

## RadiationPressureModel

5.1

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>11</b>
6.1	Models . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.2	Interactions . . . . .	12
6.2.1	Detailed Description . . . . .	12
6.3	RadiationPressure . . . . .	13
6.3.1	Detailed Description . . . . .	14
6.3.2	Macro Definition Documentation . . . . .	14
6.3.2.1	PATH . . . . .	14

<b>7</b>	<b>Namespace Documentation</b>	<b>15</b>
7.1	jeod Namespace Reference . . . . .	15
7.1.1	Detailed Description . . . . .	15
<b>8</b>	<b>Data Structure Documentation</b>	<b>17</b>
8.1	jeod::FlatPlateRadiationFacet Class Reference . . . . .	17
8.1.1	Detailed Description . . . . .	18
8.1.2	Constructor & Destructor Documentation . . . . .	18
8.1.2.1	FlatPlateRadiationFacet() [1/2] . . . . .	18
8.1.2.2	~FlatPlateRadiationFacet() . . . . .	18
8.1.2.3	FlatPlateRadiationFacet() [2/2] . . . . .	18
8.1.3	Member Function Documentation . . . . .	18
8.1.3.1	define_facet() . . . . .	18
8.1.3.2	incident_radiation() . . . . .	19
8.1.3.3	initialize_geom() . . . . .	19
8.1.3.4	operator=() . . . . .	20
8.1.3.5	radiation_pressure() . . . . .	20
8.1.4	Friends And Related Function Documentation . . . . .	20
8.1.4.1	init_attrjeod__FlatPlateRadiationFacet . . . . .	20
8.1.4.2	InputProcessor . . . . .	21
8.1.5	Field Documentation . . . . .	21
8.1.5.1	incident_flux_hat . . . . .	21
8.1.5.2	normal . . . . .	21
8.1.5.3	sin_theta . . . . .	21
8.2	jeod::FlatPlateRadiationFactory Class Reference . . . . .	22
8.2.1	Detailed Description . . . . .	22
8.2.2	Constructor & Destructor Documentation . . . . .	22
8.2.2.1	FlatPlateRadiationFactory() [1/2] . . . . .	22
8.2.2.2	~FlatPlateRadiationFactory() . . . . .	23
8.2.2.3	FlatPlateRadiationFactory() [2/2] . . . . .	23
8.2.3	Member Function Documentation . . . . .	23

8.2.3.1	<a href="#">create_facet()</a>	23
8.2.3.2	<a href="#">is_correct_factory()</a>	23
8.2.3.3	<a href="#">operator=()</a>	24
8.2.4	<a href="#">Friends And Related Function Documentation</a>	24
8.2.4.1	<a href="#">init_attrjeod__FlatPlateRadiationFactory</a>	24
8.2.4.2	<a href="#">InputProcessor</a>	24
8.3	<a href="#">jeod::RadiationBaseFacet Class Reference</a>	24
8.3.1	<a href="#">Detailed Description</a>	26
8.3.2	<a href="#">Constructor &amp; Destructor Documentation</a>	26
8.3.2.1	<a href="#">RadiationBaseFacet() [1/2]</a>	26
8.3.2.2	<a href="#">~RadiationBaseFacet()</a>	26
8.3.2.3	<a href="#">RadiationBaseFacet() [2/2]</a>	26
8.3.3	<a href="#">Member Function Documentation</a>	26
8.3.3.1	<a href="#">incident_radiation()</a>	26
8.3.3.2	<a href="#">initialize()</a>	27
8.3.3.3	<a href="#">initialize_runtime_values()</a>	27
8.3.3.4	<a href="#">interact_with_third_body()</a>	27
8.3.3.5	<a href="#">operator=()</a>	28
8.3.4	<a href="#">Friends And Related Function Documentation</a>	28
8.3.4.1	<a href="#">init_attrjeod__RadiationBaseFacet</a>	28
8.3.4.2	<a href="#">InputProcessor</a>	28
8.3.5	<a href="#">Field Documentation</a>	28
8.3.5.1	<a href="#">albedo</a>	28
8.3.5.2	<a href="#">albedo_IR</a>	29
8.3.5.3	<a href="#">albedo_vis</a>	29
8.3.5.4	<a href="#">areaxflux</a>	29
8.3.5.5	<a href="#">areaxflux_e</a>	29
8.3.5.6	<a href="#">cx_area</a>	30
8.3.5.7	<a href="#">diffuse</a>	30
8.3.5.8	<a href="#">F_absorption</a>	30

8.3.5.9	<a href="#">F_diffuse</a>	31
8.3.5.10	<a href="#">F_emission</a>	31
8.3.5.11	<a href="#">F_specular</a>	31
8.3.5.12	<a href="#">speed_of_light</a>	31
8.3.5.13	<a href="#">thermal</a>	32
8.4	<a href="#">jeod::RadiationDefaultSurface Class Reference</a>	32
8.4.1	<a href="#">Detailed Description</a>	33
8.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	33
8.4.2.1	<a href="#">RadiationDefaultSurface() [1/2]</a>	33
8.4.2.2	<a href="#">~RadiationDefaultSurface()</a>	33
8.4.2.3	<a href="#">RadiationDefaultSurface() [2/2]</a>	33
8.4.3	<a href="#">Member Function Documentation</a>	33
8.4.3.1	<a href="#">add_thermal_integrator_to()</a>	33
8.4.3.2	<a href="#">incident_radiation()</a>	34
8.4.3.3	<a href="#">initialize()</a>	34
8.4.3.4	<a href="#">operator=()</a>	35
8.4.3.5	<a href="#">set_name()</a>	35
8.4.3.6	<a href="#">thermal_update()</a>	35
8.4.4	<a href="#">Friends And Related Function Documentation</a>	35
8.4.4.1	<a href="#">init_attrjeod__RadiationDefaultSurface</a>	35
8.4.4.2	<a href="#">InputProcessor</a>	36
8.4.5	<a href="#">Field Documentation</a>	36
8.4.5.1	<a href="#">name</a>	36
8.4.5.2	<a href="#">rad_coeff</a>	36
8.4.5.3	<a href="#">surface_area</a>	36
8.4.5.4	<a href="#">temperature</a>	37
8.5	<a href="#">jeod::RadiationFacet Class Reference</a>	37
8.5.1	<a href="#">Detailed Description</a>	38
8.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	38
8.5.2.1	<a href="#">RadiationFacet() [1/2]</a>	38

8.5.2.2	<a href="#">~RadiationFacet()</a> . . . . .	38
8.5.2.3	<a href="#">RadiationFacet()</a> [2/2] . . . . .	38
8.5.3	<a href="#">Member Function Documentation</a> . . . . .	38
8.5.3.1	<a href="#">define_facet_core()</a> . . . . .	38
8.5.3.2	<a href="#">get_thermal_integrator()</a> . . . . .	39
8.5.3.3	<a href="#">initialize()</a> . . . . .	39
8.5.3.4	<a href="#">initialize_geom()</a> . . . . .	39
8.5.3.5	<a href="#">operator=()</a> . . . . .	40
8.5.3.6	<a href="#">radiation_pressure()</a> . . . . .	40
8.5.4	<a href="#">Friends And Related Function Documentation</a> . . . . .	40
8.5.4.1	<a href="#">init_attrjeod__RadiationFacet</a> . . . . .	40
8.5.4.2	<a href="#">InputProcessor</a> . . . . .	40
8.5.5	<a href="#">Field Documentation</a> . . . . .	40
8.5.5.1	<a href="#">center_pressure</a> . . . . .	41
8.5.5.2	<a href="#">crot_to_cp</a> . . . . .	41
8.5.5.3	<a href="#">two_thirds</a> . . . . .	41
8.6	<a href="#">jeod::RadiationMessages Class Reference</a> . . . . .	42
8.6.1	<a href="#">Detailed Description</a> . . . . .	42
8.6.2	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	42
8.6.2.1	<a href="#">RadiationMessages()</a> [1/2] . . . . .	42
8.6.2.2	<a href="#">RadiationMessages()</a> [2/2] . . . . .	43
8.6.3	<a href="#">Member Function Documentation</a> . . . . .	43
8.6.3.1	<a href="#">operator=()</a> . . . . .	43
8.6.4	<a href="#">Friends And Related Function Documentation</a> . . . . .	43
8.6.4.1	<a href="#">init_attrjeod__RadiationMessages</a> . . . . .	43
8.6.4.2	<a href="#">InputProcessor</a> . . . . .	43
8.6.5	<a href="#">Field Documentation</a> . . . . .	43
8.6.5.1	<a href="#">incomplete_setup_error</a> . . . . .	43
8.6.5.2	<a href="#">invalid_function_call</a> . . . . .	44
8.6.5.3	<a href="#">invalid_setup_error</a> . . . . .	44

