SurfaceModel

5.3

Generated by Doxygen 1.8.14

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

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

  circulat 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 facet.cc

  Individual facets for use in the surface model.
- file flat_plate.cc

  Flat plates for use in the surface model.
- file interaction_surface_factory.cc

  Factory that creates an interaction surface, for a specific enviornment 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.

6.3.1 Detailed Description

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 FacetStateInfo

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

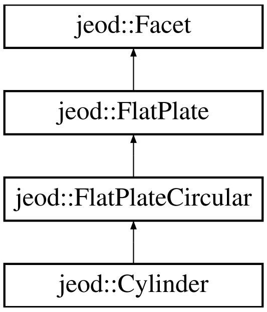
Data Structure Documentation

8.1 jeod::Cylinder Class Reference

An cylinder implementation of Facet.

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

Inheritance diagram for jeod::Cylinder:



Public Member Functions

- Cylinder ()=default
- ∼Cylinder () override=default
- Cylinder & operator= (const Cylinder &)=delete
- Cylinder (const Cylinder &)=delete

Data Fields

- double length {}

  Length of the cylinder.

Friends

- class InputProcessor
- void init_attrjeod__Cylinder ()

Additional Inherited Members

8.1.1 Detailed Description

An cylinder implementation of Facet.

Definition at line 74 of file cylinder.hh.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 Cylinder() [1/2]

```
jeod::Cylinder::Cylinder ( )  [default]
```

8.1.2.2 ∼Cylinder()

```
jeod::Cylinder::∼Cylinder ( )  [override], [default]
```

8.1.2.3 Cylinder() [2/2]

```
jeod::Cylinder::Cylinder (
          const Cylinder &  )  [delete]
```

8.1.3 Member Function Documentation

8.1.3.1 operator=()

```
Cylinder& jeod::Cylinder::operator= (
          const Cylinder &  )  [delete]
```

8.1.4 Friends And Related Function Documentation

8.1.4.1 init_attrjeod__Cylinder

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

8.1.4.2 InputProcessor

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

Definition at line 76 of file cylinder.hh.

8.1.5 Field Documentation

8.1.5.1 length

```
double jeod::Cylinder::length {}
```

Length of the cylinder.

trick_units(m)

Definition at line 85 of file cylinder.hh.

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

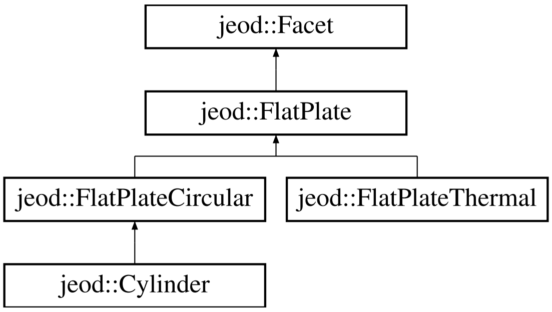
- cylinder.hh

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
- virtual ∼Facet ()=default
- Facet & operator= (const Facet &)=delete
- Facet (const Facet &)=delete
- 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.
- std::string param_name

  Name of the facet parameters, usually a material type.
- std::string name

  Name of the facet.
- std::string 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.

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 88 of file facet.hh.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 Facet() [1/2]

```
jeod::Facet::Facet ( )  [default]
```

8.2.2.2 ∼Facet()

```
virtual jeod::Facet::∼Facet ( )  [virtual], [default]
```

8.2.2.3 Facet() [2/2]

```
jeod::Facet::Facet (
             const Facet &  )  [delete]
```

8.2.3 Member Function Documentation

8.2.3.1 get_mass_body_ptr()

```
MassBody * jeod::Facet::get_mass_body_ptr ( )
```

Definition at line 119 of file facet.cc.

References mass_body_ptr.

8.2.3.2 get_mass_rel_struct()

```
MassPointState* jeod::Facet::get_mass_rel_struct ( )  [inline]
```

Getter for the mass_rel_struct element,.

Definition at line 107 of file facet.hh.

8.2.3.3 initialize_mass_connection()

```
void jeod::Facet::initialize_mass_connection (
            BaseDynManager & manager )  [virtual]
```

Definition at line 57 of file facet.cc.

