPlateCircular Class Reference	29
8.6.1	Detailed Description	30
8.6.2	Constructor & Destructor Documentation	30
8.6.2.1	FlatPlateCircular	30
8.6.2.2	~FlatPlateCircular	31
8.6.2.3	FlatPlateCircular	31
8.6.3	Member Function Documentation	31
8.6.3.1	operator=	31
8.6.4	Friends And Related Function Documentation	31
8.6.4.1	init_attrjeod__FlatPlateCircular	31
8.6.4.2	InputProcessor	31
8.6.5	Field Documentation	31
8.6.5.1	radius	31
8.7	jeod::FlatPlateThermal Class Reference	31
8.7.1	Detailed Description	32
8.7.2	Constructor & Destructor Documentation	32
8.7.2.1	FlatPlateThermal	32
8.7.2.2	~FlatPlateThermal	32
8.7.2.3	FlatPlateThermal	32
8.7.3	Member Function Documentation	32
8.7.3.1	operator=	32
8.7.4	Friends And Related Function Documentation	32
8.7.4.1	init_attrjeod__FlatPlateThermal	32
8.7.4.2	InputProcessor	32
8.7.5	Field Documentation	33
8.7.5.1	thermal	33
8.8	jeod::InteractionFacet Class Reference	33
8.8.1	Detailed Description	33
8.8.2	Constructor & Destructor Documentation	34
8.8.2.1	InteractionFacet	34
8.8.2.2	~InteractionFacet	34
8.8.2.3	InteractionFacet	34
8.8.3	Member Function Documentation	34
8.8.3.1	operator=	34
8.8.4	Friends And Related Function Documentation	34
8.8.4.1	init_attrjeod__InteractionFacet	34
8.8.4.2	InputProcessor	34
8.8.5	Field Documentation	34

8.8.5.1	<a href="#">base_facet</a>	34
8.8.5.2	<a href="#">force</a>	34
8.8.5.3	<a href="#">torque</a>	34
8.9	<a href="#">jeod::InteractionFacetFactory Class Reference</a>	35
8.9.1	<a href="#">Detailed Description</a>	35
8.9.2	<a href="#">Constructor &amp; Destructor Documentation</a>	35
8.9.2.1	<a href="#">InteractionFacetFactory</a>	35
8.9.2.2	<a href="#">~InteractionFacetFactory</a>	36
8.9.2.3	<a href="#">InteractionFacetFactory</a>	36
8.9.3	<a href="#">Member Function Documentation</a>	36
8.9.3.1	<a href="#">create_facet</a>	36
8.9.3.2	<a href="#">is_correct_factory</a>	36
8.9.3.3	<a href="#">operator=</a>	36
8.9.4	<a href="#">Friends And Related Function Documentation</a>	36
8.9.4.1	<a href="#">init_attrjeod__InteractionFacetFactory</a>	36
8.9.4.2	<a href="#">InputProcessor</a>	36
8.9.5	<a href="#">Field Documentation</a>	36
8.9.5.1	<a href="#">trick_bool</a>	36
8.10	<a href="#">jeod::InteractionSurface Class Reference</a>	37
8.10.1	<a href="#">Detailed Description</a>	37
8.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	37
8.10.2.1	<a href="#">InteractionSurface</a>	37
8.10.2.2	<a href="#">~InteractionSurface</a>	38
8.10.2.3	<a href="#">InteractionSurface</a>	38
8.10.3	<a href="#">Member Function Documentation</a>	38
8.10.3.1	<a href="#">accumulate_thermal_sources</a>	38
8.10.3.2	<a href="#">allocate_array</a>	38
8.10.3.3	<a href="#">allocate_interaction_facet</a>	38
8.10.3.4	<a href="#">operator=</a>	38
8.10.3.5	<a href="#">thermal_integrator</a>	38
8.10.4	<a href="#">Friends And Related Function Documentation</a>	38
8.10.4.1	<a href="#">init_attrjeod__InteractionSurface</a>	38
8.10.4.2	<a href="#">InputProcessor</a>	38
8.11	<a href="#">jeod::InteractionSurfaceFactory Class Reference</a>	39
8.11.1	<a href="#">Detailed Description</a>	39
8.11.2	<a href="#">Constructor &amp; Destructor Documentation</a>	40
8.11.2.1	<a href="#">InteractionSurfaceFactory</a>	40
8.11.2.2	<a href="#">~InteractionSurfaceFactory</a>	40
8.11.2.3	<a href="#">InteractionSurfaceFactory</a>	40
8.11.3	<a href="#">Member Function Documentation</a>	40

8.11.3.1	<a href="#">add_facet_factory</a>	40
8.11.3.2	<a href="#">add_facet_params</a>	40
8.11.3.3	<a href="#">create_surface</a>	40
8.11.3.4	<a href="#">create_surface</a>	41
8.11.3.5	<a href="#">operator=</a>	41
8.11.4	<a href="#">Friends And Related Function Documentation</a>	41
8.11.4.1	<a href="#">init_attrjeod__InteractionSurfaceFactory</a>	41
8.11.4.2	<a href="#">InputProcessor</a>	41
8.11.5	<a href="#">Field Documentation</a>	41
8.11.5.1	<a href="#">factories</a>	41
8.11.5.2	<a href="#">params</a>	41
8.12	<a href="#">jeod::SurfaceModel Class Reference</a>	42
8.12.1	<a href="#">Detailed Description</a>	42
8.12.2	<a href="#">Constructor &amp; Destructor Documentation</a>	43
8.12.2.1	<a href="#">SurfaceModel</a>	43
8.12.2.2	<a href="#">~SurfaceModel</a>	43
8.12.2.3	<a href="#">SurfaceModel</a>	43
8.12.3	<a href="#">Member Function Documentation</a>	43
8.12.3.1	<a href="#">add_facet</a>	43
8.12.3.2	<a href="#">add_facets</a>	43
8.12.3.3	<a href="#">initialize_mass_connections</a>	43
8.12.3.4	<a href="#">operator=</a>	43
8.12.3.5	<a href="#">update_articulation</a>	43
8.12.4	<a href="#">Friends And Related Function Documentation</a>	43
8.12.4.1	<a href="#">init_attrjeod__SurfaceModel</a>	43
8.12.4.2	<a href="#">InputProcessor</a>	43
8.12.5	<a href="#">Field Documentation</a>	44
8.12.5.1	<a href="#">articulation_active</a>	44
8.12.5.2	<a href="#">articulation_states</a>	44
8.12.5.3	<a href="#">facets</a>	44
8.12.5.4	<a href="#">struct_body_name</a>	44
8.12.5.5	<a href="#">struct_body_ptr</a>	44
8.13	<a href="#">jeod::SurfaceModelMessages Class Reference</a>	45
8.13.1	<a href="#">Detailed Description</a>	45
8.13.2	<a href="#">Constructor &amp; Destructor Documentation</a>	45
8.13.2.1	<a href="#">SurfaceModelMessages</a>	45
8.13.2.2	<a href="#">SurfaceModelMessages</a>	45
8.13.3	<a href="#">Member Function Documentation</a>	45
8.13.3.1	<a href="#">operator=</a>	45
8.13.4	<a href="#">Friends And Related Function Documentation</a>	45



8.13.4.1	<a href="#">init_attrjeod__SurfaceModelMessages</a>	45
8.13.4.2	<a href="#">InputProcessor</a>	45
8.13.5	<a href="#">Field Documentation</a>	46
8.13.5.1	<a href="#">initialization_error</a>	46
8.13.5.2	<a href="#">runtime_error</a>	46
8.13.5.3	<a href="#">setup_error</a>	46
<b>9</b>	<b><a href="#">File Documentation</a></b>	<b>47</b>
9.1	<a href="#">class_declarations.hh File Reference</a>	47
9.1.1	<a href="#">Detailed Description</a>	47
9.2	<a href="#">cylinder.cc File Reference</a>	47
9.2.1	<a href="#">Detailed Description</a>	47
9.3	<a href="#">cylinder.hh File Reference</a>	48
9.3.1	<a href="#">Detailed Description</a>	48
9.4	<a href="#">facet.cc File Reference</a>	48
9.4.1	<a href="#">Detailed Description</a>	48
9.5	<a href="#">facet.hh File Reference</a>	49
9.5.1	<a href="#">Detailed Description</a>	49
9.6	<a href="#">facet_params.cc File Reference</a>	49
9.6.1	<a href="#">Detailed Description</a>	49
9.7	<a href="#">facet_params.hh File Reference</a>	49
9.7.1	<a href="#">Detailed Description</a>	50
9.8	<a href="#">flat_plate.cc File Reference</a>	50
9.8.1	<a href="#">Detailed Description</a>	50
9.9	<a href="#">flat_plate.hh File Reference</a>	50
9.9.1	<a href="#">Detailed Description</a>	51
9.10	<a href="#">flat_plate_circular.cc File Reference</a>	51
9.10.1	<a href="#">Detailed Description</a>	51
9.11	<a href="#">flat_plate_circular.hh File Reference</a>	51
9.11.1	<a href="#">Detailed Description</a>	51
9.12	<a href="#">flat_plate_thermal.cc File Reference</a>	52
9.12.1	<a href="#">Detailed Description</a>	52
9.13	<a href="#">flat_plate_thermal.hh File Reference</a>	52
9.13.1	<a href="#">Detailed Description</a>	52
9.14	<a href="#">interaction_facet.cc File Reference</a>	52
9.14.1	<a href="#">Detailed Description</a>	53
9.15	<a href="#">interaction_facet.hh File Reference</a>	53
9.15.1	<a href="#">Detailed Description</a>	53
9.16	<a href="#">interaction_facet_factory.cc File Reference</a>	53
9.16.1	<a href="#">Detailed Description</a>	53

9.17	<a href="#">interaction_facet_factory.hh File Reference</a>	54
9.17.1	<a href="#">Detailed Description</a>	54
9.18	<a href="#">interaction_surface.cc File Reference</a>	54
9.18.1	<a href="#">Detailed Description</a>	54
9.19	<a href="#">interaction_surface.hh File Reference</a>	54
9.19.1	<a href="#">Detailed Description</a>	55
9.20	<a href="#">interaction_surface_factory.cc File Reference</a>	55
9.20.1	<a href="#">Detailed Description</a>	55
9.21	<a href="#">interaction_surface_factory.hh File Reference</a>	55
9.21.1	<a href="#">Detailed Description</a>	56
9.22	<a href="#">surface_model.cc File Reference</a>	56
9.22.1	<a href="#">Detailed Description</a>	56
9.23	<a href="#">surface_model.hh File Reference</a>	56
9.23.1	<a href="#">Detailed Description</a>	57
9.24	<a href="#">surface_model_messages.cc File Reference</a>	57
9.24.1	<a href="#">Detailed Description</a>	57
9.25	<a href="#">surface_model_messages.hh File Reference</a>	57
9.25.1	<a href="#">Detailed Description</a>	58

## Index

59

# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Models . . . . .	11
Utils . . . . .	12
SurfaceModel . . . . .	13



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">jeod</a>	Namespace jeod . . . . .	15
----------------------	--------------------------	----