8.6.5.4	<a href="#">operational_setup_error</a>	44
8.6.5.5	<a href="#">unknown_numerical_error</a>	45
8.7	<a href="#">jeod::RadiationParams Class Reference</a>	45
8.7.1	<a href="#">Detailed Description</a>	46
8.7.2	<a href="#">Constructor &amp; Destructor Documentation</a>	46
8.7.2.1	<a href="#">RadiationParams() [1/2]</a>	46
8.7.2.2	<a href="#">~RadiationParams()</a>	46
8.7.2.3	<a href="#">RadiationParams() [2/2]</a>	46
8.7.3	<a href="#">Member Function Documentation</a>	46
8.7.3.1	<a href="#">operator=()</a>	46
8.7.4	<a href="#">Friends And Related Function Documentation</a>	46
8.7.4.1	<a href="#">init_attrjeod__RadiationParams</a>	47
8.7.4.2	<a href="#">InputProcessor</a>	47
8.7.5	<a href="#">Field Documentation</a>	47
8.7.5.1	<a href="#">albedo</a>	47
8.7.5.2	<a href="#">diffuse</a>	47
8.7.5.3	<a href="#">thermal</a>	48
8.8	<a href="#">jeod::RadiationPressure Class Reference</a>	48
8.8.1	<a href="#">Detailed Description</a>	50
8.8.2	<a href="#">Constructor &amp; Destructor Documentation</a>	50
8.8.2.1	<a href="#">RadiationPressure() [1/2]</a>	50
8.8.2.2	<a href="#">~RadiationPressure()</a>	50
8.8.2.3	<a href="#">RadiationPressure() [2/2]</a>	50
8.8.3	<a href="#">Member Function Documentation</a>	50
8.8.3.1	<a href="#">add_third_body()</a>	50
8.8.3.2	<a href="#">find_third_body()</a>	51
8.8.3.3	<a href="#">initialize() [1/2]</a>	51
8.8.3.4	<a href="#">initialize() [2/2]</a>	52
8.8.3.5	<a href="#">initialize_environment()</a>	52
8.8.3.6	<a href="#">operator=()</a>	53



8.8.3.7	<a href="#">set_calculate_forces()</a>	53
8.8.3.8	<a href="#">set_third_body_active()</a>	53
8.8.3.9	<a href="#">set_third_body_inactive()</a>	54
8.8.3.10	<a href="#">third_body_adjustments()</a>	54
8.8.3.11	<a href="#">update()</a>	54
8.8.3.12	<a href="#">update_default_surface()</a>	55
8.8.3.13	<a href="#">update_facet_surface()</a>	55
8.8.4	<a href="#">Friends And Related Function Documentation</a>	55
8.8.4.1	<a href="#">init_attrjeod__RadiationPressure</a>	56
8.8.4.2	<a href="#">InputProcessor</a>	56
8.8.5	<a href="#">Field Documentation</a>	56
8.8.5.1	<a href="#">active</a>	56
8.8.5.2	<a href="#">calculate_forces</a>	56
8.8.5.3	<a href="#">default_surface_ptr</a>	57
8.8.5.4	<a href="#">dyn_manager_ptr</a>	57
8.8.5.5	<a href="#">force</a>	57
8.8.5.6	<a href="#">illum_factor</a>	57
8.8.5.7	<a href="#">initialized</a>	58
8.8.5.8	<a href="#">num_third_bodies</a>	58
8.8.5.9	<a href="#">source</a>	58
8.8.5.10	<a href="#">surface_ptr</a>	58
8.8.5.11	<a href="#">thermal</a>	59
8.8.5.12	<a href="#">third_bodies</a>	59
8.8.5.13	<a href="#">third_bodies_active</a>	59
8.8.5.14	<a href="#">torque</a>	59
8.9	<a href="#">jeod::RadiationSource Class Reference</a>	60
8.9.1	<a href="#">Detailed Description</a>	61
8.9.2	<a href="#">Member Enumeration Documentation</a>	61
8.9.2.1	<a href="#">OldShadowGeometry</a>	61
8.9.3	<a href="#">Constructor &amp; Destructor Documentation</a>	62

8.9.3.1	<a href="#">RadiationSource()</a> [1/2]	62
8.9.3.2	<a href="#">~RadiationSource()</a>	62
8.9.3.3	<a href="#">RadiationSource()</a> [2/2]	62
8.9.4	<a href="#">Member Function Documentation</a>	62
8.9.4.1	<a href="#">calculate_flux()</a>	62
8.9.4.2	<a href="#">initialize()</a>	63
8.9.4.3	<a href="#">operator=()</a>	63
8.9.4.4	<a href="#">set_name()</a>	63
8.9.5	<a href="#">Friends And Related Function Documentation</a>	63
8.9.5.1	<a href="#">init_attrjeod__RadiationSource</a>	63
8.9.5.2	<a href="#">InputProcessor</a>	64
8.9.6	<a href="#">Field Documentation</a>	64
8.9.6.1	<a href="#">bodies_active</a>	64
8.9.6.2	<a href="#">d_source_to_cg</a>	64
8.9.6.3	<a href="#">flux_hat</a>	64
8.9.6.4	<a href="#">flux_inertial</a>	65
8.9.6.5	<a href="#">flux_mag</a>	65
8.9.6.6	<a href="#">flux_struct</a>	65
8.9.6.7	<a href="#">flux_struct_hat</a>	65
8.9.6.8	<a href="#">inertial_cg</a>	66
8.9.6.9	<a href="#">inertial_frame_ptr</a>	66
8.9.6.10	<a href="#">luminosity</a>	66
8.9.6.11	<a href="#">multiple_shadow_bodies</a>	66
8.9.6.12	<a href="#">name</a>	67
8.9.6.13	<a href="#">num_bodies</a>	67
8.9.6.14	<a href="#">radius</a>	67
8.9.6.15	<a href="#">shadow_geometry</a>	67
8.9.6.16	<a href="#">solar_luminosity</a>	68
8.9.6.17	<a href="#">solar_radius</a>	68
8.9.6.18	<a href="#">source_to_cg</a>	68

8.9.6.19	<a href="#">source_to_struc_origin</a>	68
8.9.6.20	<a href="#">third_body</a>	69
8.10	<a href="#">jeod::RadiationSurface Class Reference</a>	69
8.10.1	<a href="#">Detailed Description</a>	70
8.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	70
8.10.2.1	<a href="#">RadiationSurface() [1/2]</a>	71
8.10.2.2	<a href="#">~RadiationSurface()</a>	71
8.10.2.3	<a href="#">RadiationSurface() [2/2]</a>	71
8.10.3	<a href="#">Member Function Documentation</a>	71
8.10.3.1	<a href="#">accumulate_thermal_sources()</a>	71
8.10.3.2	<a href="#">add_thermal_integrators_to()</a>	71
8.10.3.3	<a href="#">allocate_array()</a>	72
8.10.3.4	<a href="#">allocate_interaction_facet()</a>	72
8.10.3.5	<a href="#">equalize_absorption_emission()</a>	73
8.10.3.6	<a href="#">incident_radiation()</a>	73
8.10.3.7	<a href="#">initialize()</a>	73
8.10.3.8	<a href="#">initialize_runtime_values()</a>	74
8.10.3.9	<a href="#">interact_with_third_body()</a>	74
8.10.3.10	<a href="#">operator=()</a>	74
8.10.3.11	<a href="#">radiation_pressure()</a>	75
8.10.3.12	<a href="#">thermal_integrator()</a>	75
8.10.4	<a href="#">Friends And Related Function Documentation</a>	75
8.10.4.1	<a href="#">init_attrjeod__RadiationSurface</a>	75
8.10.4.2	<a href="#">InputProcessor</a>	75
8.10.5	<a href="#">Field Documentation</a>	75
8.10.5.1	<a href="#">facets</a>	76
8.10.5.2	<a href="#">force</a>	76
8.10.5.3	<a href="#">ii_facet</a>	76
8.10.5.4	<a href="#">include_conduction</a>	76
8.10.5.5	<a href="#">num_facets</a>	77

8.10.5.6	<a href="#">thermal_conduction</a>	77
8.10.5.7	<a href="#">torque</a>	77
8.11	<a href="#">jeod::RadiationSurfaceFactory Class Reference</a>	78
8.11.1	<a href="#">Detailed Description</a>	78
8.11.2	<a href="#">Constructor &amp; Destructor Documentation</a>	78
8.11.2.1	<a href="#">RadiationSurfaceFactory() [1/2]</a>	79
8.11.2.2	<a href="#">~RadiationSurfaceFactory()</a>	79
8.11.2.3	<a href="#">RadiationSurfaceFactory() [2/2]</a>	79
8.11.3	<a href="#">Member Function Documentation</a>	79
8.11.3.1	<a href="#">add_facet_params()</a>	79
8.11.3.2	<a href="#">operator=()</a>	80
8.11.4	<a href="#">Friends And Related Function Documentation</a>	80
8.11.4.1	<a href="#">init_attrjeod__RadiationSurfaceFactory</a>	80
8.11.4.2	<a href="#">InputProcessor</a>	80
8.11.5	<a href="#">Field Documentation</a>	80
8.11.5.1	<a href="#">flat_plate_radiation_factory</a>	80
8.12	<a href="#">jeod::RadiationThirdBody Class Reference</a>	80
8.12.1	<a href="#">Detailed Description</a>	83
8.12.2	<a href="#">Member Enumeration Documentation</a>	83
8.12.2.1	<a href="#">ShadowGeometry</a>	83
8.12.3	<a href="#">Constructor &amp; Destructor Documentation</a>	83
8.12.3.1	<a href="#">RadiationThirdBody() [1/2]</a>	83
8.12.3.2	<a href="#">~RadiationThirdBody()</a>	83
8.12.3.3	<a href="#">RadiationThirdBody() [2/2]</a>	83
8.12.4	<a href="#">Member Function Documentation</a>	84
8.12.4.1	<a href="#">accumulate_rad_flux()</a>	84
8.12.4.2	<a href="#">accumulate_refl_flux()</a>	84
8.12.4.3	<a href="#">calculate_shadow()</a>	84
8.12.4.4	<a href="#">convert_shadow_from_int()</a>	85
8.12.4.5	<a href="#">generate_alpha()</a>	85

