RadiationPressureModel

5.0

Generated by Doxygen 1.8.5

Wed Jun 1 2022 12:07:44

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 InteractionSurfaceFactorys.
- 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_params.cc

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

Macros

- #define PATH "interactions/radiation_pressure/"

6.3.1 Detailed Description

6.3.2 Macro Definition Documentation

6.3.2.1 #define PATH "interactions/radiation_pressure/"

Definition at line 39 of file radiation_messages.cc.

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

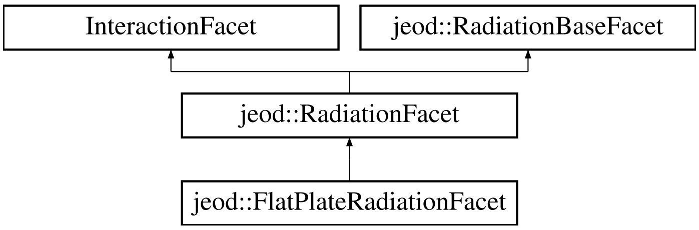
Data Structure Documentation

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 ()

  Construct a FlatPlateRadiationFacet.
- virtual ∼FlatPlateRadiationFacet ()

  Destructor for FlatPlateRadiationFacet.
- void incident_radiation (const double flux_mag, const double flux_struct_hat[3], const bool calculate_forces)

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

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

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

  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 Member Functions

- FlatPlateRadiationFacet & operator= (const FlatPlateRadiationFacet &rhs)
- FlatPlateRadiationFacet (const FlatPlateRadiationFacet &rhs)

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 52 of file flat_plate_radiation_facet.hh.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 jeod::FlatPlateRadiationFacet::FlatPlateRadiationFacet (void)

Construct a FlatPlateRadiationFacet.

Definition at line 61 of file flat_plate_radiation_facet.cc.

8.1.2.2 jeod::FlatPlateRadiationFacet::∼FlatPlateRadiationFacet (void) [virtual]

Destructor for FlatPlateRadiationFacet.

Definition at line 224 of file flat_plate_radiation_facet.cc.

8.1.2.3 jeod::FlatPlateRadiationFacet::FlatPlateRadiationFacet (const FlatPlateRadiationFacet & *rhs*) [private]

8.1.3 Member Function Documentation

8.1.3.1 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 76 of file flat_plate_radiation_facet.cc.

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

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

8.1.3.2 void jeod::FlatPlateRadiationFacet::incident_radiation (const double *flux_mag*, const double *flux_struct_hat[3]*, const bool *calculate_forces*) [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 120 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 void jeod::FlatPlateRadiationFacet::initialize_geom (double *center_grav[3]*) [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 98 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 FlatPlateRadiationFacet& jeod::FlatPlateRadiationFacet::operator= (const FlatPlateRadiationFacet & *rhs*) [private]

8.1.3.5 void jeod::FlatPlateRadiationFacet::radiation_pressure (void) [virtual]

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

Implements jeod::RadiationFacet.

Definition at line 183 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 void init_attrjeod__FlatPlateRadiationFacet ( ) [friend]

8.1.4.2 friend class InputProcessor [friend]

Definition at line 54 of file flat_plate_radiation_facet.hh.

8.1.5 Field Documentation

8.1.5.1 double jeod::FlatPlateRadiationFacet::incident_flux_hat[3]

Temporary value.

trick_units(–)

Definition at line 70 of file flat_plate_radiation_facet.hh.

8.1.5.2 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 65 of file flat_plate_radiation_facet.hh.

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

8.1.5.3 double jeod::FlatPlateRadiationFacet::sin_theta [private]

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

trick_units(–)

Definition at line 76 of file flat_plate_radiation_facet.hh.

Referenced by incident_radiation().

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

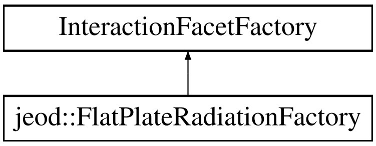
- flat_plate_radiation_facet.hh
- flat_plate_radiation_facet.cc

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 ()

Destructor for FlatPlateRadiationFactory.
- virtual InteractionFacet ∗ create_facet (Facet ∗facet, FacetParams ∗params)

  Records the data for the Flat Plate Radiation Facet.
- virtual bool is_correct_factory (Facet ∗facet)

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

Private Member Functions

- FlatPlateRadiationFactory & operator= (const FlatPlateRadiationFactory &rhs)
- FlatPlateRadiationFactory (const FlatPlateRadiationFactory &rhs)

Friends

- class InputProcessor
- void init_attrjeod__FlatPlateRadiationFactory ()

8.2.1 Detailed Description

The factory for building flat plate radiation facets.

Definition at line 51 of file flat_plate_radiation_factory.hh.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 jeod::FlatPlateRadiationFactory::FlatPlateRadiationFactory (void)

Constructor for FlatPlateRadiationFactory.

Definition at line 65 of file flat_plate_radiation_factory.cc.

8.2.2.2 jeod::FlatPlateRadiationFactory::∼FlatPlateRadiationFactory (void)

Destructor for FlatPlateRadiationFactory.

Definition at line 156 of file flat_plate_radiation_factory.cc.

8.2.2.3 jeod::FlatPlateRadiationFactory::FlatPlateRadiationFactory (const **FlatPlateRadiationFactory** & *rhs*) [private]

8.2.3 Member Function Documentation

8.2.3.1 InteractionFacet ∗ jeod::FlatPlateRadiationFactory::create_facet (Facet ∗ *facet,* FacetParams ∗ *params*) [virtual]

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 79 of file flat_plate_radiation_factory.cc.

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

8.2.3.2 bool jeod::FlatPlateRadiationFactory::is_correct_factory (Facet ∗ *facet*) [virtual]

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 138 of file flat_plate_radiation_factory.cc.

8.2.3.3 FlatPlateRadiationFactory& jeod::FlatPlateRadiationFactory::operator= (const FlatPlateRadiationFactory & *rhs*) [private]

8.2.4 Friends And Related Function Documentation

8.2.4.1 void init_attrjeod__FlatPlateRadiationFactory ( ) [friend]

8.2.4.2 friend class InputProcessor [friend]

Definition at line 53 of file flat_plate_radiation_factory.hh.