References connections_initialized, jeod::SurfaceModelMessages::initialization_error, mass_body_name, and mass_body_ptr.

8.2.3.4 operator=()

```
Facet& jeod::Facet::operator= (
            const Facet & )  [delete]
```

8.2.3.5 set_name()

```
void jeod::Facet::set_name (
            std::string name_in )  [inline]
```

Setter for the name.

Definition at line 115 of file facet.hh.

8.2.3.6 update_articulation()

```
void jeod::Facet::update_articulation ( )  [virtual]
```

Definition at line 87 of file facet.cc.

References connections_initialized, jeod::SurfaceModelMessages::initialization_error, and update_articulation_↩internal().

8.2.3.7 update_articulation_internal()

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

Reimplemented in jeod::FlatPlate.

Definition at line 130 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 init_attrjeod__Facet

```
void init_attrjeod__Facet ( )  [friend]
```

8.2.4.2 InputProcessor

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

Definition at line 90 of file facet.hh.

8.2.4.3 SurfaceModel

```
friend class SurfaceModel  [friend]
```

Definition at line 90 of file facet.hh.

8.2.5 Field Documentation

8.2.5.1 area

```
double jeod::Facet::area {}
```

Area of the plate.

trick_units(m2)

Definition at line 163 of file facet.hh.

8.2.5.2 connections_initialized

```
bool jeod::Facet::connections_initialized {}  [protected]
```

Indicates if the mass connections for this Facet have been initialized.

trick_units(–)

Definition at line 193 of file facet.hh.

Referenced by initialize_mass_connection(), and update_articulation().

8.2.5.3 int_pos

```
double jeod::Facet::int_pos[3] {}  [protected]
```

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

trick_units(m)

Definition at line 187 of file facet.hh.

Referenced by update_articulation_internal().

8.2.5.4 local_position

```
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 133 of file facet.hh.

Referenced by update_articulation_internal().

8.2.5.5 mass_body_name

```
std::string 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 153 of file facet.hh.

Referenced by initialize_mass_connection().

8.2.5.6 mass_body_ptr

```
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 abovetrick_units(–)

Definition at line 174 of file facet.hh.

Referenced by get_mass_body_ptr(), and initialize_mass_connection().

8.2.5.7 mass_rel_struct

```
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 181 of file facet.hh.

Referenced by jeod::FlatPlate::update_articulation_internal(), and update_articulation_internal().

8.2.5.8 name

```
std::string jeod::Facet::name
```

Name of the facet.

trick_units(–)

Definition at line 144 of file facet.hh.

8.2.5.9 param_name

```
std::string jeod::Facet::param_name
```

Name of the facet parameters, usually a material type.

Used to match the facet to FacetParamstrick_units(–)

Definition at line 139 of file facet.hh.

8.2.5.10 position

```
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 127 of file facet.hh.

Referenced by update_articulation_internal().

8.2.5.11 temperature

```
double jeod::Facet::temperature {}
```

Kinetic Temperature of the surface.

trick_units(K)

Definition at line 158 of file facet.hh.

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
- virtual ∼FacetParams ()=default
- FacetParams & operator= (const FacetParams &)=delete
- FacetParams (const FacetParams &)=delete
- 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.

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 80 of file facet_params.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 FacetParams() [1/2]

```
jeod::FacetParams::FacetParams ( )  [default]
```

8.3.2.2 ∼FacetParams()

```
virtual jeod::FacetParams::∼FacetParams ( )  [virtual], [default]
```

8.3.2.3 FacetParams() [2/2]

```
jeod::FacetParams::FacetParams (
             const FacetParams &  )  [delete]
```

8.3.3 Member Function Documentation

8.3.3.1 operator=()

```
FacetParams& jeod::FacetParams::operator= (
             const FacetParams &  )  [delete]
```

8.3.3.2 set_name()

```
void jeod::FacetParams::set_name (
             std::string name_in )  [inline]
```

Setter for the name.

Definition at line 96 of file facet_params.hh.

8.3.4 Friends And Related Function Documentation

8.3.4.1 init_attrjeod__FacetParams

