RadiationPressureModel

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

- Interactions

6.1.1 Detailed Description

6.2 Interactions

Modules

- RadiationPressure

6.2.1 Detailed Description

6.3 RadiationPressure

Files

- file flat_plate_radiation_facet.hh

  Individual facets for use with rad environment interaction models.
- file flat_plate_radiation_factory.hh

  Factory that creates an interaction facet, for a specific environment interaction model, from a facet model.
- file radiation_base_facet.hh

  Individual facets for use with radiation environment interaction models.
- file radiation_default_surface.hh

  Individual facets for use with radiation environment interaction models.
- file radiation_facet.hh

  Individual facets for use with radiation environment interaction models.
- file radiation_messages.hh

  Define the class RadiationMessages, the class that specifies the message IDs used in the Radiation model.
- file radiation_params.hh

  A virtual base class for radiation facet parameters, used to create interaction facets for radiation in the Interaction↩ SurfaceFactorys.
- file radiation_pressure.hh

  Radiation pressure top-level definition.
- file radiation_source.hh

  Radiation pressure parameter and variable definitions.
- file radiation_surface.hh

  Vehicle surface model for general environment interaction models.
- file radiation_surface_factory.hh

  Factory that creates an interaction surface, for a specific environment interaction model, from a surface model.
- file radiation_third_body.hh

  Define the class RadiationThirdBody.
- file flat_plate_radiation_facet.cc

  Define member functions for class FlatPlateRadiationFacet.
- file flat_plate_radiation_factory.cc

  Factory that creates a FlatPlateRadiationFacet, from a facet model.
- file radiation_base_facet.cc

  Define member functions for class RadiationBaseFacet.
- file radiation_default_surface.cc

  Default surface for use with Radiation Pressure interaction model.
- file radiation_facet.cc

  Individual facets for use with Radiation Pressure interaction models.
- file radiation_messages.cc

  Implement the class RadiationMessages.
- file radiation_pressure.cc

  Calculation of force and torque due to radiation pressure.
- file radiation_pressure__default_surface.cc

  Calculation of force and torque due to radiation pressure.
- file radiation_pressure__surface_model.cc

  Calculation of force and torque due to radiation pressure.
- file radiation_source.cc

  Definition of methods associated with Primary Sources.
- file radiation_surface.cc

Vehicle surface model for Radiation Pressure model.

- file radiation_surface_factory.cc

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

- file radiation_third_body.cc

  Provides the functionality associated with RadiationThirdBodys.

Namespaces

- jeod

  Namespace jeod.

6.3.1 Detailed Description

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class FlatPlateRadiationFacet

  A flat plate facet to be used for radiation interaction.
- class FlatPlateRadiationFactory

  The factory for building flat plate radiation facets.
- class RadiationBaseFacet

  Generic type of facet for radiation pressure.
- class RadiationDefaultSurface

  The default spherical surface for radiation pressure.
- class RadiationFacet

  Generic type of facet for radiation pressure.
- class RadiationMessages

  Provides error messages.
- class RadiationParams

  Provides a parameter list to each facet, based on the facet material properties.
- class RadiationPressure

  Radiation pressure top-level definition.
- class RadiationSource

  Provides information on the source of the incident radiation.
- class RadiationSurface

  The surface of the vehicle that interacts with the incident flux.
- class RadiationSurfaceFactory

  The factory for creating Radiation Surfaces.
- class RadiationThirdBody

  Provide information on bodies that may cause shadowing or reflected illumination.

7.1.1 Detailed Description

Namespace jeod.

Chapter 8

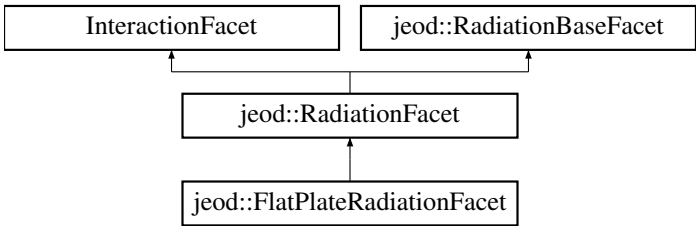
Data Structure Documentation

8.1 jeod::FlatPlateRadiationFacet Class Reference

A flat plate facet to be used for radiation interaction.

`#include <flat_plate_radiation_facet.hh>`

Inheritance diagram for jeod::FlatPlateRadiationFacet:



Public Member Functions

- FlatPlateRadiationFacet ()=default
- ~FlatPlateRadiationFacet () override=default
- FlatPlateRadiationFacet & operator= (const FlatPlateRadiationFacet &)=delete
- FlatPlateRadiationFacet (const FlatPlateRadiationFacet &)=delete
- void incident_radiation (const double flux_mag, const double flux_struct_hat[3], const bool calculate_forces) override

  Calculation of force and torque due to radiation pressure.
- void initialize_geom (double center_grav[3]) override

  Initializes the Facet for use in the model.
- void define_facet (FlatPlate *flat_plate)

  Defines the facet data values.
- void radiation_pressure () override

  Calculates the radiative emission force, accumulated force, and torque acting on a facet.

Data Fields

- double * normal {}

  Unit vector normal to the plate surface, pointing outward (structural frame).
- double incident_flux_hat [3] {}

  Temporary value.

Private Attributes

- double sin_theta {}

  Theta is the angle between the plate and the radiation vector.

Friends

- class InputProcessor
- void init_attrjeod__FlatPlateRadiationFacet ()

Additional Inherited Members

8.1.1 Detailed Description

A flat plate facet to be used for radiation interaction.

Definition at line 86 of file flat_plate_radiation_facet.hh.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 FlatPlateRadiationFacet() [1/2]

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

8.1.2.2 ~FlatPlateRadiationFacet()

```
jeod::FlatPlateRadiationFacet::~FlatPlateRadiationFacet ( )  [override], [default]
```

8.1.2.3 FlatPlateRadiationFacet() [2/2]

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

8.1.3 Member Function Documentation

8.1.3.1 define_facet()

```
void jeod::FlatPlateRadiationFacet::define_facet (
            FlatPlate * flat_plate )
```

Defines the facet data values.

Parameters

in	*flat_plate*	pointer to the flat plate object

Definition at line 54 of file flat_plate_radiation_facet.cc.

References jeod::RadiationFacet::center_pressure, and normal.

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

8.1.3.2 incident_radiation()

```
void jeod::FlatPlateRadiationFacet::incident_radiation (
             const double flux_mag,
             const double flux_struct_hat[3],
             const bool calculate_forces )  [override], [virtual]
```

Calculation of force and torque due to radiation pressure.

Assumptions and Limitations

- Only called when flux_mag $>$ 0

Parameters

in	*flux_mag*	incident flux (per unit area) Units: N/m
in	*flux_struct_hat*	the flux unit vector in structural frame
in	*calculate_forces*	on/off flag for whether to calculate forces.

Implements jeod::RadiationBaseFacet.

Definition at line 85 of file flat_plate_radiation_facet.cc.

References jeod::RadiationBaseFacet::albedo, jeod::RadiationBaseFacet::areaxflux, jeod::RadiationBaseFacet::areaxflux_e, jeod::RadiationBaseFacet::cx_area, jeod::RadiationBaseFacet::diffuse, jeod::RadiationBaseFacet::F_absorption, jeod::RadiationBaseFacet::F_diffuse, jeod::RadiationBaseFacet::F_specular, normal, sin_theta, jeod::RadiationBaseFacet::speed_of_light, jeod::RadiationBaseFacet::thermal, and jeod::RadiationFacet::two_thirds.

8.1.3.3 initialize_geom()

```
void jeod::FlatPlateRadiationFacet::initialize_geom (
             double center_grav[3] )  [override], [virtual]
```

Initializes the Facet for use in the model.

Parameters

in	*center_grav*	center of gravity position Units: M

Implements jeod::RadiationFacet.

Definition at line 68 of file flat_plate_radiation_facet.cc.

References jeod::RadiationFacet::center_pressure, jeod::RadiationFacet::crot_to_cp, jeod::RadiationFacet::initialize(), and jeod::RadiationBaseFacet::thermal.

8.1.3.4 operator=()

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

8.1.3.5 radiation_pressure()

```
void jeod::FlatPlateRadiationFacet::radiation_pressure ( )  [override], [virtual]
```

Calculates the radiative emission force, accumulated force, and torque acting on a facet.

Implements jeod::RadiationFacet.

Definition at line 135 of file flat_plate_radiation_facet.cc.

References jeod::RadiationFacet::crot_to_cp, jeod::RadiationBaseFacet::F_absorption, jeod::RadiationBaseFacet::F_diffuse, jeod::RadiationBaseFacet::F_emission, jeod::RadiationBaseFacet::F_specular, normal, jeod::RadiationBaseFacet::speed_of_light, jeod::RadiationBaseFacet::thermal, jeod::RadiationFacet::two_thirds, and jeod::RadiationMessages::unknown_numerical_error.

8.1.4 Friends And Related Function Documentation

8.1.4.1 init_attrjeod__FlatPlateRadiationFacet

```
void init_attrjeod__FlatPlateRadiationFacet ( )  [friend]
```

8.1.4.2 InputProcessor

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

Definition at line 88 of file flat_plate_radiation_facet.hh.

8.1.5 Field Documentation

8.1.5.1 incident_flux_hat

```
double jeod::FlatPlateRadiationFacet::incident_flux_hat[3] {}
```

Temporary value.

trick_units(–)

Definition at line 101 of file flat_plate_radiation_facet.hh.

8.1.5.2 normal

```
double* jeod::FlatPlateRadiationFacet::normal {}
```

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

once the radiation surface is initialized, it points to the normal found in FlatPlatetrick_units(–)

Definition at line 96 of file flat_plate_radiation_facet.hh.

Referenced by define_facet(), incident_radiation(), and radiation_pressure().

8.1.5.3 sin_theta

```
double jeod::FlatPlateRadiationFacet::sin_theta {}  [private]
```

Theta is the angle between the plate and the radiation vector.

trick_units(–)

Definition at line 107 of file flat_plate_radiation_facet.hh.

Referenced by incident_radiation().

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

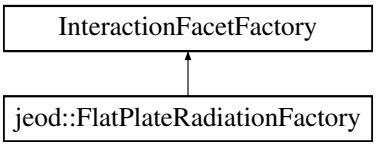
- flat_plate_radiation_facet.hh
- flat_plate_radiation_facet.cc

8.2 jeod::FlatPlateRadiationFactory Class Reference

The factory for building flat plate radiation facets.

`#include <flat_plate_radiation_factory.hh>`

Inheritance diagram for jeod::FlatPlateRadiationFactory:



Public Member Functions

- FlatPlateRadiationFactory ()

  Constructor for FlatPlateRadiationFactory.
- ~FlatPlateRadiationFactory () override=default
- FlatPlateRadiationFactory & operator= (const FlatPlateRadiationFactory &)=delete
- FlatPlateRadiationFactory (const FlatPlateRadiationFactory &)=delete
- InteractionFacet * create_facet (Facet *facet, FacetParams *params) override

  Records the data for the Flat Plate Radiation Facet.
- bool is_correct_factory (Facet *facet) override

  Tests to ensure that the factory can function on the facet as intended.

Friends

- class InputProcessor
- void init_attrjeod__FlatPlateRadiationFactory ()

8.2.1 Detailed Description

The factory for building flat plate radiation facets.

Definition at line 84 of file flat_plate_radiation_factory.hh.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 FlatPlateRadiationFactory() [1/2]

`jeod::FlatPlateRadiationFactory::FlatPlateRadiationFactory ( )`

Constructor for FlatPlateRadiationFactory.

Definition at line 59 of file flat_plate_radiation_factory.cc.

8.2.2.2 ~FlatPlateRadiationFactory()

```
jeod::FlatPlateRadiationFactory::~FlatPlateRadiationFactory ( )  [override], [default]
```

8.2.2.3 FlatPlateRadiationFactory() [2/2]

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

8.2.3 Member Function Documentation

8.2.3.1 create_facet()

```
InteractionFacet * jeod::FlatPlateRadiationFactory::create_facet (
            Facet * facet,
            FacetParams * params )  [override]
```

Records the data for the Flat Plate Radiation Facet.

Returns

pointer to the interaction facet that this function creates.

Parameters

in	*facet*	pointer to the facet
in	*params*	pointer to the set of parameters for the facet.

Definition at line 70 of file flat_plate_radiation_factory.cc.

References jeod::FlatPlateRadiationFacet::define_facet(), and jeod::RadiationMessages::invalid_setup_error.

8.2.3.2 is_correct_factory()

```
bool jeod::FlatPlateRadiationFactory::is_correct_factory (
            Facet * facet )  [override]
```

Tests to ensure that the factory can function on the facet as intended.

Returns

Boolean, is this the correct factory?

Parameters

in	*facet*	pointer to the facet being manipulated by the factory

Definition at line 130 of file flat_plate_radiation_factory.cc.

8.2.3.3 operator=()

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

8.2.4 Friends And Related Function Documentation

8.2.4.1 init_attrjeod__FlatPlateRadiationFactory

```
void init_attrjeod__FlatPlateRadiationFactory ( )  [friend]
```

8.2.4.2 InputProcessor

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

Definition at line 86 of file flat_plate_radiation_factory.hh.