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

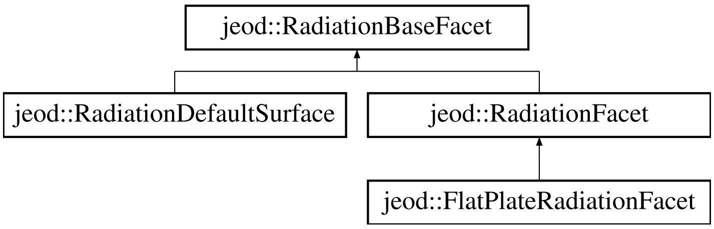
- flat_plate_radiation_factory.hh
- flat_plate_radiation_factory.cc

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 ()

  Construct a RadiationBaseFacet.
- virtual ∼RadiationBaseFacet ()

  Destructor for RadiationBaseFacet.
- virtual void initialize (void)

  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 (void)

  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

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

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

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

  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 const double speed_of_light = 299792458.0

  Speed of light in vacuum.

Private Member Functions

- RadiationBaseFacet & operator= (const RadiationBaseFacet &rhs)
- RadiationBaseFacet (const RadiationBaseFacet &rhs)

Friends

- class InputProcessor
- void init_attrjeod__RadiationBaseFacet ()

8.3.1 Detailed Description

Generic type of facet for radiation pressure.

Definition at line 51 of file radiation_base_facet.hh.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 jeod::RadiationBaseFacet::RadiationBaseFacet (void)

Construct a RadiationBaseFacet.

Definition at line 59 of file radiation_base_facet.cc.

8.3.2.2 jeod::RadiationBaseFacet::∼RadiationBaseFacet (void) [virtual]

Destructor for RadiationBaseFacet.

Definition at line 160 of file radiation_base_facet.cc.

8.3.2.3 jeod::RadiationBaseFacet::RadiationBaseFacet (const **RadiationBaseFacet** & *rhs*) [private]

8.3.3 Member Function Documentation

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

initializes the base surface

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

Definition at line 79 of file radiation_base_facet.cc.

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

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

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

To initialize the values during each update run.

Definition at line 147 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 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 125 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 RadiationBaseFacet& jeod::RadiationBaseFacet::operator= (const RadiationBaseFacet & *rhs*) [private]

8.3.4 Friends And Related Function Documentation

8.3.4.1 void init_attrjeod__RadiationBaseFacet ( ) [friend]

8.3.4.2 friend class InputProcessor [friend]

Definition at line 53 of file radiation_base_facet.hh.

8.3.5 Field Documentation

8.3.5.1 double jeod::RadiationBaseFacet::albedo

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

trick_units(--)

Definition at line 63 of file radiation_base_facet.hh.

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

8.3.5.2 double jeod::RadiationBaseFacet::albedo_IR

Fraction of incident IR radiation that is immediately reflected.

trick_units(--)

Definition at line 73 of file radiation_base_facet.hh.

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

8.3.5.3 double jeod::RadiationBaseFacet::albedo_vis

Fraction of incident visible radiation that is immediately reflected.

trick_units(–)

Definition at line 68 of file radiation_base_facet.hh.

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

8.3.5.4 double jeod::RadiationBaseFacet::areaxflux

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

Highly variable.trick_units(–)

Definition at line 103 of file radiation_base_facet.hh.

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

8.3.5.5 double jeod::RadiationBaseFacet::areaxflux_e

product of energy flux and cross-sectional area

trick_units(–)

Definition at line 108 of file radiation_base_facet.hh.

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

8.3.5.6 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 94 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_-radiation(), jeod::RadiationDefaultSurface::initialize(), and jeod::RadiationDefaultSurface::RadiationDefault-Surface().

8.3.5.7 double jeod::RadiationBaseFacet::diffuse

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

trick_units(–)

Definition at line 79 of file radiation_base_facet.hh.

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

8.3.5.8 double jeod::RadiationBaseFacet::F_absorption[3]

Force due to photon absorption from ONLY ONE source.

trick_units(–)

Definition at line 113 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_radiation(), initialize_runtime_values(), jeod::FlatPlateRadiationFacet::radiation_pressure(), jeod::RadiationDefaultSurface::RadiationDefaultSurface(), and jeod::RadiationPressure::update_default_surface().

8.3.5.9 double jeod::RadiationBaseFacet::F_diffuse[3]

Force due to photon diffuse reflection from ONLY ONE source.

trick_units(--)

Definition at line 123 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_radiation(), initialize_runtime_values(), jeod::FlatPlateRadiationFacet::radiation_pressure(), jeod::RadiationDefaultSurface::RadiationDefaultSurface(), and jeod::RadiationPressure::update_default_surface().

8.3.5.10 double jeod::RadiationBaseFacet::F_emission[3]

Force due to photon (thermal) emission.

trick_units(--)

Definition at line 128 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::radiation_pressure(), and jeod::RadiationDefaultSurface::RadiationDefaultSurface().

8.3.5.11 double jeod::RadiationBaseFacet::F_specular[3]

Force due to photon specular reflection from ONLY ONE source.

trick_units(--)

Definition at line 118 of file radiation_base_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_radiation(), initialize_runtime_values(), jeod::FlatPlateRadiationFacet::radiation_pressure(), jeod::RadiationDefaultSurface::RadiationDefaultSurface(), and jeod::RadiationPressure::update_default_surface().

8.3.5.12 const 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 136 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 ThermalFacetRider jeod::RadiationBaseFacet::thermal

thermal characteristics of the facet.

trick_units(--)

Definition at line 84 of file radiation_base_facet.hh.

Referenced by jeod::RadiationSurface::accumulate_thermal_sources(), jeod::RadiationDefaultSurface::add_thermal_integrator_to(), jeod::RadiationFacet::define_facet_core(), jeod::RadiationSurface::equalize_absorption_emission(), jeod::RadiationFacet::get_thermal_integrator(), jeod::FlatPlateRadiationFacet::incident_radiation(), jeod::RadiationDefaultSurface::incident_radiation(), jeod::RadiationFacet::initialize(), jeod::RadiationSurface::initialize(), jeod::RadiationDefaultSurface::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:

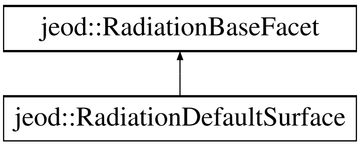
- radiation_base_facet.hh
- radiation_base_facet.cc

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 ()

  Constructor for RadiationDefaultSurface.
- virtual ∼RadiationDefaultSurface ()

  Destructor for RadiationDefaultSurface.
- virtual void initialize (void)

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

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

  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

  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.

Private Member Functions

- RadiationDefaultSurface & operator= (const RadiationDefaultSurface &rhs)
- RadiationDefaultSurface (const RadiationDefaultSurface &rhs)

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 55 of file radiation_default_surface.hh.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 jeod::RadiationDefaultSurface::RadiationDefaultSurface (void)

Constructor for RadiationDefaultSurface.

Definition at line 69 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_emission, jeod::RadiationBaseFacet::F_specular, rad_coeff, surface_area, and temperature.