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

jeod::Facet . . . . .	19
jeod::FlatPlate . . . . .	27
jeod::FlatPlateCircular . . . . .	29
jeod::Cylinder . . . . .	17
jeod::FlatPlateThermal . . . . .	31
jeod::FacetParams . . . . .	23
jeod::FacetStateInfo . . . . .	25
jeod::InteractionFacet . . . . .	33
jeod::InteractionFacetFactory . . . . .	35
jeod::InteractionSurface . . . . .	37
jeod::InteractionSurfaceFactory . . . . .	39
jeod::SurfaceModel . . . . .	42
jeod::SurfaceModelMessages . . . . .	45





## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::Cylinder</a>	
An cylinder implementation of <a href="#">Facet</a>	17
<a href="#">jeod::Facet</a>	
A general base class for all surface model facets	19
<a href="#">jeod::FacetParams</a>	
General base class for all parameters associated with facets in the surface model	23
<a href="#">jeod::FacetStateInfo</a>	
This is a structure used only in the surface model to aid in relative state calculations for articulation	25
<a href="#">jeod::FlatPlate</a>	
A <a href="#">FlatPlate</a> implementation of <a href="#">Facet</a>	27
<a href="#">jeod::FlatPlateCircular</a>	
An circular flat plate implementation of <a href="#">Facet</a>	29
<a href="#">jeod::FlatPlateThermal</a>	
A <a href="#">FlatPlate</a> implementation of <a href="#">Facet</a> , with thermal information	31
<a href="#">jeod::InteractionFacet</a>	
A base class for an interaction specific facet	33
<a href="#">jeod::InteractionFacetFactory</a>	
A factory to create a specific interaction facet from a general facet	35
<a href="#">jeod::InteractionSurface</a>	
A base class for interaction specific surfaces	37
<a href="#">jeod::InteractionSurfaceFactory</a>	
A base class for creating specific interaction surfaces from general surfaces	39
<a href="#">jeod::SurfaceModel</a>	
A general, non-interaction specific surface that can be used to create surfaces suitable for specific interactions	42
<a href="#">jeod::SurfaceModelMessages</a>	
Messages associated with the use of the surface model	45



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">class_declarations.hh</a>	Forward declarations of classes defined for JEOD 2.0 surface model . . . . .	47
<a href="#">cylinder.cc</a>	Cylinders for use in the surface model . . . . .	47
<a href="#">cylinder.hh</a>	Cylinders for use in the surface model and the contact model . . . . .	48
<a href="#">facet.cc</a>	Individual facets for use in the surface model . . . . .	48
<a href="#">facet.hh</a>	Individual facets for use in the surface model . . . . .	49
<a href="#">facet_params.cc</a>	A pure virtual base class for facet parameters, used to create interaction facets in the Interaction-SurfaceFactorys . . . . .	49
<a href="#">facet_params.hh</a>	A virtual base class for facet parameters, used to create interaction facets in the Interaction-SurfaceFactorys . . . . .	49
<a href="#">flat_plate.cc</a>	Flat plates for use in the surface model . . . . .	50
<a href="#">flat_plate.hh</a>	Flat plates for use in the surface model . . . . .	50
<a href="#">flat_plate_circular.cc</a>	Circular flat plates for use in the surface model . . . . .	51
<a href="#">flat_plate_circular.hh</a>	Circular flat plates for use in the surface model and the contact model . . . . .	51
<a href="#">flat_plate_thermal.cc</a>	Flat plates for use in the surface model, with the thermal rider . . . . .	52
<a href="#">flat_plate_thermal.hh</a>	Flat plates for use in the surface model, including a thermal portion . . . . .	52
<a href="#">interaction_facet.cc</a>	Individual facets for use with environment interaction models . . . . .	52
<a href="#">interaction_facet.hh</a>	Individual facets for use with specific environment interaction models . . . . .	53
<a href="#">interaction_facet_factory.cc</a>	Factory that creates an interaction facet, for a specific environment interaction model, from a facet model . . . . .	53
<a href="#">interaction_facet_factory.hh</a>	Factory that creates an interaction facet, for a specific environment interaction model, from a facet model . . . . .	54

<a href="#">interaction_surface.cc</a>	
Vehicle surface model for general environment interaction models . . . . .	54
<a href="#">interaction_surface.hh</a>	
Vehicle surface model for general environment interaction models . . . . .	54
<a href="#">interaction_surface_factory.cc</a>	
Factory that creates an interaction surface, for a specific environment interaction model, from a surface model . . . . .	55
<a href="#">interaction_surface_factory.hh</a>	
Factory that creates an interaction surface, for a specific environment interaction model, from a surface model . . . . .	55
<a href="#">surface_model.cc</a>	
Vehicle surface model for general environment interaction models . . . . .	56
<a href="#">surface_model.hh</a>	
Vehicle surface model for general environment interaction models . . . . .	56
<a href="#">surface_model_messages.cc</a>	
Implement surface_model_messages . . . . .	57
<a href="#">surface_model_messages.hh</a>	
Implement surface_model_messages . . . . .	57

## Chapter 6

# Module Documentation

### 6.1 Models

#### Modules

- [Utils](#)

#### 6.1.1 Detailed Description

## 6.2 Utils

### Modules

- [SurfaceModel](#)

### 6.2.1 Detailed Description

## 6.3 SurfaceModel

### Files

- file [class\\_declarations.hh](#)  
*Forward declarations of classes defined for JEOD 2.0 surface model.*
- file [cylinder.hh](#)  
*cylinders for use in the surface model and the contact model*
- file [facet.hh](#)  
*Individual facets for use in the surface model.*
- file [facet\\_params.hh](#)  
*A virtual base class for facet parameters, used to create interaction facets in the InteractionSurfaceFactorys.*
- file [flat\\_plate.hh](#)  
*Flat plates for use in the surface model.*
- file [flat\\_plate\\_circular.hh](#)  
*circular flat plates for use in the surface model and the contact model*
- file [flat\\_plate\\_thermal.hh](#)  
*Flat plates for use in the surface model, including a thermal portion.*
- file [interaction\\_facet.hh](#)  
*Individual facets for use with specific environment interaction models.*
- file [interaction\\_facet\\_factory.hh](#)  
*Factory that creates an interaction facet, for a specific environment interaction model, from a facet model.*
- file [interaction\\_surface.hh](#)  
*Vehicle surface model for general environment interaction models.*
- file [interaction\\_surface\\_factory.hh](#)  
*Factory that creates an interaction surface, for a specific environment interaction model, from a surface model.*
- file [surface\\_model.hh](#)  
*Vehicle surface model for general environment interaction models.*
- file [surface\\_model\\_messages.hh](#)  
*Implement surface\_model\_messages.*
- file [cylinder.cc](#)  
*cylinders for use in the surface model*
- file [facet.cc](#)  
*Individual facets for use in the surface model.*
- file [facet\\_params.cc](#)  
*A pure virtual base class for facet parameters, used to create interaction facets in the InteractionSurfaceFactorys.*
- file [flat\\_plate.cc](#)  
*Flat plates for use in the surface model.*
- file [flat\\_plate\\_circular.cc](#)  
*circular flat plates for use in the surface model*
- file [flat\\_plate\\_thermal.cc](#)  
*Flat plates for use in the surface model, with the thermal rider.*
- file [interaction\\_facet.cc](#)  
*Individual facets for use with environment interaction models.*
- file [interaction\\_facet\\_factory.cc](#)  
*Factory that creates an interaction facet, for a specific environment interaction model, from a facet model.*
- file [interaction\\_surface.cc](#)  
*Vehicle surface model for general environment interaction models.*
- file [interaction\\_surface\\_factory.cc](#)  
*Factory that creates an interaction surface, for a specific environment interaction model, from a surface model.*
- file [surface\\_model.cc](#)  
*Vehicle surface model for general environment interaction models.*
- file [surface\\_model\\_messages.cc](#)  
*Implement surface\_model\_messages.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

- `#define PATH "utils/surface_model/"`

### 6.3.1 Detailed Description

### 6.3.2 Macro Definition Documentation

#### 6.3.2.1 `#define PATH "utils/surface_model/"`

Definition at line 39 of file `surface_model_messages.cc`.



## Chapter 7

# Namespace Documentation

### 7.1 jeod Namespace Reference

Namespace jeod.

#### Data Structures

- class [Cylinder](#)  
*An cylinder implementation of [Facet](#).*
- class [Facet](#)  
*A general base class for all surface model facets.*
- class [FacetParams](#)  
*General base class for all parameters associated with facets in the surface model.*
- class [FlatPlate](#)  
*A [FlatPlate](#) implementation of [Facet](#).*
- class [FlatPlateCircular](#)  
*An circular flat plate implementation of [Facet](#).*
- class [FlatPlateThermal](#)  
*A [FlatPlate](#) implementation of [Facet](#), with thermal information.*
- class [InteractionFacet](#)  
*A base class for an interaction specific facet.*
- class [InteractionFacetFactory](#)  
*A factory to create a specific interaction facet from a general facet.*
- class [InteractionSurface](#)  
*A base class for interaction specific surfaces.*
- class [InteractionSurfaceFactory](#)  
*A base class for creating specific interaction surfaces from general surfaces.*
- struct [FacetStateInfo](#)  
*This is a structure used only in the surface model to aid in relative state calculations for articulation.*
- class [SurfaceModel](#)  
*A general, non-interaction specific surface that can be used to create surfaces suitable for specific interactions.*
- class [SurfaceModelMessages](#)  
*Messages associated with the use of the surface model.*

#### 7.1.1 Detailed Description

Namespace jeod.



## Chapter 8

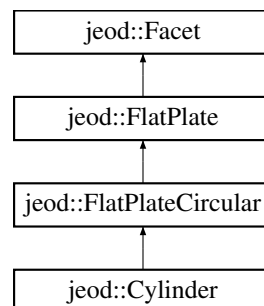
# Data Structure Documentation

### 8.1 jeod::Cylinder Class Reference

An cylinder implementation of [Facet](#).

```
#include <cylinder.hh>
```

Inheritance diagram for jeod::Cylinder:



#### Public Member Functions

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

#### Data Fields

- double [length](#)  
*Length of the cylinder.*

#### Private Member Functions

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

## Friends

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

## Additional Inherited Members

### 8.1.1 Detailed Description

An cylinder implementation of [Facet](#).

Definition at line 42 of file cylinder.hh.

### 8.1.2 Constructor & Destructor Documentation

#### 8.1.2.1 `jeod::Cylinder::Cylinder ( void )`

Default Constructor.

Definition at line 42 of file cylinder.cc.

#### 8.1.2.2 `jeod::Cylinder::~~Cylinder ( void ) [virtual]`

Destructor.

Definition at line 54 of file cylinder.cc.

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

### 8.1.3 Member Function Documentation

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

### 8.1.4 Friends And Related Function Documentation

#### 8.1.4.1 `void init_attrjeod__Cylinder ( ) [friend]`

#### 8.1.4.2 `friend class InputProcessor [friend]`

Definition at line 44 of file cylinder.hh.

### 8.1.5 Field Documentation

#### 8.1.5.1 `double jeod::Cylinder::length`

Length of the cylinder.

`trick_units(m)`

Definition at line 57 of file cylinder.hh.