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

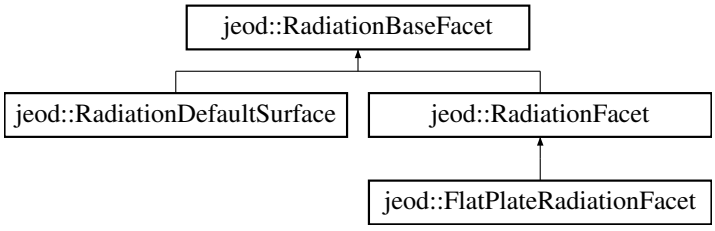
- flat_plate_radiation_factory.hh
- flat_plate_radiation_factory.cc

8.3 jeod::RadiationBaseFacet Class Reference

Generic type of facet for radiation pressure.

```
#include <radiation_base_facet.hh>
```

Inheritance diagram for jeod::RadiationBaseFacet:



Public Member Functions

- RadiationBaseFacet ()=default
- virtual ~RadiationBaseFacet ()=default
- RadiationBaseFacet & operator= (const RadiationBaseFacet &)=delete
- RadiationBaseFacet (const RadiationBaseFacet &)=delete
- virtual void initialize ()

  initializes the base surface
- virtual void interact_with_third_body (RadiationThirdBody *third_body, const bool calc_forces)

  Calculation of force, torques and irradiance due to ThirdBody flux.
- virtual void initialize_runtime_values ()

  To initialize the values during each update run.
- virtual void incident_radiation (const double flux_mag, const double flux_hat[3], const bool calc_forc)=0

  Calculates the effect on the facet of the incident radiation.

Data Fields

- double albedo {-1.0}

  Usable value of albedo, set to either albedo_IR or albedo_vis, depending on situation.
- double albedo_vis {-1.0}

  Fraction of incident visible radiation that is immediately reflected.
- double albedo_IR {-1.0}

  Fraction of incident IR radiation that is immediately reflected.
- double diffuse {-1.0}

  Fraction of reflected radiation that is reflected diffusely (balance reflected specularly)
- ThermalFacetRider thermal

  thermal characteristics of the facet.
- double cx_area {}

  cross-sectional area projected perpendicular to the radiation vector.
- double areaxflux {}

  product of momentum flux and cross-sectional area (cx_area).
- double areaxflux_e {}

  product of energy flux and cross-sectional area
- double F_absorption [3] {}

  Force due to photon absorption from ONLY ONE source.
- double F_specular [3] {}

  Force due to photon specular reflection from ONLY ONE source.
- double F_diffuse [3] {}

  Force due to photon diffuse reflection from ONLY ONE source.
- double F_emission [3] {}

  Force due to photon (thermal) emission.

Static Protected Attributes

- static constexpr double speed_of_light {299792458.0}

  Speed of light in vacuum.

Friends

- class InputProcessor
- void init_attrjeod__RadiationBaseFacet ()

8.3.1 Detailed Description

Generic type of facet for radiation pressure.

Definition at line 86 of file radiation_base_facet.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 RadiationBaseFacet() [1/2]

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

8.3.2.2 ~RadiationBaseFacet()

```
virtual jeod::RadiationBaseFacet::~RadiationBaseFacet ( )  [virtual], [default]
```

8.3.2.3 RadiationBaseFacet() [2/2]

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

8.3.3 Member Function Documentation

8.3.3.1 incident_radiation()

```
virtual void jeod::RadiationBaseFacet::incident_radiation (
            const double flux_mag,
            const double flux_hat[3],
            const bool calc_forc )  [pure virtual]
```

Calculates the effect on the facet of the incident radiation.

Parameters

in	*flux_mag*	Magnitude of the incident flux
in	*flux_hat*	unit vector associated with the incident flux vector
in	*calc_forc*	true/false, do forces get calculated

Implemented in jeod::RadiationDefaultSurface, and jeod::FlatPlateRadiationFacet.

Referenced by jeod::RadiationSurface::incident_radiation().

8.3.3.2 initialize()

```
void jeod::RadiationBaseFacet::initialize ( )  [virtual]
```

initializes the base surface

Reimplemented in jeod::RadiationFacet, and jeod::RadiationDefaultSurface.

Definition at line 50 of file radiation_base_facet.cc.

References albedo, albedo_IR, albedo_vis, and jeod::RadiationMessages::invalid_setup_error.

Referenced by jeod::RadiationDefaultSurface::initialize(), and jeod::RadiationFacet::initialize().

8.3.3.3 initialize_runtime_values()

```
void jeod::RadiationBaseFacet::initialize_runtime_values ( )  [virtual]
```

To initialize the values during each update run.

Definition at line 110 of file radiation_base_facet.cc.

References F_absorption, F_diffuse, F_specular, and thermal.

Referenced by jeod::RadiationSurface::initialize_runtime_values(), and jeod::RadiationPressure::update_default↩
_surface().

8.3.3.4 interact_with_third_body()

```
void jeod::RadiationBaseFacet::interact_with_third_body (
            RadiationThirdBody * third_body_ptr,
            const bool calculate_forces )  [virtual]
```

Calculation of force, torques and irradiance due to ThirdBody flux.

Parameters

in,out	*third_body_ptr*	Third body that emits radiation
in	*calculate_forces*	Calculate forces on this facet if true.

Definition at line 95 of file radiation_base_facet.cc.

References jeod::RadiationThirdBody::accumulate_rad_flux(), jeod::RadiationThirdBody::accumulate_refl_flux(), albedo, albedo_IR, and albedo_vis.

Referenced by jeod::RadiationSurface::interact_with_third_body(), and jeod::RadiationPressure::update_default_surface().

8.3.3.5 operator=()

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

8.3.4 Friends And Related Function Documentation

8.3.4.1 init_attrjeod__RadiationBaseFacet

```
void init_attrjeod__RadiationBaseFacet ( )  [friend]
```

8.3.4.2 InputProcessor

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

Definition at line 88 of file radiation_base_facet.hh.

8.3.5 Field Documentation

8.3.5.1 albedo

```
double jeod::RadiationBaseFacet::albedo {-1.0}
```

Usable value of albedo, set to either albedo_IR or albedo_vis, depending on situation.

trick_units(–)

Definition at line 95 of file radiation_base_facet.hh.

Referenced by jeod::RadiationFacet::define_facet_core(), jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_radiation(), jeod::RadiationDefaultSurface::initialize(), jeod::RadiationFacet::initialize(), initialize(), and interact_with_third_body().

8.3.5.2 albedo_IR

```
double jeod::RadiationBaseFacet::albedo_IR {-1.0}
```

Fraction of incident IR radiation that is immediately reflected.

trick_units(–)

Definition at line 105 of file radiation_base_facet.hh.

Referenced by jeod::RadiationFacet::initialize(), initialize(), and interact_with_third_body().

8.3.5.3 albedo_vis

```
double jeod::RadiationBaseFacet::albedo_vis {-1.0}
```

Fraction of incident visible radiation that is immediately reflected.

trick_units(–)

Definition at line 100 of file radiation_base_facet.hh.

Referenced by jeod::RadiationFacet::initialize(), initialize(), and interact_with_third_body().

8.3.5.4 areaxflux

```
double jeod::RadiationBaseFacet::areaxflux {}
```

product of momentum flux and cross-sectional area (cx_area).

Highly variable.trick_units(–)

Definition at line 132 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), and jeod::RadiationDefaultSurface::incident_↩radiation().

8.3.5.5 areaxflux_e

```
double jeod::RadiationBaseFacet::areaxflux_e {}
```

product of energy flux and cross-sectional area

trick_units(–)

Definition at line 137 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), and jeod::RadiationDefaultSurface::incident_↩radiation().

8.3.5.6 cx_area

```
double jeod::RadiationBaseFacet::cx_area {}
```

cross-sectional area projected perpendicular to the radiation vector.

For facets that have an orientation, this is intended to be a variable value, dependent upon the orientation. For spherical surfaces, this can be set at initialization.trick_units(m2)

Definition at line 124 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_↩ radiation(), and jeod::RadiationDefaultSurface::initialize().

8.3.5.7 diffuse

```
double jeod::RadiationBaseFacet::diffuse {-1.0}
```

Fraction of reflected radiation that is reflected diffusely (balance reflected specularly)

trick_units(–)

Definition at line 111 of file radiation_base_facet.hh.

Referenced by jeod::RadiationFacet::define_facet_core(), jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_radiation(), jeod::RadiationDefaultSurface::initialize(), and jeod::↩ RadiationFacet::initialize().

8.3.5.8 F_absorption

```
double jeod::RadiationBaseFacet::F_absorption[3] {}
```

Force due to photon absorption from ONLY ONE source.

trick_units(–)

Definition at line 142 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_↩ radiation(), initialize_runtime_values(), jeod::FlatPlateRadiationFacet::radiation_pressure(), and jeod::Radiation↩ Pressure::update_default_surface().

8.3.5.9 F_diffuse

```
double jeod::RadiationBaseFacet::F_diffuse[3] {}
```

Force due to photon diffuse reflection from ONLY ONE source.

trick_units(–)

Definition at line 152 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_↩ radiation(), initialize_runtime_values(), jeod::FlatPlateRadiationFacet::radiation_pressure(), and jeod::Radiation↩ Pressure::update_default_surface().

8.3.5.10 F_emission

```
double jeod::RadiationBaseFacet::F_emission[3] {}
```

Force due to photon (thermal) emission.

trick_units(–)

Definition at line 157 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::radiation_pressure().

8.3.5.11 F_specular

```
double jeod::RadiationBaseFacet::F_specular[3] {}
```

Force due to photon specular reflection from ONLY ONE source.

trick_units(–)

Definition at line 147 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_↩ radiation(), initialize_runtime_values(), jeod::FlatPlateRadiationFacet::radiation_pressure(), and jeod::Radiation↩ Pressure::update_default_surface().

8.3.5.12 speed_of_light

```
constexpr double jeod::RadiationBaseFacet::speed_of_light {299792458.0}  [static], [protected]
```

Speed of light in vacuum.

trick_io(∗o) trick_units(m/s)

Definition at line 164 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_↩ radiation(), and jeod::FlatPlateRadiationFacet::radiation_pressure().

8.3.5.13 thermal

`ThermalFacetRider jeod::RadiationBaseFacet::thermal`

thermal characteristics of the facet.

trick_units(–)

Definition at line 116 of file radiation_base_facet.hh.

Referenced by jeod::RadiationSurface::accumulate_thermal_sources(), jeod::RadiationDefaultSurface::add_↩ thermal_integrator_to(), jeod::RadiationSurface::add_thermal_integrators_to(), jeod::RadiationFacet::define_↩ facet_core(), jeod::RadiationSurface::equalize_absorption_emission(), jeod::FlatPlateRadiationFacet::incident_↩ radiation(), jeod::RadiationDefaultSurface::incident_radiation(), jeod::RadiationDefaultSurface::initialize(), jeod↩ ::RadiationFacet::initialize(), jeod::RadiationSurface::initialize(), jeod::FlatPlateRadiationFacet::initialize_geom(), initialize_runtime_values(), jeod::FlatPlateRadiationFacet::radiation_pressure(), jeod::RadiationSurface::thermal↩ _integrator(), and jeod::RadiationDefaultSurface::thermal_update().

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

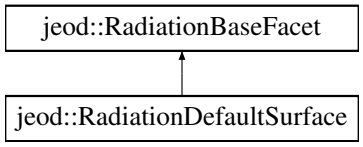
- radiation_base_facet.hh
- radiation_base_facet.cc

8.4 jeod::RadiationDefaultSurface Class Reference

The default spherical surface for radiation pressure.

`#include <radiation_default_surface.hh>`

Inheritance diagram for jeod::RadiationDefaultSurface:



Public Member Functions

- RadiationDefaultSurface ()=default
- ~RadiationDefaultSurface () override=default
- RadiationDefaultSurface & operator= (const RadiationDefaultSurface &)=delete
- RadiationDefaultSurface (const RadiationDefaultSurface &)=delete
- void initialize () override

  initializes the default surface
- void incident_radiation (const double flux_mag, const double flux_struc_hat[3], const bool calculate_forces) override

  Calculation of force and torque due to radiation pressure.
- void thermal_update ()

  Test for necessity of maintaining thermal updates, and performs them as necessary.
- void add_thermal_integrator_to (DynBody *dyn_body)

  Adds the thermal integrator of this surface to the integration group of the DynBody.
- void set_name (std::string name_in)

  Setter for the name.

Data Fields

- double rad_coeff {-1.0}
  The radiation-equivalent of a drag coefficient.
- double temperature {}
  The value of the surface kinetic temperature.
- std::string name
  The name of the surface.
- double surface_area {}
  surface area of the default sphere.

Friends

- class InputProcessor
- void init_attrjeod__RadiationDefaultSurface ()

Additional Inherited Members

8.4.1 Detailed Description

The default spherical surface for radiation pressure.

Definition at line 89 of file radiation_default_surface.hh.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 RadiationDefaultSurface() [1/2]

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

8.4.2.2 ~RadiationDefaultSurface()

```
jeod::RadiationDefaultSurface::~RadiationDefaultSurface ( )  [override], [default]
```

8.4.2.3 RadiationDefaultSurface() [2/2]

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

8.4.3 Member Function Documentation

8.4.3.1 add_thermal_integrator_to()

```
void jeod::RadiationDefaultSurface::add_thermal_integrator_to (
            DynBody * dyn_body )
```

Adds the thermal integrator of this surface to the integration group of the DynBody.

Parameters

in,out	*dyn_body*	Body associated with this thermal object.

Definition at line 268 of file radiation_default_surface.cc.

References jeod::RadiationBaseFacet::thermal.

8.4.3.2 incident_radiation()

```
void jeod::RadiationDefaultSurface::incident_radiation (
            const double flux_mag,
            const double flux_struc_hat[3],
            const bool calculate_forces ) [override], [virtual]
```

Calculation of force and torque due to radiation pressure.

Assumptions and Limitations

- Assumes that flux magnitude is positive.
- Will only be called if flux magnitude is positive.
- Assumes spherical surface.

Parameters

in	*flux_mag*	the magnitude of the incident flux
in	*flux_struc_hat*	the unit vector in structural frame for the flux vector.
in	*calculate_forces*	boolean indicating whether to calculate forces.

Implements jeod::RadiationBaseFacet.

Definition at line 214 of file radiation_default_surface.cc.

References jeod::RadiationBaseFacet::albedo, jeod::RadiationBaseFacet::areaxflux, jeod::RadiationBaseFacet::areaxflux_e, jeod::RadiationBaseFacet::cx_area, jeod::RadiationBaseFacet::diffuse, jeod::RadiationBaseFacet::F_absorption, jeod::RadiationBaseFacet::F_diffuse, jeod::RadiationBaseFacet::F_specular, jeod::RadiationBaseFacet::speed_of_light, and jeod::RadiationBaseFacet::thermal.