8.12.4.6	<a href="#">get_added_to_model()</a>	86
8.12.4.7	<a href="#">initialize()</a>	86
8.12.4.8	<a href="#">is_interactive()</a>	86
8.12.4.9	<a href="#">operator=()</a>	87
8.12.4.10	<a href="#">process_third_body()</a>	87
8.12.4.11	<a href="#">set_added_to_model()</a>	87
8.12.4.12	<a href="#">set_name()</a>	88
8.12.4.13	<a href="#">test_for_state_update()</a>	88
8.12.4.14	<a href="#">update_third_body_state()</a>	88
8.12.5	<a href="#">Friends And Related Function Documentation</a>	88
8.12.5.1	<a href="#">init_attrjeod__RadiationThirdBody</a>	89
8.12.5.2	<a href="#">InputProcessor</a>	89
8.12.6	<a href="#">Field Documentation</a>	89
8.12.6.1	<a href="#">active</a>	89
8.12.6.2	<a href="#">added_to_model</a>	89
8.12.6.3	<a href="#">d_source_to_third</a>	90
8.12.6.4	<a href="#">force_state_update</a>	90
8.12.6.5	<a href="#">illum_factor</a>	90
8.12.6.6	<a href="#">initialized</a>	90
8.12.6.7	<a href="#">last_update_time</a>	91
8.12.6.8	<a href="#">local_frame_ptr</a>	91
8.12.6.9	<a href="#">name</a>	91
8.12.6.10	<a href="#">output_flux</a>	91
8.12.6.11	<a href="#">planet_link</a>	92
8.12.6.12	<a href="#">primary_source_ptr</a>	92
8.12.6.13	<a href="#">r_minus</a>	92
8.12.6.14	<a href="#">r_par</a>	92
8.12.6.15	<a href="#">r_perp</a>	93
8.12.6.16	<a href="#">r_plus</a>	93
8.12.6.17	<a href="#">r_ratio</a>	93
8.12.6.18	<a href="#">radius</a>	93
8.12.6.19	<a href="#">shadow_geometry</a>	94
8.12.6.20	<a href="#">source_to_third_hat_inrtl</a>	94
8.12.6.21	<a href="#">source_to_third_inrtl</a>	94
8.12.6.22	<a href="#">third_to_cg_inrtl</a>	94
8.12.6.23	<a href="#">update_interval</a>	94

<b>9 File Documentation</b>	<b>95</b>
9.1 flat_plate_radiation_facet.cc File Reference	95
9.1.1 Detailed Description	95
9.2 flat_plate_radiation_facet.hh File Reference	95
9.2.1 Detailed Description	96
9.3 flat_plate_radiation_factory.cc File Reference	96
9.3.1 Detailed Description	96
9.4 flat_plate_radiation_factory.hh File Reference	96
9.4.1 Detailed Description	97
9.5 radiation_base_facet.cc File Reference	97
9.5.1 Detailed Description	97
9.6 radiation_base_facet.hh File Reference	97
9.6.1 Detailed Description	98
9.7 radiation_default_surface.cc File Reference	98
9.7.1 Detailed Description	98
9.8 radiation_default_surface.hh File Reference	98
9.8.1 Detailed Description	99
9.9 radiation_facet.cc File Reference	99
9.9.1 Detailed Description	99
9.10 radiation_facet.hh File Reference	99
9.10.1 Detailed Description	100
9.11 radiation_messages.cc File Reference	100
9.11.1 Detailed Description	100
9.12 radiation_messages.hh File Reference	100
9.12.1 Detailed Description	101
9.13 radiation_params.hh File Reference	101
9.13.1 Detailed Description	101
9.14 radiation_pressure.cc File Reference	101
9.14.1 Detailed Description	102
9.15 radiation_pressure.hh File Reference	102

9.15.1 Detailed Description . . . . .	102
9.16 radiation_pressure__default_surface.cc File Reference . . . . .	102
9.16.1 Detailed Description . . . . .	103
9.17 radiation_pressure__surface_model.cc File Reference . . . . .	103
9.17.1 Detailed Description . . . . .	103
9.18 radiation_source.cc File Reference . . . . .	103
9.18.1 Detailed Description . . . . .	104
9.19 radiation_source.hh File Reference . . . . .	104
9.19.1 Detailed Description . . . . .	104
9.20 radiation_surface.cc File Reference . . . . .	104
9.20.1 Detailed Description . . . . .	105
9.21 radiation_surface.hh File Reference . . . . .	105
9.21.1 Detailed Description . . . . .	105
9.22 radiation_surface_factory.cc File Reference . . . . .	105
9.22.1 Detailed Description . . . . .	106
9.23 radiation_surface_factory.hh File Reference . . . . .	106
9.23.1 Detailed Description . . . . .	106
9.24 radiation_third_body.cc File Reference . . . . .	106
9.24.1 Detailed Description . . . . .	107
9.25 radiation_third_body.hh File Reference . . . . .	107
9.25.1 Detailed Description . . . . .	107
<b>Index</b>	<b>109</b>





# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Models . . . . .	11
Interactions . . . . .	12
RadiationPressure . . . . .	13



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

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



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

FacetParams	
jeod::RadiationParams . . . . .	45
InteractionFacet	
jeod::RadiationFacet . . . . .	37
jeod::FlatPlateRadiationFacet . . . . .	17
InteractionFacetFactory	
jeod::FlatPlateRadiationFactory . . . . .	22
InteractionSurface	
jeod::RadiationSurface . . . . .	69
InteractionSurfaceFactory	
jeod::RadiationSurfaceFactory . . . . .	78
jeod::RadiationBaseFacet . . . . .	24
jeod::RadiationDefaultSurface . . . . .	32
jeod::RadiationFacet . . . . .	37
jeod::RadiationMessages . . . . .	42
jeod::RadiationPressure . . . . .	48
jeod::RadiationSource . . . . .	60
jeod::RadiationThirdBody . . . . .	80



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::FlatPlateRadiationFacet</a>	
A flat plate facet to be used for radiation interaction . . . . .	17
<a href="#">jeod::FlatPlateRadiationFactory</a>	
The factory for building flat plate radiation facets . . . . .	22
<a href="#">jeod::RadiationBaseFacet</a>	
Generic type of facet for radiation pressure . . . . .	24
<a href="#">jeod::RadiationDefaultSurface</a>	
The default spherical surface for radiation pressure . . . . .	32
<a href="#">jeod::RadiationFacet</a>	
Generic type of facet for radiation pressure . . . . .	37
<a href="#">jeod::RadiationMessages</a>	
Provides error messages . . . . .	42
<a href="#">jeod::RadiationParams</a>	
Provides a parameter list to each facet, based on the facet material properties . . . . .	45
<a href="#">jeod::RadiationPressure</a>	
Radiation pressure top-level definition . . . . .	48
<a href="#">jeod::RadiationSource</a>	
Provides information on the source of the incident radiation . . . . .	60
<a href="#">jeod::RadiationSurface</a>	
The surface of the vehicle that interacts with the incident flux . . . . .	69
<a href="#">jeod::RadiationSurfaceFactory</a>	
The factory for creating Radiation Surfaces . . . . .	78
<a href="#">jeod::RadiationThirdBody</a>	
Provide information on bodies that may cause shadowing or reflected illumination . . . . .	80





## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">flat_plate_radiation_facet.cc</a>	Define member functions for class FlatPlateRadiationFacet . . . . .	95
<a href="#">flat_plate_radiation_facet.hh</a>	Individual facets for use with rad environment interaction models . . . . .	95
<a href="#">flat_plate_radiation_factory.cc</a>	Factory that creates a FlatPlateRadiationFacet, from a facet model . . . . .	96
<a href="#">flat_plate_radiation_factory.hh</a>	Factory that creates an interaction facet, for a specific environment interaction model, from a facet model . . . . .	96
<a href="#">radiation_base_facet.cc</a>	Define member functions for class RadiationBaseFacet . . . . .	97
<a href="#">radiation_base_facet.hh</a>	Individual facets for use with radiation environment interaction models . . . . .	97
<a href="#">radiation_default_surface.cc</a>	Default surface for use with Radiation Pressure interaction model . . . . .	98
<a href="#">radiation_default_surface.hh</a>	Individual facets for use with radiation environment interaction models . . . . .	98
<a href="#">radiation_facet.cc</a>	Individual facets for use with Radiation Pressure interaction models . . . . .	99
<a href="#">radiation_facet.hh</a>	Individual facets for use with radiation environment interaction models . . . . .	99
<a href="#">radiation_messages.cc</a>	Implement the class RadiationMessages . . . . .	100
<a href="#">radiation_messages.hh</a>	Define the class RadiationMessages, the class that specifies the message IDs used in the Radiation model . . . . .	100
<a href="#">radiation_params.hh</a>	A virtual base class for radiation facet parameters, used to create interaction facets for radiation in the InteractionSurfaceFactorys . . . . .	101
<a href="#">radiation_pressure.cc</a>	Calculation of force and torque due to radiation pressure . . . . .	101
<a href="#">radiation_pressure.hh</a>	Radiation pressure top-level definition . . . . .	102
<a href="#">radiation_pressure__default_surface.cc</a>	Calculation of force and torque due to radiation pressure . . . . .	102