```
void init_attrjeod__FacetParams ( )  [friend]
```

8.3.4.2 InputProcessor

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

Definition at line 82 of file facet_params.hh.

8.3.5 Field Documentation

8.3.5.1 name

```
std::string jeod::FacetParams::name
```

Name that will be used to match FacetParams to facets.

trick_units(–)

Definition at line 91 of file facet_params.hh.

Referenced by jeod::InteractionSurfaceFactory::add_facet_params().

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

- facet_params.hh

8.4 jeod::FacetStateInfo Class 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

  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 95 of file surface_model.hh.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 FacetStateInfo() [1/3]

```
jeod::FacetStateInfo::FacetStateInfo ( )  [default]
```

Default constructor to keep the memory manager happy.

8.4.2.2 FacetStateInfo() [2/3]

```
jeod::FacetStateInfo::FacetStateInfo (
            MassBody ∗ new_mass_body )  [inline], [explicit]
```

FacetStateInfo non-default constructor.

Parameters

new_mass_body	The mass body to which this object will refer.

Definition at line 120 of file surface_model.hh.

8.4.2.3 FacetStateInfo() [3/3]

```
jeod::FacetStateInfo::FacetStateInfo (
             MassBody & new_mass_body )  [inline], [explicit]
```

FacetStateInfo non-default constructor.

Parameters

new_mass_body	The mass body to which this object will refer.

Definition at line 129 of file surface_model.hh.

8.4.3 Member Function Documentation

8.4.3.1 operator==()

```
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

rhs	Object to be compared with this object.

Definition at line 139 of file surface_model.hh.

8.4.4 Friends And Related Function Documentation

8.4.4.1 init_attrjeod__FacetStateInfo

```
void init_attrjeod__FacetStateInfo ( )  [friend]
```

8.4.4.2 InputProcessor

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

Definition at line 97 of file surface_model.hh.

8.4.5 Field Documentation

8.4.5.1 mass_body

```
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 109 of file surface_model.hh.

8.4.5.2 mass_state

```
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 103 of file surface_model.hh.

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

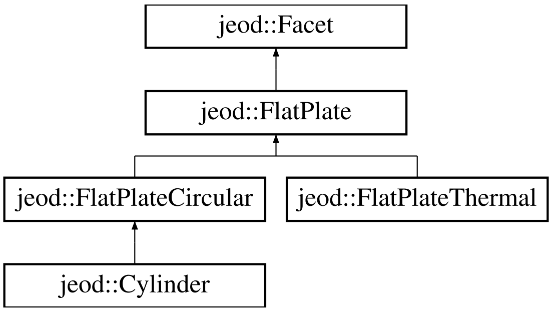
- surface_model.hh

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
- ∼FlatPlate () override=default
- FlatPlate & operator= (const FlatPlate &)=delete
- FlatPlate (const FlatPlate &)=delete

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

- void update_articulation_internal () override

Friends

- class InputProcessor
- void init_attrjeod__FlatPlate ()

Additional Inherited Members

8.5.1 Detailed Description

A FlatPlate implementation of Facet.

Definition at line 83 of file flat_plate.hh.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 FlatPlate() [1/2]

```
jeod::FlatPlate::FlatPlate ( )  [default]
```

8.5.2.2 ∼FlatPlate()

```
jeod::FlatPlate::∼FlatPlate ( )  [override], [default]
```

8.5.2.3 FlatPlate() [2/2]

```
jeod::FlatPlate::FlatPlate (
              const FlatPlate &  )  [delete]
```

8.5.3 Member Function Documentation

8.5.3.1 operator=()

```
FlatPlate& jeod::FlatPlate::operator= (
              const FlatPlate &  )  [delete]
```

8.5.3.2 update_articulation_internal()

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

Reimplemented from jeod::Facet.

Definition at line 44 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 init_attrjeod__FlatPlate

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

8.5.4.2 InputProcessor

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

Definition at line 85 of file flat_plate.hh.

8.5.5 Field Documentation

8.5.5.1 local_normal

```
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 107 of file flat_plate.hh.

Referenced by update_articulation_internal().

8.5.5.2 normal

```
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 functionaity 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 101 of file flat_plate.hh.