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

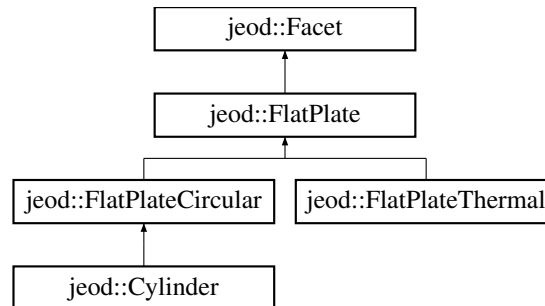
- [cylinder.hh](#)
- [cylinder.cc](#)

## 8.2 jeod::Facet Class Reference

A general base class for all surface model facets.

```
#include <facet.hh>
```

Inheritance diagram for jeod::Facet:



### Public Member Functions

- [Facet](#) ()  
*Default Constructor.*
- virtual [~Facet](#) ()  
*Destructor.*
- virtual void [initialize\\_mass\\_connection](#) (BaseDynManager &manager)
- virtual void [update\\_articulation](#) ()
- MassBody \* [get\\_mass\\_body\\_ptr](#) ()
- MassPointState \* [get\\_mass\\_rel\\_struct](#) ()  
*Getter for the mass\_rel\_struct element.*
- void [set\\_name](#) (std::string name\_in)  
*Setter for the name.*

### Data Fields

- double [position](#) [3]  
*Position of the facet in the vehicle structural frame.*
- double [local\\_position](#) [3]  
*Position of the facet in the structure frame of the MassBody this facet is associated with.*
- char \* [param\\_name](#)  
*Name of the facet parameters, usually a material type.*
- std::string [name](#)  
*Name of the facet.*
- char \* [mass\\_body\\_name](#)  
*The name of the MassBody this facet is associated with.*
- double [temperature](#)  
*Kinetic Temperature of the surface.*
- double [area](#)  
*Area of the plate.*

### Protected Member Functions

- virtual void [update\\_articulation\\_internal](#) ()

## Protected Attributes

- `MassBody * mass_body_ptr`  
*A pointer to the MassBody this facet is associated with.*
- `MassPointState * mass_rel_struct`  
*The relative state between the mass body this facet is associated with, and the user set structure point in the mass tree.*
- `double int_pos [3]`  
*An intermediate position, used as a working variable when doing calculations.*
- `bool connections_initialized`  
*Indicates if the mass connections for this Facet have been initialized.*

## Private Member Functions

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

## Friends

- `class InputProcessor`
- `class SurfaceModel`
- `void init_attrjeod__Facet ()`

### 8.2.1 Detailed Description

A general base class for all surface model facets.

Definition at line 54 of file facet.hh.

### 8.2.2 Constructor & Destructor Documentation

#### 8.2.2.1 `jeod::Facet::Facet ( void )`

Default Constructor.

Definition at line 55 of file facet.cc.

References `area`, `int_pos`, `local_position`, `position`, and `temperature`.

#### 8.2.2.2 `jeod::Facet::~~Facet ( void ) [virtual]`

Destructor.

Definition at line 76 of file facet.cc.

#### 8.2.2.3 `jeod::Facet::Facet ( const Facet & rhs ) [private]`

### 8.2.3 Member Function Documentation

#### 8.2.3.1 `MassBody * jeod::Facet::get_mass_body_ptr ( void )`

Definition at line 168 of file facet.cc.

References `mass_body_ptr`.

### 8.2.3.2 MassPointState\* jeod::Facet::get\_mass\_rel\_struct ( ) [inline]

Getter for the mass\_rel\_struct element,.

Definition at line 78 of file facet.hh.

References mass\_rel\_struct.

### 8.2.3.3 void jeod::Facet::initialize\_mass\_connection ( BaseDynManager & manager ) [virtual]

Definition at line 92 of file facet.cc.

References connections\_initialized, jeod::SurfaceModelMessages::initialization\_error, mass\_body\_name, and mass\_body\_ptr.

### 8.2.3.4 Facet& jeod::Facet::operator= ( const Facet & rhs ) [private]

### 8.2.3.5 void jeod::Facet::set\_name ( std::string name\_in ) [inline]

Setter for the name.

Definition at line 83 of file facet.hh.

References name.

### 8.2.3.6 void jeod::Facet::update\_articulation ( void ) [virtual]

Definition at line 129 of file facet.cc.

References connections\_initialized, jeod::SurfaceModelMessages::initialization\_error, and update\_articulation\_internal().

### 8.2.3.7 void jeod::Facet::update\_articulation\_internal ( void ) [protected],[virtual]

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

Definition at line 183 of file facet.cc.

References int\_pos, local\_position, mass\_rel\_struct, and position.

Referenced by update\_articulation(), and jeod::FlatPlate::update\_articulation\_internal().

## 8.2.4 Friends And Related Function Documentation

### 8.2.4.1 void init\_attrjeod\_\_Facet ( ) [friend]

### 8.2.4.2 friend class InputProcessor [friend]

Definition at line 56 of file facet.hh.

### 8.2.4.3 friend class SurfaceModel [friend]

Definition at line 58 of file facet.hh.

## 8.2.5 Field Documentation

**8.2.5.1 double jeod::Facet::area**

Area of the plate.

trick\_units(m2)

Definition at line 132 of file facet.hh.

Referenced by Facet().

**8.2.5.2 bool jeod::Facet::connections\_initialized** [protected]

Indicates if the mass connections for this [Facet](#) have been initialized.

trick\_units(-)

Definition at line 163 of file facet.hh.

Referenced by initialize\_mass\_connection(), and update\_articulation().

**8.2.5.3 double jeod::Facet::int\_pos[3]** [protected]

An intermediate position, used as a working variable when doing calculations.

trick\_units(m)

Definition at line 157 of file facet.hh.

Referenced by Facet(), and update\_articulation\_internal().

**8.2.5.4 double jeod::Facet::local\_position[3]**

Position of the facet in the structure frame of the MassBody this facet is associated with.

Used in articulation, contact, etc.trick\_units(m)

Definition at line 101 of file facet.hh.

Referenced by Facet(), and update\_articulation\_internal().

**8.2.5.5 char\* jeod::Facet::mass\_body\_name**

The name of the MassBody this facet is associated with.

This is used, only for specific applications, such as contact and articulation. Otherwise it is optional. This is used to find and cache a pointer to the mass\_body, stored below in 'mass\_body\_ptr'.trick\_units(-)

Definition at line 122 of file facet.hh.

Referenced by initialize\_mass\_connection().

**8.2.5.6 MassBody\* jeod::Facet::mass\_body\_ptr** [protected]

A pointer to the MassBody this facet is associated with.

This is used, only for specific applications, such as contact and articulation. Otherwise it is optional. This pointer is cached from the tree of MassBodies for the mass\_body\_name set above.trick\_units(-)

Definition at line 144 of file facet.hh.

Referenced by get\_mass\_body\_ptr(), and initialize\_mass\_connection().



**8.2.5.7** MassPointState\* jeod::Facet::mass\_rel\_struct [protected]

The relative state between the mass body this facet is associated with, and the user set structure point in the mass tree.

Only used in certain applications, such as articulation.trick\_units(-)

Definition at line 151 of file facet.hh.

Referenced by get\_mass\_rel\_struct(), jeod::FlatPlate::update\_articulation\_internal(), and update\_articulation\_internal().

**8.2.5.8** std::string jeod::Facet::name

Name of the facet.

trick\_units(-)

Definition at line 113 of file facet.hh.

Referenced by set\_name().

**8.2.5.9** char\* jeod::Facet::param\_name

Name of the facet parameters, usually a material type.

Used to match the facet to FacetParamstrick\_units(-)

Definition at line 108 of file facet.hh.

**8.2.5.10** double jeod::Facet::position[3]

Position of the facet in the vehicle structural frame.

The reference point of the facet is positioned on a class by class basis (example: for flat plate it is the position of the center of pressure)trick\_units(m)

Definition at line 95 of file facet.hh.

Referenced by Facet(), and update\_articulation\_internal().

**8.2.5.11** double jeod::Facet::temperature

Kinetic Temperature of the surface.

trick\_units(K)

Definition at line 127 of file facet.hh.

Referenced by Facet().

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

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

**8.3 jeod::FacetParams Class Reference**

General base class for all parameters associated with facets in the surface model.

```
#include <facet_params.hh>
```

## Public Member Functions

- [FacetParams](#) ()  
*Default Constructor.*
- virtual [~FacetParams](#) ()  
*Destructor.*
- void [set\\_name](#) (std::string name\_in)  
*Setter for the name.*

## Data Fields

- std::string [name](#)  
*Name that will be used to match [FacetParams](#) to facets.*

## Private Member Functions

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

## Friends

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

### 8.3.1 Detailed Description

General base class for all parameters associated with facets in the surface model.

Definition at line 49 of file facet\_params.hh.

### 8.3.2 Constructor & Destructor Documentation

#### 8.3.2.1 jeod::FacetParams::FacetParams ( void )

Default Constructor.

Definition at line 47 of file facet\_params.cc.

#### 8.3.2.2 jeod::FacetParams::~FacetParams ( void ) [virtual]

Destructor.

Definition at line 61 of file facet\_params.cc.

#### 8.3.2.3 jeod::FacetParams::FacetParams ( const FacetParams & rhs ) [private]

### 8.3.3 Member Function Documentation

#### 8.3.3.1 FacetParams& jeod::FacetParams::operator= ( const FacetParams & rhs ) [private]

#### 8.3.3.2 void jeod::FacetParams::set\_name ( std::string name\_in ) [inline]

Setter for the name.

Definition at line 71 of file facet\_params.hh.

### 8.3.4 Friends And Related Function Documentation

8.3.4.1 void init\_attrjeod\_\_FacetParams ( ) [friend]

8.3.4.2 friend class InputProcessor [friend]

Definition at line 51 of file facet\_params.hh.

### 8.3.5 Field Documentation

8.3.5.1 std::string jeod::FacetParams::name

Name that will be used to match [FacetParams](#) to facets.

trick\_units(—)

Definition at line 65 of file facet\_params.hh.

Referenced by jeod::InteractionSurfaceFactory::add\_facet\_params().

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

- [facet\\_params.hh](#)
- [facet\\_params.cc](#)

## 8.4 jeod::FacetStateInfo Struct Reference

This is a structure used only in the surface model to aid in relative state calculations for articulation.

```
#include <surface_model.hh>
```

### Public Member Functions

- [FacetStateInfo](#) ()  
*Default constructor to keep the memory manager happy.*
- [FacetStateInfo](#) (MassBody \*new\_mass\_body)  
*[FacetStateInfo](#) non-default constructor.*
- [FacetStateInfo](#) (MassBody &new\_mass\_body)  
*[FacetStateInfo](#) non-default constructor.*
- bool [operator==](#) (const [FacetStateInfo](#) &rhs) const  
*Compare this [FacetStateInfo](#) object to another.*

### Data Fields

- MassPointState [mass\\_state](#)  
*The resulting relative mass point state between the structural body named in struct\_body\_name and the MassBody pointed to in this structure's mass\_body.*
- MassBody \* [mass\\_body](#)  
*The MassBody object whose state, relative and w.r.t.*

## Friends

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

### 8.4.1 Detailed Description

This is a structure used only in the surface model to aid in relative state calculations for articulation.

For each mass body that needs a relative state calculated w.r.t. the mass body named in struct `body_name`, one of these objects will be instantiated. That way, the relative state information must only be calculated once per mass body.