Referenced by jeod::RadiationPressure::update_default_surface().

8.4.3.3 initialize()

```
void jeod::RadiationDefaultSurface::initialize ( ) [override], [virtual]
```

initializes the default surface

Reimplemented from jeod::RadiationBaseFacet.

Definition at line 58 of file radiation_default_surface.cc.

References jeod::RadiationBaseFacet::albedo, jeod::RadiationBaseFacet::cx_area, jeod::RadiationBaseFacet::diffuse, jeod::RadiationBaseFacet::initialize(), jeod::RadiationMessages::invalid_setup_error, jeod::RadiationMessages::operational_setup_error, rad_coeff, surface_area, temperature, and jeod::RadiationBaseFacet::thermal.

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

8.4.3.4 operator=()

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

8.4.3.5 set_name()

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

Setter for the name.

Definition at line 148 of file radiation_default_surface.hh.

8.4.3.6 thermal_update()

```
void jeod::RadiationDefaultSurface::thermal_update ( )
```

Test for necessity of maintaining thermal updates, and performs them as necessary.

Definition at line 251 of file radiation_default_surface.cc.

References temperature, and jeod::RadiationBaseFacet::thermal.

Referenced by jeod::RadiationPressure::update_default_surface().

8.4.4 Friends And Related Function Documentation

8.4.4.1 init_attrjeod__RadiationDefaultSurface

```
void init_attrjeod__RadiationDefaultSurface ( )  [friend]
```

8.4.4.2 InputProcessor

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

Definition at line 91 of file radiation_default_surface.hh.

8.4.5 Field Documentation

8.4.5.1 name

```
std::string jeod::RadiationDefaultSurface::name
```

The name of the surface.

This is optional.trick_units(–)

Definition at line 116 of file radiation_default_surface.hh.

8.4.5.2 rad_coeff

```
double jeod::RadiationDefaultSurface::rad_coeff {-1.0}
```

The radiation-equivalent of a drag coefficient.

It is the value by which the area must be multiplied in order to generate the same force if the reflecting surface were considered a perfectly absorbing surface. For a sphere, this has a value between 1.0 (perfectly absorbing, or perfect specular reflection) to 1.4444 (all diffuse reflection). Specify either: rad_coeff OR (albedo AND diffuse). NOTE 1 - this value is used at initialization only; changes to its value mid-simulation cannot be effected. NOTE 2 - the values albedo and diffuse are inherited from RadiationBaseFacet.trick_units(–)

Definition at line 106 of file radiation_default_surface.hh.

Referenced by initialize().

8.4.5.3 surface_area

```
double jeod::RadiationDefaultSurface::surface_area {}
```

surface area of the default sphere.

trick_units(m2)

Definition at line 125 of file radiation_default_surface.hh.

Referenced by initialize().

8.4.5.4 temperature

```
double jeod::RadiationDefaultSurface::temperature {}
```

The value of the surface kinetic temperature.

trick_units(K)

Definition at line 111 of file radiation_default_surface.hh.

Referenced by initialize(), and thermal_update().

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

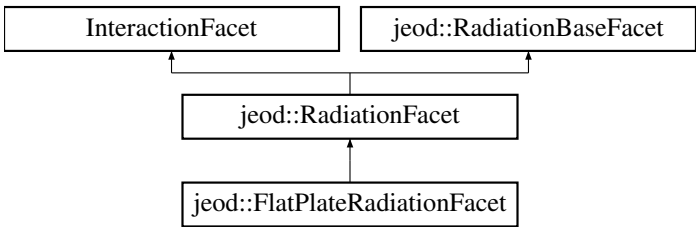
- radiation_default_surface.hh
- radiation_default_surface.cc

8.5 jeod::RadiationFacet Class Reference

Generic type of facet for radiation pressure.

```
#include <radiation_facet.hh>
```

Inheritance diagram for jeod::RadiationFacet:



Public Member Functions

- RadiationFacet ()=default
- ~RadiationFacet () override=default
- RadiationFacet & operator= (const RadiationFacet &)=delete
- RadiationFacet (const RadiationFacet &)=delete
- virtual void radiation_pressure ()=0

  Calculates the effect of radiation pressure on the facet.
- virtual void initialize_geom (double cg[3])=0

  Initialize the facet geometry.
- void initialize () override

  Run sanity checks on input variables.
- void define_facet_core (Facet *facet, const ThermalFacetRider &facet_thermal, RadiationParams *params)

  Defines the inherent facet values.
- ThermalIntegrableObject & get_thermal_integrator ()

  Get the integrator for thermal characteristics.

Data Fields

- double crot_to_cp [3] {}
  position of center of pressure w.r.t.
- double * center_pressure {}
  Center of pressure (in structural frame).

Static Protected Attributes

- static const double two_thirds = 2.0 / 3.0
  quite literally, 2/3.

Friends

- class InputProcessor
- void init_attrjeod__RadiationFacet ()

8.5.1 Detailed Description

Generic type of facet for radiation pressure.

Definition at line 87 of file radiation_facet.hh.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 RadiationFacet() [1/2]

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

8.5.2.2 ~RadiationFacet()

```
jeod::RadiationFacet::~RadiationFacet ( )  [override], [default]
```

8.5.2.3 RadiationFacet() [2/2]

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

8.5.3 Member Function Documentation

8.5.3.1 define_facet_core()

```
void jeod::RadiationFacet::define_facet_core (
            Facet * facet,
            const ThermalFacetRider & facet_thermal,
            RadiationParams * params )
```

Defines the inherent facet values.

Parameters

in	*facet*	pointer to the facet
in	*facet_thermal*	reference to the thermal components of the facet.
in	*params*	pointer to the list of material parameters for the facet.

Definition at line 86 of file radiation_facet.cc.

References jeod::RadiationParams::albedo, jeod::RadiationBaseFacet::albedo, jeod::RadiationParams::diffuse, jeod::RadiationBaseFacet::diffuse, jeod::RadiationParams::thermal, and jeod::RadiationBaseFacet::thermal.

8.5.3.2 get_thermal_integrator()

```
ThermalIntegrableObject& jeod::RadiationFacet::get_thermal_integrator ( )  [inline]
```

Get the integrator for thermal characteristics.

Returns

Integrable object that integrates temperature.

Definition at line 144 of file radiation_facet.hh.

8.5.3.3 initialize()

```
void jeod::RadiationFacet::initialize ( )  [override], [virtual]
```

Run sanity checks on input variables.

Reimplemented from jeod::RadiationBaseFacet.

Definition at line 56 of file radiation_facet.cc.

References jeod::RadiationBaseFacet::albedo, jeod::RadiationBaseFacet::albedo_IR, jeod::RadiationBaseFacet::albedo_vis, jeod::RadiationBaseFacet::diffuse, jeod::RadiationBaseFacet::initialize(), jeod::RadiationMessages::invalid_setup_error, and jeod::RadiationBaseFacet::thermal.

Referenced by jeod::FlatPlateRadiationFacet::initialize_geom().

8.5.3.4 initialize_geom()

```
virtual void jeod::RadiationFacet::initialize_geom (
            double cg[3] )  [pure virtual]
```

Initialize the facet geometry.

Parameters

cg	Center of mass.

Implemented in jeod::FlatPlateRadiationFacet.

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

8.5.3.5 operator=()

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

8.5.3.6 radiation_pressure()

```
virtual void jeod::RadiationFacet::radiation_pressure ( )  [pure virtual]
```

Calculates the effect of radiation pressure on the facet.

Implemented in jeod::FlatPlateRadiationFacet.

Referenced by jeod::RadiationSurface::radiation_pressure().

8.5.4 Friends And Related Function Documentation

8.5.4.1 init_attrjeod__RadiationFacet

```
void init_attrjeod__RadiationFacet ( )  [friend]
```

8.5.4.2 InputProcessor

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

Definition at line 90 of file radiation_facet.hh.

8.5.5 Field Documentation

8.5.5.1 center_pressure

```
double* jeod::RadiationFacet::center_pressure {}
```

Center of pressure (in structural frame).

Potentially variable, depending on the topology of the facet. For a flat plate facet, this just points to the position found in FlatPlatetrick_units(m)

Definition at line 106 of file radiation_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::define_facet(), and jeod::FlatPlateRadiationFacet::initialize_geom().

8.5.5.2 crot_to_cp

```
double jeod::RadiationFacet::crot_to_cp[3] {}
```

position of center of pressure w.r.t.

center of rotation (usually center of mass, or center of gravity), expressed in structural reference frame. Potentially variable, depending on the topology of the facet and the configuration of the vehicle.trick_units(m)

Definition at line 99 of file radiation_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::initialize_geom(), and jeod::FlatPlateRadiationFacet::radiation_↩ pressure().

8.5.5.3 two_thirds

```
const double jeod::RadiationFacet::two_thirds = 2.0 / 3.0  [static], [protected]
```

quite literally, 2/3.

Occurs frequently with diffuse reflection and emission, so it is calculated once.trick_io(∗o) trick_units(–)

Definition at line 115 of file radiation_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), and jeod::FlatPlateRadiationFacet::radiation_↩ pressure().

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

- radiation_facet.hh
- radiation_facet.cc

8.6 jeod::RadiationMessages Class Reference

Provides error messages.

`#include <radiation_messages.hh>`

Public Member Functions

- RadiationMessages ()=delete
- RadiationMessages (const RadiationMessages &)=delete
- RadiationMessages & operator= (const RadiationMessages &)=delete

Static Public Attributes

- static const char * incomplete_setup_error = "interactions/radiation_pressure/" "incomplete_setup_error"
  The setup was not fully defined.
- static const char * invalid_setup_error = "interactions/radiation_pressure/" "invalid_setup_error"
  The setup is invalid.
- static const char * operational_setup_error = "interactions/radiation_pressure/" "operational_setup_error"
  Something internal went wrong in the setup methods without obvious user error.
- static const char * invalid_function_call = "interactions/radiation_pressure/" "invalid_function_call"
  A function was called before it was fully implemented.
- static const char * unknown_numerical_error = "interactions/radiation_pressure/" "unknown_numerical_error"
  Something went horribly wrong.

Friends

- class InputProcessor
- void init_attrjeod__RadiationMessages ()

8.6.1 Detailed Description

Provides error messages.

Definition at line 84 of file radiation_messages.hh.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 RadiationMessages() [1/2]

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

8.6.2.2 RadiationMessages() [2/2]

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

8.6.3 Member Function Documentation

8.6.3.1 operator=()

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

8.6.4 Friends And Related Function Documentation

8.6.4.1 init_attrjeod__RadiationMessages

```
void init_attrjeod__RadiationMessages ( )  [friend]
```

8.6.4.2 InputProcessor

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

Definition at line 86 of file radiation_messages.hh.

8.6.5 Field Documentation

8.6.5.1 incomplete_setup_error

```
char const * jeod::RadiationMessages::incomplete_setup_error = "interactions/radiation_↩
pressure/" "incomplete_setup_error"  [static]
```

The setup was not fully defined.

trick_units(–)

Definition at line 92 of file radiation_messages.hh.

Referenced by jeod::RadiationPressure::add_third_body(), and jeod::RadiationSource::initialize().

8.6.5.2 invalid_function_call

```
char const * jeod::RadiationMessages::invalid_function_call = "interactions/radiation_pressure/"
"invalid_function_call"  [static]
```

A function was called before it was fully implemented.

trick_units(–)

Definition at line 105 of file radiation_messages.hh.

Referenced by jeod::RadiationPressure::find_third_body(), jeod::RadiationPressure::set_third_body_active(), and jeod::RadiationPressure::set_third_body_inactive().

8.6.5.3 invalid_setup_error

```
char const * jeod::RadiationMessages::invalid_setup_error = "interactions/radiation_pressure/"
"invalid_setup_error"  [static]
```

The setup is invalid.

trick_units(–)

Definition at line 96 of file radiation_messages.hh.

Referenced by jeod::RadiationSurfaceFactory::add_facet_params(), jeod::RadiationPressure::add_third_↩ body(), jeod::RadiationSurface::allocate_interaction_facet(), jeod::RadiationThirdBody::calculate_shadow(), jeod::RadiationThirdBody::convert_shadow_from_int(), jeod::FlatPlateRadiationFactory::create_facet(), jeod::↩ RadiationDefaultSurface::initialize(), jeod::RadiationFacet::initialize(), jeod::RadiationSurface::initialize(), jeod::↩ RadiationBaseFacet::initialize(), jeod::RadiationThirdBody::initialize(), jeod::RadiationThirdBody::process_third_↩ body(), and jeod::RadiationThirdBody::update_third_body_state().

8.6.5.4 operational_setup_error

```
char const * jeod::RadiationMessages::operational_setup_error = "interactions/radiation_↩
pressure/" "operational_setup_error"  [static]
```

Something internal went wrong in the setup methods without obvious user error.

trick_units(–)

Definition at line 101 of file radiation_messages.hh.

Referenced by jeod::RadiationSurface::allocate_array(), jeod::RadiationSurface::allocate_interaction_facet(), jeod::RadiationDefaultSurface::initialize(), jeod::RadiationSurface::initialize(), and jeod::RadiationThirdBody↩ ::process_third_body().

8.6.5.5 unknown_numerical_error

```
char const * jeod::RadiationMessages::unknown_numerical_error = "interactions/radiation_↩
pressure/" "unknown_numerical_error"  [static]
```

Something went horribly wrong.

trick_units(–)

Definition at line 110 of file radiation_messages.hh.

Referenced by jeod::FlatPlateRadiationFacet::radiation_pressure().