Referenced by update_articulation_internal().

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

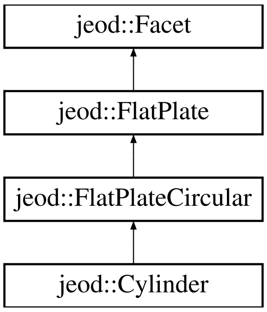
- flat_plate.hh
- flat_plate.cc

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
- ∼FlatPlateCircular () override=default
- FlatPlateCircular & operator= (const FlatPlateCircular &)=delete
- FlatPlateCircular (const FlatPlateCircular &)=delete

Data Fields

- double radius {}

  Radius of the plate.

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 74 of file flat_plate_circular.hh.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 FlatPlateCircular() [1/2]

```
jeod::FlatPlateCircular::FlatPlateCircular ( )  [default]
```

8.6.2.2 ∼FlatPlateCircular()

```
jeod::FlatPlateCircular::∼FlatPlateCircular ( )  [override], [default]
```

8.6.2.3 FlatPlateCircular() [2/2]

```
jeod::FlatPlateCircular::FlatPlateCircular (
                const FlatPlateCircular &  )  [delete]
```

8.6.3 Member Function Documentation

8.6.3.1 operator=()

```
FlatPlateCircular& jeod::FlatPlateCircular::operator= (
                const FlatPlateCircular &  )  [delete]
```

8.6.4 Friends And Related Function Documentation

8.6.4.1 init_attrjeod__FlatPlateCircular

```
void init_attrjeod__FlatPlateCircular ( )  [friend]
```

8.6.4.2 InputProcessor

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

Definition at line 76 of file flat_plate_circular.hh.

8.6.5 Field Documentation

8.6.5.1 radius

```
double jeod::FlatPlateCircular::radius {}
```

Radius of the plate.

trick_units(m)

Definition at line 85 of file flat_plate_circular.hh.

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

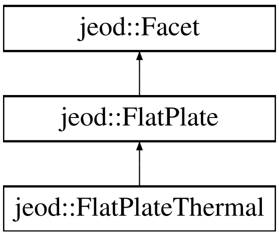
- flat_plate_circular.hh

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 ()=default
- ∼FlatPlateThermal () override=default
- FlatPlateThermal & operator= (const FlatPlateThermal &)=delete
- FlatPlateThermal (const FlatPlateThermal &)=delete

Data Fields

- ThermalFacetRider thermal

  Thermal characteristics rider.

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 80 of file flat_plate_thermal.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 FlatPlateThermal() [1/2]

```
jeod::FlatPlateThermal::FlatPlateThermal ( )  [default]
```

8.7.2.2 ∼FlatPlateThermal()

```
jeod::FlatPlateThermal::∼FlatPlateThermal ( )  [override], [default]
```

8.7.2.3 FlatPlateThermal() [2/2]

```
jeod::FlatPlateThermal::FlatPlateThermal (
            const FlatPlateThermal &  )  [delete]
```

8.7.3 Member Function Documentation

8.7.3.1 operator=()

```
FlatPlateThermal& jeod::FlatPlateThermal::operator= (
            const FlatPlateThermal &  )  [delete]
```

8.7.4 Friends And Related Function Documentation

8.7.4.1 init_attrjeod__FlatPlateThermal

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

8.7.4.2 InputProcessor

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

Definition at line 82 of file flat_plate_thermal.hh.

8.7.5 Field Documentation

8.7.5.1 thermal

```
ThermalFacetRider jeod::FlatPlateThermal::thermal
```

Thermal characteristics rider.

trick_units(–)

Definition at line 91 of file flat_plate_thermal.hh.

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

- flat_plate_thermal.hh

8.8 jeod::InteractionFacet Class Reference

A base class for an interaction specific facet.

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

Public Member Functions

- InteractionFacet ()=default
- virtual ∼InteractionFacet ()=default
- InteractionFacet & operator= (const InteractionFacet &)=delete
- InteractionFacet (const InteractionFacet &)=delete

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.

Friends

- class InputProcessor
- void init_attrjeod__InteractionFacet ()

8.8.1 Detailed Description

A base class for an interaction specific facet.