<a href="#">radiation_pressure__surface_model.cc</a>	
Calculation of force and torque due to radiation pressure . . . . .	103
<a href="#">radiation_source.cc</a>	
Definition of methods associated with Primary Sources . . . . .	103
<a href="#">radiation_source.hh</a>	
Radiation pressure parameter and variable definitions . . . . .	104
<a href="#">radiation_surface.cc</a>	
Vehicle surface model for Radiation Pressure model . . . . .	104
<a href="#">radiation_surface.hh</a>	
Vehicle surface model for general environment interaction models . . . . .	105
<a href="#">radiation_surface_factory.cc</a>	
Factory that creates an interaction surface, for a specific environment interaction model, from a surface model . . . . .	105
<a href="#">radiation_surface_factory.hh</a>	
Factory that creates an interaction surface, for a specific environment interaction model, from a surface model . . . . .	106
<a href="#">radiation_third_body.cc</a>	
Provides the functionality associated with RadiationThirdBodys . . . . .	106
<a href="#">radiation_third_body.hh</a>	
Define the class RadiationThirdBody . . . . .	107

## Chapter 6

# Module Documentation

### 6.1 Models

#### Modules

- [Interactions](#)

#### 6.1.1 Detailed Description

## 6.2 Interactions

### Modules

- [RadiationPressure](#)

### 6.2.1 Detailed Description

## 6.3 RadiationPressure

### Files

- file [flat\\_plate\\_radiation\\_facet.hh](#)  
*Individual facets for use with rad environment interaction models.*
- file [flat\\_plate\\_radiation\\_factory.hh](#)  
*Factory that creates an interaction facet, for a specific environment interaction model, from a facet model.*
- file [radiation\\_base\\_facet.hh](#)  
*Individual facets for use with radiation environment interaction models.*
- file [radiation\\_default\\_surface.hh](#)  
*Individual facets for use with radiation environment interaction models.*
- file [radiation\\_facet.hh](#)  
*Individual facets for use with radiation environment interaction models.*
- file [radiation\\_messages.hh](#)  
*Define the class RadiationMessages, the class that specifies the message IDs used in the Radiation model.*
- file [radiation\\_params.hh](#)  
*A virtual base class for radiation facet parameters, used to create interaction facets for radiation in the Interaction↔SurfaceFactorys.*
- file [radiation\\_pressure.hh](#)  
*Radiation pressure top-level definition.*
- file [radiation\\_source.hh](#)  
*Radiation pressure parameter and variable definitions.*
- file [radiation\\_surface.hh](#)  
*Vehicle surface model for general environment interaction models.*
- file [radiation\\_surface\\_factory.hh](#)  
*Factory that creates an interaction surface, for a specific environment interaction model, from a surface model.*
- file [radiation\\_third\\_body.hh](#)  
*Define the class RadiationThirdBody.*
- file [flat\\_plate\\_radiation\\_facet.cc](#)  
*Define member functions for class FlatPlateRadiationFacet.*
- file [flat\\_plate\\_radiation\\_factory.cc](#)  
*Factory that creates a FlatPlateRadiationFacet, from a facet model.*
- file [radiation\\_base\\_facet.cc](#)  
*Define member functions for class RadiationBaseFacet.*
- file [radiation\\_default\\_surface.cc](#)  
*Default surface for use with Radiation Pressure interaction model.*
- file [radiation\\_facet.cc](#)  
*Individual facets for use with Radiation Pressure interaction models.*
- file [radiation\\_messages.cc](#)  
*Implement the class RadiationMessages.*
- file [radiation\\_pressure.cc](#)  
*Calculation of force and torque due to radiation pressure.*
- file [radiation\\_pressure\\_\\_default\\_surface.cc](#)  
*Calculation of force and torque due to radiation pressure.*
- file [radiation\\_pressure\\_\\_surface\\_model.cc](#)  
*Calculation of force and torque due to radiation pressure.*
- file [radiation\\_source.cc](#)  
*Definition of methods associated with Primary Sources.*
- file [radiation\\_surface.cc](#)

*Vehicle surface model for Radiation Pressure model.*

- file [radiation\\_surface\\_factory.cc](#)

*Factory that creates an interaction surface, for a specific environment 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 PATH

```
#define PATH "interactions/radiation_pressure/"
```

Definition at line 36 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

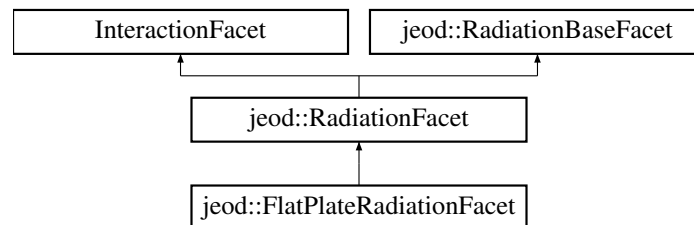
# Data Structure Documentation

### 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](#) ()=default
- [~FlatPlateRadiationFacet](#) () override=default
- [FlatPlateRadiationFacet](#) & [operator=](#) (const [FlatPlateRadiationFacet](#) &)=delete
- [FlatPlateRadiationFacet](#) (const [FlatPlateRadiationFacet](#) &)=delete
- void [incident\\_radiation](#) (const double flux\_mag, const double flux\_struct\_hat[3], const bool calculate\_forces) override  
*Calculation of force and torque due to radiation pressure.*
- void [initialize\\_geom](#) (double center\_grav[3]) override  
*Initializes the Facet for use in the model.*
- void [define\\_facet](#) (FlatPlate \*flat\_plate)  
*Defines the facet data values.*
- void [radiation\\_pressure](#) () override  
*Calculates the radiative emission force, accumulated force, and torque acting on a facet.*

## Data Fields

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

## Private Attributes

- double `sin_theta` {}  
*Theta is the angle between the plate and the radiation vector.*

## Friends

- class `InputProcessor`
- void `init_attrjeod__FlatPlateRadiationFacet` ()

## Additional Inherited Members

### 8.1.1 Detailed Description

A flat plate facet to be used for radiation interaction.

Definition at line 86 of file `flat_plate_radiation_facet.hh`.

### 8.1.2 Constructor & Destructor Documentation

#### 8.1.2.1 FlatPlateRadiationFacet() [1/2]

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

#### 8.1.2.2 ~FlatPlateRadiationFacet()

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

#### 8.1.2.3 FlatPlateRadiationFacet() [2/2]

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

### 8.1.3 Member Function Documentation

#### 8.1.3.1 define\_facet()

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

Defines the facet data values.

## Parameters

in	<i>flat_plate</i>	pointer to the flat plate object
----	-------------------	----------------------------------

Definition at line 54 of file flat\_plate\_radiation\_facet.cc.

References [jeod::RadiationFacet::center\\_pressure](#), and [normal](#).

Referenced by [jeod::FlatPlateRadiationFactory::create\\_facet\(\)](#).

## 8.1.3.2 incident\_radiation()

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

Calculation of force and torque due to radiation pressure.

## Assumptions and Limitations

- Only called when `flux_mag > 0`

## Parameters

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

Implements [jeod::RadiationBaseFacet](#).

Definition at line 85 of file flat\_plate\_radiation\_facet.cc.

References [jeod::RadiationBaseFacet::albedo](#), [jeod::RadiationBaseFacet::areaxflux](#), [jeod::RadiationBaseFacet::areaxflux\\_e](#), [jeod::RadiationBaseFacet::cx\\_area](#), [jeod::RadiationBaseFacet::diffuse](#), [jeod::RadiationBaseFacet::F\\_absorption](#), [jeod::RadiationBaseFacet::F\\_diffuse](#), [jeod::RadiationBaseFacet::F\\_specular](#), [normal](#), [sin\\_theta](#), [jeod::RadiationBaseFacet::speed\\_of\\_light](#), [jeod::RadiationBaseFacet::thermal](#), and [jeod::RadiationFacet::two thirds](#).

## 8.1.3.3 initialize\_geom()

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

Initializes the Facet for use in the model.

## Parameters

in	<i>center_grav</i>	center of gravity position Units: M
----	--------------------	--

Implements [jeod::RadiationFacet](#).

Definition at line 68 of file flat\_plate\_radiation\_facet.cc.

References [jeod::RadiationFacet::center\\_pressure](#), [jeod::RadiationFacet::crot\\_to\\_cp](#), [jeod::RadiationFacet::initialize\(\)](#), and [jeod::RadiationBaseFacet::thermal](#).

#### 8.1.3.4 operator=()

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

#### 8.1.3.5 radiation\_pressure()

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

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

Implements [jeod::RadiationFacet](#).

Definition at line 135 of file flat\_plate\_radiation\_facet.cc.