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

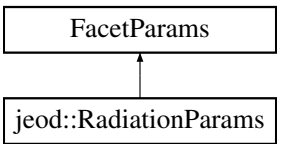
- radiation_messages.hh
- radiation_messages.cc

8.7 jeod::RadiationParams Class Reference

Provides a parameter list to each facet, based on the facet material properties.

```
#include <radiation_params.hh>
```

Inheritance diagram for jeod::RadiationParams:



Public Member Functions

- RadiationParams ()=default
- ~RadiationParams () override=default
- RadiationParams & operator= (const RadiationParams &)=delete
- RadiationParams (const RadiationParams &)=delete

Data Fields

- double albedo {}

  Fraction of incident radiation that is immediately reflected.
- double diffuse {}

  Fraction of reflected radiation that is reflected diffusely (balance reflected specularly)
- ThermalParams thermal

  Additional thermal parameters.

Friends

- class InputProcessor
- void init_attrjeod__RadiationParams ()

8.7.1 Detailed Description

Provides a parameter list to each facet, based on the facet material properties.

Definition at line 79 of file radiation_params.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 RadiationParams() [1/2]

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

8.7.2.2 ~RadiationParams()

```
jeod::RadiationParams::~RadiationParams ( )  [override], [default]
```

8.7.2.3 RadiationParams() [2/2]

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

8.7.3 Member Function Documentation

8.7.3.1 operator=()

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

8.7.4 Friends And Related Function Documentation

8.7.4.1 init_attrjeod__RadiationParams

```
void init_attrjeod__RadiationParams ( )  [friend]
```

8.7.4.2 InputProcessor

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

Definition at line 81 of file radiation_params.hh.

8.7.5 Field Documentation

8.7.5.1 albedo

```
double jeod::RadiationParams::albedo {}
```

Fraction of incident radiation that is immediately reflected.

trick_units(–)

Definition at line 85 of file radiation_params.hh.

Referenced by jeod::RadiationFacet::define_facet_core().

8.7.5.2 diffuse

```
double jeod::RadiationParams::diffuse {}
```

Fraction of reflected radiation that is reflected diffusely (balance reflected specularly)

trick_units(–)

Definition at line 91 of file radiation_params.hh.

Referenced by jeod::RadiationFacet::define_facet_core().

8.7.5.3 thermal

`ThermalParams jeod::RadiationParams::thermal`

Additional thermal parameters.

trick_units(–)

Definition at line 96 of file radiation_params.hh.

Referenced by jeod::RadiationFacet::define_facet_core().

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

- radiation_params.hh

8.8 jeod::RadiationPressure Class Reference

Radiation pressure top-level definition.

`#include <radiation_pressure.hh>`

Public Member Functions

- RadiationPressure ()
- virtual ~RadiationPressure ()

  Destructor for RadiationPressure.
- RadiationPressure (const RadiationPressure &)=delete
- RadiationPressure & operator= (const RadiationPressure &)=delete
- void initialize (DynManager &dyn_manager, RadiationSurface *surf_ptr, double center_grav[3])

  Initialize the radiation pressure model when using a RadiationSurface (i.e.
- void initialize (DynManager &dyn_manager, RadiationDefaultSurface *surf_ptr)

  Initialize the radiation pressure model when using a RadiationDefaultSurface (i.e.
- void update (RefFrame &vehicle_structural_frame, double center_grav[3], double scale_factor, double time)

  Updates the model at each time step.
- void add_third_body (RadiationThirdBody *third_body_ptr)

  Adds a third body to the vector storage of third bodies.
- void set_third_body_active (const std::string &third_body_name)

  Sets a Third Body to be active when it previously was not.
- void set_third_body_inactive (const std::string &third_body_name)

  Sets a Third Body to be inactive when it previously was not.
- void set_calculate_forces (bool calc_forces)

  Sets the value calculate_forces.

Data Fields

- bool active {true}

  Is radiation pressure desired?
- double force [3] {}

  Net force due to radiation.
- double torque [3] {}

  Net torque due to radiation.
- double illum_factor {}

  fraction of primary flux that is not eclipsed
- RadiationSource source

  Collection of data for radiation source.
- ThermalModelRider thermal

  Rider to allow dynamic thermal variation on facets in this model.

Protected Member Functions

- void update_default_surface ()

  Used to update the model when the surface is a default surface.
- void update_facet_surface ()

  Used to update the model when the surface comprises facets.
- void initialize_environment (DynManager *dyn_mgr_ptr)

  Initializes the source and third bodies.
- void third_body_adjustments (double time, RefFrame &vehicle_structural_frame)

  Used to modify the flux for shadowing.
- int find_third_body (const std::string &third_body_name)

  Finds the index of a third body based on its name.

Protected Attributes

- bool initialized {}

  Has model been initialized?
- bool calculate_forces {true}

  Flag to indicate whether forces and torques should be calculated.
- bool third_bodies_active {}

  Flag indicates whether there are any of the third bodies currently active.
- unsigned int num_third_bodies {}

  Number of ThirdBodies available.
- JeodPointerVector< RadiationThirdBody >::type third_bodies

  Planetary bodies that provide shadowing or indirect, reflected, illumination.
- RadiationSurface * surface_ptr {}

  The surface over which the radiation pressure will be collected.
- RadiationDefaultSurface * default_surface_ptr {}

  If no surface is defined, use a DefaultSurface; this is the pointer to that default surface,.
- DynManager * dyn_manager_ptr {}

  Pointer to the Dynamics Manager.

Friends

- class InputProcessor
- void init_attrjeod__RadiationPressure ()

8.8.1 Detailed Description

Radiation pressure top-level definition.

Definition at line 90 of file radiation_pressure.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 RadiationPressure() [1/2]

```
jeod::RadiationPressure::RadiationPressure ( )
```

Definition at line 60 of file radiation_pressure.cc.

References third_bodies.

8.8.2.2 ~RadiationPressure()

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

Destructor for RadiationPressure.

Definition at line 70 of file radiation_pressure.cc.

References third_bodies.

8.8.2.3 RadiationPressure() [2/2]

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

8.8.3 Member Function Documentation

8.8.3.1 add_third_body()

```
void jeod::RadiationPressure::add_third_body (
            RadiationThirdBody * third_body_ptr )
```

Adds a third body to the vector storage of third bodies.

Parameters

in	*third_body_ptr*	pointer to ThirdBody

Definition at line 156 of file radiation_pressure.cc.

References jeod::RadiationThirdBody::active, dyn_manager_ptr, jeod::RadiationThirdBody::get_added_to_model(), jeod::RadiationMessages::incomplete_setup_error, jeod::RadiationThirdBody::initialize(), initialized, jeod::RadiationMessages::invalid_setup_error, jeod::RadiationThirdBody::name, num_third_bodies, jeod::RadiationThirdBody::set_added_to_model(), third_bodies, and third_bodies_active.

Referenced by initialize().

8.8.3.2 find_third_body()

```
int jeod::RadiationPressure::find_third_body (
            const std::string & third_body_name )  [protected]
```

Finds the index of a third body based on its name.

Returns

void

Parameters

in	*third_body_name*	Name of ThirdBody

Definition at line 358 of file radiation_pressure.cc.

References jeod::RadiationMessages::invalid_function_call, num_third_bodies, and third_bodies.

Referenced by set_third_body_active(), and set_third_body_inactive().

8.8.3.3 initialize() [1/2]

```
void jeod::RadiationPressure::initialize (
            DynManager & dyn_mgr,
            RadiationSurface * surf_ptr,
            double center_grav[3] )
```

Initialize the radiation pressure model when using a RadiationSurface (i.e.

with facets).

Parameters

in	*dyn_mgr*	The dynamics manager
in	*surf_ptr*	pointer to the radiation surface
in	*center_grav*	position of center of mass Units: M

Definition at line 70 of file radiation_pressure__surface_model.cc.

References add_third_body(), dyn_manager_ptr, jeod::RadiationSurface::initialize(), initialize_environment(), jeod::RadiationSource::num_bodies, jeod::RadiationSource::shadow_geometry, source, surface_ptr, thermal, third_bodies, and jeod::RadiationSource::third_body.

8.8.3.4 initialize() [2/2]

```
void jeod::RadiationPressure::initialize (
            DynManager & dyn_mgr,
            RadiationDefaultSurface * surf_ptr )
```

Initialize the radiation pressure model when using a RadiationDefaultSurface (i.e.

spherical, uniform properties.)

Parameters

in	*dyn_mgr*	The dynamics manager
in	*surf_ptr*	pointer to the radiation surface

Definition at line 69 of file radiation_pressure__default_surface.cc.

References add_third_body(), default_surface_ptr, dyn_manager_ptr, jeod::RadiationDefaultSurface::initialize(), initialize_environment(), jeod::RadiationSource::num_bodies, jeod::RadiationSource::shadow_geometry, source, third_bodies, and jeod::RadiationSource::third_body.

8.8.3.5 initialize_environment()

```
void jeod::RadiationPressure::initialize_environment (
            DynManager * dyn_mgr_ptr )  [protected]
```

Initializes the source and third bodies.

Parameters

in	*dyn_mgr_ptr*	The dynamics manager

Definition at line 79 of file radiation_pressure.cc.

References jeod::RadiationSource::initialize(), initialized, num_third_bodies, source, and third_bodies.

Referenced by initialize().

8.8.3.6 operator=()

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

8.8.3.7 set_calculate_forces()

```
void jeod::RadiationPressure::set_calculate_forces (
            bool value )
```

Sets the value calculate_forces.

Parameters

in	*value*	whether forces are needed

Definition at line 382 of file radiation_pressure.cc.

References calculate_forces, force, and torque.

8.8.3.8 set_third_body_active()

```
void jeod::RadiationPressure::set_third_body_active (
            const std::string & third_body_name )
```

Sets a Third Body to be active when it previously was not.

Parameters

in	*third_body_name*	Name of ThirdBody

Definition at line 270 of file radiation_pressure.cc.

References active, dyn_manager_ptr, find_third_body(), jeod::RadiationMessages::invalid_function_call, third_↩ bodies, and third_bodies_active.

8.8.3.9 set_third_body_inactive()

```
void jeod::RadiationPressure::set_third_body_inactive (
            const std::string & third_body_name )
```

Sets a Third Body to be inactive when it previously was not.

Parameters

in	*third_body_name*	Name of ThirdBody

Definition at line 310 of file radiation_pressure.cc.

References active, dyn_manager_ptr, find_third_body(), jeod::RadiationMessages::invalid_function_call, num_↩ third_bodies, third_bodies, and third_bodies_active.

8.8.3.10 third_body_adjustments()

```
void jeod::RadiationPressure::third_body_adjustments (
            double real_time,
            RefFrame & veh_struc_frame )  [protected]
```

Used to modify the flux for shadowing.

Parameters

in	*real_time*	The current time, e.g. TAI or UT1. Units: s
in	*veh_struc_frame*	Structure frame of vehicle

Definition at line 136 of file radiation_pressure.cc.

References active, jeod::RadiationSource::flux_inertial, jeod::RadiationSource::flux_mag, jeod::RadiationSource↩ ::flux_struc, illum_factor, num_third_bodies, source, and third_bodies.

Referenced by update().

8.8.3.11 update()

```
void jeod::RadiationPressure::update (
            RefFrame & veh_struc_frame,
            double center_grav[3],
            double scale_factor,
            double real_time )
```

Updates the model at each time step.

Parameters

in	*veh_struc_frame*	The vehicle structural reference frame.
in	*center_grav*	position of center of mass Units: M
in	*scale_factor*	the scale between the simulator time and the dynamic time.
in	*real_time*	The current time, e.g. TAI or UT1. Units: s

Definition at line 97 of file radiation_pressure.cc.

References active, jeod::RadiationSource::calculate_flux(), source, surface_ptr, third_bodies_active, third_body_↩ adjustments(), update_default_surface(), and update_facet_surface().

8.8.3.12 update_default_surface()

```
void jeod::RadiationPressure::update_default_surface ( )  [protected]
```

Used to update the model when the surface is a default surface.

Definition at line 90 of file radiation_pressure__default_surface.cc.

References active, calculate_forces, default_surface_ptr, jeod::RadiationBaseFacet::F_absorption, jeod::↩ RadiationBaseFacet::F_diffuse, jeod::RadiationBaseFacet::F_specular, jeod::RadiationSource::flux_mag, jeod↩ ::RadiationSource::flux_struc_hat, force, jeod::RadiationDefaultSurface::incident_radiation(), jeod::Radiation↩ BaseFacet::initialize_runtime_values(), jeod::RadiationBaseFacet::interact_with_third_body(), num_third_bodies, source, jeod::RadiationDefaultSurface::thermal_update(), third_bodies, and torque.

Referenced by update().

8.8.3.13 update_facet_surface()

```
void jeod::RadiationPressure::update_facet_surface ( )  [protected]
```

Used to update the model when the surface comprises facets.

Definition at line 98 of file radiation_pressure__surface_model.cc.

References active, calculate_forces, jeod::RadiationSurface::equalize_absorption_emission(), jeod::Radiation↩ Source::flux_mag, jeod::RadiationSource::flux_struc_hat, force, jeod::RadiationSurface::force, jeod::Radiation↩ Surface::incident_radiation(), jeod::RadiationSurface::initialize_runtime_values(), jeod::RadiationSurface::interact↩ _with_third_body(), num_third_bodies, jeod::RadiationSurface::radiation_pressure(), source, surface_ptr, thermal, third_bodies, torque, and jeod::RadiationSurface::torque.

Referenced by update().

8.8.4 Friends And Related Function Documentation

8.8.4.1 init_attrjeod__RadiationPressure

```
void init_attrjeod__RadiationPressure ( )  [friend]
```

8.8.4.2 InputProcessor

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

Definition at line 92 of file radiation_pressure.hh.

8.8.5 Field Documentation

8.8.5.1 active

```
bool jeod::RadiationPressure::active {true}
```

Is radiation pressure desired?

trick_units(–)

Definition at line 97 of file radiation_pressure.hh.

Referenced by set_third_body_active(), set_third_body_inactive(), third_body_adjustments(), update(), update_↩ default_surface(), and update_facet_surface().

8.8.5.2 calculate_forces

```
bool jeod::RadiationPressure::calculate_forces {true}  [protected]
```

Flag to indicate whether forces and torques should be calculated.

trick_units(–)

Definition at line 133 of file radiation_pressure.hh.