Definition at line 80 of file interaction_facet.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 InteractionFacet() [1/2]

```
jeod::InteractionFacet::InteractionFacet ( )  [default]
```

8.8.2.2 ∼InteractionFacet()

```
virtual jeod::InteractionFacet::∼InteractionFacet ( )  [virtual], [default]
```

8.8.2.3 InteractionFacet() [2/2]

```
jeod::InteractionFacet::InteractionFacet (
            const InteractionFacet &  )  [delete]
```

8.8.3 Member Function Documentation

8.8.3.1 operator=()

```
InteractionFacet& jeod::InteractionFacet::operator= (
             const InteractionFacet &  )  [delete]
```

8.8.4 Friends And Related Function Documentation

8.8.4.1 init_attrjeod__InteractionFacet

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

8.8.4.2 InputProcessor

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

Definition at line 82 of file interaction_facet.hh.

8.8.5 Field Documentation

8.8.5.1 base_facet

```
Facet* jeod::InteractionFacet::base_facet {}
```

The original facet from which this interaction facet was created.

trick_units(–)

Definition at line 101 of file interaction_facet.hh.

8.8.5.2 force

```
double jeod::InteractionFacet::force[3] {}
```

The force on the facet caused by the environment interaction.

trick_units(N)

Definition at line 91 of file interaction_facet.hh.

8.8.5.3 torque

```
double jeod::InteractionFacet::torque[3] {}
```

The torque on the facet caused by the environment interaction.

trick_units(N∗m)

Definition at line 96 of file interaction_facet.hh.

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

- interaction_facet.hh

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
- virtual ∼InteractionFacetFactory ()=default
- InteractionFacetFactory & operator= (const InteractionFacetFactory &)=delete
- InteractionFacetFactory (const InteractionFacetFactory &)=delete
- 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.

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 85 of file interaction_facet_factory.hh.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 InteractionFacetFactory() [1/2]

```
jeod::InteractionFacetFactory::InteractionFacetFactory ( )  [default]
```

8.9.2.2 ∼InteractionFacetFactory()

```
virtual jeod::InteractionFacetFactory::∼InteractionFacetFactory ( )  [virtual], [default]
```

8.9.2.3 InteractionFacetFactory() [2/2]

```
jeod::InteractionFacetFactory::InteractionFacetFactory (
             const InteractionFacetFactory &  )  [delete]
```

8.9.3 Member Function Documentation

8.9.3.1 create_facet()

```
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	*facet*	The facet the InteractionFacet is created from
in	*params*	The parameter object to be added.

8.9.3.2 is_correct_factory()

```
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	*facet*	The facet that is being checked

8.9.3.3 operator=()

```
InteractionFacetFactory& jeod::InteractionFacetFactory::operator= (
            const InteractionFacetFactory & )  [delete]
```

8.9.4 Friends And Related Function Documentation

8.9.4.1 init_attrjeod__InteractionFacetFactory

```
void init_attrjeod__InteractionFacetFactory ( )  [friend]
```

8.9.4.2 InputProcessor

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

Definition at line 87 of file interaction_facet_factory.hh.

8.9.5 Field Documentation

8.9.5.1 trick_bool

```
bool jeod::InteractionFacetFactory::trick_bool {}  [protected]
```

Unused data field to expedite dynamic allocation in Trick environment.

trick_units(–)

Definition at line 119 of file interaction_facet_factory.hh.

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

- interaction_facet_factory.hh

8.10 jeod::InteractionSurface Class Reference

A base class for interaction specific surfaces.

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

Public Member Functions

- InteractionSurface ()=default
- virtual ∼InteractionSurface ()=default
- InteractionSurface & operator= (const InteractionSurface &)=delete
- InteractionSurface (const InteractionSurface &)=delete
- virtual void accumulate_thermal_sources ()

  Adds all thermal sources together.
- virtual void thermal_integrator ()

  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.

Friends

- class InputProcessor
- void init_attrjeod__InteractionSurface ()

8.10.1 Detailed Description

A base class for interaction specific surfaces.