References [jeod::RadiationFacet::crot\\_to\\_cp](#), [jeod::RadiationBaseFacet::F\\_absorption](#), [jeod::RadiationBaseFacet::F\\_diffuse](#), [jeod::RadiationBaseFacet::F\\_emission](#), [jeod::RadiationBaseFacet::F\\_specular](#), [normal](#), [jeod::RadiationBaseFacet::speed\\_of\\_light](#), [jeod::RadiationBaseFacet::thermal](#), [jeod::RadiationFacet::two\\_thirds](#), and [jeod::RadiationMessages::unknown\\_numerical\\_error](#).

### 8.1.4 Friends And Related Function Documentation

#### 8.1.4.1 init\_attrjeod\_FlatPlateRadiationFacet

```
void init_attrjeod_FlatPlateRadiationFacet ( ) [friend]
```

#### 8.1.4.2 InputProcessor

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

Definition at line 88 of file flat\_plate\_radiation\_facet.hh.

### 8.1.5 Field Documentation

#### 8.1.5.1 incident\_flux\_hat

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

Temporary value.

trick\_units(-)

Definition at line 101 of file flat\_plate\_radiation\_facet.hh.

#### 8.1.5.2 normal

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

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

once the radiation surface is initialized, it points to the normal found in FlatPlatetrick\_units(-)

Definition at line 96 of file flat\_plate\_radiation\_facet.hh.

Referenced by define\_facet(), incident\_radiation(), and radiation\_pressure().

#### 8.1.5.3 sin\_theta

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

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

trick\_units(-)

Definition at line 107 of file flat\_plate\_radiation\_facet.hh.

Referenced by incident\_radiation().

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

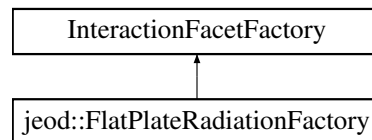
- [flat\\_plate\\_radiation\\_facet.hh](#)
- [flat\\_plate\\_radiation\\_facet.cc](#)

## 8.2 jeod::FlatPlateRadiationFactory Class Reference

The factory for building flat plate radiation facets.

```
#include <flat_plate_radiation_factory.hh>
```

Inheritance diagram for jeod::FlatPlateRadiationFactory:



### Public Member Functions

- [FlatPlateRadiationFactory](#) ()  
*Constructor for [FlatPlateRadiationFactory](#).*
- [~FlatPlateRadiationFactory](#) () override=default
- [FlatPlateRadiationFactory](#) & [operator=](#) (const [FlatPlateRadiationFactory](#) &)=delete
- [FlatPlateRadiationFactory](#) (const [FlatPlateRadiationFactory](#) &)=delete
- InteractionFacet \* [create\\_facet](#) (Facet \*facet, FacetParams \*params) override  
*Records the data for the Flat Plate Radiation Facet.*
- bool [is\\_correct\\_factory](#) (Facet \*facet) override  
*Tests to ensure that the factory can function on the facet as intended.*

### Friends

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

### 8.2.1 Detailed Description

The factory for building flat plate radiation facets.

Definition at line 84 of file flat\_plate\_radiation\_factory.hh.

### 8.2.2 Constructor & Destructor Documentation

#### 8.2.2.1 FlatPlateRadiationFactory() [1/2]

```
jeod::FlatPlateRadiationFactory::FlatPlateRadiationFactory ( )
```

Constructor for [FlatPlateRadiationFactory](#).

Definition at line 59 of file flat\_plate\_radiation\_factory.cc.

## 8.2.2.2 ~FlatPlateRadiationFactory()

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

## 8.2.2.3 FlatPlateRadiationFactory() [2/2]

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

## 8.2.3 Member Function Documentation

## 8.2.3.1 create\_facet()

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

Records the data for the Flat Plate Radiation Facet.

## Returns

pointer to the interaction facet that this function creates.

## Parameters

in	<i>facet</i>	pointer to the facet
in	<i>params</i>	pointer to the set of parameters for the facet.

Definition at line 70 of file flat\_plate\_radiation\_factory.cc.

References jeod::FlatPlateRadiationFacet::define\_facet(), and jeod::RadiationMessages::invalid\_setup\_error.

## 8.2.3.2 is\_correct\_factory()

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

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

## Returns

Boolean, is this the correct factory?

## Parameters

in	<i>facet</i>	pointer to the facet being manipulated by the factory
----	--------------	---

Definition at line 130 of file flat\_plate\_radiation\_factory.cc.

## 8.2.3.3 operator=()

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

## 8.2.4 Friends And Related Function Documentation

## 8.2.4.1 init\_attrjeod\_\_FlatPlateRadiationFactory

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

## 8.2.4.2 InputProcessor

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

Definition at line 86 of file flat\_plate\_radiation\_factory.hh.

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

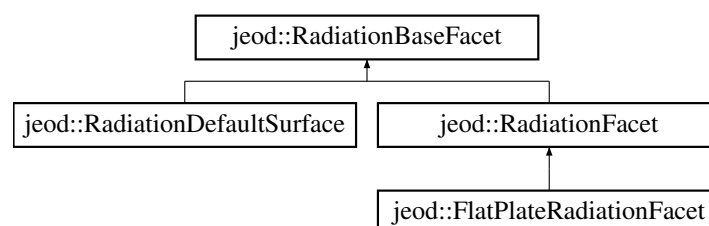
- [flat\\_plate\\_radiation\\_factory.hh](#)
- [flat\\_plate\\_radiation\\_factory.cc](#)

## 8.3 jeod::RadiationBaseFacet Class Reference

Generic type of facet for radiation pressure.

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

Inheritance diagram for jeod::RadiationBaseFacet:





## Public Member Functions

- [RadiationBaseFacet](#) ()=default
- virtual [~RadiationBaseFacet](#) ()=default
- [RadiationBaseFacet](#) & [operator=](#) (const [RadiationBaseFacet](#) &)=delete
- [RadiationBaseFacet](#) (const [RadiationBaseFacet](#) &)=delete
- virtual void [initialize](#) ()  
*initializes the base surface*
- virtual void [interact\\_with\\_third\\_body](#) ([RadiationThirdBody](#) \*third\_body, const bool calc\_forces)  
*Calculation of force, torques and irradiance due to ThirdBody flux.*
- virtual void [initialize\\_runtime\\_values](#) ()  
*To initialize the values during each update run.*
- virtual void [incident\\_radiation](#) (const double flux\_mag, const double flux\_hat[3], const bool calc\_forc)=0  
*Calculates the effect on the facet of the incident radiation.*

## Data Fields

- double [albedo](#) {-1.0}  
*Usable value of albedo, set to either albedo\_IR or albedo\_vis, depending on situation.*
- double [albedo\\_vis](#) {-1.0}  
*Fraction of incident visible radiation that is immediately reflected.*
- double [albedo\\_IR](#) {-1.0}  
*Fraction of incident IR radiation that is immediately reflected.*
- double [diffuse](#) {-1.0}  
*Fraction of reflected radiation that is reflected diffusely (balance reflected specularly)*
- ThermalFacetRider [thermal](#)  
*thermal characteristics of the facet.*
- double [cx\\_area](#) {}  
*cross-sectional area projected perpendicular to the radiation vector.*
- double [areaxflux](#) {}  
*product of momentum flux and cross-sectional area (cx\_area).*
- double [areaxflux\\_e](#) {}  
*product of energy flux and cross-sectional area*
- double [F\\_absorption](#) [3] {}  
*Force due to photon absorption from ONLY ONE source.*
- double [F\\_specular](#) [3] {}  
*Force due to photon specular reflection from ONLY ONE source.*
- double [F\\_diffuse](#) [3] {}  
*Force due to photon diffuse reflection from ONLY ONE source.*
- double [F\\_emission](#) [3] {}  
*Force due to photon (thermal) emission.*

## Static Protected Attributes

- static constexpr double [speed\\_of\\_light](#) {299792458.0}  
*Speed of light in vacuum.*

## Friends

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

### 8.3.1 Detailed Description

Generic type of facet for radiation pressure.

Definition at line 86 of file radiation\_base\_facet.hh.

### 8.3.2 Constructor & Destructor Documentation

#### 8.3.2.1 RadiationBaseFacet() [1/2]

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

#### 8.3.2.2 ~RadiationBaseFacet()

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

#### 8.3.2.3 RadiationBaseFacet() [2/2]

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

### 8.3.3 Member Function Documentation

#### 8.3.3.1 incident\_radiation()

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

Calculates the effect on the facet of the incident radiation.

#### Parameters

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

Implemented in [jeod::RadiationDefaultSurface](#), and [jeod::FlatPlateRadiationFacet](#).

Referenced by [jeod::RadiationSurface::incident\\_radiation\(\)](#).

#### 8.3.3.2 initialize()

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

initializes the base surface

Reimplemented in [jeod::RadiationFacet](#), and [jeod::RadiationDefaultSurface](#).

Definition at line 50 of file `radiation_base_facet.cc`.

References `albedo`, `albedo_IR`, `albedo_vis`, and [jeod::RadiationMessages::invalid\\_setup\\_error](#).

Referenced by [jeod::RadiationDefaultSurface::initialize\(\)](#), and [jeod::RadiationFacet::initialize\(\)](#).

#### 8.3.3.3 initialize\_runtime\_values()

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

To initialize the values during each update run.