Referenced by set_calculate_forces(), update_default_surface(), and update_facet_surface().

8.8.5.3 default_surface_ptr

```
RadiationDefaultSurface* jeod::RadiationPressure::default_surface_ptr {}  [protected]
```

If no surface is defined, use a DefaultSurface; this is the pointer to that default surface,.

trick_units(–)

Definition at line 163 of file radiation_pressure.hh.

Referenced by initialize(), and update_default_surface().

8.8.5.4 dyn_manager_ptr

```
DynManager* jeod::RadiationPressure::dyn_manager_ptr {}  [protected]
```

Pointer to the Dynamics Manager.

trick_units(–)

Definition at line 168 of file radiation_pressure.hh.

Referenced by add_third_body(), initialize(), set_third_body_active(), and set_third_body_inactive().

8.8.5.5 force

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

Net force due to radiation.

trick_units(N)

Definition at line 102 of file radiation_pressure.hh.

Referenced by set_calculate_forces(), update_default_surface(), and update_facet_surface().

8.8.5.6 illum_factor

```
double jeod::RadiationPressure::illum_factor {}
```

fraction of primary flux that is not eclipsed

trick_units(–)

Definition at line 112 of file radiation_pressure.hh.

Referenced by third_body_adjustments().

8.8.5.7 initialized

```
bool jeod::RadiationPressure::initialized {}  [protected]
```

Has model been initialized?

trick_units(–)

Definition at line 128 of file radiation_pressure.hh.

Referenced by add_third_body(), and initialize_environment().

8.8.5.8 num_third_bodies

```
unsigned int jeod::RadiationPressure::num_third_bodies {}  [protected]
```

Number of ThirdBodies available.

trick_units(count)

Definition at line 144 of file radiation_pressure.hh.

Referenced by add_third_body(), find_third_body(), initialize_environment(), set_third_body_inactive(), third_body_adjustments(), update_default_surface(), and update_facet_surface().

8.8.5.9 source

```
RadiationSource jeod::RadiationPressure::source
```

Collection of data for radiation source.

trick_units(–)

Definition at line 117 of file radiation_pressure.hh.

Referenced by initialize(), initialize_environment(), third_body_adjustments(), update(), update_default_surface(), and update_facet_surface().

8.8.5.10 surface_ptr

```
RadiationSurface* jeod::RadiationPressure::surface_ptr {}  [protected]
```

The surface over which the radiation pressure will be collected.

If this is NULL, use the default radiation surface (which is a whole separate entity from a RadiationSurface.trick_units(–)

Definition at line 157 of file radiation_pressure.hh.

Referenced by initialize(), update(), and update_facet_surface().

8.8.5.11 thermal

```
ThermalModelRider jeod::RadiationPressure::thermal
```

Rider to allow dynamic thermal variation on facets in this model.

trick_units(–)

Definition at line 122 of file radiation_pressure.hh.

Referenced by initialize(), and update_facet_surface().

8.8.5.12 third_bodies

```
JeodPointerVector<RadiationThirdBody>::type jeod::RadiationPressure::third_bodies  [protected]
```

Planetary bodies that provide shadowing or indirect, reflected, illumination.

trick_io(**)

Definition at line 150 of file radiation_pressure.hh.

Referenced by add_third_body(), find_third_body(), initialize(), initialize_environment(), RadiationPressure(), set↩_third_body_active(), set_third_body_inactive(), third_body_adjustments(), update_default_surface(), update_↩facet_surface(), and ~RadiationPressure().

8.8.5.13 third_bodies_active

```
bool jeod::RadiationPressure::third_bodies_active {}  [protected]
```

Flag indicates whether there are any of the third bodies currently active.

trick_units(–)

Definition at line 139 of file radiation_pressure.hh.

Referenced by add_third_body(), set_third_body_active(), set_third_body_inactive(), and update().

8.8.5.14 torque

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

Net torque due to radiation.

trick_units(N*m)

Definition at line 107 of file radiation_pressure.hh.

Referenced by set_calculate_forces(), update_default_surface(), and update_facet_surface().

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

- radiation_pressure.hh
- radiation_pressure.cc
- radiation_pressure__default_surface.cc
- radiation_pressure__surface_model.cc

8.9 jeod::RadiationSource Class Reference

Provides information on the source of the incident radiation.

`#include <radiation_source.hh>`

Public Types

- enum OldShadowGeometry { Cylindrical = 0, Cyl = 1, Conical = 2, Con = 3 }

  distinguishes between conical and cylindrical shadowing.

Public Member Functions

- RadiationSource ()=default
- virtual ~RadiationSource ()=default
- RadiationSource (const RadiationSource &)=delete
- RadiationSource & operator= (const RadiationSource &)=delete
- virtual void initialize (DynManager *dyn_manager_ptr)

  Initializes the source object for use in the Radiation Pressure model.
- virtual void calculate_flux (RefFrame &veh_struc_frame, const double center_grav[3])

  calculates the flux vector from the vehicle's position.
- void set_name (std::string name_in)

  Setter for the name.

Data Fields

- const double solar_luminosity {3.827E+26}

  Solar Luminosity.
- const double solar_radius {6.98E+08}

  Mean solar radius.
- std::string name {"Sun"}

  The name of the source of the illumination (usually Sun)
- double flux_hat [3] {}

  (inertial) unit vector associated with primary radiative flux.
- double flux_mag {}

  magnitude of principle radiative flux vector
- double d_source_to_cg {}

  distance from source to vehicle cg.
- double luminosity {solar_luminosity}

  Luminosity of primary source.
- double radius {solar_radius}

  Radius of primary source.
- double flux_inertial [3] {}

  (N/M/s) Power per unit area in the inertial reference frame
- double flux_struc [3] {}

  *(N/M*s) Power per unit area in the vehicle structural reference frame*
- double flux_struc_hat [3] {}

  Unit vector representing flux vector in the vehicle structural reference frame.
- double inertial_cg [3] {}

vehicle cg position w.r.t vehicle strucural origin, expressed in inertial RF.

- double source_to_cg [3] {}

  vehicle cg position relative to the source
- double source_to_struc_origin [3] {}

  vehicle struc frame origin position relative to source.
- bool multiple_shadow_bodies {}

  flags that more than one body are casting shadows on the vehicle.
- RefFrame * inertial_frame_ptr {}

  the inertial reference frame associated with this source
- bool bodies_active {}

  can there be 3rd-body interference, either shadowing or reflection.
- unsigned int num_bodies {}

  number of ThirdBodies available.
- RadiationThirdBody ** third_body {}

  Planetary bodies that provide shadowing or indirect, reflected, illumination.
- OldShadowGeometry shadow_geometry {}

  Flag indicating cylindrical / conical shadow geometry.

Friends

- class InputProcessor
- void init_attrjeod__RadiationSource ()

8.9.1 Detailed Description

Provides information on the source of the incident radiation.

Definition at line 90 of file radiation_source.hh.

8.9.2 Member Enumeration Documentation

8.9.2.1 OldShadowGeometry

`enum jeod::RadiationSource::OldShadowGeometry`

distinguishes between conical and cylindrical shadowing.

NOTE this is provided for backward-compatibility and should not be used.

Enumerator

Cylindrical	planet casts a cylindrical shadow
Cyl	planet casts a cylindrical shadow
Conical	planet casts a conical shadow
Con	planet casts a conical shadow

Definition at line 200 of file radiation_source.hh.

8.9.3 Constructor & Destructor Documentation

8.9.3.1 RadiationSource() [1/2]

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

8.9.3.2 ~RadiationSource()

```
virtual jeod::RadiationSource::~RadiationSource ( )  [virtual], [default]
```

8.9.3.3 RadiationSource() [2/2]

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

8.9.4 Member Function Documentation

8.9.4.1 calculate_flux()

```
void jeod::RadiationSource::calculate_flux (
            RefFrame & veh_struc_frame,
            const double center_gravity[3] )  [virtual]
```

calculates the flux vector from the vehicle's position.

Parameters

in	*veh_struc_frame*	the vehicle structural reference frame
in	*center_gravity*	position of the center of mass Units: M

Definition at line 72 of file radiation_source.cc.

References d_source_to_cg, flux_hat, flux_inertial, flux_mag, flux_struc, flux_struc_hat, inertial_cg, inertial_↩ frame_ptr, luminosity, source_to_cg, and source_to_struc_origin.

Referenced by jeod::RadiationPressure::update().

8.9.4.2 initialize()

```
void jeod::RadiationSource::initialize (
            DynManager * dyn_mgr_ptr )  [virtual]
```

Initializes the source object for use in the Radiation Pressure model.

Parameters

in	*dyn_mgr_ptr*	pointer to the dynamics manager

Definition at line 117 of file radiation_source.cc.

References jeod::RadiationMessages::incomplete_setup_error, inertial_frame_ptr, and name.

Referenced by jeod::RadiationPressure::initialize_environment().

8.9.4.3 operator=()

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

8.9.4.4 set_name()

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

Setter for the name.

Definition at line 227 of file radiation_source.hh.

8.9.5 Friends And Related Function Documentation

8.9.5.1 init_attrjeod__RadiationSource

```
void init_attrjeod__RadiationSource ( )  [friend]
```

8.9.5.2 InputProcessor

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

Definition at line 92 of file radiation_source.hh.

8.9.6 Field Documentation

8.9.6.1 bodies_active

```
bool jeod::RadiationSource::bodies_active {}
```

can there be 3rd-body interference, either shadowing or reflection.

trick_units(–)

Definition at line 183 of file radiation_source.hh.

8.9.6.2 d_source_to_cg

```
double jeod::RadiationSource::d_source_to_cg {}
```

distance from source to vehicle cg.

trick_units(m)

Definition at line 121 of file radiation_source.hh.

Referenced by calculate_flux(), and jeod::RadiationThirdBody::calculate_shadow().

8.9.6.3 flux_hat

```
double jeod::RadiationSource::flux_hat[3] {}
```

(inertial) unit vector associated with primary radiative flux.

trick_units(–)

Definition at line 111 of file radiation_source.hh.

Referenced by calculate_flux().

8.9.6.4 flux_inertial

```
double jeod::RadiationSource::flux_inertial[3] {}
```

(N/M/s) Power per unit area in the inertial reference frame

trick_units(–)

Definition at line 136 of file radiation_source.hh.

Referenced by calculate_flux(), and jeod::RadiationPressure::third_body_adjustments().

8.9.6.5 flux_mag

```
double jeod::RadiationSource::flux_mag {}
```

magnitude of principle radiative flux vector

trick_units(N/m2)

Definition at line 116 of file radiation_source.hh.

Referenced by calculate_flux(), jeod::RadiationPressure::third_body_adjustments(), jeod::RadiationPressure↩ ::update_default_surface(), and jeod::RadiationPressure::update_facet_surface().

8.9.6.6 flux_struc

```
double jeod::RadiationSource::flux_struc[3] {}
```

(N/M∗s) Power per unit area in the vehicle structural reference frame

trick_units(–)

Definition at line 141 of file radiation_source.hh.

Referenced by calculate_flux(), and jeod::RadiationPressure::third_body_adjustments().

8.9.6.7 flux_struc_hat

```
double jeod::RadiationSource::flux_struc_hat[3] {}
```

Unit vector representing flux vector in the vehicle structural reference frame.

trick_units(–)

Definition at line 147 of file radiation_source.hh.

Referenced by calculate_flux(), jeod::RadiationPressure::update_default_surface(), and jeod::RadiationPressure↩ ::update_facet_surface().

8.9.6.8 inertial_cg

```
double jeod::RadiationSource::inertial_cg[3] {}
```

vehicle cg position w.r.t vehicle strucural origin, expressed in inertial RF.

trick_units(m)

Definition at line 153 of file radiation_source.hh.

Referenced by calculate_flux().

8.9.6.9 inertial_frame_ptr

```
RefFrame* jeod::RadiationSource::inertial_frame_ptr {}
```

the inertial reference frame associated with this source

trick_units(–)

Definition at line 173 of file radiation_source.hh.

Referenced by calculate_flux(), initialize(), jeod::RadiationThirdBody::initialize(), and jeod::RadiationThirdBody↩ ::update_third_body_state().

8.9.6.10 luminosity

```
double jeod::RadiationSource::luminosity {solar_luminosity}
```

Luminosity of primary source.

trick_units(–)

Definition at line 126 of file radiation_source.hh.

Referenced by calculate_flux().

8.9.6.11 multiple_shadow_bodies

```
bool jeod::RadiationSource::multiple_shadow_bodies {}
```

flags that more than one body are casting shadows on the vehicle.

trick_units(–)

Definition at line 168 of file radiation_source.hh.

8.9.6.12 name

```
std::string jeod::RadiationSource::name {"Sun"}
```

The name of the source of the illumination (usually Sun)

trick_units(–)

Definition at line 106 of file radiation_source.hh.

Referenced by initialize().

8.9.6.13 num_bodies

```
unsigned int jeod::RadiationSource::num_bodies {}
```

number of ThirdBodies available.

trick_units(count)

Definition at line 188 of file radiation_source.hh.

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

8.9.6.14 radius

```
double jeod::RadiationSource::radius {solar_radius}
```

Radius of primary source.

trick_units(m)

Definition at line 131 of file radiation_source.hh.

Referenced by jeod::RadiationThirdBody::calculate_shadow(), and jeod::RadiationThirdBody::initialize().

8.9.6.15 shadow_geometry

```
OldShadowGeometry jeod::RadiationSource::shadow_geometry {}
```

Flag indicating cylindrical / conical shadow geometry.

trick_units(–)

Definition at line 211 of file radiation_source.hh.

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

8.9.6.16 solar_luminosity

```
const double jeod::RadiationSource::solar_luminosity {3.827E+26}
```

Solar Luminosity.

trick_units(–)

Definition at line 96 of file radiation_source.hh.

8.9.6.17 solar_radius

```
const double jeod::RadiationSource::solar_radius {6.98E+08}
```

Mean solar radius.

trick_units(m)

Definition at line 101 of file radiation_source.hh.