Definition at line 84 of file interaction_surface.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 InteractionSurface() [1/2]

```
jeod::InteractionSurface::InteractionSurface ( )  [default]
```

8.10.2.2 ∼InteractionSurface()

```
virtual jeod::InteractionSurface::∼InteractionSurface ( )  [virtual], [default]
```

8.10.2.3 InteractionSurface() [2/2]

```
jeod::InteractionSurface::InteractionSurface (
            const InteractionSurface &  )  [delete]
```

8.10.3 Member Function Documentation

8.10.3.1 accumulate_thermal_sources()

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

Adds all thermal sources together.

Definition at line 100 of file interaction_surface.hh.

8.10.3.2 allocate_array()

```
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

in	*size*	Size of the array to be allocated Units: cnt

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

8.10.3.3 allocate_interaction_facet()

```
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

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

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

8.10.3.4 operator=()

```
InteractionSurface& jeod::InteractionSurface::operator= (
            const InteractionSurface &  )  [delete]
```

8.10.3.5 thermal_integrator()

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

Integrates thermal sources to get temperature.

Definition at line 109 of file interaction_surface.hh.

8.10.4 Friends And Related Function Documentation

8.10.4.1 init_attrjeod__InteractionSurface

```
void init_attrjeod__InteractionSurface ( ) [friend]
```

8.10.4.2 InputProcessor

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

Definition at line 86 of file interaction_surface.hh.

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

- interaction_surface.hh

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.
- InteractionSurfaceFactory & operator= (const InteractionSurfaceFactory &)=delete
- InteractionSurfaceFactory (const InteractionSurfaceFactory &)=delete
- 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.

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 88 of file interaction_surface_factory.hh.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 InteractionSurfaceFactory() [1/2]

```
jeod::InteractionSurfaceFactory::InteractionSurfaceFactory ( )
```

Default Constructor.

Definition at line 58 of file interaction_surface_factory.cc.

References factories, and params.

8.11.2.2 ∼InteractionSurfaceFactory()

```
jeod::InteractionSurfaceFactory::∼InteractionSurfaceFactory ( )  [virtual]
```

Destructor.

Definition at line 71 of file interaction_surface_factory.cc.

References factories, and params.

8.11.2.3 InteractionSurfaceFactory() [2/2]

```
jeod::InteractionSurfaceFactory::InteractionSurfaceFactory (
             const InteractionSurfaceFactory &  )  [delete]
```

8.11.3 Member Function Documentation

8.11.3.1 add_facet_factory()

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

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

Parameters

in	*to_add*	The interaction facet factory to add

Definition at line 183 of file interaction_surface_factory.cc.

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

8.11.3.2 add_facet_params()

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

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

Parameters

in	*to_add*	The facet parameters to add

Definition at line 203 of file interaction_surface_factory.cc.

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

8.11.3.3 create_surface() [1/2]

```
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

in	*surface*	The surface model used to create the interaction surface
out	*inter_surface*	Where the interaction surface will be produced

Definition at line 86 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.

8.11.3.4 create_surface() [2/2]

```
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 105 of file interaction_surface_factory.hh.

8.11.3.5 operator=()

```
InteractionSurfaceFactory& jeod::InteractionSurfaceFactory::operator= (
             const InteractionSurfaceFactory & )  [delete]
```

8.11.4 Friends And Related Function Documentation

8.11.4.1 init_attrjeod__InteractionSurfaceFactory

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

8.11.4.2 InputProcessor

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

Definition at line 90 of file interaction_surface_factory.hh.

8.11.5 Field Documentation

8.11.5.1 factories

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

A vector of interaction facet factories to be used.

Matched to facets by typetrick_io(∗∗)

Definition at line 114 of file interaction_surface_factory.hh.

Referenced by add_facet_factory(), create_surface(), InteractionSurfaceFactory(), and ∼InteractionSurface←Factory().

8.11.5.2 params

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

A vector of FacetParams to be used.

trick_io(∗∗)

Definition at line 119 of file interaction_surface_factory.hh.

Referenced by add_facet_params(), create_surface(), InteractionSurfaceFactory(), and ∼InteractionSurface←Factory().