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

### 8.4.2 Constructor & Destructor Documentation

#### 8.4.2.1 `jeod::FacetStateInfo::FacetStateInfo ( ) [inline]`

Default constructor to keep the memory manager happy.

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

#### 8.4.2.2 `jeod::FacetStateInfo::FacetStateInfo ( MassBody * new_mass_body ) [inline]`

[FacetStateInfo](#) non-default constructor.

Parameters

<code>new_mass_body</code>	The mass body to which this object will refer.
----------------------------	--

Definition at line 85 of file `surface_model.hh`.

#### 8.4.2.3 `jeod::FacetStateInfo::FacetStateInfo ( MassBody & new_mass_body ) [inline]`

[FacetStateInfo](#) non-default constructor.

Parameters

<code>new_mass_body</code>	The mass body to which this object will refer.
----------------------------	--

Definition at line 94 of file `surface_model.hh`.

### 8.4.3 Member Function Documentation

#### 8.4.3.1 `bool jeod::FacetStateInfo::operator==( const FacetStateInfo & rhs ) const [inline]`

Compare this [FacetStateInfo](#) object to another.

The two are 'equal' if they refer to the same mass body.

Parameters

<code>rhs</code>	Object to be compared with this object.
------------------	---

Definition at line 104 of file `surface_model.hh`.

References `mass_body`.

### 8.4.4 Friends And Related Function Documentation

8.4.4.1 void init\_attrjeod\_\_FacetStateInfo ( ) [friend]

8.4.4.2 friend class InputProcessor [friend]

Definition at line 62 of file surface\_model.hh.

### 8.4.5 Field Documentation

8.4.5.1 MassBody\* jeod::FacetStateInfo::mass\_body

The MassBody object whose state, relative and w.r.t.

the MassBody named in struct\_body\_name, is being calculatedtrick\_io(\*\*)

Definition at line 74 of file surface\_model.hh.

Referenced by operator==(), and jeod::SurfaceModel::update\_articulation().

8.4.5.2 MassPointState jeod::FacetStateInfo::mass\_state

The resulting relative mass point state between the structural body named in struct\_body\_name and the MassBody pointed to in this structure's mass\_body.

trick\_io(\*\*)

Definition at line 68 of file surface\_model.hh.

Referenced by jeod::SurfaceModel::update\_articulation().

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

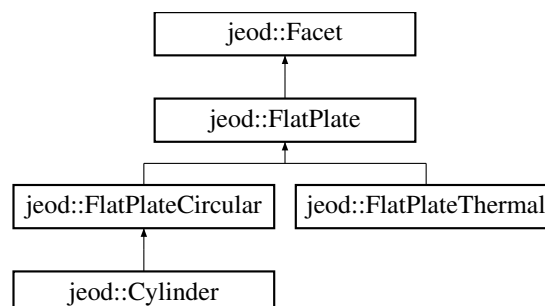
- [surface\\_model.hh](#)

## 8.5 jeod::FlatPlate Class Reference

A [FlatPlate](#) implementation of [Facet](#).

```
#include <flat_plate.hh>
```

Inheritance diagram for jeod::FlatPlate:



### Public Member Functions

- [FlatPlate](#) ()

*Default Constructor.*

- virtual [~FlatPlate](#) ()

*Destructor.*

## Data Fields

- double [normal](#) [3]

*normal of the plate, pointing outward of the craft, with respect to the vehicle structural frame.*

- double [local\\_normal](#) [3]

*The normal of the plate with respect to the structural frame associated with the mass body named in mass\_body\_name.*

## Protected Member Functions

- virtual void [update\\_articulation\\_internal](#) ()

## Private Member Functions

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

## Friends

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

## Additional Inherited Members

### 8.5.1 Detailed Description

A [FlatPlate](#) implementation of [Facet](#).

Definition at line 49 of file flat\_plate.hh.

### 8.5.2 Constructor & Destructor Documentation

#### 8.5.2.1 [jeod::FlatPlate::FlatPlate](#) ( void )

Default Constructor.

Definition at line 42 of file flat\_plate.cc.

References [local\\_normal](#), and [normal](#).

#### 8.5.2.2 [jeod::FlatPlate::~~FlatPlate](#) ( void ) [virtual]

Destructor.

Definition at line 55 of file flat\_plate.cc.

8.5.2.3 `jeod::FlatPlate::FlatPlate ( const FlatPlate & rhs )` `[private]`

### 8.5.3 Member Function Documentation

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

8.5.3.2 `void jeod::FlatPlate::update_articulation_internal ( void )` `[protected]`, `[virtual]`

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

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

References `local_normal`, `jeod::Facet::mass_rel_struct`, `normal`, and `jeod::Facet::update_articulation_internal()`.

### 8.5.4 Friends And Related Function Documentation

8.5.4.1 `void init_attrjeod__FlatPlate ( )` `[friend]`

8.5.4.2 `friend class InputProcessor` `[friend]`

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

### 8.5.5 Field Documentation

8.5.5.1 `double jeod::FlatPlate::local_normal[3]`

The normal of the plate with respect to the structural frame associated with the mass body named in `mass_body_name`.

`trick_units(-)`

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

Referenced by `FlatPlate()`, and `update_articulation_internal()`.

8.5.5.2 `double jeod::FlatPlate::normal[3]`

normal of the plate, pointing outward of the craft, with respect to the vehicle structural frame.

If the mass tying functionality is turned on, this will be in the structural frame associated with the mass body named in `struct_mass_name`. If mass tying functionality is not turned on, this parameter should be set directly. If it is turned on, then the `local_normal` should be set and this variable should be automatically calculated.`trick_units(-)`

Definition at line 71 of file `flat_plate.hh`.

Referenced by `FlatPlate()`, and `update_articulation_internal()`.

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

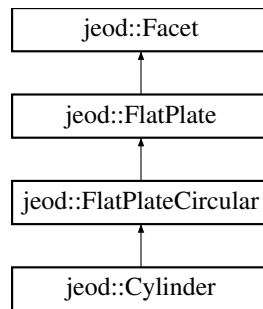
- [flat\\_plate.hh](#)
- [flat\\_plate.cc](#)

## 8.6 jeod::FlatPlateCircular Class Reference

An circular flat plate implementation of [Facet](#).

```
#include <flat_plate_circular.hh>
```

Inheritance diagram for `jeod::FlatPlateCircular`:



## Public Member Functions

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

## Data Fields

- double [radius](#)  
*Radius of the plate.*

## Private Member Functions

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

## Friends

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

## Additional Inherited Members

### 8.6.1 Detailed Description

An circular flat plate implementation of [Facet](#).

Definition at line 42 of file `flat_plate_circular.hh`.

### 8.6.2 Constructor & Destructor Documentation

#### 8.6.2.1 `jeod::FlatPlateCircular::FlatPlateCircular ( void )`

Default Constructor.

Definition at line 42 of file `flat_plate_circular.cc`.



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

Destructor.

Definition at line 54 of file `flat_plate_circular.cc`.

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

### 8.6.3 Member Function Documentation

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

### 8.6.4 Friends And Related Function Documentation

8.6.4.1 `void init_attrjeod__FlatPlateCircular ( ) [friend]`

8.6.4.2 `friend class InputProcessor [friend]`

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

### 8.6.5 Field Documentation

8.6.5.1 `double jeod::FlatPlateCircular::radius`

Radius of the plate.

`trick_units(m)`

Definition at line 57 of file `flat_plate_circular.hh`.

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

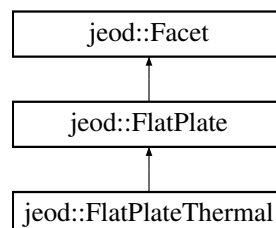
- [flat\\_plate\\_circular.hh](#)
- [flat\\_plate\\_circular.cc](#)

## 8.7 jeod::FlatPlateThermal Class Reference

A [FlatPlate](#) implementation of [Facet](#), with thermal information.

```
#include <flat_plate_thermal.hh>
```

Inheritance diagram for `jeod::FlatPlateThermal`:



### Public Member Functions

- [FlatPlateThermal \( \)](#)  
*DefaultConstructor.*

- virtual `~FlatPlateThermal()`  
*Destructor.*

## Data Fields

- ThermalFacetRider `thermal`  
*Thermal characteristics rider.*

## Private Member Functions

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

## Friends

- class `InputProcessor`
- void `init_attrjeod__FlatPlateThermal()`

## Additional Inherited Members

### 8.7.1 Detailed Description

A `FlatPlate` implementation of `Facet`, with thermal information.

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

### 8.7.2 Constructor & Destructor Documentation

#### 8.7.2.1 `jeod::FlatPlateThermal::FlatPlateThermal ( void )`

DefaultConstructor.

Definition at line 42 of file `flat_plate_thermal.cc`.

#### 8.7.2.2 `jeod::FlatPlateThermal::~~FlatPlateThermal ( void ) [virtual]`

Destructor.

Definition at line 53 of file `flat_plate_thermal.cc`.

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

### 8.7.3 Member Function Documentation

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

### 8.7.4 Friends And Related Function Documentation

#### 8.7.4.1 `void init_attrjeod__FlatPlateThermal ( ) [friend]`

#### 8.7.4.2 `friend class InputProcessor [friend]`

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

### 8.7.5 Field Documentation

#### 8.7.5.1 ThermalFacetRider jeod::FlatPlateThermal::thermal

Thermal characteristics rider.

trick\_units(-)

Definition at line 64 of file flat\_plate\_thermal.hh.

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

- [flat\\_plate\\_thermal.hh](#)
- [flat\\_plate\\_thermal.cc](#)

## 8.8 jeod::InteractionFacet Class Reference

A base class for an interaction specific facet.

```
#include <interaction_facet.hh>
```

### Public Member Functions

- [InteractionFacet](#) ()  
*Default constructor.*
- virtual [~InteractionFacet](#) ()  
*Destructor.*

### Data Fields

- double [force](#) [3]  
*The force on the facet caused by the environment interaction.*
- double [torque](#) [3]  
*The torque on the facet caused by the environment interaction.*
- [Facet](#) \* [base\\_facet](#)  
*The original facet from which this interaction facet was created.*

### Private Member Functions

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

### Friends

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

#### 8.8.1 Detailed Description

A base class for an interaction specific facet.

Definition at line 50 of file interaction\_facet.hh.

## 8.8.2 Constructor & Destructor Documentation

### 8.8.2.1 `jeod::InteractionFacet::InteractionFacet ( void )`

Default constructor.

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

References `force`, and `torque`.

### 8.8.2.2 `jeod::InteractionFacet::~~InteractionFacet ( void ) [virtual]`

Destructor.

Definition at line 64 of file `interaction_facet.cc`.

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

## 8.8.3 Member Function Documentation

### 8.8.3.1 `InteractionFacet& jeod::InteractionFacet::operator= ( const InteractionFacet & rhs ) [private]`

## 8.8.4 Friends And Related Function Documentation

### 8.8.4.1 `void init_attrjeod__InteractionFacet ( ) [friend]`

### 8.8.4.2 `friend class InputProcessor [friend]`

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

## 8.8.5 Field Documentation

### 8.8.5.1 `Facet* jeod::InteractionFacet::base_facet`

The original facet from which this interaction facet was created.

`trick_units(-)`

Definition at line 75 of file `interaction_facet.hh`.