Definition at line 110 of file `radiation_base_facet.cc`.

References `F_absorption`, `F_diffuse`, `F_specular`, and `thermal`.

Referenced by [jeod::RadiationSurface::initialize\\_runtime\\_values\(\)](#), and [jeod::RadiationPressure::update\\_default←\\_surface\(\)](#).

#### 8.3.3.4 interact\_with\_third\_body()

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

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

##### Parameters

in, out	<i>third_body_ptr</i>	Third body that emits radiation
in	<i>calculate_forces</i>	Calculate forces on this facet if true.

Definition at line 95 of file `radiation_base_facet.cc`.

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

Referenced by `jeod::RadiationSurface::interact_with_third_body()`, and `jeod::RadiationPressure::update_default←_surface()`.

#### 8.3.3.5 operator=()

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

### 8.3.4 Friends And Related Function Documentation

#### 8.3.4.1 init\_attrjeod\_\_RadiationBaseFacet

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

#### 8.3.4.2 InputProcessor

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

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

### 8.3.5 Field Documentation

#### 8.3.5.1 albedo

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

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

`trick_units(-)`

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

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

#### 8.3.5.2 albedo\_IR

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

Fraction of incident IR radiation that is immediately reflected.

trick\_units(-)

Definition at line 105 of file radiation\_base\_facet.hh.

Referenced by jeod::RadiationFacet::initialize(), initialize(), and interact\_with\_third\_body().

#### 8.3.5.3 albedo\_vis

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

Fraction of incident visible radiation that is immediately reflected.

trick\_units(-)

Definition at line 100 of file radiation\_base\_facet.hh.

Referenced by jeod::RadiationFacet::initialize(), initialize(), and interact\_with\_third\_body().

#### 8.3.5.4 areaxflux

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

product of momentum flux and cross-sectional area (cx\_area).

Highly variable.trick\_units(-)

Definition at line 132 of file radiation\_base\_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident\_radiation(), and jeod::RadiationDefaultSurface::incident\_↔ radiation().

#### 8.3.5.5 areaxflux\_e

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

product of energy flux and cross-sectional area

trick\_units(-)

Definition at line 137 of file radiation\_base\_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident\_radiation(), and jeod::RadiationDefaultSurface::incident\_↔ radiation().

### 8.3.5.6 cx\_area

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

cross-sectional area projected perpendicular to the radiation vector.

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

Definition at line 124 of file radiation\_base\_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident\_radiation(), jeod::RadiationDefaultSurface::incident\_radiation(), and jeod::RadiationDefaultSurface::initialize().

### 8.3.5.7 diffuse

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

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

trick\_units(-)

Definition at line 111 of file radiation\_base\_facet.hh.

Referenced by jeod::RadiationFacet::define\_facet\_core(), jeod::FlatPlateRadiationFacet::incident\_radiation(), jeod::RadiationDefaultSurface::incident\_radiation(), jeod::RadiationDefaultSurface::initialize(), and jeod::RadiationFacet::initialize().

### 8.3.5.8 F\_absorption

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

Force due to photon absorption from ONLY ONE source.

trick\_units(-)

Definition at line 142 of file radiation\_base\_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident\_radiation(), jeod::RadiationDefaultSurface::incident\_radiation(), initialize\_runtime\_values(), jeod::FlatPlateRadiationFacet::radiation\_pressure(), and jeod::RadiationPressure::update\_default\_surface().

#### 8.3.5.9 F\_diffuse

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

Force due to photon diffuse reflection from ONLY ONE source.

trick\_units(—)

Definition at line 152 of file radiation\_base\_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident\_radiation(), jeod::RadiationDefaultSurface::incident\_radiation(), initialize\_runtime\_values(), jeod::FlatPlateRadiationFacet::radiation\_pressure(), and jeod::RadiationPressure::update\_default\_surface().

#### 8.3.5.10 F\_emission

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

Force due to photon (thermal) emission.

trick\_units(—)

Definition at line 157 of file radiation\_base\_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::radiation\_pressure().

#### 8.3.5.11 F\_specular

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

Force due to photon specular reflection from ONLY ONE source.

trick\_units(—)

Definition at line 147 of file radiation\_base\_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident\_radiation(), jeod::RadiationDefaultSurface::incident\_radiation(), initialize\_runtime\_values(), jeod::FlatPlateRadiationFacet::radiation\_pressure(), and jeod::RadiationPressure::update\_default\_surface().

#### 8.3.5.12 speed\_of\_light

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

Speed of light in vacuum.

trick\_io(\*o) trick\_units(m/s)

Definition at line 164 of file radiation\_base\_facet.hh.

Referenced by jeod::FlatPlateRadiationFacet::incident\_radiation(), jeod::RadiationDefaultSurface::incident\_radiation(), and jeod::FlatPlateRadiationFacet::radiation\_pressure().

### 8.3.5.13 thermal

ThermalFacetRider jeod::RadiationBaseFacet::thermal

thermal characteristics of the facet.

trick\_units(-)

Definition at line 116 of file radiation\_base\_facet.hh.

Referenced by jeod::RadiationSurface::accumulate\_thermal\_sources(), jeod::RadiationDefaultSurface::add\_thermal\_integrator\_to(), jeod::RadiationSurface::add\_thermal\_integrators\_to(), jeod::RadiationFacet::define\_facet\_core(), jeod::RadiationSurface::equalize\_absorption\_emission(), jeod::FlatPlateRadiationFacet::incident\_radiation(), jeod::RadiationDefaultSurface::incident\_radiation(), jeod::RadiationDefaultSurface::initialize(), jeod::RadiationFacet::initialize(), jeod::RadiationSurface::initialize(), jeod::FlatPlateRadiationFacet::initialize\_geom(), initialize\_runtime\_values(), jeod::FlatPlateRadiationFacet::radiation\_pressure(), jeod::RadiationSurface::thermal\_integrator(), and jeod::RadiationDefaultSurface::thermal\_update().

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

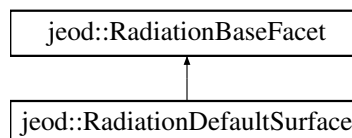
- [radiation\\_base\\_facet.hh](#)
- [radiation\\_base\\_facet.cc](#)

## 8.4 jeod::RadiationDefaultSurface Class Reference

The default spherical surface for radiation pressure.

```
#include <radiation_default_surface.hh>
```

Inheritance diagram for jeod::RadiationDefaultSurface:



### Public Member Functions

- [RadiationDefaultSurface](#) ()=default
- [~RadiationDefaultSurface](#) () override=default
- [RadiationDefaultSurface & operator=](#) (const [RadiationDefaultSurface](#) &)=delete
- [RadiationDefaultSurface](#) (const [RadiationDefaultSurface](#) &)=delete
- void [initialize](#) () override
  - initializes the default surface*
- void [incident\\_radiation](#) (const double flux\_mag, const double flux\_struc\_hat[3], const bool calculate\_forces) override
  - Calculation of force and torque due to radiation pressure.*
- void [thermal\\_update](#) ()
  - Test for necessity of maintaining thermal updates, and performs them as necessary.*
- void [add\\_thermal\\_integrator\\_to](#) (DynBody \*dyn\_body)
  - Adds the thermal integrator of this surface to the integration group of the DynBody.*
- void [set\\_name](#) (std::string name\_in)
  - Setter for the name.*



## Data Fields

- double [rad\\_coeff](#) {-1.0}  
*The radiation-equivalent of a drag coefficient.*
- double [temperature](#) {}  
*The value of the surface kinetic temperature.*
- std::string [name](#)  
*The name of the surface.*
- double [surface\\_area](#) {}  
*surface area of the default sphere.*

## Friends

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

## Additional Inherited Members

### 8.4.1 Detailed Description

The default spherical surface for radiation pressure.  
Definition at line 89 of file radiation\_default\_surface.hh.

### 8.4.2 Constructor & Destructor Documentation

#### 8.4.2.1 RadiationDefaultSurface() [1/2]

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

#### 8.4.2.2 ~RadiationDefaultSurface()

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

#### 8.4.2.3 RadiationDefaultSurface() [2/2]

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

### 8.4.3 Member Function Documentation

#### 8.4.3.1 add\_thermal\_integrator\_to()

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

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

**Parameters**

<i>in, out</i>	<i>dyn_body</i>	Body associated with this thermal object.
----------------	-----------------	---

Definition at line 268 of file radiation\_default\_surface.cc.

References `jeod::RadiationBaseFacet::thermal`.

**8.4.3.2 incident\_radiation()**

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

Calculation of force and torque due to radiation pressure.

**Assumptions and Limitations**

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

**Parameters**

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

Implements [jeod::RadiationBaseFacet](#).

Definition at line 214 of file radiation\_default\_surface.cc.

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

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

**8.4.3.3 initialize()**

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

initializes the default surface

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

Definition at line 58 of file radiation\_default\_surface.cc.

References [jeod::RadiationBaseFacet::albedo](#), [jeod::RadiationBaseFacet::cx\\_area](#), [jeod::RadiationBaseFacet::diffuse](#), [jeod::RadiationBaseFacet::initialize\(\)](#), [jeod::RadiationMessages::invalid\\_setup\\_error](#), [jeod::RadiationMessages::operational\\_setup\\_error](#), [rad\\_coeff](#), [surface\\_area](#), [temperature](#), and [jeod::RadiationBaseFacet::thermal](#).