8.4.2.2 jeod::RadiationDefaultSurface::∼RadiationDefaultSurface (void) [virtual]

Destructor for RadiationDefaultSurface.

Definition at line 303 of file radiation_default_surface.cc.

8.4.2.3 jeod::RadiationDefaultSurface::RadiationDefaultSurface (const RadiationDefaultSurface & *rhs*) [private]

8.4.3 Member Function Documentation

8.4.3.1 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 293 of file radiation_default_surface.cc.

References jeod::RadiationBaseFacet::thermal.

8.4.3.2 void jeod::RadiationDefaultSurface::incident_radiation (const double *flux_mag,* const double *flux_struc_hat[3],* const bool *calculate_forces*) [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 222 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 void jeod::RadiationDefaultSurface::initialize (void) [virtual]

initializes the default surface

Reimplemented from jeod::RadiationBaseFacet.

Definition at line 93 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 RadiationDefaultSurface& jeod::RadiationDefaultSurface::operator= (const RadiationDefaultSurface & *rhs*) [private]

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

Setter for the name.

Definition at line 126 of file radiation_default_surface.hh.

8.4.3.6 void jeod::RadiationDefaultSurface::thermal_update (void)

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

Definition at line 274 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 void init_attrjeod__RadiationDefaultSurface ( ) [friend]

8.4.4.2 friend class InputProcessor [friend]

Definition at line 57 of file radiation_default_surface.hh.

8.4.5 Field Documentation

8.4.5.1 std::string jeod::RadiationDefaultSurface::name

The name of the surface.

This is optional.trick_units(–)

Definition at line 85 of file radiation_default_surface.hh.

8.4.5.2 double jeod::RadiationDefaultSurface::rad_coeff

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 75 of file radiation_default_surface.hh.

Referenced by initialize(), and RadiationDefaultSurface().

8.4.5.3 double jeod::RadiationDefaultSurface::surface_area

surface area of the default sphere.

trick_units(m2)

Definition at line 95 of file radiation_default_surface.hh.

Referenced by initialize(), and RadiationDefaultSurface().

8.4.5.4 double jeod::RadiationDefaultSurface::temperature

The value of the surface kinetic temperature.

trick_units(K)

Definition at line 80 of file radiation_default_surface.hh.

Referenced by initialize(), RadiationDefaultSurface(), and thermal_update().

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

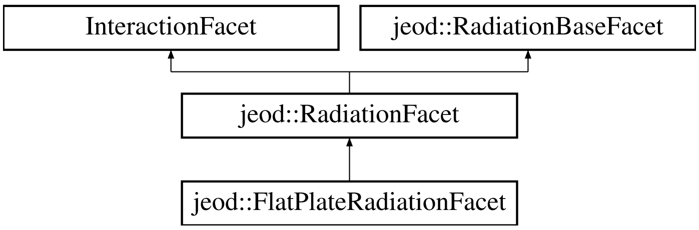
- radiation_default_surface.hh
- radiation_default_surface.cc

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 ()

  Constructor for Radiationfacet.
- virtual ∼RadiationFacet ()

  Destructor for RadiationFacet.
- 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.
- virtual void initialize ()

  Run sanity checks on input variables.
- void define_facet_core (Facet ∗facet, 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.

Private Member Functions

- RadiationFacet & operator= (const RadiationFacet &rhs)
- RadiationFacet (const RadiationFacet &rhs)

Friends

- class InputProcessor
- void init_attrjeod__RadiationFacet ()

8.5.1 Detailed Description

Generic type of facet for radiation pressure.

Definition at line 53 of file radiation_facet.hh.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 jeod::RadiationFacet::RadiationFacet (void)

Constructor for Radiationfacet.

Definition at line 61 of file radiation_facet.cc.

References center_pressure, and crot_to_cp.

8.5.2.2 jeod::RadiationFacet::∼RadiationFacet (void) [virtual]

Destructor for RadiationFacet.

Definition at line 123 of file radiation_facet.cc.

8.5.2.3 jeod::RadiationFacet::RadiationFacet (const RadiationFacet & *rhs*) [private]

8.5.3 Member Function Documentation

8.5.3.1 void jeod::RadiationFacet::define_facet_core (Facet ∗ *facet,* 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 102 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 ThermalIntegrableObject& jeod::RadiationFacet::get_thermal_integrator ( ) [inline]

Get the integrator for thermal characteristics.

Returns

Integrable object that integrates temperature.

Definition at line 120 of file radiation_facet.hh.

References jeod::RadiationBaseFacet::thermal.

8.5.3.3 void jeod::RadiationFacet::initialize (void) [virtual]

Run sanity checks on input variables.

Reimplemented from jeod::RadiationBaseFacet.

Definition at line 72 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 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 RadiationFacet& jeod::RadiationFacet::operator= (const RadiationFacet & *rhs*) `[private]`

8.5.3.6 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 void init_attrjeod__RadiationFacet ( ) `[friend]`

8.5.4.2 friend class InputProcessor `[friend]`

Definition at line 55 of file radiation_facet.hh.

8.5.5 Field Documentation

8.5.5.1 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 74 of file radiation_facet.hh.

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

8.5.5.2 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 67 of file radiation_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::initialize_geom(), jeod::FlatPlateRadiationFacet::radiation_pressure(), and RadiationFacet().

8.5.5.3 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 84 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>
```

Static Public Attributes

- static char const ∗ incomplete_setup_error

  The setup was not fully defined.
- static char const ∗ invalid_setup_error

  The setup is invalid.
- static char const ∗ operational_setup_error

  Something internal went wrong in the setup methods without obvious user error.
- static char const ∗ invalid_function_call

  A function was called before it was fully implemented.
- static char const ∗ unknown_numerical_error

  Something went horribly wrong.

Private Member Functions

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

Friends

- class InputProcessor
- void init_attrjeod__RadiationMessages ()

8.6.1 Detailed Description

Provides error messages.

Definition at line 52 of file radiation_messages.hh.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 jeod::RadiationMessages::RadiationMessages (void) [private]

8.6.2.2 jeod::RadiationMessages::RadiationMessages (const RadiationMessages &) [private]

8.6.3 Member Function Documentation

8.6.3.1 RadiationMessages& jeod::RadiationMessages::operator= (const RadiationMessages &) [private]

8.6.4 Friends And Related Function Documentation

8.6.4.1 void init_attrjeod__RadiationMessages ( ) [friend]

8.6.4.2 friend class InputProcessor [friend]

Definition at line 55 of file radiation_messages.hh.

8.6.5 Field Documentation

8.6.5.1 char const ∗ jeod::RadiationMessages::incomplete_setup_error [static]

Initial value:

```
=
    "interactions/radiation_pressure/"  "incomplete_setup_error"
```

The setup was not fully defined.

trick_units(–)

Definition at line 64 of file radiation_messages.hh.

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

8.6.5.2 char const ∗ jeod::RadiationMessages::invalid_function_call [static]

Initial value:

```
=
    "interactions/radiation_pressure/"  "invalid_function_call"
```

A function was called before it was fully implemented.

trick_units(–)

Definition at line 77 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 char const ∗ jeod::RadiationMessages::invalid_setup_error [static]

Initial value:

```
=
    "interactions/radiation_pressure/"  "invalid_setup_error"
```

The setup is invalid.

trick_units(–)

Definition at line 68 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::RadiationFacet::initialize(), jeod::RadiationSurface::initialize(), jeod::RadiationDefaultSurface::initialize(), jeod::RadiationBaseFacet::initialize(), jeod::RadiationThirdBody::initialize(), jeod::RadiationThirdBody::process_third_body(), and jeod::RadiationThirdBody::update_third_body_state().

8.6.5.4 char const ∗ jeod::RadiationMessages::operational_setup_error [static]

Initial value:

```
=
     "interactions/radiation_pressure/"  "operational_setup_error"
```

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

trick_units(–)

Definition at line 73 of file radiation_messages.hh.

Referenced by jeod::RadiationSurface::allocate_array(), jeod::RadiationSurface::allocate_interaction_facet(), jeod-::RadiationSurface::initialize(), jeod::RadiationDefaultSurface::initialize(), and jeod::RadiationThirdBody::process_-third_body().

8.6.5.5 char const ∗ jeod::RadiationMessages::unknown_numerical_error [static]

Initial value:

```
=
     "interactions/radiation_pressure/"  "unknown_numerical_error"
```

Something went horribly wrong.

trick_units(–)

Definition at line 82 of file radiation_messages.hh.

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

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

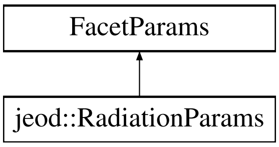
- radiation_messages.hh
- radiation_messages.cc

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 ()

  Constructor for RadiationParams.
- virtual ∼RadiationParams ()

  Destructor for RadiationParams.

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.

Private Member Functions

- RadiationParams & operator= (const RadiationParams &rhs)
- RadiationParams (const RadiationParams &rhs)

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 48 of file radiation_params.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 jeod::RadiationParams::RadiationParams (void)

Constructor for RadiationParams.

Definition at line 44 of file radiation_params.cc.

References albedo, and diffuse.

8.7.2.2 jeod::RadiationParams::∼RadiationParams (void) [virtual]

Destructor for RadiationParams.

Definition at line 54 of file radiation_params.cc.

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

8.7.3 Member Function Documentation

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

8.7.4 Friends And Related Function Documentation

8.7.4.1 void init_attrjeod__RadiationParams ( ) `[friend]`

8.7.4.2 friend class InputProcessor `[friend]`

Definition at line 50 of file radiation_params.hh.

8.7.5 Field Documentation

8.7.5.1 double jeod::RadiationParams::albedo

Fraction of incident radiation that is immediately reflected.

trick_units(--)

Definition at line 58 of file radiation_params.hh.

Referenced by jeod::RadiationFacet::define_facet_core(), and RadiationParams().

8.7.5.2 double jeod::RadiationParams::diffuse

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

trick_units(--)

Definition at line 64 of file radiation_params.hh.

Referenced by jeod::RadiationFacet::define_facet_core(), and RadiationParams().

8.7.5.3 ThermalParams jeod::RadiationParams::thermal

Additional thermal parameters.

trick_units(--)

Definition at line 69 of file radiation_params.hh.

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

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

- radiation_params.hh
- radiation_params.cc

8.8 jeod::RadiationPressure Class Reference

Radiation pressure top-level definition.

`#include <radiation_pressure.hh>`

Public Member Functions

- RadiationPressure ()

  Constructor for RadiationPressure.
- virtual ∼RadiationPressure ()

  Destructor for RadiationPressure.
- 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 char ∗third_body_name)

  Sets a Third Body to be active when it previously was not.
- void set_third_body_inactive (const char ∗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

  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 (void)

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

  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 char ∗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

  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.

Private Member Functions

- RadiationPressure (const RadiationPressure &)
- RadiationPressure & operator= (const RadiationPressure &)

Friends

- class InputProcessor
- void init_attrjeod__RadiationPressure ()

8.8.1 Detailed Description

Radiation pressure top-level definition.

Definition at line 58 of file radiation_pressure.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 jeod::RadiationPressure::RadiationPressure (void)

Constructor for RadiationPressure.

Definition at line 72 of file radiation_pressure.cc.

References active, calculate_forces, default_surface_ptr, dyn_manager_ptr, force, illum_factor, initialized, num_third_bodies, surface_ptr, third_bodies, third_bodies_active, and torque.

8.8.2.2 jeod::RadiationPressure::∼RadiationPressure (void) [virtual]

Destructor for RadiationPressure.

Definition at line 396 of file radiation_pressure.cc.

References third_bodies.

8.8.2.3 jeod::RadiationPressure::RadiationPressure (const RadiationPressure &) [private]

8.8.3 Member Function Documentation

8.8.3.1 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 186 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 int jeod::RadiationPressure::find_third_body (const char ∗ *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 360 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 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 79 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 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 75 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 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 97 of file radiation_pressure.cc.

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

Referenced by initialize().

8.8.3.6 RadiationPressure& jeod::RadiationPressure::operator= (const RadiationPressure &) [private]

8.8.3.7 void jeod::RadiationPressure::set_calculate_forces (bool *value*)

Sets the value calculate_forces.

Parameters

in	*value*	whether forces are needed

Definition at line 381 of file radiation_pressure.cc.

References calculate_forces, force, and torque.

8.8.3.8 void jeod::RadiationPressure::set_third_body_active (const char ∗ *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 282 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 void jeod::RadiationPressure::set_third_body_inactive (const char ∗ *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 319 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 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

Definition at line 162 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 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 118 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 void jeod::RadiationPressure::update_default_surface (void) [protected]

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

Definition at line 100 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::RadiationBaseFacet::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 void jeod::RadiationPressure::update_facet_surface (void) [protected]

Used to update the model when the surface comprises facets.

Definition at line 115 of file radiation_pressure__surface_model.cc.

References active, calculate_forces, jeod::RadiationSurface::equalize_absorption_emission(), jeod::RadiationSource::flux_mag, jeod::RadiationSource::flux_struc_hat, force, jeod::RadiationSurface::force, jeod::RadiationSurface::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 void init_attrjeod__RadiationPressure ( ) [friend]

8.8.4.2 friend class InputProcessor [friend]

Definition at line 60 of file radiation_pressure.hh.

8.8.5 Field Documentation

8.8.5.1 bool jeod::RadiationPressure::active

Is radiation pressure desired?

trick_units(–)

Definition at line 68 of file radiation_pressure.hh.

Referenced by RadiationPressure(), set_third_body_active(), set_third_body_inactive(), third_body_adjustments(), update(), update_default_surface(), and update_facet_surface().

8.8.5.2 bool jeod::RadiationPressure::calculate_forces [protected]

Flag to indicate whether forces and torques should be calculated.

trick_units(–)

Definition at line 105 of file radiation_pressure.hh.

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

8.8.5.3 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 135 of file radiation_pressure.hh.

Referenced by initialize(), RadiationPressure(), and update_default_surface().

8.8.5.4 DynManager* jeod::RadiationPressure::dyn_manager_ptr [protected]

Pointer to the Dynamics Manager.

trick_units(–)

Definition at line 140 of file radiation_pressure.hh.

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

8.8.5.5 double jeod::RadiationPressure::force[3]

Net force due to radiation.

trick_units(N)

Definition at line 73 of file radiation_pressure.hh.

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

8.8.5.6 double jeod::RadiationPressure::illum_factor

fraction of primary flux that is not eclipsed

trick_units(–)

Definition at line 83 of file radiation_pressure.hh.

Referenced by RadiationPressure(), and third_body_adjustments().

8.8.5.7 bool jeod::RadiationPressure::initialized [protected]

Has model been initialized?

trick_units(–)

Definition at line 100 of file radiation_pressure.hh.

Referenced by add_third_body(), initialize_environment(), and RadiationPressure().

8.8.5.8 unsigned int jeod::RadiationPressure::num_third_bodies [protected]

Number of ThirdBodies available.

trick_units(count)

Definition at line 116 of file radiation_pressure.hh.

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

8.8.5.9 RadiationSource jeod::RadiationPressure::source

Collection of data for radiation source.

trick_units(–)

Definition at line 88 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 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 129 of file radiation_pressure.hh.

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

8.8.5.11 ThermalModelRider jeod::RadiationPressure::thermal

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

trick_units(–)

Definition at line 93 of file radiation_pressure.hh.

Referenced by initialize(), and update_facet_surface().

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

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

trick_io(**)

Definition at line 122 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 bool jeod::RadiationPressure::third_bodies_active [protected]

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

trick_units(–)

Definition at line 111 of file radiation_pressure.hh.

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

8.8.5.14 double jeod::RadiationPressure::torque[3]

Net torque due to radiation.

trick_units(N*m)

Definition at line 78 of file radiation_pressure.hh.

Referenced by RadiationPressure(), 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 ()

  Constructor for RadiationSource.
- virtual ∼RadiationSource ()

  destructor for RadiationSource
- 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

  Solar Luminosity.
- const double solar_radius

  Mean solar radius.
- std::string name

  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

  Luminosity of primary source.
- double 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.

Private Member Functions

- RadiationSource (const RadiationSource &)
- RadiationSource & operator= (const RadiationSource &)

Friends

- class InputProcessor
- void init_attrjeod__RadiationSource ()

8.9.1 Detailed Description

Provides information on the source of the incident radiation.

Definition at line 57 of file radiation_source.hh.

8.9.2 Member Enumeration Documentation

8.9.2.1 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 174 of file radiation_source.hh.

8.9.3 Constructor & Destructor Documentation

8.9.3.1 jeod::RadiationSource::RadiationSource (void)

Constructor for RadiationSource.

Definition at line 66 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, multiple_shadow_bodies, name, radius, solar_luminosity, solar_radius, source_to_cg, and source_to_struc_origin.

8.9.3.2 jeod::RadiationSource::∼RadiationSource (void) [virtual]

destructor for RadiationSource

Definition at line 191 of file radiation_source.cc.

8.9.3.3 jeod::RadiationSource::RadiationSource (const **RadiationSource** &) [private]

8.9.4 Member Function Documentation