### 8.8.5.2 `double jeod::InteractionFacet::force[3]`

The force on the facet caused by the environment interaction.

`trick_units(N)`

Definition at line 65 of file `interaction_facet.hh`.

Referenced by `InteractionFacet()`.

### 8.8.5.3 `double jeod::InteractionFacet::torque[3]`

The torque on the facet caused by the environment interaction.

`trick_units(N*m)`

Definition at line 70 of file `interaction_facet.hh`.

Referenced by `InteractionFacet()`.

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

- [interaction\\_facet.hh](#)
- [interaction\\_facet.cc](#)

## 8.9 jeod::InteractionFacetFactory Class Reference

A factory to create a specific interaction facet from a general facet.

```
#include <interaction_facet_factory.hh>
```

### Public Member Functions

- [InteractionFacetFactory](#) ()  
*Default Constructor.*
- virtual [~InteractionFacetFactory](#) ()  
*Destructor.*
- virtual [InteractionFacet](#) \* [create\\_facet](#) ([Facet](#) \*facet, [FacetParams](#) \*params)=0  
*A pure virtual function that creates a specific interaction facet from a base facet with the given [FacetParams](#).*
- virtual bool [is\\_correct\\_factory](#) ([Facet](#) \*facet)=0  
*A pure virtual function.*

### Protected Attributes

- bool [trick\\_bool](#)  
*Unused data field to expedite dynamic allocation in Trick environment.*

### Private Member Functions

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

### Friends

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

#### 8.9.1 Detailed Description

A factory to create a specific interaction facet from a general facet.

Definition at line 57 of file [interaction\\_facet\\_factory.hh](#).

#### 8.9.2 Constructor & Destructor Documentation

##### 8.9.2.1 jeod::InteractionFacetFactory::InteractionFacetFactory ( void )

Default Constructor.

Definition at line 44 of file [interaction\\_facet\\_factory.cc](#).

8.9.2.2 `jeod::InteractionFacetFactory::~~InteractionFacetFactory ( void ) [virtual]`

Destructor.

Definition at line 58 of file `interaction_facet_factory.cc`.

8.9.2.3 `jeod::InteractionFacetFactory::InteractionFacetFactory ( const InteractionFacetFactory & rhs ) [private]`

### 8.9.3 Member Function Documentation

8.9.3.1 `virtual InteractionFacet* jeod::InteractionFacetFactory::create_facet ( Facet * facet, FacetParams * params ) [pure virtual]`

A pure virtual function that creates a specific interaction facet from a base facet with the given [FacetParams](#).

This defines interface for all classes that inherit from [InteractionFacetFactory](#)

#### Returns

The new interaction facet

#### Parameters

in	<i>facet</i>	The facet the <a href="#">InteractionFacet</a> is created from
in	<i>params</i>	The parameter object to be added.

8.9.3.2 `virtual bool jeod::InteractionFacetFactory::is_correct_factory ( Facet * facet ) [pure virtual]`

A pure virtual function.

Returns true or false: is the given facet the type this factory is meant to use?

#### Returns

true or false. Is the given facet the correct type for this factory?

#### Parameters

in	<i>facet</i>	The facet that is being checked
----	--------------	---------------------------------

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

### 8.9.4 Friends And Related Function Documentation

8.9.4.1 `void init_attrjeod__InteractionFacetFactory ( ) [friend]`

8.9.4.2 `friend class InputProcessor [friend]`

Definition at line 59 of file `interaction_facet_factory.hh`.

### 8.9.5 Field Documentation

8.9.5.1 `bool jeod::InteractionFacetFactory::trick_bool [protected]`

Unused data field to expedite dynamic allocation in Trick environment.

trick\_units(-)

Definition at line 101 of file interaction\_facet\_factory.hh.

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

- [interaction\\_facet\\_factory.hh](#)
- [interaction\\_facet\\_factory.cc](#)

## 8.10 jeod::InteractionSurface Class Reference

A base class for interaction specific surfaces.

```
#include <interaction_surface.hh>
```

### Public Member Functions

- [InteractionSurface](#) ()  
*Default Constructor.*
- virtual [~InteractionSurface](#) ()  
*Destructor.*
- virtual void [accumulate\\_thermal\\_sources](#) (void)  
*Adds all thermal sources together.*
- virtual void [thermal\\_integrator](#) (void)  
*Integrates thermal sources to get temperature.*
- virtual void [allocate\\_array](#) (unsigned int size)=0  
*A pure virtual function that will allocate the array of pointers to the correct interaction facet type, of the given size.*
- virtual void [allocate\\_interaction\\_facet](#) (Facet \*facet, [InteractionFacetFactory](#) \*factory, [FacetParams](#) \*params, unsigned int index)=0  
*A pure virtual function that will allocate the interaction facet, from the given facet, using the given facet parameters, and place it in the allocated array of interaction facets at the given index.*

### Private Member Functions

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

### Friends

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

#### 8.10.1 Detailed Description

A base class for interaction specific surfaces.

Definition at line 55 of file interaction\_surface.hh.

#### 8.10.2 Constructor & Destructor Documentation

##### 8.10.2.1 jeod::InteractionSurface::InteractionSurface ( void )

Default Constructor.

Definition at line 40 of file interaction\_surface.cc.

8.10.2.2 `jeod::InteractionSurface::~~InteractionSurface ( void ) [virtual]`

Destructor.

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

8.10.2.3 `jeod::InteractionSurface::InteractionSurface ( const InteractionSurface & rhs ) [private]`

### 8.10.3 Member Function Documentation

8.10.3.1 `virtual void jeod::InteractionSurface::accumulate_thermal_sources ( void ) [inline],[virtual]`

Adds all thermal sources together.

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

8.10.3.2 `virtual void jeod::InteractionSurface::allocate_array ( unsigned int size ) [pure virtual]`

A pure virtual function that will allocate the array of pointers to the correct interaction facet type, of the given size.

Parameters

<i>in</i>	<i>size</i>	Size of the array to be allocated Units: cnt
-----------	-------------	---

Referenced by `jeod::InteractionSurfaceFactory::create_surface()`.

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

A pure virtual function that will allocate the interaction facet, from the given facet, using the given facet parameters, and place it in the allocated array of interaction facets at the given index.

Parameters

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

Referenced by `jeod::InteractionSurfaceFactory::create_surface()`.

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

8.10.3.5 `virtual void jeod::InteractionSurface::thermal_integrator ( void ) [inline],[virtual]`

Integrates thermal sources to get temperature.

Definition at line 88 of file `interaction_surface.hh`.

### 8.10.4 Friends And Related Function Documentation

8.10.4.1 `void init_attrjeod__InteractionSurface ( ) [friend]`

8.10.4.2 `friend class InputProcessor [friend]`

Definition at line 57 of file `interaction_surface.hh`.



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

- [interaction\\_surface.hh](#)
- [interaction\\_surface.cc](#)

## 8.11 jeod::InteractionSurfaceFactory Class Reference

A base class for creating specific interaction surfaces from general surfaces.

```
#include <interaction_surface_factory.hh>
```

### Public Member Functions

- [InteractionSurfaceFactory](#) ()  
*Default Constructor.*
- virtual [~InteractionSurfaceFactory](#) ()  
*Destructor.*
- virtual void [create\\_surface](#) ([SurfaceModel](#) \*surface, [InteractionSurface](#) \*inter\_surface)  
*Creates an interaction surface, in the inter\_surface parameter, from the given [SurfaceModel](#).*
- void [create\\_surface](#) ([SurfaceModel](#) &surface, [InteractionSurface](#) &inter\_surface)  
*Convenience version of create\_surface which can be called from the input file.*
- virtual void [add\\_facet\\_factory](#) ([InteractionFacetFactory](#) \*to\_add)  
*Used to add an interaction facet factory for use in the surface factory.*
- virtual void [add\\_facet\\_params](#) ([FacetParams](#) \*to\_add)  
*Add a set of facet parameters for use in the interaction surface factory.*

### Data Fields

- [JeodPointerVector](#)  
< [InteractionFacetFactory](#) >  
::type [factories](#)  
*A vector of interaction facet factories to be used.*
- [JeodPointerVector](#)< [FacetParams](#) >  
::type [params](#)  
*A vector of [FacetParams](#) to be used.*

### Private Member Functions

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

### Friends

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

#### 8.11.1 Detailed Description

A base class for creating specific interaction surfaces from general surfaces.

Definition at line 56 of file [interaction\\_surface\\_factory.hh](#).

## 8.11.2 Constructor & Destructor Documentation

### 8.11.2.1 `jeod::InteractionSurfaceFactory::InteractionSurfaceFactory ( void )`

Default Constructor.

Definition at line 61 of file `interaction_surface_factory.cc`.

References `factories`, and `params`.

### 8.11.2.2 `jeod::InteractionSurfaceFactory::~~InteractionSurfaceFactory ( void ) [virtual]`

Destructor.

Definition at line 75 of file `interaction_surface_factory.cc`.

References `factories`, and `params`.

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

## 8.11.3 Member Function Documentation

### 8.11.3.1 `void jeod::InteractionSurfaceFactory::add_facet_factory ( InteractionFacetFactory * to_add ) [virtual]`

Used to add an interaction facet factory for use in the surface factory.

Parameters

<code>in</code>	<code>to_add</code>	The interaction facet factory to add
-----------------	---------------------	--------------------------------------

Definition at line 200 of file `interaction_surface_factory.cc`.

References `factories`, and `jeod::SurfaceModelMessages::setup_error`.

### 8.11.3.2 `void jeod::InteractionSurfaceFactory::add_facet_params ( FacetParams * to_add ) [virtual]`

Add a set of facet parameters for use in the interaction surface factory.

Parameters

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

Definition at line 222 of file `interaction_surface_factory.cc`.

References `jeod::FacetParams::name`, `params`, and `jeod::SurfaceModelMessages::setup_error`.

### 8.11.3.3 `void jeod::InteractionSurfaceFactory::create_surface ( SurfaceModel * surface, InteractionSurface * inter_surface ) [virtual]`

Creates an interaction surface, in the `inter_surface` parameter, from the given [SurfaceModel](#).

The [InteractionSurfaceFactory](#) should contain all necessary [InteractionFacetFactories](#) and [FacetParams](#) already

Parameters

<code>in</code>	<code>surface</code>	The surface model used to create the interaction surface
-----------------	----------------------	--

out	inter_surface	Where the interaction surface will be produced
-----	---------------	--

Definition at line 93 of file interaction\_surface\_factory.cc.

References `jeod::InteractionSurface::allocate_array()`, `jeod::InteractionSurface::allocate_interaction_facet()`, `jeod::SurfaceModel::facets`, `factories`, `jeod::SurfaceModelMessages::initialization_error`, and `params`.

Referenced by `create_surface()`.

**8.11.3.4** `void jeod::InteractionSurfaceFactory::create_surface ( SurfaceModel & surface, InteractionSurface & inter_surface ) [inline]`

Convenience version of `create_surface` which can be called from the input file.

Parameters

surface	Surface model.
inter_surface	Reference to the interaction surface.

Definition at line 78 of file interaction\_surface\_factory.hh.

References `create_surface()`.