8.9.6.18 source_to_cg

```
double jeod::RadiationSource::source_to_cg[3] {}
```

vehicle cg position relative to the source

trick_units(m)

Definition at line 158 of file radiation_source.hh.

Referenced by calculate_flux(), and jeod::RadiationThirdBody::calculate_shadow().

8.9.6.19 source_to_struc_origin

```
double jeod::RadiationSource::source_to_struc_origin[3] {}
```

vehicle struc frame origin position relative to source.

trick_units(m)

Definition at line 163 of file radiation_source.hh.

Referenced by calculate_flux().

8.9.6.20 third_body

```
RadiationThirdBody** jeod::RadiationSource::third_body {}
```

Planetary bodies that provide shadowing or indirect, reflected, illumination.

trick_units(–)

Definition at line 194 of file radiation_source.hh.

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

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

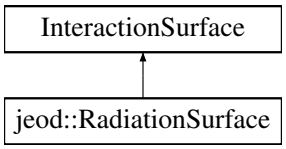
- radiation_source.hh
- radiation_source.cc

8.10 jeod::RadiationSurface Class Reference

The surface of the vehicle that interacts with the incident flux.

```
#include <radiation_surface.hh>
```

Inheritance diagram for jeod::RadiationSurface:



Public Member Functions

- RadiationSurface ()

  Constructor for RadiationSurface.
- ~RadiationSurface () override

  Destructor for RadiationSurface.
- RadiationSurface & operator= (const RadiationSurface &)=delete
- RadiationSurface (const RadiationSurface &)=delete
- void initialize (double center_grav[3])

  Initializes the radiation surface.
- void allocate_array (unsigned int size) override

  Allocates memory for an array of radiation facets.
- void allocate_interaction_facet (Facet *facet, InteractionFacetFactory *factory, FacetParams *params, unsigned int index) override

  Turns facet memory into radiaiton facet memory.
- void initialize_runtime_values ()

  To initialize the values during each update run.
- void incident_radiation (double flux_mag, const double flux_struc_hat[3], bool calculate_forces)

  systematically calls the method to calculate the interaction on each facet.

- void interact_with_third_body (RadiationThirdBody *third_body_ptr, const bool calculate_forces)

  systematically calls the method to calculate the interaction of each facet with ThirdBody flux.
- void accumulate_thermal_sources () override

  systematically calls the method to accumulate thermal sources on each facet.
- void thermal_integrator () override

  systematically calls the method to integrate the temperature variation for each facet.
- void equalize_absorption_emission ()

  systematically calls the method to ensure that the same for each facet.
- void radiation_pressure ()

  systematically calls the method to wrap up the radiation pressure calculation on each facet and accumulate forces and torques over all facets.
- void add_thermal_integrators_to (DynBody *dyn_body)

  To attach thermal integrators from each facet to the integration group of a DynBody.

Data Fields

- RadiationFacet ** facets {}

  Array of pointers to the Radiation Facets.
- unsigned int num_facets {}

  number of facets in this surface.
- bool include_conduction {}

  Flag to instruct model to include conduction between plates in the thermal calculation.
- double ** thermal_conduction {}

  Conductivity values (Watts per Kelvin) between facets.
- double force [3] {}

  Force resulting from all radiative interactions.
- double torque [3] {}

  Torque resulting from all radiative interactions.
- unsigned int ii_facet {}

  Simple counter, used repeatedly.

Friends

- class InputProcessor
- void init_attrjeod__RadiationSurface ()

8.10.1 Detailed Description

The surface of the vehicle that interacts with the incident flux.

Definition at line 88 of file radiation_surface.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 RadiationSurface() [1/2]

```
jeod::RadiationSurface::RadiationSurface ( )
```

Constructor for RadiationSurface.

Definition at line 60 of file radiation_surface.cc.

8.10.2.2 ~RadiationSurface()

```
jeod::RadiationSurface::~RadiationSurface ( )  [override]
```

Destructor for RadiationSurface.

Definition at line 337 of file radiation_surface.cc.

References facets, and num_facets.

8.10.2.3 RadiationSurface() [2/2]

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

8.10.3 Member Function Documentation

8.10.3.1 accumulate_thermal_sources()

```
void jeod::RadiationSurface::accumulate_thermal_sources ( )  [override]
```

systematically calls the method to accumulate thermal sources on each facet.

Definition at line 255 of file radiation_surface.cc.

References facets, ii_facet, num_facets, and jeod::RadiationBaseFacet::thermal.

8.10.3.2 add_thermal_integrators_to()

```
void jeod::RadiationSurface::add_thermal_integrators_to (
            DynBody * dyn_body )
```

To attach thermal integrators from each facet to the integration group of a DynBody.

Parameters

in,out	*dyn_body*	Body to which integrators are to be added.

Definition at line 325 of file radiation_surface.cc.

References facets, num_facets, and jeod::RadiationBaseFacet::thermal.

8.10.3.3 allocate_array()

```
void jeod::RadiationSurface::allocate_array (
            unsigned int size )  [override]
```

Allocates memory for an array of radiation facets.

Parameters

in	*size*	Size of array

Definition at line 134 of file radiation_surface.cc.

References facets, ii_facet, num_facets, and jeod::RadiationMessages::operational_setup_error.

8.10.3.4 allocate_interaction_facet()

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

Turns facet memory into radiaiton facet memory.

Parameters

in	*facet*	pointer to the facet
in	*factory*	pointer to the facet factory
in	*params*	pointer to the generic facet parameter.
in	*index*	index value in the facet list.

Definition at line 166 of file radiation_surface.cc.

References facets, jeod::RadiationMessages::invalid_setup_error, num_facets, and jeod::RadiationMessages↩ ::operational_setup_error.

8.10.3.5 equalize_absorption_emission()

```
void jeod::RadiationSurface::equalize_absorption_emission ( )
```

systematically calls the method to ensure that the same for each facet.

Definition at line 284 of file radiation_surface.cc.

References facets, ii_facet, num_facets, and jeod::RadiationBaseFacet::thermal.

Referenced by jeod::RadiationPressure::update_facet_surface().

8.10.3.6 incident_radiation()

```
void jeod::RadiationSurface::incident_radiation (
             double flux_mag,
             const double flux_struc_hat[3],
             bool calculate_forces )
```

systematically calls the method to calculate the interaction on each facet.

Parameters

in	*flux_mag*	Magnitude of incident flux
in	*flux_struc_hat*	unit vector of incident flux
in	*calculate_forces*	boolean indicating whether to calculate forces.

Definition at line 230 of file radiation_surface.cc.

References facets, ii_facet, jeod::RadiationBaseFacet::incident_radiation(), and num_facets.

Referenced by jeod::RadiationPressure::update_facet_surface().

8.10.3.7 initialize()

```
void jeod::RadiationSurface::initialize (
             double center_grav[3] )
```

Initializes the radiation surface.

Parameters

in	*center_grav*	position of center of mass Units: M

Definition at line 69 of file radiation_surface.cc.

References facets, ii_facet, jeod::RadiationFacet::initialize_geom(), jeod::RadiationMessages::invalid_setup_error, num_facets, jeod::RadiationMessages::operational_setup_error, and jeod::RadiationBaseFacet::thermal.

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

8.10.3.8 initialize_runtime_values()

```
void jeod::RadiationSurface::initialize_runtime_values ( )
```

To initialize the values during each update run.

Definition at line 312 of file radiation_surface.cc.

References facets, ii_facet, jeod::RadiationBaseFacet::initialize_runtime_values(), and num_facets.

Referenced by jeod::RadiationPressure::update_facet_surface().

8.10.3.9 interact_with_third_body()

```
void jeod::RadiationSurface::interact_with_third_body (
            RadiationThirdBody * third_body_ptr,
            const bool calculate_forces )
```

systematically calls the method to calculate the interaction of each facet with ThirdBody flux.

Parameters

in,out	*third_body_ptr*	Third body that emits radiation.
in	*calculate_forces*	Calculate forces on the surface if true.

Definition at line 244 of file radiation_surface.cc.

References facets, ii_facet, jeod::RadiationBaseFacet::interact_with_third_body(), and num_facets.

Referenced by jeod::RadiationPressure::update_facet_surface().

8.10.3.10 operator=()

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

8.10.3.11 radiation_pressure()

```
void jeod::RadiationSurface::radiation_pressure ( )
```

systematically calls the method to wrap up the radiation pressure calculation on each facet and accumulate forces and torques over all facets.

Definition at line 296 of file radiation_surface.cc.

References facets, force, ii_facet, num_facets, jeod::RadiationFacet::radiation_pressure(), and torque.

Referenced by jeod::RadiationPressure::update_facet_surface().

8.10.3.12 thermal_integrator()

```
void jeod::RadiationSurface::thermal_integrator ( )  [override]
```

systematically calls the method to integrate the temperature variation for each facet.

Definition at line 266 of file radiation_surface.cc.

References facets, ii_facet, num_facets, and jeod::RadiationBaseFacet::thermal.

8.10.4 Friends And Related Function Documentation

8.10.4.1 init_attrjeod__RadiationSurface

```
void init_attrjeod__RadiationSurface ( )  [friend]
```

8.10.4.2 InputProcessor

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

Definition at line 90 of file radiation_surface.hh.

8.10.5 Field Documentation

8.10.5.1 facets

```
RadiationFacet** jeod::RadiationSurface::facets {}
```

Array of pointers to the Radiation Facets.

trick_units(–)

Definition at line 94 of file radiation_surface.hh.

Referenced by accumulate_thermal_sources(), add_thermal_integrators_to(), allocate_array(), allocate_↩ interaction_facet(), equalize_absorption_emission(), incident_radiation(), initialize(), initialize_runtime_values(), interact_with_third_body(), radiation_pressure(), thermal_integrator(), and ∼RadiationSurface().

8.10.5.2 force

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

Force resulting from all radiative interactions.

trick_units(–)

Definition at line 118 of file radiation_surface.hh.

Referenced by radiation_pressure(), and jeod::RadiationPressure::update_facet_surface().

8.10.5.3 ii_facet

```
unsigned int jeod::RadiationSurface::ii_facet {}
```

Simple counter, used repeatedly.

trick_units(–)

Definition at line 128 of file radiation_surface.hh.

Referenced by accumulate_thermal_sources(), allocate_array(), equalize_absorption_emission(), incident_↩ radiation(), initialize(), initialize_runtime_values(), interact_with_third_body(), radiation_pressure(), and thermal_↩ integrator().

8.10.5.4 include_conduction

```
bool jeod::RadiationSurface::include_conduction {}
```

Flag to instruct model to include conduction between plates in the thermal calculation.

Requires a thermal_conduction matrix.trick_units(–)

Definition at line 105 of file radiation_surface.hh.

8.10.5.5 num_facets

```
unsigned int jeod::RadiationSurface::num_facets {}
```

number of facets in this surface.

trick_units(–)

Definition at line 99 of file radiation_surface.hh.

Referenced by accumulate_thermal_sources(), add_thermal_integrators_to(), allocate_array(), allocate_↩ interaction_facet(), equalize_absorption_emission(), incident_radiation(), initialize(), initialize_runtime_values(), interact_with_third_body(), radiation_pressure(), thermal_integrator(), and ∼RadiationSurface().

8.10.5.6 thermal_conduction

```
double** jeod::RadiationSurface::thermal_conduction {}
```

Conductivity values (Watts per Kelvin) between facets.

Assumed symmetric, only the values [small][large] will be considered (e.g. thermal_conduction[2][3] gives the conduction between facets 2 and 3, whereas thermal_conduction[3][2] will never be seen.trick_units(–)

Definition at line 113 of file radiation_surface.hh.

8.10.5.7 torque

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

Torque resulting from all radiative interactions.

trick_units(–)

Definition at line 123 of file radiation_surface.hh.

Referenced by radiation_pressure(), and jeod::RadiationPressure::update_facet_surface().

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

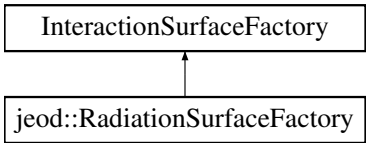
- radiation_surface.hh
- radiation_surface.cc

8.11 jeod::RadiationSurfaceFactory Class Reference

The factory for creating Radiation Surfaces.

`#include <radiation_surface_factory.hh>`

Inheritance diagram for jeod::RadiationSurfaceFactory:



Public Member Functions

- RadiationSurfaceFactory ()

  Constructor.
- ~RadiationSurfaceFactory () override

  Destructor.
- RadiationSurfaceFactory & operator= (const RadiationSurfaceFactory &)=delete
- RadiationSurfaceFactory (const RadiationSurfaceFactory &)=delete
- void add_facet_params (FacetParams *to_add) override

  Add a set of radiation parameters to those available to Radiation Surface.

Protected Attributes

- FlatPlateRadiationFactory flat_plate_radiation_factory

  The factory to build flat plate thermal facets.

Friends

- class InputProcessor
- void init_attrjeod__RadiationSurfaceFactory ()

8.11.1 Detailed Description

The factory for creating Radiation Surfaces.

Definition at line 85 of file radiation_surface_factory.hh.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 RadiationSurfaceFactory() [1/2]

```
jeod::RadiationSurfaceFactory::RadiationSurfaceFactory ( )
```

Constructor.

Definition at line 52 of file radiation_surface_factory.cc.

References flat_plate_radiation_factory.

8.11.2.2 ~RadiationSurfaceFactory()

```
jeod::RadiationSurfaceFactory::~RadiationSurfaceFactory ( )  [override]
```

Destructor.

Definition at line 91 of file radiation_surface_factory.cc.

8.11.2.3 RadiationSurfaceFactory() [2/2]

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

8.11.3 Member Function Documentation

8.11.3.1 add_facet_params()

```
void jeod::RadiationSurfaceFactory::add_facet_params (
            FacetParams * to_add )  [override]
```

Add a set of radiation parameters to those available to Radiation Surface.

Parameters

in	*to_add*	pointer to the parameter list to add.