8.9.4.1 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 104 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 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 168 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 **RadiationSource**& jeod::RadiationSource::operator= (const **RadiationSource** &) [private]

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

Setter for the name.

Definition at line 200 of file radiation_source.hh.

References name.

8.9.5 Friends And Related Function Documentation

8.9.5.1 void init_attrjeod__RadiationSource ( ) [friend]

8.9.5.2 friend class InputProcessor [friend]

Definition at line 58 of file radiation_source.hh.

8.9.6 Field Documentation

8.9.6.1 bool jeod::RadiationSource::bodies_active

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

trick_units(–)

Definition at line 156 of file radiation_source.hh.

8.9.6.2 double jeod::RadiationSource::d_source_to_cg

distance from source to vehicle cg.

trick_units(m)

Definition at line 92 of file radiation_source.hh.

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

8.9.6.3 double jeod::RadiationSource::flux_hat[3]

(inertial) unit vector associated with primary radiative flux.

trick_units(–)

Definition at line 81 of file radiation_source.hh.

Referenced by calculate_flux(), and RadiationSource().

8.9.6.4 double jeod::RadiationSource::flux_inertial[3]

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

trick_units(–)

Definition at line 107 of file radiation_source.hh.

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

8.9.6.5 double jeod::RadiationSource::flux_mag

magnitude of principle radiative flux vector

trick_units(N/m2)

Definition at line 86 of file radiation_source.hh.

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

8.9.6.6 double jeod::RadiationSource::flux_struc[3]

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

trick_units(–)

Definition at line 112 of file radiation_source.hh.

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

8.9.6.7 double jeod::RadiationSource::flux_struc_hat[3]

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

trick_units(–)

Definition at line 118 of file radiation_source.hh.

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

8.9.6.8 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 124 of file radiation_source.hh.

Referenced by calculate_flux(), and RadiationSource().

8.9.6.9 RefFrame∗ jeod::RadiationSource::inertial_frame_ptr

the inertial reference frame associated with this source

trick_units(–)

Definition at line 144 of file radiation_source.hh.

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

8.9.6.10 double jeod::RadiationSource::luminosity

Luminosity of primary source.

trick_units(–)

Definition at line 97 of file radiation_source.hh.

Referenced by calculate_flux(), and RadiationSource().

8.9.6.11 bool jeod::RadiationSource::multiple_shadow_bodies

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

trick_units(–)

Definition at line 139 of file radiation_source.hh.

Referenced by RadiationSource().

8.9.6.12 std::string jeod::RadiationSource::name

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

trick_units(–)

Definition at line 76 of file radiation_source.hh.

Referenced by initialize(), RadiationSource(), and set_name().

8.9.6.13 unsigned int jeod::RadiationSource::num_bodies

number of ThirdBodies available.

trick_units(count)

Definition at line 161 of file radiation_source.hh.

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

8.9.6.14 double jeod::RadiationSource::radius

Radius of primary source.

trick_units(m)

Definition at line 102 of file radiation_source.hh.

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

8.9.6.15 OldShadowGeometry jeod::RadiationSource::shadow_geometry

Flag indicating cylindrical / conical shadow geometry.

trick_units(–)

Definition at line 184 of file radiation_source.hh.

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

8.9.6.16 const double jeod::RadiationSource::solar_luminosity

Solar Luminosity.

trick_units(–)

Definition at line 66 of file radiation_source.hh.

Referenced by RadiationSource().

8.9.6.17 const double jeod::RadiationSource::solar_radius

Mean solar radius.

trick_units(m)

Definition at line 71 of file radiation_source.hh.

Referenced by RadiationSource().

8.9.6.18 double jeod::RadiationSource::source_to_cg[3]

vehicle cg position relative to the source

trick_units(m)

Definition at line 129 of file radiation_source.hh.

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

8.9.6.19 double jeod::RadiationSource::source_to_struc_origin[3]

vehicle struc frame origin position relative to source.

trick_units(m)

Definition at line 134 of file radiation_source.hh.

Referenced by calculate_flux(), and RadiationSource().

8.9.6.20 RadiationThirdBody∗∗ jeod::RadiationSource::third_body

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

trick_units(–)

Definition at line 167 of file radiation_source.hh.

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

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

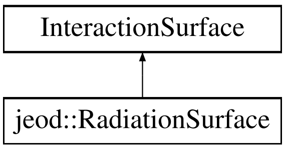
- radiation_source.hh
- radiation_source.cc

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.
- virtual ∼RadiationSurface ()

  Destructor for RadiationSurface.
- void initialize (double center_grav[3])

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

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

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

  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 (void)

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

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

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

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.

Private Member Functions

- RadiationSurface & operator= (const RadiationSurface &rhs)
- RadiationSurface (const RadiationSurface &rhs)

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 55 of file radiation_surface.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 jeod::RadiationSurface::RadiationSurface (void)

Constructor for RadiationSurface.

Definition at line 67 of file radiation_surface.cc.

8.10.2.2 jeod::RadiationSurface::∼RadiationSurface (void) [virtual]

Destructor for RadiationSurface.

Definition at line 376 of file radiation_surface.cc.

References facets, ii_facet, and num_facets.

8.10.2.3 jeod::RadiationSurface::RadiationSurface (const RadiationSurface & *rhs*) [private]

8.10.3 Member Function Documentation

8.10.3.1 void jeod::RadiationSurface::accumulate_thermal_sources (void)

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

Definition at line 278 of file radiation_surface.cc.

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

8.10.3.2 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 365 of file radiation_surface.cc.

8.10.3.3 void jeod::RadiationSurface::allocate_array (unsigned int *size*)

Allocates memory for an array of radiation facets.

Parameters

in	*size*	Size of array

Definition at line 142 of file radiation_surface.cc.

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

8.10.3.4 void jeod::RadiationSurface::allocate_interaction_facet (Facet ∗ *facet,* InteractionFacetFactory ∗ *factory,* FacetParams ∗ *params,* unsigned int *index*)

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 176 of file radiation_surface.cc.

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

8.10.3.5 void jeod::RadiationSurface::equalize_absorption_emission (void)

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

Definition at line 313 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 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 241 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 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 87 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 void jeod::RadiationSurface::initialize_runtime_values (void)

To initialize the values during each update run.

Definition at line 349 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 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 263 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 RadiationSurface& jeod::RadiationSurface::operator= (const RadiationSurface & *rhs*) [private]

8.10.3.11 void jeod::RadiationSurface::radiation_pressure (void)

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 328 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 void jeod::RadiationSurface::thermal_integrator (void)

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

Definition at line 293 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 void init_attrjeod__RadiationSurface ( ) [friend]

8.10.4.2 friend class InputProcessor [friend]

Definition at line 57 of file radiation_surface.hh.

8.10.5 Field Documentation

8.10.5.1 RadiationFacet** jeod::RadiationSurface::facets

Array of pointers to the Radiation Facets.

trick_units(–)

Definition at line 64 of file radiation_surface.hh.

Referenced by accumulate_thermal_sources(), 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 double jeod::RadiationSurface::force[3]

Force resulting from all radiative interactions.

trick_units(–)

Definition at line 88 of file radiation_surface.hh.

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

8.10.5.3 unsigned int jeod::RadiationSurface::ii_facet

Simple counter, used repeatedly.

trick_units(–)

Definition at line 98 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(), thermal_integrator(), and ~RadiationSurface().

8.10.5.4 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 75 of file radiation_surface.hh.

8.10.5.5 unsigned int jeod::RadiationSurface::num_facets

number of facets in this surface.

trick_units(–)

Definition at line 69 of file radiation_surface.hh.

Referenced by accumulate_thermal_sources(), 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 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 83 of file radiation_surface.hh.

8.10.5.7 double jeod::RadiationSurface::torque[3]

Torque resulting from all radiative interactions.

trick_units(–)

Definition at line 93 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:

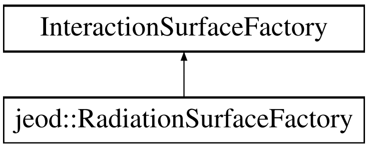
- radiation_surface.hh
- radiation_surface.cc

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.
- virtual ∼RadiationSurfaceFactory ()

  Destructor.

- virtual void add_facet_params (FacetParams ∗to_add)

  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.

Private Member Functions

- RadiationSurfaceFactory & operator= (const RadiationSurfaceFactory &rhs)
- RadiationSurfaceFactory (const RadiationSurfaceFactory &rhs)

Friends

- class InputProcessor
- void init_attrjeod__RadiationSurfaceFactory ()

8.11.1 Detailed Description

The factory for creating Radiation Surfaces.

Definition at line 50 of file radiation_surface_factory.hh.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 jeod::RadiationSurfaceFactory::RadiationSurfaceFactory (void)

Constructor.

Definition at line 56 of file radiation_surface_factory.cc.

References flat_plate_radiation_factory.

8.11.2.2 jeod::RadiationSurfaceFactory::∼RadiationSurfaceFactory (void) [virtual]

Destructor.

Definition at line 100 of file radiation_surface_factory.cc.

8.11.2.3 jeod::RadiationSurfaceFactory::RadiationSurfaceFactory (const RadiationSurfaceFactory & *rhs*) [private]

8.11.3 Member Function Documentation

8.11.3.1 void jeod::RadiationSurfaceFactory::add_facet_params (FacetParams ∗ *to_add*) [virtual]

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 70 of file radiation_surface_factory.cc.