**8.11.3.5** `InteractionSurfaceFactory& jeod::InteractionSurfaceFactory::operator= ( const InteractionSurfaceFactory & rhs ) [private]`

## 8.11.4 Friends And Related Function Documentation

**8.11.4.1** `void init_attrjeod__InteractionSurfaceFactory ( ) [friend]`

**8.11.4.2** `friend class InputProcessor [friend]`

Definition at line 58 of file interaction\_surface\_factory.hh.

## 8.11.5 Field Documentation

**8.11.5.1** `JeodPointerVector<InteractionFacetFactory>::type jeod::InteractionSurfaceFactory::factories`

A vector of interaction facet factories to be used.

Matched to facets by `typetrick_io(**)`

Definition at line 87 of file interaction\_surface\_factory.hh.

Referenced by `add_facet_factory()`, `create_surface()`, `InteractionSurfaceFactory()`, and `~InteractionSurfaceFactory()`.

**8.11.5.2** `JeodPointerVector<FacetParams>::type jeod::InteractionSurfaceFactory::params`

A vector of [FacetParams](#) to be used.

`trick_io(**)`

Definition at line 92 of file interaction\_surface\_factory.hh.

Referenced by `add_facet_params()`, `create_surface()`, `InteractionSurfaceFactory()`, and `~InteractionSurfaceFactory()`.

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

- [interaction\\_surface\\_factory.hh](#)

- [interaction\\_surface\\_factory.cc](#)

## 8.12 jeod::SurfaceModel Class Reference

A general, non-interaction specific surface that can be used to create surfaces suitable for specific interactions.

```
#include <surface_model.hh>
```

### Public Member Functions

- [SurfaceModel](#) ()  
*Default constructor.*
- [~SurfaceModel](#) ()  
*Destructor.*
- void [add\\_facets](#) ([Facet](#) \*\*new\_facets, unsigned int num\_new\_facets)
- void [add\\_facet](#) ([Facet](#) \*new\_facet)
- void [initialize\\_mass\\_connections](#) (BaseDynManager &manager)
- void [update\\_articulation](#) ()

### Data Fields

- bool [articulation\\_active](#)  
*Is the articulation active? If yes, facet information will be updated from the previously supplied mass tree.*
- char \* [struct\\_body\\_name](#)  
*The name of the MassBody representing the overall structural frame of the vehicle associated with this surface model.*
- JeodPointerVector< [Facet](#) >::type [facets](#)  
*The facets that make up the surface.*

### Protected Attributes

- MassBody \* [struct\\_body\\_ptr](#)  
*A pointer to the MassBody named by struct\_body\_name.*
- JeodObjectList< [FacetStateInfo](#) >  
::type [articulation\\_states](#)  
*The set of states used to update the articulation of each facet.*

### Private Member Functions

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

### Friends

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

#### 8.12.1 Detailed Description

A general, non-interaction specific surface that can be used to create surfaces suitable for specific interactions.

Definition at line 114 of file [surface\\_model.hh](#).

## 8.12.2 Constructor & Destructor Documentation

### 8.12.2.1 jeod::SurfaceModel::SurfaceModel ( void )

Default constructor.

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

References `articulation_states`, and `facets`.

### 8.12.2.2 jeod::SurfaceModel::~~SurfaceModel ( void )

Destructor.

Definition at line 77 of file `surface_model.cc`.

References `articulation_states`, and `facets`.

### 8.12.2.3 jeod::SurfaceModel::SurfaceModel ( const SurfaceModel & rhs ) [private]

## 8.12.3 Member Function Documentation

### 8.12.3.1 void jeod::SurfaceModel::add\_facet ( Facet \* new\_facet )

Definition at line 125 of file `surface_model.cc`.

References `facets`, and `jeod::SurfaceModelMessages::setup_error`.

### 8.12.3.2 void jeod::SurfaceModel::add\_facets ( Facet \*\* new\_facets, unsigned int num\_new\_facets )

Definition at line 90 of file `surface_model.cc`.

References `facets`, and `jeod::SurfaceModelMessages::setup_error`.

### 8.12.3.3 void jeod::SurfaceModel::initialize\_mass\_connections ( BaseDynManager & manager )

Definition at line 150 of file `surface_model.cc`.

References `articulation_states`, `facets`, `jeod::SurfaceModelMessages::initialization_error`, `struct_body_name`, and `struct_body_ptr`.

### 8.12.3.4 SurfaceModel& jeod::SurfaceModel::operator= ( const SurfaceModel & rhs ) [private]

### 8.12.3.5 void jeod::SurfaceModel::update\_articulation ( void )

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

References `articulation_active`, `articulation_states`, `facets`, `jeod::FacetStateInfo::mass_body`, `jeod::FacetStateInfo::mass_state`, `jeod::SurfaceModelMessages::runtime_error`, `struct_body_name`, and `struct_body_ptr`.

## 8.12.4 Friends And Related Function Documentation

### 8.12.4.1 void init\_attrjeod\_\_SurfaceModel ( ) [friend]

### 8.12.4.2 friend class InputProcessor [friend]

Definition at line 116 of file `surface_model.hh`.

### 8.12.5 Field Documentation

#### 8.12.5.1 `bool jeod::SurfaceModel::articulation_active`

Is the articulation active? If yes, facet information will be updated from the previously supplied mass tree.

If not, nothing will update. This defaults to `false` `trick_units(-)`

Definition at line 131 of file `surface_model.hh`.

Referenced by `update_articulation()`.

#### 8.12.5.2 `JeodObjectList<FacetStateInfo>::type jeod::SurfaceModel::articulation_states` `[protected]`

The set of states used to update the articulation of each facet.

`trick_io(**)`

Definition at line 172 of file `surface_model.hh`.

Referenced by `initialize_mass_connections()`, `SurfaceModel()`, `update_articulation()`, and `~SurfaceModel()`.

#### 8.12.5.3 `JeodPointerVector<Facet>::type jeod::SurfaceModel::facets`

The facets that make up the surface.

`trick_io(**)`

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

Referenced by `add_facet()`, `add_facets()`, `jeod::InteractionSurfaceFactory::create_surface()`, `initialize_mass_connections()`, `SurfaceModel()`, `update_articulation()`, and `~SurfaceModel()`.

#### 8.12.5.4 `char* jeod::SurfaceModel::struct_body_name`

The name of the `MassBody` representing the overall structural frame of the vehicle associated with this surface model.

All states of all contained facets will be relative to the structural frame of this `MassBody`. This name is only required for specific applications, such as contact and articulation `trick_units(-)`

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

Referenced by `initialize_mass_connections()`, and `update_articulation()`.

#### 8.12.5.5 `MassBody* jeod::SurfaceModel::struct_body_ptr` `[protected]`

A pointer to the `MassBody` named by `struct_body_name`.

This pointer will be set, using `struct_body_name`, by searching the `DynManager` object supplied to the `initialize_mass_connections` function. This pointer is only used for specific applications, such as contact and articulation `trick_units(-)`

Definition at line 166 of file `surface_model.hh`.

Referenced by `initialize_mass_connections()`, and `update_articulation()`.

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

- [surface\\_model.hh](#)
- [surface\\_model.cc](#)

## 8.13 jeod::SurfaceModelMessages Class Reference

Messages associated with the use of the surface model.

```
#include <surface_model_messages.hh>
```

### Static Public Attributes

- static char const \* [initialization\\_error](#)  
*Represents an error in initialization.*
- static char const \* [setup\\_error](#)  
*Represents an error in setup of the surface model.*
- static char const \* [runtime\\_error](#)  
*Represents an error during the runtime of the surface model.*

### Private Member Functions

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

### Friends

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

#### 8.13.1 Detailed Description

Messages associated with the use of the surface model.

Definition at line 54 of file `surface_model_messages.hh`.

#### 8.13.2 Constructor & Destructor Documentation

8.13.2.1 `jeod::SurfaceModelMessages::SurfaceModelMessages ( void )` `[private]`

8.13.2.2 `jeod::SurfaceModelMessages::SurfaceModelMessages ( const SurfaceModelMessages & rhs )` `[private]`

#### 8.13.3 Member Function Documentation

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

#### 8.13.4 Friends And Related Function Documentation

8.13.4.1 `void init_attrjeod__SurfaceModelMessages ( )` `[friend]`

8.13.4.2 `friend class InputProcessor` `[friend]`

Definition at line 56 of file `surface_model_messages.hh`.

### 8.13.5 Field Documentation

#### 8.13.5.1 `char const * jeod::SurfaceModelMessages::initialization_error` [static]

##### Initial value:

```
=
    "utils/surface_model/" "initialization_error"
```

Represents an error in initialization.

`trick_units(-)`

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

Referenced by `jeod::InteractionSurfaceFactory::create_surface()`, `jeod::Facet::initialize_mass_connection()`, `jeod::SurfaceModel::initialize_mass_connections()`, and `jeod::Facet::update_articulation()`.

#### 8.13.5.2 `char const * jeod::SurfaceModelMessages::runtime_error` [static]

##### Initial value:

```
=
    "utils/surface_model/" "runtime_error"
```

Represents an error during the runtime of the surface model.

`trick_units(-)`

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

Referenced by `jeod::SurfaceModel::update_articulation()`.

#### 8.13.5.3 `char const * jeod::SurfaceModelMessages::setup_error` [static]

##### Initial value:

```
=
    "utils/surface_model/" "setup_error"
```

Represents an error in setup of the surface model.

`trick_units(-)`

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

Referenced by `jeod::SurfaceModel::add_facet()`, `jeod::InteractionSurfaceFactory::add_facet_factory()`, `jeod::InteractionSurfaceFactory::add_facet_params()`, and `jeod::SurfaceModel::add_facets()`.

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

- [surface\\_model\\_messages.hh](#)
- [surface\\_model\\_messages.cc](#)



## Chapter 9

# File Documentation

### 9.1 `class_declarations.hh` File Reference

Forward declarations of classes defined for JEOD 2.0 surface model.

#### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.1.1 Detailed Description

Forward declarations of classes defined for JEOD 2.0 surface model.

Definition in file [class\\_declarations.hh](#).

### 9.2 `cylinder.cc` File Reference

cylinders for use in the surface model

```
#include "../include/cylinder.hh"  
#include "utils/math/include/vector3.hh"
```

#### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.2.1 Detailed Description

cylinders for use in the surface model

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

## 9.3 cylinder.hh File Reference

cylinders for use in the surface model and the contact model

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

### Data Structures

- class [jeod::Cylinder](#)  
*An cylinder implementation of [Facet](#).*

### Namespaces

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

#### 9.3.1 Detailed Description

cylinders for use in the surface model and the contact model

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

## 9.4 facet.cc File Reference

Individual facets for use in the surface model.

```
#include <cstdint>
#include "dynamics/dyn_manager/include/base_dyn_manager.hh"
#include "dynamics/mass/include/mass.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/math/include/vector3.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/surface_model_messages.hh"
#include "../include/facet.hh"
```

### Namespaces

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

#### 9.4.1 Detailed Description

Individual facets for use in the surface model.

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

## 9.5 facet.hh File Reference

Individual facets for use in the surface model.

```
#include <string>
#include <utility>
#include "utils/sim_interface/include/jeod_class.hh"
#include "dynamics/mass/include/mass_point_state.hh"
```

### Data Structures

- class [jeod::Facet](#)  
*A general base class for all surface model facets.*

### Namespaces

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

#### 9.5.1 Detailed Description

Individual facets for use in the surface model.