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.
- SurfaceModel & operator= (const SurfaceModel &)=delete
- SurfaceModel (const SurfaceModel &)=delete
- 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.
- std::string 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.
- JeodPointerList< FacetStateInfo >::type articulation_states

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

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 149 of file surface_model.hh.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 SurfaceModel() [1/2]

```
jeod::SurfaceModel::SurfaceModel ( )
```

Default constructor.

Definition at line 56 of file surface_model.cc.

References articulation_states, and facets.

8.12.2.2 ∼SurfaceModel()

```
jeod::SurfaceModel::∼SurfaceModel ( )
```

Destructor.

Definition at line 69 of file surface_model.cc.

References articulation_states, and facets.

8.12.2.3 SurfaceModel() [2/2]

```
jeod::SurfaceModel::SurfaceModel (
            const SurfaceModel & ) [delete]
```

8.12.3 Member Function Documentation

8.12.3.1 add_facet()

```
void jeod::SurfaceModel::add_facet (
            Facet * new_facet )
```

Definition at line 121 of file surface_model.cc.

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

8.12.3.2 add_facets()

```
void jeod::SurfaceModel::add_facets (
            Facet ** new_facets,
            unsigned int num_new_facets )
```

Definition at line 87 of file surface_model.cc.

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

8.12.3.3 initialize_mass_connections()

```
void jeod::SurfaceModel::initialize_mass_connections (
             BaseDynManager & manager )
```

Definition at line 143 of file surface_model.cc.

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

8.12.3.4 operator=()

```
SurfaceModel& jeod::SurfaceModel::operator= (
             const SurfaceModel & )  [delete]
```

8.12.3.5 update_articulation()

```
void jeod::SurfaceModel::update_articulation ( )
```

Definition at line 202 of file surface_model.cc.

References articulation_active, articulation_states, facets, jeod::SurfaceModelMessages::runtime_error, struct_body_name, and struct_body_ptr.

8.12.4 Friends And Related Function Documentation

8.12.4.1 init_attrjeod__SurfaceModel

```
void init_attrjeod__SurfaceModel ( )  [friend]
```

8.12.4.2 InputProcessor

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

Definition at line 151 of file surface_model.hh.

8.12.5 Field Documentation

8.12.5.1 articulation_active

```
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 falsetrick_units(–)

Definition at line 162 of file surface_model.hh.

Referenced by update_articulation().

8.12.5.2 articulation_states

```
JeodPointerList<FacetStateInfo>::type jeod::SurfaceModel::articulation_states  [protected]
```

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

trick_io(∗∗)

Definition at line 201 of file surface_model.hh.

Referenced by initialize_mass_connections(), SurfaceModel(), update_articulation(), and ∼SurfaceModel().

8.12.5.3 facets

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

The facets that make up the surface.

trick_io(∗∗)

Definition at line 185 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 struct_body_name

```
std::string 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 articulationtrick_units(–)

Definition at line 180 of file surface_model.hh.

Referenced by initialize_mass_connections(), and update_articulation().

8.12.5.5 struct_body_ptr

```
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 articulationtrick_units(–)

Definition at line 195 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>
```

Public Member Functions

- SurfaceModelMessages ()=delete
- SurfaceModelMessages (const SurfaceModelMessages &rhs)=delete
- SurfaceModelMessages & operator= (const SurfaceModelMessages &rhs)=delete

Static Public Attributes

- static const char ∗ initialization_error = "utils/surface_model/" "initialization_error"

  Represents an error in initialization.
- static const char ∗ setup_error = "utils/surface_model/" "setup_error"

  Represents an error in setup of the surface model.
- static const char ∗ runtime_error = "utils/surface_model/" "runtime_error"

  Represents an error during the runtime of the surface model.

Friends

- class InputProcessor
- void init_attrjeod__SurfaceModelMessages ()

8.13.1 Detailed Description

Messages associated with the use of the surface model.