Definition at line 64 of file radiation_surface_factory.cc.

References jeod::RadiationMessages::invalid_setup_error.

8.11.3.2 operator=()

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

8.11.4 Friends And Related Function Documentation

8.11.4.1 init_attrjeod__RadiationSurfaceFactory

```
void init_attrjeod__RadiationSurfaceFactory ( )  [friend]
```

8.11.4.2 InputProcessor

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

Definition at line 87 of file radiation_surface_factory.hh.

8.11.5 Field Documentation

8.11.5.1 flat_plate_radiation_factory

```
FlatPlateRadiationFactory jeod::RadiationSurfaceFactory::flat_plate_radiation_factory  [protected]
```

The factory to build flat plate thermal facets.

trick_units(–)

Definition at line 103 of file radiation_surface_factory.hh.

Referenced by RadiationSurfaceFactory().

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

- radiation_surface_factory.hh
- radiation_surface_factory.cc

8.12 jeod::RadiationThirdBody Class Reference

Provide information on bodies that may cause shadowing or reflected illumination.

```
#include <radiation_third_body.hh>
```

Public Types

- enum ShadowGeometry { Cylindrical = 0, Cyl = 1, Conical = 2, Con = 3 }

  distinguishes between conical and cylindrical shadowing

Public Member Functions

- RadiationThirdBody ()=default
- virtual ~RadiationThirdBody ()=default
- RadiationThirdBody (const RadiationThirdBody &)=delete
- RadiationThirdBody & operator= (const RadiationThirdBody &)=delete
- virtual void initialize (DynManager *dyn_mgr_ptr)

  initializes the third body
- void convert_shadow_from_int (int geometry)

  convert from the old style framework in which the shadow geometry was owned by the RadiationSource (same for all third bodies) to the new style framework where the geometry is owned by each ThirdBody
- virtual double process_third_body (double real_time, RefFrame &veh_struc_frame)

  Primary executable for handling the Third Body effects.
- void set_name (std::string name_in)

  Setter for the name.
- virtual void accumulate_refl_flux (RadiationBaseFacet *veh_surf_elem, bool calculate_forces)

  To provide base class null implementation.
- virtual void accumulate_rad_flux (RadiationBaseFacet *veh_surf_elem, bool calculate_forces)

  To provide base class null implementation.
- virtual bool is_interactive ()

  Identifies this class as one that does not produce a radiaiton field.
- bool get_added_to_model ()

  Returns information on whether the body has already been used in one RadiationPressure model.
- void set_added_to_model (bool value)

  Sets the added_to_model value.

Data Fields

- RadiationSource * primary_source_ptr {}

  Pointer to the primary illumination source.
- ShadowGeometry shadow_geometry {Conical}

  Flag indicating cylindrical / conical shadow geometry.
- bool force_state_update {true}

  Flag to indicate whether to enforce an update to the third-body body-state.
- std::string name

  Name of planet.
- bool active {true}

  Model on/off flag.
- double update_interval {}

  Time interval between third body state updates.
- double radius {}

  Radius of RadiationThirdBody.
- double output_flux [3] {}

  Calculated value.
- RefFrame * local_frame_ptr {}

  Pointer to the local reference frame, for determining position of this object.

Protected Member Functions

- double generate_alpha (double rho_adj, double delta)

  generate_alpha(rho_adj, delta: generates an approximate alpha-value given a relative disk size and fraction of maximum possible eclipse
- bool test_for_state_update (double time)

  Tests for necessity of updating third body state, and calls appropriate update method (polymorphic) if needed.
- void calculate_shadow ()

  Calculates the effect of shadowing by a third body.
- virtual bool update_third_body_state ()

  Updates the state of the Third Body.

Protected Attributes

- bool added_to_model {}

  Flag to indicate whether body has already been added to a model.
- bool initialized {}

  Flag to indicate that initialization has been complete.
- double illum_factor {1.0}

  Value between 0 and 1 representing the fraction of possible flux that is actually incident after shadowing effects.
- double last_update_time {}

  Time at which the third body state was last updated.
- Planet ∗ planet_link {}

  Pointer to the associated Planet object.
- double r_plus {}

  The value (radius + R_source)
- double r_minus {}

  The value (radius - R_source)
- double r_ratio {}

  The ratio (radius / R_source)
- double r_par {}

  Component of vector from RadiationThirdBody to vehicle that is parallel to vector from primary source to RadiationThirdBody.
- double r_perp {}

  Component of vector from RadiationThirdBody to vehicle that is perpendicular to vector from primary source to RadiationThirdBody.
- double third_to_cg_inrtl [3] {}

  Vector from RadiationThirdBody to vehicle center of gravity, in inertial RF.
- double source_to_third_inrtl [3] {}

  Vector from the primary source to the RadiationThirdBody in inertial RF.
- double d_source_to_third {}

  Magnitude of source_to_third.
- double source_to_third_hat_inrtl [3] {}

  Unit vector of source_to_third.

Friends

- class InputProcessor
- void init_attrjeod__RadiationThirdBody ()

8.12.1 Detailed Description

Provide information on bodies that may cause shadowing or reflected illumination.

Definition at line 84 of file radiation_third_body.hh.

8.12.2 Member Enumeration Documentation

8.12.2.1 ShadowGeometry

```
enum jeod::RadiationThirdBody::ShadowGeometry
```

distinguishes between conical and cylindrical shadowing

Enumerator

Cylindrical	planet casts a cylindrical shadow
Cyl	planet casts a cylindrical shadow
Conical	planet casts a conical shadow
Con	planet casts a conical shadow

Definition at line 90 of file radiation_third_body.hh.

8.12.3 Constructor & Destructor Documentation

8.12.3.1 RadiationThirdBody() [1/2]

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

8.12.3.2 ~RadiationThirdBody()

```
virtual jeod::RadiationThirdBody::~RadiationThirdBody ( )  [virtual], [default]
```

8.12.3.3 RadiationThirdBody() [2/2]

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

8.12.4 Member Function Documentation

8.12.4.1 accumulate_rad_flux()

```
virtual void jeod::RadiationThirdBody::accumulate_rad_flux (
            RadiationBaseFacet * veh_surf_elem,
            bool calculate_forces )  [inline], [virtual]
```

To provide base class null implementation.

Parameters

veh_surf_elem	The facet of the vehicle.
calculate_forces	pass-through flag.

Definition at line 261 of file radiation_third_body.hh.

Referenced by jeod::RadiationBaseFacet::interact_with_third_body().

8.12.4.2 accumulate_refl_flux()

```
virtual void jeod::RadiationThirdBody::accumulate_refl_flux (
            RadiationBaseFacet * veh_surf_elem,
            bool calculate_forces )  [inline], [virtual]
```

To provide base class null implementation.

Parameters

veh_surf_elem	The facet of the vehicle.
calculate_forces	pass-through flag.

Definition at line 253 of file radiation_third_body.hh.

Referenced by jeod::RadiationBaseFacet::interact_with_third_body().

8.12.4.3 calculate_shadow()

```
void jeod::RadiationThirdBody::calculate_shadow ( )  [protected]
```

Calculates the effect of shadowing by a third body.

Definition at line 190 of file radiation_third_body.cc.

References Con, Conical, Cyl, Cylindrical, jeod::RadiationSource::d_source_to_cg, d_source_to_third, generate_alpha(), illum_factor, jeod::RadiationMessages::invalid_setup_error, name, primary_source_ptr, r_minus, r_par, r_perp, r_plus, r_ratio, jeod::RadiationSource::radius, radius, shadow_geometry, jeod::RadiationSource::source_to_cg, source_to_third_hat_inrtl, source_to_third_inrtl, and third_to_cg_inrtl.

Referenced by process_third_body().

8.12.4.4 convert_shadow_from_int()

```
void jeod::RadiationThirdBody::convert_shadow_from_int (
             int old_shadow_geometry )
```

convert from the old style framework in which the shadow geometry was owned by the RadiationSource (same for all third bodies) to the new style framework where the geometry is owned by each ThirdBody

Parameters

in	*old_shadow_geometry*	integer representation of the shadow geometry enumeration.

Definition at line 350 of file radiation_third_body.cc.

References jeod::RadiationMessages::invalid_setup_error, and shadow_geometry.

8.12.4.5 generate_alpha()

```
double jeod::RadiationThirdBody::generate_alpha (
             double rho_adj,
             double delta )  [protected]
```

generate_alpha(rho_adj, delta: generates an approximate alpha-value given a relative disk size and fraction of maximum possible eclipse

Parameters

in	*rho_adj*	relative disk size
in	*delta*	fraction of eclipse achieved.

Definition at line 331 of file radiation_third_body.cc.

Referenced by calculate_shadow().

8.12.4.6 get_added_to_model()

```
bool jeod::RadiationThirdBody::get_added_to_model ( )  [inline]
```

Returns information on whether the body has already been used in one RadiationPressure model.

Protects against multiple use of one ThirdBody instance.

Returns

added_to_model

Definition at line 279 of file radiation_third_body.hh.

Referenced by jeod::RadiationPressure::add_third_body().

8.12.4.7 initialize()

```
void jeod::RadiationThirdBody::initialize (
            DynManager * dyn_mgr_ptr )  [virtual]
```

initializes the third body

Parameters

in	*dyn_mgr_ptr*	pointer to the dynamics manager

Definition at line 57 of file radiation_third_body.cc.

References active, jeod::RadiationSource::inertial_frame_ptr, initialized, jeod::RadiationMessages::invalid_setup_error, local_frame_ptr, name, planet_link, primary_source_ptr, r_minus, r_plus, r_ratio, jeod::RadiationSource::radius, and radius.

Referenced by jeod::RadiationPressure::add_third_body().

8.12.4.8 is_interactive()

```
virtual bool jeod::RadiationThirdBody::is_interactive ( )  [inline], [virtual]
```

Identifies this class as one that does not produce a radiaiton field.

Returns

false

Definition at line 268 of file radiation_third_body.hh.

8.12.4.9 operator=()

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

8.12.4.10 process_third_body()

```
double jeod::RadiationThirdBody::process_third_body (
            double real_time,
            RefFrame & veh_struc_frame )  [virtual]
```

Primary executable for handling the Third Body effects.

Returns

shading fraction

Parameters

in	*real_time*	Current time. Units: s
in	*veh_struc_frame*	Unused.

Definition at line 371 of file radiation_third_body.cc.

References active, calculate_shadow(), illum_factor, initialized, jeod::RadiationMessages::invalid_setup_error, name, jeod::RadiationMessages::operational_setup_error, and test_for_state_update().

8.12.4.11 set_added_to_model()

```
void jeod::RadiationThirdBody::set_added_to_model (
            bool value )  [inline]
```

Sets the added_to_model value.

Parameters

value	New value

Definition at line 288 of file radiation_third_body.hh.

Referenced by jeod::RadiationPressure::add_third_body().

8.12.4.12 set_name()

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

Setter for the name.

Definition at line 243 of file radiation_third_body.hh.

8.12.4.13 test_for_state_update()

```
bool jeod::RadiationThirdBody::test_for_state_update (
            double real_time )  [protected]
```

Tests for necessity of updating third body state, and calls appropriate update method (polymorphic) if needed.

Parameters

in	*real_time*	Current time Units: s

Definition at line 418 of file radiation_third_body.cc.

References force_state_update, last_update_time, update_interval, and update_third_body_state().

Referenced by process_third_body().

8.12.4.14 update_third_body_state()

```
bool jeod::RadiationThirdBody::update_third_body_state ( )  [protected], [virtual]
```

Updates the state of the Third Body.

Because the ThirdBody may be associated with multiple vehicles, and because its state typically changes much more slowly than the vehicle anyway, it is not always necessary to update the ThirdBodyState.

Definition at line 440 of file radiation_third_body.cc.

References active, d_source_to_third, jeod::RadiationSource::inertial_frame_ptr, jeod::RadiationMessages↩ ::invalid_setup_error, local_frame_ptr, name, primary_source_ptr, source_to_third_hat_inrtl, and source_to_↩ third_inrtl.

Referenced by test_for_state_update().

8.12.5 Friends And Related Function Documentation

8.12.5.1 init_attrjeod__RadiationThirdBody

```
void init_attrjeod__RadiationThirdBody ( )  [friend]
```

8.12.5.2 InputProcessor

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

Definition at line 86 of file radiation_third_body.hh.

8.12.6 Field Documentation

8.12.6.1 active

```
bool jeod::RadiationThirdBody::active {true}
```

Model on/off flag.

trick_units(–)

Definition at line 123 of file radiation_third_body.hh.

Referenced by jeod::RadiationPressure::add_third_body(), initialize(), process_third_body(), and update_third_↩ body_state().

8.12.6.2 added_to_model

```
bool jeod::RadiationThirdBody::added_to_model {}  [protected]
```

Flag to indicate whether body has already been added to a model.

trick_units(–)

Definition at line 153 of file radiation_third_body.hh.

8.12.6.3 d_source_to_third

```
double jeod::RadiationThirdBody::d_source_to_third {}  [protected]
```

Magnitude of source_to_third.

trick_units(m)

Definition at line 220 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and update_third_body_state().

8.12.6.4 force_state_update

```
bool jeod::RadiationThirdBody::force_state_update {true}
```

Flag to indicate whether to enforce an update to the third-body body-state.

Default: true Self-resets to false after each use.trick_units(–)

Definition at line 113 of file radiation_third_body.hh.

Referenced by test_for_state_update().

8.12.6.5 illum_factor

```
double jeod::RadiationThirdBody::illum_factor {1.0}  [protected]
```

Value between 0 and 1 representing the fraction of possible flux that is actually incident after shadowing effects.

trick_units(–)

Definition at line 164 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and process_third_body().

8.12.6.6 initialized

```
bool jeod::RadiationThirdBody::initialized {}  [protected]
```

Flag to indicate that initialization has been complete.

trick_units(–)

Definition at line 158 of file radiation_third_body.hh.

Referenced by initialize(), and process_third_body().