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

## 9.6 facet\_params.cc File Reference

A pure virtual base class for facet parameters, used to create interaction facets in the InteractionSurfaceFactorys.

```
#include <cstdlib>
#include "../include/facet_params.hh"
```

### Namespaces

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

#### 9.6.1 Detailed Description

A pure virtual base class for facet parameters, used to create interaction facets in the InteractionSurfaceFactorys.

Definition in file [facet\\_params.cc](#).

## 9.7 facet\_params.hh File Reference

A virtual base class for facet parameters, used to create interaction facets in the InteractionSurfaceFactorys.

```
#include <string>
#include <utility>
#include "utils/sim_interface/include/jeod_class.hh"
```

## Data Structures

- class [jeod::FacetParams](#)

*General base class for all parameters associated with facets in the surface model.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.7.1 Detailed Description

A virtual base class for facet parameters, used to create interaction facets in the InteractionSurfaceFactorys.

Definition in file [facet\\_params.hh](#).

## 9.8 flat\_plate.cc File Reference

Flat plates for use in the surface model.

```
#include "../include/flat_plate.hh"
#include "utils/math/include/vector3.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.8.1 Detailed Description

Flat plates for use in the surface model.

Definition in file [flat\\_plate.cc](#).

## 9.9 flat\_plate.hh File Reference

Flat plates for use in the surface model.

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

## Data Structures

- class [jeod::FlatPlate](#)

*A [FlatPlate](#) implementation of [Facet](#).*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.9.1 Detailed Description

Flat plates for use in the surface model.

Definition in file [flat\\_plate.hh](#).

## 9.10 flat\_plate\_circular.cc File Reference

circular flat plates for use in the surface model

```
#include "../include/flat_plate_circular.hh"
#include "utils/math/include/vector3.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.10.1 Detailed Description

circular flat plates for use in the surface model

Definition in file [flat\\_plate\\_circular.cc](#).

## 9.11 flat\_plate\_circular.hh File Reference

circulat flat plates for use in the surface model and the contact model

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

### Data Structures

- class [jeod::FlatPlateCircular](#)

*An circular flat plate implementation of [Facet](#).*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.11.1 Detailed Description

circulat flat plates for use in the surface model and the contact model

Definition in file [flat\\_plate\\_circular.hh](#).

## 9.12 flat\_plate\_thermal.cc File Reference

Flat plates for use in the surface model, with the thermal rider.

```
#include "../include/flat_plate_thermal.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.12.1 Detailed Description

Flat plates for use in the surface model, with the thermal rider.

Definition in file [flat\\_plate\\_thermal.cc](#).

## 9.13 flat\_plate\_thermal.hh File Reference

Flat plates for use in the surface model, including a thermal portion.

```
#include "interactions/thermal_rider/include/thermal_facet_rider.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "flat_plate.hh"
```

### Data Structures

- class [jeod::FlatPlateThermal](#)

*A [FlatPlate](#) implementation of [Facet](#), with thermal information.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.13.1 Detailed Description

Flat plates for use in the surface model, including a thermal portion.

Definition in file [flat\\_plate\\_thermal.hh](#).

## 9.14 interaction\_facet.cc File Reference

Individual facets for use with environment interaction models.

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

## Namespaces

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

### 9.14.1 Detailed Description

Individual facets for use with environment interaction models.

Definition in file [interaction\\_facet.cc](#).

## 9.15 interaction\_facet.hh File Reference

Individual facets for use with specific environment interaction models.

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

## Data Structures

- class [jeod::InteractionFacet](#)  
*A base class for an interaction specific facet.*

## Namespaces

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

### 9.15.1 Detailed Description

Individual facets for use with specific environment interaction models.

Definition in file [interaction\\_facet.hh](#).

## 9.16 interaction\_facet\_factory.cc File Reference

Factory that creates an interaction facet, for a specific environment interaction model, from a facet model.

```
#include "../include/interaction_facet_factory.hh"  
#include "../include/interaction_facet.hh"
```

## Namespaces

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

### 9.16.1 Detailed Description

Factory that creates an interaction facet, for a specific environment interaction model, from a facet model.

Definition in file [interaction\\_facet\\_factory.cc](#).

## 9.17 interaction\_facet\_factory.hh File Reference

Factory that creates an interaction facet, for a specific environment interaction model, from a facet model.

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

### Data Structures

- class [jeod::InteractionFacetFactory](#)  
*A factory to create a specific interaction facet from a general facet.*

### Namespaces

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

#### 9.17.1 Detailed Description

Factory that creates an interaction facet, for a specific environment interaction model, from a facet model. This is a pure virtual class, and the pure virtual functions must be implemented in any instantiable, inheriting class

Definition in file [interaction\\_facet\\_factory.hh](#).

## 9.18 interaction\_surface.cc File Reference

Vehicle surface model for general environment interaction models.

```
#include "../include/interaction_surface.hh"
```

### Namespaces

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

#### 9.18.1 Detailed Description

Vehicle surface model for general environment interaction models.

Definition in file [interaction\\_surface.cc](#).

## 9.19 interaction\_surface.hh File Reference

Vehicle surface model for general environment interaction models.

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



## Data Structures

- class [jeod::InteractionSurface](#)  
*A base class for interaction specific surfaces.*

## Namespaces

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

### 9.19.1 Detailed Description

Vehicle surface model for general environment interaction models. This is a pure virtual function and has methods that must be implemented in any inherited, instantiable class

Definition in file [interaction\\_surface.hh](#).

## 9.20 interaction\_surface\_factory.cc File Reference

Factory that creates an interaction surface, for a specific environment interaction model, from a surface model.

```
#include <cstdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/interaction_surface_factory.hh"
#include "../include/interaction_facet_factory.hh"
#include "../include/interaction_surface.hh"
#include "../include/facet.hh"
#include "../include/facet_params.hh"
#include "../include/surface_model.hh"
#include "../include/surface_model_messages.hh"
```

## Namespaces

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

### 9.20.1 Detailed Description

Factory that creates an interaction surface, for a specific environment interaction model, from a surface model.

Definition in file [interaction\\_surface\\_factory.cc](#).

## 9.21 interaction\_surface\_factory.hh File Reference

Factory that creates an interaction surface, for a specific environment interaction model, from a surface model.

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

## Data Structures

- class [jeod::InteractionSurfaceFactory](#)  
*A base class for creating specific interaction surfaces from general surfaces.*

## Namespaces

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

### 9.21.1 Detailed Description

Factory that creates an interaction surface, for a specific environment interaction model, from a surface model.

Definition in file [interaction\\_surface\\_factory.hh](#).

## 9.22 surface\_model.cc File Reference

Vehicle surface model for general environment interaction models.

```
#include <cstdint>
#include <algorithm>
#include "dynamics/dyn_manager/include/base_dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "dynamics/mass/include/mass.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/surface_model.hh"
#include "../include/surface_model_messages.hh"
#include "../include/facet.hh"
```

## Namespaces

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

### 9.22.1 Detailed Description

Vehicle surface model for general environment interaction models.

Definition in file [surface\\_model.cc](#).

## 9.23 surface\_model.hh File Reference

Vehicle surface model for general environment interaction models.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/container/include/pointer_vector.hh"
#include "utils/container/include/object_vector.hh"
#include "utils/container/include/object_list.hh"
#include "dynamics/mass/include/mass_point_state.hh"
```

## Data Structures

- struct [jeod::FacetStateInfo](#)  
*This is a structure used only in the surface model to aid in relative state calculations for articulation.*
- class [jeod::SurfaceModel](#)  
*A general, non-interaction specific surface that can be used to create surfaces suitable for specific interactions.*

## Namespaces

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

### 9.23.1 Detailed Description

Vehicle surface model for general environment interaction models.

Definition in file [surface\\_model.hh](#).

## 9.24 surface\_model\_messages.cc File Reference

Implement surface\_model\_messages.

```
#include "../include/surface_model_messages.hh"
```

## Namespaces

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

## Macros

- #define [PATH](#) "utils/surface\_model/"

### 9.24.1 Detailed Description

Implement surface\_model\_messages.

Definition in file [surface\\_model\\_messages.cc](#).

## 9.25 surface\_model\_messages.hh File Reference

Implement surface\_model\_messages.

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

## Data Structures

- class [jeod::SurfaceModelMessages](#)  
*Messages associated with the use of the surface model.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.25.1 Detailed Description

Implement `surface_model_messages`.

Definition in file [surface\\_model\\_messages.hh](#).

# Index

- ~Cylinder
  - jeod::Cylinder, [18](#)
- ~Facet
  - jeod::Facet, [20](#)
- ~FacetParams
  - jeod::FacetParams, [24](#)
- ~FlatPlate
  - jeod::FlatPlate, [28](#)
- ~FlatPlateCircular
  - jeod::FlatPlateCircular, [30](#)
- ~FlatPlateThermal
  - jeod::FlatPlateThermal, [32](#)
- ~InteractionFacet
  - jeod::InteractionFacet, [34](#)
- ~InteractionFacetFactory
  - jeod::InteractionFacetFactory, [35](#)
- ~InteractionSurface
  - jeod::InteractionSurface, [37](#)
- ~InteractionSurfaceFactory
  - jeod::InteractionSurfaceFactory, [40](#)
- ~SurfaceModel
  - jeod::SurfaceModel, [43](#)
- accumulate\_thermal\_sources
  - jeod::InteractionSurface, [38](#)
- add\_facet
  - jeod::SurfaceModel, [43](#)
- add\_facet\_factory
  - jeod::InteractionSurfaceFactory, [40](#)
- add\_facet\_params
  - jeod::InteractionSurfaceFactory, [40](#)
- add\_facets
  - jeod::SurfaceModel, [43](#)
- allocate\_array
  - jeod::InteractionSurface, [38](#)
- allocate\_interaction\_facet
  - jeod::InteractionSurface, [38](#)
- area
  - jeod::Facet, [21](#)
- articulation\_active
  - jeod::SurfaceModel, [44](#)
- articulation\_states
  - jeod::SurfaceModel, [44](#)
- base\_facet
  - jeod::InteractionFacet, [34](#)
- class\_declarations.hh, [47](#)
- connections\_initialized
  - jeod::Facet, [22](#)
- create\_facet
  - jeod::InteractionFacetFactory, [36](#)
- create\_surface
  - jeod::InteractionSurfaceFactory, [40](#), [41](#)
- Cylinder
  - jeod::Cylinder, [18](#)
- cylinder.cc, [47](#)
- cylinder.hh, [48](#)
- Facet
  - jeod::Facet, [20](#)
- facet.cc, [48](#)
- facet.hh, [49](#)
- facet\_params.cc, [49](#)
- facet\_params.hh, [49](#)
- FacetParams
  - jeod::FacetParams, [24](#)
- FacetStateInfo
  - jeod::FacetStateInfo, [26](#)
- facets
  - jeod::SurfaceModel, [44](#)
- factories
  - jeod::InteractionSurfaceFactory, [41](#)
- flat\_plate.cc, [50](#)
- flat\_plate.hh, [50](#)
- flat\_plate\_circular.cc, [51](#)
- flat\_plate\_circular.hh, [51](#)
- flat\_plate\_thermal.cc, [52](#)
- flat\_plate\_thermal.hh, [52](#)
- FlatPlate
  - jeod::FlatPlate, [28](#)
- FlatPlateCircular
  - jeod::FlatPlateCircular, [30](#), [31](#)
- FlatPlateThermal
  - jeod::FlatPlateThermal, [32](#)
- force
  - jeod::InteractionFacet, [34](#)
- get\_mass\_body\_ptr
  - jeod::Facet, [20](#)
- get\_mass\_rel\_struct
  - jeod::Facet, [20](#)
- init\_attrjeod\_\_Cylinder
  - jeod::Cylinder, [18](#)
- init\_attrjeod\_\_Facet
  - jeod::Facet, [21](#)
- init\_attrjeod\_\_FacetParams
  - jeod::FacetParams, [25](#)
- init\_attrjeod\_\_FacetStateInfo