Referenced by [jeod::RadiationPressure::initialize\(\)](#).

#### 8.4.3.4 operator=()

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

#### 8.4.3.5 set\_name()

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

Setter for the name.

Definition at line 148 of file radiation\_default\_surface.hh.

#### 8.4.3.6 thermal\_update()

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

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

Definition at line 251 of file radiation\_default\_surface.cc.

References [temperature](#), and [jeod::RadiationBaseFacet::thermal](#).

Referenced by [jeod::RadiationPressure::update\\_default\\_surface\(\)](#).

### 8.4.4 Friends And Related Function Documentation

#### 8.4.4.1 init\_attrjeod\_\_RadiationDefaultSurface

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

#### 8.4.4.2 InputProcessor

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

Definition at line 91 of file radiation\_default\_surface.hh.

### 8.4.5 Field Documentation

#### 8.4.5.1 name

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

The name of the surface.

This is optional.trick\_units(-)

Definition at line 116 of file radiation\_default\_surface.hh.

#### 8.4.5.2 rad\_coeff

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

The radiation-equivalent of a drag coefficient.

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

Definition at line 106 of file radiation\_default\_surface.hh.

Referenced by initialize().

#### 8.4.5.3 surface\_area

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

surface area of the default sphere.

trick\_units(m2)

Definition at line 125 of file radiation\_default\_surface.hh.

Referenced by initialize().

## 8.4.5.4 temperature

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

The value of the surface kinetic temperature.

trick\_units(K)

Definition at line 111 of file radiation\_default\_surface.hh.

Referenced by initialize(), and thermal\_update().

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

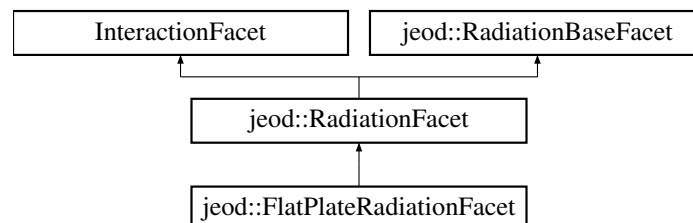
- [radiation\\_default\\_surface.hh](#)
- [radiation\\_default\\_surface.cc](#)

## 8.5 jeod::RadiationFacet Class Reference

Generic type of facet for radiation pressure.

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

Inheritance diagram for jeod::RadiationFacet:



## Public Member Functions

- [RadiationFacet](#) ()=default
- [~RadiationFacet](#) () override=default
- [RadiationFacet & operator=](#) (const [RadiationFacet](#) &)=delete
- [RadiationFacet](#) (const [RadiationFacet](#) &)=delete
- virtual void [radiation\\_pressure](#) ()=0  
*Calculates the effect of radiation pressure on the facet.*
- virtual void [initialize\\_geom](#) (double cg[3])=0  
*Initialize the facet geometry.*
- void [initialize](#) () override  
*Run sanity checks on input variables.*
- void [define\\_facet\\_core](#) (Facet \*facet, const ThermalFacetRider &facet\_thermal, [RadiationParams](#) \*params)  
*Defines the inherent facet values.*
- ThermalIntegrableObject & [get\\_thermal\\_integrator](#) ()  
*Get the integrator for thermal characteristics.*

## Data Fields

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

## Static Protected Attributes

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

## Friends

- class `InputProcessor`
- void `init_attrjeod__RadiationFacet` ()

### 8.5.1 Detailed Description

Generic type of facet for radiation pressure.

Definition at line 87 of file `radiation_facet.hh`.

### 8.5.2 Constructor & Destructor Documentation

#### 8.5.2.1 RadiationFacet() [1/2]

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

#### 8.5.2.2 ~RadiationFacet()

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

#### 8.5.2.3 RadiationFacet() [2/2]

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

### 8.5.3 Member Function Documentation

#### 8.5.3.1 define\_facet\_core()

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

Defines the inherent facet values.

## Parameters

in	<i>facet</i>	pointer to the facet
in	<i>facet_thermal</i>	reference to the thermal components of the facet.
in	<i>params</i>	pointer to the list of material parameters for the facet.

Definition at line 86 of file radiation\_facet.cc.

References [jeod::RadiationParams::albedo](#), [jeod::RadiationBaseFacet::albedo](#), [jeod::RadiationParams::diffuse](#), [jeod::RadiationBaseFacet::diffuse](#), [jeod::RadiationParams::thermal](#), and [jeod::RadiationBaseFacet::thermal](#).

## 8.5.3.2 get\_thermal\_integrator()

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

Get the integrator for thermal characteristics.

## Returns

Integrable object that integrates temperature.

Definition at line 144 of file radiation\_facet.hh.

## 8.5.3.3 initialize()

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

Run sanity checks on input variables.

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

Definition at line 56 of file radiation\_facet.cc.

References [jeod::RadiationBaseFacet::albedo](#), [jeod::RadiationBaseFacet::albedo\\_IR](#), [jeod::RadiationBaseFacet::albedo\\_vis](#), [jeod::RadiationBaseFacet::diffuse](#), [jeod::RadiationBaseFacet::initialize\(\)](#), [jeod::RadiationMessages::invalid\\_setup\\_error](#), and [jeod::RadiationBaseFacet::thermal](#).

Referenced by [jeod::FlatPlateRadiationFacet::initialize\\_geom\(\)](#).

## 8.5.3.4 initialize\_geom()

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

Initialize the facet geometry.

**Parameters**

<i>cg</i>	Center of mass.
-----------	-----------------

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

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

**8.5.3.5 operator=()**

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

**8.5.3.6 radiation\_pressure()**

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

Calculates the effect of radiation pressure on the facet.

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

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

**8.5.4 Friends And Related Function Documentation****8.5.4.1 init\_attrjeod\_\_RadiationFacet**

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

**8.5.4.2 InputProcessor**

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

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

**8.5.5 Field Documentation**



#### 8.5.5.1 center\_pressure

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

Center of pressure (in structural frame).

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

Definition at line 106 of file `radiation_facet.hh`.

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

#### 8.5.5.2 crot\_to\_cp

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

position of center of pressure w.r.t.

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

Definition at line 99 of file `radiation_facet.hh`.

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

#### 8.5.5.3 two\_thirds

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

quite literally, 2/3.

Occurs frequently with diffuse reflection and emission, so it is calculated once. `trick_io(*o) trick_units(-)`

Definition at line 115 of file `radiation_facet.hh`.

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

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

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

## 8.6 jeod::RadiationMessages Class Reference

Provides error messages.

```
#include <radiation_messages.hh>
```

### Public Member Functions

- [RadiationMessages](#) ()=delete
- [RadiationMessages](#) (const [RadiationMessages](#) &)=delete
- [RadiationMessages](#) & operator= (const [RadiationMessages](#) &)=delete

### Static Public Attributes

- static const char \* [incomplete\\_setup\\_error](#) = "interactions/radiation\_pressure/" "incomplete\_setup\_error"  
*The setup was not fully defined.*
- static const char \* [invalid\\_setup\\_error](#) = "interactions/radiation\_pressure/" "invalid\_setup\_error"  
*The setup is invalid.*
- static const char \* [operational\\_setup\\_error](#) = "interactions/radiation\_pressure/" "operational\_setup\_error"  
*Something internal went wrong in the setup methods without obvious user error.*
- static const char \* [invalid\\_function\\_call](#) = "interactions/radiation\_pressure/" "invalid\_function\_call"  
*A function was called before it was fully implemented.*
- static const char \* [unknown\\_numerical\\_error](#) = "interactions/radiation\_pressure/" "unknown\_numerical\_error"  
*Something went horribly wrong.*

### Friends

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

#### 8.6.1 Detailed Description

Provides error messages.

Definition at line 84 of file radiation\_messages.hh.

#### 8.6.2 Constructor & Destructor Documentation

##### 8.6.2.1 RadiationMessages() [1/2]

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

### 8.6.2.2 RadiationMessages() [2/2]

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

## 8.6.3 Member Function Documentation

### 8.6.3.1 operator=()

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

## 8.6.4 Friends And Related Function Documentation

### 8.6.4.1 init\_attrjeod\_\_RadiationMessages

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

### 8.6.4.2 InputProcessor

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

Definition at line 86 of file radiation\_messages.hh.

## 8.6.5 Field Documentation

### 8.6.5.1 incomplete\_setup\_error

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

The setup was not fully defined.

```
trick_units(-)
```

Definition at line 92 of file radiation\_messages.hh.

Referenced by jeod::RadiationPressure::add\_third\_body(), and jeod::RadiationSource::initialize().

### 8.6.5.2 invalid\_function\_call

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

A function was called before it was fully implemented.

trick\_units(-)

Definition at line 105 of file radiation\_messages.hh.

Referenced by jeod::RadiationPressure::find\_third\_body(), jeod::RadiationPressure::set\_third\_body\_active(), and jeod::RadiationPressure::set\_third\_body\_inactive().

### 8.6.5.3 invalid\_setup\_error

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

The setup is invalid.

trick\_units(-)

Definition at line 96 of file radiation\_messages.hh.