8.12.6.7 last_update_time

```
double jeod::RadiationThirdBody::last_update_time {}  [protected]
```

Time at which the third body state was last updated.

trick_units(s)

Definition at line 169 of file radiation_third_body.hh.

Referenced by test_for_state_update().

8.12.6.8 local_frame_ptr

```
RefFrame* jeod::RadiationThirdBody::local_frame_ptr {}
```

Pointer to the local reference frame, for determining position of this object.

trick_units(–)

Definition at line 146 of file radiation_third_body.hh.

Referenced by initialize(), and update_third_body_state().

8.12.6.9 name

```
std::string jeod::RadiationThirdBody::name
```

Name of planet.

trick_units(–)

Definition at line 118 of file radiation_third_body.hh.

Referenced by jeod::RadiationPressure::add_third_body(), calculate_shadow(), initialize(), process_third_body(), and update_third_body_state().

8.12.6.10 output_flux

```
double jeod::RadiationThirdBody::output_flux[3] {}
```

Calculated value.

Set only in derived class RadiationReflectingThirdBody.trick_units(–)

Definition at line 140 of file radiation_third_body.hh.

8.12.6.11 planet_link

```
Planet* jeod::RadiationThirdBody::planet_link {}  [protected]
```

Pointer to the associated Planet object.

trick_units(–)

Definition at line 175 of file radiation_third_body.hh.

Referenced by initialize().

8.12.6.12 primary_source_ptr

```
RadiationSource* jeod::RadiationThirdBody::primary_source_ptr {}
```

Pointer to the primary illumination source.

trick_units(–)

Definition at line 101 of file radiation_third_body.hh.

Referenced by calculate_shadow(), initialize(), and update_third_body_state().

8.12.6.13 r_minus

```
double jeod::RadiationThirdBody::r_minus {}  [protected]
```

The value (radius - R_source)

trick_units(–)

Definition at line 185 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and initialize().

8.12.6.14 r_par

```
double jeod::RadiationThirdBody::r_par {}  [protected]
```

Component of vector from RadiationThirdBody to vehicle that is parallel to vector from primary source to RadiationThirdBody.

trick_units(m)

Definition at line 197 of file radiation_third_body.hh.

Referenced by calculate_shadow().

8.12.6.15 r_perp

```
double jeod::RadiationThirdBody::r_perp {}  [protected]
```

Component of vector from RadiationThirdBody to vehicle that is perpendicular to vector from primary source to RadiationThirdBody.

trick_units(–)

Definition at line 203 of file radiation_third_body.hh.

Referenced by calculate_shadow().

8.12.6.16 r_plus

```
double jeod::RadiationThirdBody::r_plus {}  [protected]
```

The value (radius + R_source)

trick_units(–)

Definition at line 180 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and initialize().

8.12.6.17 r_ratio

```
double jeod::RadiationThirdBody::r_ratio {}  [protected]
```

The ratio (radius / R_source)

trick_units(–)

Definition at line 190 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and initialize().

8.12.6.18 radius

```
double jeod::RadiationThirdBody::radius {}
```

Radius of RadiationThirdBody.

Note - this is set automatically if RTB is a planet, but must be set manually if RTB is a DynBody.trick_units(m)

Definition at line 134 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and initialize().

8.12.6.19 shadow_geometry

```
ShadowGeometry jeod::RadiationThirdBody::shadow_geometry {Conical}
```

Flag indicating cylindrical / conical shadow geometry.

trick_units(–)

Definition at line 106 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and convert_shadow_from_int().

8.12.6.20 source_to_third_hat_inrtl

```
double jeod::RadiationThirdBody::source_to_third_hat_inrtl[3] {}  [protected]
```

Unit vector of source_to_third.

trick_units(–)

Definition at line 225 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and update_third_body_state().

8.12.6.21 source_to_third_inrtl

```
double jeod::RadiationThirdBody::source_to_third_inrtl[3] {}  [protected]
```

Vector from the primary source to the RadiationThirdBody in inertial RF.

trick_units(m)

Definition at line 215 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and update_third_body_state().

8.12.6.22 third_to_cg_inrtl

```
double jeod::RadiationThirdBody::third_to_cg_inrtl[3] {}  [protected]
```

Vector from RadiationThirdBody to vehicle center of gravity, in inertial RF.

trick_units(m)

Definition at line 209 of file radiation_third_body.hh.

Referenced by calculate_shadow().

8.12.6.23 update_interval

```
double jeod::RadiationThirdBody::update_interval {}
```

Time interval between third body state updates.

trick_units(s)

Definition at line 128 of file radiation_third_body.hh.

Referenced by test_for_state_update().

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

- radiation_third_body.hh
- radiation_third_body.cc

Chapter 9

File Documentation

9.1 flat_plate_radiation_facet.cc File Reference

Define member functions for class FlatPlateRadiationFacet.

```
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/surface_model/include/flat_plate.hh"
#include "../include/flat_plate_radiation_facet.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_third_body.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.1.1 Detailed Description

Define member functions for class FlatPlateRadiationFacet.

9.2 flat_plate_radiation_facet.hh File Reference

Individual facets for use with rad environment interaction models.

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

Data Structures

- class jeod::FlatPlateRadiationFacet

  A flat plate facet to be used for radiation interaction.

Namespaces

- jeod

  Namespace jeod.

9.2.1 Detailed Description

Individual facets for use with rad environment interaction models.

9.3 flat_plate_radiation_factory.cc File Reference

Factory that creates a FlatPlateRadiationFacet, from a facet model.

```
#include <cstddef>
#include <typeinfo>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/surface_model/include/flat_plate_thermal.hh"
#include "../include/flat_plate_radiation_facet.hh"
#include "../include/flat_plate_radiation_factory.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_params.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.3.1 Detailed Description

Factory that creates a FlatPlateRadiationFacet, from a facet model.

9.4 flat_plate_radiation_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"
#include "utils/surface_model/include/interaction_facet_factory.hh"
#include "flat_plate_radiation_facet.hh"
```

Data Structures

- class jeod::FlatPlateRadiationFactory

  The factory for building flat plate radiation facets.

Namespaces

- jeod

  Namespace jeod.

9.4.1 Detailed Description

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

9.5 radiation_base_facet.cc File Reference

Define member functions for class RadiationBaseFacet.

```
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/radiation_base_facet.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_third_body.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.5.1 Detailed Description

Define member functions for class RadiationBaseFacet.

9.6 radiation_base_facet.hh File Reference

Individual facets for use with radiation environment interaction models.

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

Data Structures

- class jeod::RadiationBaseFacet

  Generic type of facet for radiation pressure.

Namespaces

- jeod

  Namespace jeod.

9.6.1 Detailed Description

Individual facets for use with radiation environment interaction models.

9.7 radiation_default_surface.cc File Reference

Default surface for use with Radiation Pressure interaction model.

```
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "interactions/thermal_rider/include/thermal_facet_rider.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/radiation_default_surface.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_third_body.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.7.1 Detailed Description

Default surface for use with Radiation Pressure interaction model.

9.8 radiation_default_surface.hh File Reference

Individual facets for use with radiation environment interaction models.

```
#include <string>
#include <utility>
#include "interactions/thermal_rider/include/thermal_facet_rider.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "radiation_base_facet.hh"
```

Data Structures

- class jeod::RadiationDefaultSurface

  The default spherical surface for radiation pressure.

Namespaces

- jeod

  Namespace jeod.

9.8.1 Detailed Description

Individual facets for use with radiation environment interaction models.

9.9 radiation_facet.cc File Reference

Individual facets for use with Radiation Pressure interaction models.

```
#include "interactions/thermal_rider/include/thermal_facet_rider.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/surface_model/include/facet.hh"
#include "../include/radiation_facet.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_params.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.9.1 Detailed Description

Individual facets for use with Radiation Pressure interaction models.

9.10 radiation_facet.hh File Reference

Individual facets for use with radiation environment interaction models.

```
#include "interactions/thermal_rider/include/thermal_facet_rider.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/surface_model/include/interaction_facet.hh"
#include "radiation_base_facet.hh"
#include "utils/surface_model/include/facet.hh"
```

Data Structures

- class jeod::RadiationFacet

  Generic type of facet for radiation pressure.

Namespaces

- jeod

  Namespace jeod.

9.10.1 Detailed Description

Individual facets for use with radiation environment interaction models.

9.11 radiation_messages.cc File Reference

Implement the class RadiationMessages.

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

Namespaces

- jeod

  Namespace jeod.

Macros

- #define MAKE_RADIATION_MESSAGE_CODE(id) JEOD_MAKE_MESSAGE_CODE(RadiationMessages, "interactions/radiation_pressure/", id)

9.11.1 Detailed Description

Implement the class RadiationMessages.

9.11.2 Macro Definition Documentation

9.11.2.1 MAKE_RADIATION_MESSAGE_CODE

```
#define MAKE_RADIATION_MESSAGE_CODE(
              id ) JEOD_MAKE_MESSAGE_CODE(RadiationMessages, "interactions/radiation_pressure/",
id)
```

Definition at line 43 of file radiation_messages.cc.

9.12 radiation_messages.hh File Reference

Define the class RadiationMessages, the class that specifies the message IDs used in the Radiation model.

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

Data Structures

- class jeod::RadiationMessages
  Provides error messages.

Namespaces

- jeod
  Namespace jeod.

9.12.1 Detailed Description

Define the class RadiationMessages, the class that specifies the message IDs used in the Radiation model.

9.13 radiation_params.hh File Reference

A virtual base class for radiation facet parameters, used to create interaction facets for radiation in the Interaction↩ SurfaceFactorys.

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

Data Structures

- class jeod::RadiationParams
  Provides a parameter list to each facet, based on the facet material properties.

Namespaces

- jeod
  Namespace jeod.

9.13.1 Detailed Description

A virtual base class for radiation facet parameters, used to create interaction facets for radiation in the Interaction↩ SurfaceFactorys.

9.14 radiation_pressure.cc File Reference

Calculation of force and torque due to radiation pressure.

```
#include <cstddef>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "interactions/thermal_rider/include/thermal_facet_rider.hh"
#include "utils/math/include/vector3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_pressure.hh"
#include "../include/radiation_source.hh"
#include "../include/radiation_third_body.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.14.1 Detailed Description

Calculation of force and torque due to radiation pressure.

9.15 radiation_pressure.hh File Reference

Radiation pressure top-level definition.

```
#include "interactions/thermal_rider/include/thermal_model_rider.hh"
#include "utils/container/include/pointer_vector.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "radiation_source.hh"
#include "radiation_default_surface.hh"
#include "radiation_surface.hh"
```

Data Structures

- class jeod::RadiationPressure

  Radiation pressure top-level definition.

Namespaces

- jeod

  Namespace jeod.

9.15.1 Detailed Description

Radiation pressure top-level definition.

9.16 radiation_pressure__default_surface.cc File Reference

Calculation of force and torque due to radiation pressure.

```
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "interactions/thermal_rider/include/thermal_model_rider.hh"
#include "utils/math/include/vector3.hh"
#include "../include/radiation_default_surface.hh"
#include "../include/radiation_pressure.hh"
#include "../include/radiation_source.hh"
#include "../include/radiation_third_body.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.16.1 Detailed Description

Calculation of force and torque due to radiation pressure.

9.17 radiation_pressure__surface_model.cc File Reference

Calculation of force and torque due to radiation pressure.

```
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "interactions/thermal_rider/include/thermal_model_rider.hh"
#include "utils/math/include/vector3.hh"
#include "../include/radiation_pressure.hh"
#include "../include/radiation_source.hh"
#include "../include/radiation_surface.hh"
#include "../include/radiation_third_body.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.17.1 Detailed Description

Calculation of force and torque due to radiation pressure.

9.18 radiation_source.cc File Reference

Definition of methods associated with Primary Sources.

```
#include <cstddef>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/math/include/vector3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame.hh"
#include "../include/radiation_source.hh"
#include "../include/radiation_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.18.1 Detailed Description

Definition of methods associated with Primary Sources.

9.19 radiation_source.hh File Reference

Radiation pressure parameter and variable definitions.

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

Data Structures

- class jeod::RadiationSource

  Provides information on the source of the incident radiation.

Namespaces

- jeod

  Namespace jeod.

9.19.1 Detailed Description

Radiation pressure parameter and variable definitions.

9.20 radiation_surface.cc File Reference

Vehicle surface model for Radiation Pressure model.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/math/include/vector3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/surface_model/include/facet.hh"
#include "utils/surface_model/include/interaction_facet.hh"
#include "utils/surface_model/include/interaction_facet_factory.hh"
#include "../include/radiation_facet.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_surface.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.20.1 Detailed Description

Vehicle surface model for Radiation Pressure model.

9.21 radiation_surface.hh File Reference

Vehicle surface model for general environment interaction models.

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

Data Structures

- class jeod::RadiationSurface

  The surface of the vehicle that interacts with the incident flux.

Namespaces

- jeod

  Namespace jeod.

9.21.1 Detailed Description

Vehicle surface model for general environment interaction models.

9.22 radiation_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/message/include/message_handler.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_params.hh"
#include "../include/radiation_surface_factory.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.22.1 Detailed Description

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

9.23 radiation_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/surface_model/include/interaction_surface_factory.hh"
#include "flat_plate_radiation_factory.hh"
```

Data Structures

- class jeod::RadiationSurfaceFactory

  The factory for creating Radiation Surfaces.

Namespaces

- jeod

  Namespace jeod.

9.23.1 Detailed Description

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

9.24 radiation_third_body.cc File Reference

Provides the functionality associated with RadiationThirdBodys.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_source.hh"
#include "../include/radiation_third_body.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.24.1 Detailed Description

Provides the functionality associated with RadiationThirdBodys.

9.25 radiation_third_body.hh File Reference

Define the class RadiationThirdBody.

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

Data Structures

- class jeod::RadiationThirdBody

  Provide information on bodies that may cause shadowing or reflected illumination.

Namespaces

- jeod

  Namespace jeod.

9.25.1 Detailed Description

Define the class RadiationThirdBody.

Index