- jeod::FacetStateInfo, 27
- init\_attrjeod\_\_FlatPlate
  - jeod::FlatPlate, 29
- init\_attrjeod\_\_FlatPlateCircular
  - jeod::FlatPlateCircular, 31
- init\_attrjeod\_\_FlatPlateThermal
  - jeod::FlatPlateThermal, 32
- init\_attrjeod\_\_InteractionFacet
  - jeod::InteractionFacet, 34
- init\_attrjeod\_\_InteractionFacetFactory
  - jeod::InteractionFacetFactory, 36
- init\_attrjeod\_\_InteractionSurface
  - jeod::InteractionSurface, 38
- init\_attrjeod\_\_InteractionSurfaceFactory
  - jeod::InteractionSurfaceFactory, 41
- init\_attrjeod\_\_SurfaceModel
  - jeod::SurfaceModel, 43
- init\_attrjeod\_\_SurfaceModelMessages
  - jeod::SurfaceModelMessages, 45
- initialization\_error
  - jeod::SurfaceModelMessages, 46
- initialize\_mass\_connection
  - jeod::Facet, 21
- initialize\_mass\_connections
  - jeod::SurfaceModel, 43
- InputProcessor
  - jeod::Cylinder, 18
  - jeod::Facet, 21
  - jeod::FacetParams, 25
  - jeod::FacetStateInfo, 27
  - jeod::FlatPlate, 29
  - jeod::FlatPlateCircular, 31
  - jeod::FlatPlateThermal, 32
  - jeod::InteractionFacet, 34
  - jeod::InteractionFacetFactory, 36
  - jeod::InteractionSurface, 38
  - jeod::InteractionSurfaceFactory, 41
  - jeod::SurfaceModel, 43
  - jeod::SurfaceModelMessages, 45
- int\_pos
  - jeod::Facet, 22
- interaction\_facet.cc, 52
- interaction\_facet.hh, 53
- interaction\_facet\_factory.cc, 53
- interaction\_facet\_factory.hh, 54
- interaction\_surface.cc, 54
- interaction\_surface.hh, 54
- interaction\_surface\_factory.cc, 55
- interaction\_surface\_factory.hh, 55
- InteractionFacet
  - jeod::InteractionFacet, 34
- InteractionFacetFactory
  - jeod::InteractionFacetFactory, 35, 36
- InteractionSurface
  - jeod::InteractionSurface, 37, 38
- InteractionSurfaceFactory
  - jeod::InteractionSurfaceFactory, 40
- is\_correct\_factory
  - jeod::InteractionFacetFactory, 36
- jeod, 15
- jeod::Cylinder, 17
  - ~Cylinder, 18
  - Cylinder, 18
  - init\_attrjeod\_\_Cylinder, 18
  - InputProcessor, 18
  - length, 18
  - operator=, 18
- jeod::Facet, 19
  - ~Facet, 20
  - area, 21
  - connections\_initialized, 22
  - Facet, 20
  - get\_mass\_body\_ptr, 20
  - get\_mass\_rel\_struct, 20
  - init\_attrjeod\_\_Facet, 21
  - initialize\_mass\_connection, 21
  - InputProcessor, 21
  - int\_pos, 22
  - local\_position, 22
  - mass\_body\_name, 22
  - mass\_body\_ptr, 22
  - mass\_rel\_struct, 22
  - name, 23
  - operator=, 21
  - param\_name, 23
  - position, 23
  - set\_name, 21
  - SurfaceModel, 21
  - temperature, 23
  - update\_articulation, 21
  - update\_articulation\_internal, 21
- jeod::FacetParams, 23
  - ~FacetParams, 24
  - FacetParams, 24
  - init\_attrjeod\_\_FacetParams, 25
  - InputProcessor, 25
  - name, 25
  - operator=, 24
  - set\_name, 24
- jeod::FacetStateInfo, 25
  - FacetStateInfo, 26
  - init\_attrjeod\_\_FacetStateInfo, 27
  - InputProcessor, 27
  - mass\_body, 27
  - mass\_state, 27
  - operator==, 26
- jeod::FlatPlate, 27
  - ~FlatPlate, 28
  - FlatPlate, 28
  - init\_attrjeod\_\_FlatPlate, 29
  - InputProcessor, 29
  - local\_normal, 29
  - normal, 29
  - operator=, 29
  - update\_articulation\_internal, 29
- jeod::FlatPlateCircular, 29

- ~FlatPlateCircular, [30](#)
- FlatPlateCircular, [30](#), [31](#)
- init\_attrjeod\_\_FlatPlateCircular, [31](#)
- InputProcessor, [31](#)
- operator=, [31](#)
- radius, [31](#)
- jeod::FlatPlateThermal, [31](#)
  - ~FlatPlateThermal, [32](#)
  - FlatPlateThermal, [32](#)
  - init\_attrjeod\_\_FlatPlateThermal, [32](#)
  - InputProcessor, [32](#)
  - operator=, [32](#)
  - thermal, [33](#)
- jeod::InteractionFacet, [33](#)
  - ~InteractionFacet, [34](#)
  - base\_facet, [34](#)
  - force, [34](#)
  - init\_attrjeod\_\_InteractionFacet, [34](#)
  - InputProcessor, [34](#)
  - InteractionFacet, [34](#)
  - operator=, [34](#)
  - torque, [34](#)
- jeod::InteractionFacetFactory, [35](#)
  - ~InteractionFacetFactory, [35](#)
  - create\_facet, [36](#)
  - init\_attrjeod\_\_InteractionFacetFactory, [36](#)
  - InputProcessor, [36](#)
  - InteractionFacetFactory, [35](#), [36](#)
  - is\_correct\_factory, [36](#)
  - operator=, [36](#)
  - trick\_bool, [36](#)
- jeod::InteractionSurface, [37](#)
  - ~InteractionSurface, [37](#)
  - accumulate\_thermal\_sources, [38](#)
  - allocate\_array, [38](#)
  - allocate\_interaction\_facet, [38](#)
  - init\_attrjeod\_\_InteractionSurface, [38](#)
  - InputProcessor, [38](#)
  - InteractionSurface, [37](#), [38](#)
  - operator=, [38](#)
  - thermal\_integrator, [38](#)
- jeod::InteractionSurfaceFactory, [39](#)
  - ~InteractionSurfaceFactory, [40](#)
  - add\_facet\_factory, [40](#)
  - add\_facet\_params, [40](#)
  - create\_surface, [40](#), [41](#)
  - factories, [41](#)
  - init\_attrjeod\_\_InteractionSurfaceFactory, [41](#)
  - InputProcessor, [41](#)
  - InteractionSurfaceFactory, [40](#)
  - operator=, [41](#)
  - params, [41](#)
- jeod::SurfaceModel, [42](#)
  - ~SurfaceModel, [43](#)
  - add\_facet, [43](#)
  - add\_facets, [43](#)
  - articulation\_active, [44](#)
  - articulation\_states, [44](#)
  - facets, [44](#)
  - init\_attrjeod\_\_SurfaceModel, [43](#)
  - initialize\_mass\_connections, [43](#)
  - InputProcessor, [43](#)
  - operator=, [43](#)
  - struct\_body\_name, [44](#)
  - struct\_body\_ptr, [44](#)
  - SurfaceModel, [43](#)
  - update\_articulation, [43](#)
- jeod::SurfaceModelMessages, [45](#)
  - init\_attrjeod\_\_SurfaceModelMessages, [45](#)
  - initialization\_error, [46](#)
  - InputProcessor, [45](#)
  - operator=, [45](#)
  - runtime\_error, [46](#)
  - setup\_error, [46](#)
  - SurfaceModelMessages, [45](#)
- length
  - jeod::Cylinder, [18](#)
- local\_normal
  - jeod::FlatPlate, [29](#)
- local\_position
  - jeod::Facet, [22](#)
- mass\_body
  - jeod::FacetStateInfo, [27](#)
- mass\_body\_name
  - jeod::Facet, [22](#)
- mass\_body\_ptr
  - jeod::Facet, [22](#)
- mass\_rel\_struct
  - jeod::Facet, [22](#)
- mass\_state
  - jeod::FacetStateInfo, [27](#)
- Models, [11](#)
- name
  - jeod::Facet, [23](#)
  - jeod::FacetParams, [25](#)
- normal
  - jeod::FlatPlate, [29](#)
- operator=
  - jeod::Cylinder, [18](#)
  - jeod::Facet, [21](#)
  - jeod::FacetParams, [24](#)
  - jeod::FlatPlate, [29](#)
  - jeod::FlatPlateCircular, [31](#)
  - jeod::FlatPlateThermal, [32](#)
  - jeod::InteractionFacet, [34](#)
  - jeod::InteractionFacetFactory, [36](#)
  - jeod::InteractionSurface, [38](#)
  - jeod::InteractionSurfaceFactory, [41](#)
  - jeod::SurfaceModel, [43](#)
  - jeod::SurfaceModelMessages, [45](#)
- operator==
  - jeod::FacetStateInfo, [26](#)
- PATH

- SurfaceModel, [14](#)
- param\_name
  - jeod::Facet, [23](#)
- params
  - jeod::InteractionSurfaceFactory, [41](#)
- position
  - jeod::Facet, [23](#)
- radius
  - jeod::FlatPlateCircular, [31](#)
- runtime\_error
  - jeod::SurfaceModelMessages, [46](#)
- set\_name
  - jeod::Facet, [21](#)
  - jeod::FacetParams, [24](#)
- setup\_error
  - jeod::SurfaceModelMessages, [46](#)
- struct\_body\_name
  - jeod::SurfaceModel, [44](#)
- struct\_body\_ptr
  - jeod::SurfaceModel, [44](#)
- surface\_model.cc, [56](#)
- surface\_model.hh, [56](#)
- surface\_model\_messages.cc, [57](#)
- surface\_model\_messages.hh, [57](#)
- SurfaceModel, [13](#)
  - jeod::Facet, [21](#)
  - jeod::SurfaceModel, [43](#)
  - PATH, [14](#)
- SurfaceModelMessages
  - jeod::SurfaceModelMessages, [45](#)
- temperature
  - jeod::Facet, [23](#)
- thermal
  - jeod::FlatPlateThermal, [33](#)
- thermal\_integrator
  - jeod::InteractionSurface, [38](#)
- torque
  - jeod::InteractionFacet, [34](#)
- trick\_bool
  - jeod::InteractionFacetFactory, [36](#)
- update\_articulation
  - jeod::Facet, [21](#)
  - jeod::SurfaceModel, [43](#)
- update\_articulation\_internal
  - jeod::Facet, [21](#)
  - jeod::FlatPlate, [29](#)
- Utils, [12](#)