References jeod::RadiationMessages::invalid_setup_error.

8.11.3.2 RadiationSurfaceFactory& jeod::RadiationSurfaceFactory::operator= (const RadiationSurfaceFactory & *rhs*) [private]

8.11.4 Friends And Related Function Documentation

8.11.4.1 void init_attrjeod__RadiationSurfaceFactory ( ) [friend]

8.11.4.2 friend class InputProcessor [friend]

Definition at line 52 of file radiation_surface_factory.hh.

8.11.5 Field Documentation

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

The factory to build flat plate thermal facets.

trick_units(–)

Definition at line 76 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 ()

  Constructor.
- virtual ∼RadiationThirdBody ()
- virtual void initialize (DynManager ∗dyn_mgr_ptr)

  initializes the third body
- void calculate_shadow (void)

  Calculates the effect of shadowing by a 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 time, RefFrame &vehicle_structural_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

  Flag indicating cylindrical / conical shadow geometry.
- bool force_state_update

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

  Name of planet.
- bool active

  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, 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.
- virtual bool update_third_body_state (void)

  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

  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.

Private Member Functions

- RadiationThirdBody (const RadiationThirdBody &)
- RadiationThirdBody & operator= (const RadiationThirdBody &)

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 49 of file radiation_third_body.hh.

8.12.2 Member Enumeration Documentation

8.12.2.1 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 57 of file radiation_third_body.hh.

8.12.3 Constructor & Destructor Documentation

8.12.3.1 jeod::RadiationThirdBody::RadiationThirdBody ( )

Constructor.

Definition at line 58 of file radiation_third_body.cc.

References output_flux, source_to_third_hat_inrtl, source_to_third_inrtl, and third_to_cg_inrtl.

8.12.3.2 virtual jeod::RadiationThirdBody::∼RadiationThirdBody ( ) [inline], [virtual]

Definition at line 203 of file radiation_third_body.hh.

8.12.3.3 jeod::RadiationThirdBody::RadiationThirdBody (const RadiationThirdBody &) [private]

8.12.4 Member Function Documentation

8.12.4.1 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 237 of file radiation_third_body.hh.

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

8.12.4.2 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 229 of file radiation_third_body.hh.

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

8.12.4.3 void jeod::RadiationThirdBody::calculate_shadow (void)

Calculates the effect of shadowing by a third body.

NOTE - This method is intended to be an internal call so should be protected However, doing so would be a change to the API. Delay for JEOD4.0.

Definition at line 206 of file radiation_third_body.cc.

References active, Con, Conical, Cyl, Cylindrical, jeod::RadiationSource::d_source_to_cg, d_source_to_third, generate_alpha(), illum_factor, initialized, jeod::RadiationMessages::invalid_setup_error, name, primary_source_ptr, r_minus, r_par, r_perp, r_plus, r_ratio, radius, jeod::RadiationSource::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 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 409 of file radiation_third_body.cc.

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

8.12.4.5 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	*relative*	disk size
in	*fraction*	of eclipse achieved.

Definition at line 376 of file radiation_third_body.cc.

Referenced by calculate_shadow().

8.12.4.6 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 252 of file radiation_third_body.hh.

References added_to_model.

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

8.12.4.7 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 93 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, radius, and jeod::RadiationSource::radius.

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

8.12.4.8 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 244 of file radiation_third_body.hh.

8.12.4.9 RadiationThirdBody& jeod::RadiationThirdBody::operator= (const RadiationThirdBody &) [private]

8.12.4.10 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 431 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 void jeod::RadiationThirdBody::set_added_to_model (bool *value*) [inline]

Sets the added_to_model value.

Parameters

value	New value

Definition at line 258 of file radiation_third_body.hh.

References added_to_model.

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

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

Setter for the name.

Definition at line 219 of file radiation_third_body.hh.

References name.

8.12.4.13 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 474 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 bool jeod::RadiationThirdBody::update_third_body_state (void) [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 498 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 void init_attrjeod__RadiationThirdBody ( ) [friend]

8.12.5.2 friend class InputProcessor [friend]

Definition at line 51 of file radiation_third_body.hh.

8.12.6 Field Documentation

8.12.6.1 bool jeod::RadiationThirdBody::active

Model on/off flag.

trick_units(–)

Definition at line 89 of file radiation_third_body.hh.

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

8.12.6.2 bool jeod::RadiationThirdBody::added_to_model [protected]

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

trick_units(–)

Definition at line 121 of file radiation_third_body.hh.

Referenced by get_added_to_model(), and set_added_to_model().

8.12.6.3 double jeod::RadiationThirdBody::d_source_to_third [protected]

Magnitude of source_to_third.

trick_units(m)

Definition at line 191 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and update_third_body_state().

8.12.6.4 bool jeod::RadiationThirdBody::force_state_update

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 79 of file radiation_third_body.hh.

Referenced by test_for_state_update().

8.12.6.5 double jeod::RadiationThirdBody::illum_factor [protected]

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

trick_units(–)

Definition at line 132 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and process_third_body().

8.12.6.6 bool jeod::RadiationThirdBody::initialized [protected]

Flag to indicate that initialization has been complete.

trick_units(–)

Definition at line 126 of file radiation_third_body.hh.

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

8.12.6.7 double jeod::RadiationThirdBody::last_update_time [protected]

Time at which the third body state was last updated.

trick_units(s)

Definition at line 137 of file radiation_third_body.hh.

Referenced by test_for_state_update().

8.12.6.8 RefFrame∗ jeod::RadiationThirdBody::local_frame_ptr

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

trick_units(–)

Definition at line 112 of file radiation_third_body.hh.

Referenced by initialize(), and update_third_body_state().

8.12.6.9 std::string jeod::RadiationThirdBody::name

Name of planet.

trick_units(–)

Definition at line 84 of file radiation_third_body.hh.

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

8.12.6.10 double jeod::RadiationThirdBody::output_flux[3]

Calculated value.

Set only in derived class RadiationReflectingThirdBody.trick_units(–)

Definition at line 106 of file radiation_third_body.hh.

Referenced by RadiationThirdBody().

8.12.6.11 Planet∗ jeod::RadiationThirdBody::planet_link [protected]

Pointer to the associated Planet object.

trick_units(–)

Definition at line 143 of file radiation_third_body.hh.

Referenced by initialize().

8.12.6.12 RadiationSource∗ jeod::RadiationThirdBody::primary_source_ptr

Pointer to the primary illumination source.

trick_units(–)

Definition at line 67 of file radiation_third_body.hh.

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

8.12.6.13 double jeod::RadiationThirdBody::r_minus [protected]

The value (radius - R_source)

trick_units(–)

Definition at line 153 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and initialize().

8.12.6.14 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 166 of file radiation_third_body.hh.

Referenced by calculate_shadow().

8.12.6.15 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 172 of file radiation_third_body.hh.

Referenced by calculate_shadow().

8.12.6.16 double jeod::RadiationThirdBody::r_plus [protected]

The value (radius + R_source)

trick_units(–)

Definition at line 148 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and initialize().

8.12.6.17 double jeod::RadiationThirdBody::r_ratio [protected]

The ratio (radius / R_source)

trick_units(–)

Definition at line 158 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and initialize().

8.12.6.18 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 100 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and initialize().

8.12.6.19 ShadowGeometry jeod::RadiationThirdBody::shadow_geometry

Flag indicating cylindrical / conical shadow geometry.

trick_units(–)

Definition at line 72 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and convert_shadow_from_int().

8.12.6.20 double jeod::RadiationThirdBody::source_to_third_hat_inrtl[3] [protected]

Unit vector of source_to_third.

trick_units(–)

Definition at line 196 of file radiation_third_body.hh.

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

8.12.6.21 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 186 of file radiation_third_body.hh.

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

8.12.6.22 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 178 of file radiation_third_body.hh.

Referenced by calculate_shadow(), and RadiationThirdBody().

8.12.6.23 double jeod::RadiationThirdBody::update_interval

Time interval between third body state updates.

trick_units(s)

Definition at line 94 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.

Definition in file flat_plate_radiation_facet.cc.

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.

Definition in file flat_plate_radiation_facet.hh.

9.3 flat_plate_radiation_factory.cc File Reference

Factory that creates a FlatPlateRadiationFacet, from a facet model.

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

Namespaces

- jeod

  Namespace jeod.

9.3.1 Detailed Description

Factory that creates a FlatPlateRadiationFacet, from a facet model.

Definition in file flat_plate_radiation_factory.cc.

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.

Definition in file flat_plate_radiation_factory.hh.

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.

Definition in file radiation_base_facet.cc.

9.6 radiation_base_facet.hh File Reference

Individual facets for use with radiation environment interaction models.

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

Definition in file radiation_base_facet.hh.

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.

Definition in file radiation_default_surface.cc.

9.8 radiation_default_surface.hh File Reference

Individual facets for use with radiation environment interaction models.

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

Definition in file radiation_default_surface.hh.

9.9 radiation_facet.cc File Reference

Individual facets for use with Radiation Pressure interaction models.