Definition at line 89 of file surface_model_messages.hh.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 SurfaceModelMessages() [1/2]

```
jeod::SurfaceModelMessages::SurfaceModelMessages ( )  [delete]
```

8.13.2.2 SurfaceModelMessages() [2/2]

```
jeod::SurfaceModelMessages::SurfaceModelMessages (
             const SurfaceModelMessages & rhs )  [delete]
```

8.13.3 Member Function Documentation

8.13.3.1 operator=()

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

8.13.4 Friends And Related Function Documentation

8.13.4.1 init_attrjeod__SurfaceModelMessages

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

8.13.4.2 InputProcessor

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

Definition at line 91 of file surface_model_messages.hh.

8.13.5 Field Documentation

8.13.5.1 initialization_error

```
char const * jeod::SurfaceModelMessages::initialization_error = "utils/surface_model/" "initialization↩
_error"  [static]
```

Represents an error in initialization.

trick_units(--)

Definition at line 99 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 runtime_error

```
char const * jeod::SurfaceModelMessages::runtime_error = "utils/surface_model/" "runtime_↩
error"  [static]
```

Represents an error during the runtime of the surface model.

trick_units(--)

Definition at line 109 of file surface_model_messages.hh.

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

8.13.5.3 setup_error

```
char const * jeod::SurfaceModelMessages::setup_error = "utils/surface_model/" "setup_error"
[static]
```

Represents an error in setup of the surface model.

trick_units(--)

Definition at line 104 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.

9.2 cylinder.hh File Reference

cylinders for use in the surface model and the contact model

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

Data Structures

- class jeod::Cylinder

    An cylinder implementation of Facet.

Namespaces

- jeod

    Namespace jeod.

9.2.1 Detailed Description

cylinders for use in the surface model and the contact model

9.3 facet.cc File Reference

Individual facets for use in the surface model.

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

Namespaces

- jeod

  Namespace jeod.

9.3.1 Detailed Description

Individual facets for use in the surface model.

9.4 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.

9.4.1 Detailed Description

Individual facets for use in the surface model.

9.5 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.

9.5.1 Detailed Description

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

9.6 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.

9.6.1 Detailed Description

Flat plates for use in the surface model.

9.7 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.

9.7.1 Detailed Description

Flat plates for use in the surface model.

9.8 flat_plate_circular.hh File Reference

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

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

Data Structures

- class jeod::FlatPlateCircular

  An circular flat plate implementation of Facet.

Namespaces

- jeod

  Namespace jeod.

9.8.1 Detailed Description

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

9.9 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.

9.9.1 Detailed Description

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

9.10 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.

9.10.1 Detailed Description

Individual facets for use with specific environment interaction models.

9.11 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.

9.11.1 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

9.12 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.

9.12.1 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 inheritied, instantiable class

9.13 interaction_surface_factory.cc File Reference

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

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

Namespaces

- jeod

  Namespace jeod.

9.13.1 Detailed Description

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

9.14 interaction_surface_factory.hh File Reference

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

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

Data Structures

- class jeod::InteractionSurfaceFactory

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

Namespaces

- jeod

  Namespace jeod.

9.14.1 Detailed Description

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

9.15 surface_model.cc File Reference

Vehicle surface model for general environment interaction models.

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

Namespaces

- jeod

  Namespace jeod.

9.15.1 Detailed Description

Vehicle surface model for general environment interaction models.

9.16 surface_model.hh File Reference

Vehicle surface model for general environment interaction models.

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

Data Structures

- class 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.

9.16.1 Detailed Description

Vehicle surface model for general environment interaction models.

9.17 surface_model_messages.cc File Reference

Implement surface_model_messages.

```
#include "utils/message/include/make_message_code.hh"
#include "../include/surface_model_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define MAKE_SURFACEMODEL_MESSAGE_CODE(id) JEOD_MAKE_MESSAGE_CODE(Surface↩ ModelMessages, "utils/surface_model/", id)

9.17.1 Detailed Description

Implement surface_model_messages.

9.17.2 Macro Definition Documentation

9.17.2.1 MAKE_SURFACEMODEL_MESSAGE_CODE

```
#define MAKE_SURFACEMODEL_MESSAGE_CODE(
              id ) JEOD_MAKE_MESSAGE_CODE(SurfaceModelMessages, "utils/surface_model/", id)
```

Definition at line 43 of file surface_model_messages.cc.

9.18 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.

9.18.1 Detailed Description

Implement surface_model_messages.

Index