```
#include "utils/math/include/vector3.hh"
#include "utils/surface_model/include/facet.hh"
#include "interactions/thermal_rider/include/thermal_facet_rider.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/radiation_facet.hh"
#include "../include/radiation_params.hh"
#include "../include/radiation_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.9.1 Detailed Description

Individual facets for use with Radiation Pressure interaction models.

Definition in file radiation_facet.cc.

9.10 radiation_facet.hh File Reference

Individual facets for use with radiation environment interaction models.

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

Definition in file radiation_facet.hh.

9.11 radiation_messages.cc File Reference

Implement the class RadiationMessages.

```
#include "../include/radiation_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define PATH "interactions/radiation_pressure/"

9.11.1 Detailed Description

Implement the class RadiationMessages.

Definition in 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.

Definition in file radiation_messages.hh.

9.13 radiation_params.cc File Reference

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

```
#include "../include/radiation_params.hh"
```

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 InteractionSurfaceFactorys.

Definition in file radiation_params.cc.

9.14 radiation_params.hh File Reference

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

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

9.14.1 Detailed Description

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

Definition in file radiation_params.hh.

9.15 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 "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "interactions/thermal_rider/include/thermal_facet_rider.hh"
#include "../include/radiation_pressure.hh"
#include "../include/radiation_third_body.hh"
#include "../include/radiation_source.hh"
#include "../include/radiation_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.15.1 Detailed Description

Calculation of force and torque due to radiation pressure.

Definition in file radiation_pressure.cc.

9.16 radiation_pressure.hh File Reference

Radiation pressure top-level definition.

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

Data Structures

- class jeod::RadiationPressure

  Radiation pressure top-level definition.

Namespaces

- jeod

  Namespace jeod.

9.16.1 Detailed Description

Radiation pressure top-level definition.

Definition in file radiation_pressure.hh.

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

Namespaces

- jeod

    Namespace jeod.

9.17.1 Detailed Description

Calculation of force and torque due to radiation pressure.

Definition in file radiation_pressure__default_surface.cc.

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

Namespaces

- jeod

    Namespace jeod.

9.18.1 Detailed Description

Calculation of force and torque due to radiation pressure.

Definition in file radiation_pressure__surface_model.cc.

9.19 radiation_source.cc File Reference

Definition of methods associated with Primary Sources.

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

Namespaces

- jeod

  Namespace jeod.

9.19.1 Detailed Description

Definition of methods associated with Primary Sources.

Definition in file radiation_source.cc.

9.20 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.

9.20.1 Detailed Description

Radiation pressure parameter and variable definitions.

Definition in file radiation_source.hh.

9.21 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/interaction_facet_factory.hh"
#include "utils/surface_model/include/interaction_facet.hh"
#include "utils/surface_model/include/facet.hh"
#include "../include/radiation_facet.hh"
#include "../include/radiation_surface.hh"
#include "../include/radiation_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.21.1 Detailed Description

Vehicle surface model for Radiation Pressure model.

Definition in file radiation_surface.cc.

9.22 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.

9.22.1 Detailed Description

Vehicle surface model for general environment interaction models.

Definition in file radiation_surface.hh.

9.23 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_surface_factory.hh"
#include "../include/radiation_params.hh"
#include "../include/radiation_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.23.1 Detailed Description

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

Definition in file radiation_surface_factory.cc.

9.24 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.

9.24.1 Detailed Description

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

Definition in file radiation_surface_factory.hh.

9.25 radiation_third_body.cc File Reference

Provides the functionality associated with RadiationThirdBodys.

```
#include <cstddef>
#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_third_body.hh"
#include "../include/radiation_source.hh"
#include "../include/radiation_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

9.25.1 Detailed Description

Provides the functionality associated with RadiationThirdBodys.

Definition in file radiation_third_body.cc.

9.26 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.

9.26.1 Detailed Description

Define the class RadiationThirdBody.

Definition in file radiation_third_body.hh.

Index