Referenced by jeod::RadiationSurfaceFactory::add\_facet\_params(), jeod::RadiationPressure::add\_third\_body(), jeod::RadiationSurface::allocate\_interaction\_facet(), jeod::RadiationThirdBody::calculate\_shadow(), jeod::RadiationThirdBody::convert\_shadow\_from\_int(), jeod::FlatPlateRadiationFactory::create\_facet(), jeod::RadiationDefaultSurface::initialize(), jeod::RadiationFacet::initialize(), jeod::RadiationSurface::initialize(), jeod::RadiationBaseFacet::initialize(), jeod::RadiationThirdBody::initialize(), jeod::RadiationThirdBody::process\_third\_body(), and jeod::RadiationThirdBody::update\_third\_body\_state().

### 8.6.5.4 operational\_setup\_error

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

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

trick\_units(-)

Definition at line 101 of file radiation\_messages.hh.

Referenced by jeod::RadiationSurface::allocate\_array(), jeod::RadiationSurface::allocate\_interaction\_facet(), jeod::RadiationDefaultSurface::initialize(), jeod::RadiationSurface::initialize(), and jeod::RadiationThirdBody::process\_third\_body().

## 8.6.5.5 unknown\_numerical\_error

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

Something went horribly wrong.

trick\_units(-)

Definition at line 110 of file radiation\_messages.hh.

Referenced by jeod::FlatPlateRadiationFacet::radiation\_pressure().

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

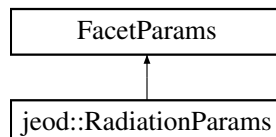
- [radiation\\_messages.hh](#)
- [radiation\\_messages.cc](#)

## 8.7 jeod::RadiationParams Class Reference

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

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

Inheritance diagram for jeod::RadiationParams:



### Public Member Functions

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

### Data Fields

- double [albedo](#) {}  
*Fraction of incident radiation that is immediately reflected.*
- double [diffuse](#) {}  
*Fraction of reflected radiation that is reflected diffusely (balance reflected specularly)*
- ThermalParams [thermal](#)  
*Additional thermal parameters.*

## Friends

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

### 8.7.1 Detailed Description

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

Definition at line 79 of file radiation\_params.hh.

### 8.7.2 Constructor & Destructor Documentation

#### 8.7.2.1 RadiationParams() [1/2]

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

#### 8.7.2.2 ~RadiationParams()

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

#### 8.7.2.3 RadiationParams() [2/2]

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

### 8.7.3 Member Function Documentation

#### 8.7.3.1 operator=()

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

### 8.7.4 Friends And Related Function Documentation

#### 8.7.4.1 init\_attrjeod\_\_RadiationParams

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

#### 8.7.4.2 InputProcessor

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

Definition at line 81 of file radiation\_params.hh.

### 8.7.5 Field Documentation

#### 8.7.5.1 albedo

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

Fraction of incident radiation that is immediately reflected.

trick\_units(-)

Definition at line 85 of file radiation\_params.hh.

Referenced by jeod::RadiationFacet::define\_facet\_core().

#### 8.7.5.2 diffuse

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

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

trick\_units(-)

Definition at line 91 of file radiation\_params.hh.

Referenced by jeod::RadiationFacet::define\_facet\_core().

### 8.7.5.3 thermal

ThermalParams jeod::RadiationParams::thermal

Additional thermal parameters.

trick\_units(-)

Definition at line 96 of file radiation\_params.hh.

Referenced by jeod::RadiationFacet::define\_facet\_core().

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

- [radiation\\_params.hh](#)

## 8.8 jeod::RadiationPressure Class Reference

Radiation pressure top-level definition.

```
#include <radiation_pressure.hh>
```

### Public Member Functions

- [RadiationPressure](#) ()  
*Constructor for [RadiationPressure](#).*
- virtual [~RadiationPressure](#) ()  
*Destructor for [RadiationPressure](#).*
- [RadiationPressure](#) (const [RadiationPressure](#) &)=delete
- [RadiationPressure](#) & operator= (const [RadiationPressure](#) &)=delete
- void [initialize](#) (DynManager &dyn\_manager, [RadiationSurface](#) \*surf\_ptr, double center\_grav[3])  
*Initialize the radiation pressure model when using a [RadiationSurface](#) (i.e.*
- void [initialize](#) (DynManager &dyn\_manager, [RadiationDefaultSurface](#) \*surf\_ptr)  
*Initialize the radiation pressure model when using a [RadiationDefaultSurface](#) (i.e.*
- void [update](#) (RefFrame &vehicle\_structural\_frame, double center\_grav[3], double scale\_factor, double time)  
*Updates the model at each time step.*
- void [add\\_third\\_body](#) ([RadiationThirdBody](#) \*third\_body\_ptr)  
*Adds a third body to the vector storage of third bodies.*
- void [set\\_third\\_body\\_active](#) (const std::string &third\_body\_name)  
*Sets a Third Body to be active when it previously was not.*
- void [set\\_third\\_body\\_inactive](#) (const std::string &third\_body\_name)  
*Sets a Third Body to be inactive when it previously was not.*
- void [set\\_calculate\\_forces](#) (bool calc\_forces)  
*Sets the value calculate\_forces.*



## Data Fields

- bool [active](#) {true}  
*Is radiation pressure desired?*
- double [force](#) [3] {}  
*Net force due to radiation.*
- double [torque](#) [3] {}  
*Net torque due to radiation.*
- double [illum\\_factor](#) {}  
*fraction of primary flux that is not eclipsed*
- [RadiationSource](#) [source](#)  
*Collection of data for radiation source.*
- ThermalModelRider [thermal](#)  
*Rider to allow dynamic thermal variation on facets in this model.*

## Protected Member Functions

- void [update\\_default\\_surface](#) ()  
*Used to update the model when the surface is a default surface.*
- void [update\\_facet\\_surface](#) ()  
*Used to update the model when the surface comprises facets.*
- void [initialize\\_environment](#) (DynManager \*dyn\_mgr\_ptr)  
*Initializes the source and third bodies.*
- void [third\\_body\\_adjustments](#) (double time, RefFrame &vehicle\_structural\_frame)  
*Used to modify the flux for shadowing.*
- int [find\\_third\\_body](#) (const std::string &third\_body\_name)  
*Finds the index of a third body based on its name.*

## Protected Attributes

- bool [initialized](#) {}  
*Has model been initialized?*
- bool [calculate\\_forces](#) {true}  
*Flag to indicate whether forces and torques should be calculated.*
- bool [third\\_bodies\\_active](#) {}  
*Flag indicates whether there are any of the third bodies currently active.*
- unsigned int [num\\_third\\_bodies](#) {}  
*Number of ThirdBodies available.*
- JeodPointerVector< [RadiationThirdBody](#) >::type [third\\_bodies](#)  
*Planetary bodies that provide shadowing or indirect, reflected, illumination.*
- [RadiationSurface](#) \* [surface\\_ptr](#) {}  
*The surface over which the radiation pressure will be collected.*
- [RadiationDefaultSurface](#) \* [default\\_surface\\_ptr](#) {}  
*If no surface is defined, use a DefaultSurface; this is the pointer to that default surface,.*
- DynManager \* [dyn\\_manager\\_ptr](#) {}  
*Pointer to the Dynamics Manager.*

## Friends

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

### 8.8.1 Detailed Description

Radiation pressure top-level definition.

Definition at line 90 of file [radiation\\_pressure.hh](#).

### 8.8.2 Constructor & Destructor Documentation

#### 8.8.2.1 RadiationPressure() [1/2]

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

Constructor for [RadiationPressure](#).

Definition at line 68 of file [radiation\\_pressure.cc](#).

References [third\\_bodies](#).

#### 8.8.2.2 ~RadiationPressure()

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

Destructor for [RadiationPressure](#).

Definition at line 395 of file [radiation\\_pressure.cc](#).

References [third\\_bodies](#).

#### 8.8.2.3 RadiationPressure() [2/2]

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

### 8.8.3 Member Function Documentation

#### 8.8.3.1 add\_third\_body()

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

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

## Parameters

in	<i>third_body_ptr</i>	pointer to ThirdBody
----	-----------------------	----------------------

Definition at line 156 of file radiation\_pressure.cc.

References jeod::RadiationThirdBody::active, dyn\_manager\_ptr, jeod::RadiationThirdBody::get\_added\_to\_model(), jeod::RadiationMessages::incomplete\_setup\_error, jeod::RadiationThirdBody::initialize(), initialized, jeod::RadiationMessages::invalid\_setup\_error, jeod::RadiationThirdBody::name, num\_third\_bodies, jeod::RadiationThirdBody::set\_added\_to\_model(), third\_bodies, and third\_bodies\_active.

Referenced by initialize().

## 8.8.3.2 find\_third\_body()

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

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

## Returns

void

## Parameters

in	<i>third_body_name</i>	Name of ThirdBody
----	------------------------	-------------------

Definition at line 358 of file radiation\_pressure.cc.

References jeod::RadiationMessages::invalid\_function\_call, num\_third\_bodies, and third\_bodies.

Referenced by set\_third\_body\_active(), and set\_third\_body\_inactive().

## 8.8.3.3 initialize() [1/2]

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

Initialize the radiation pressure model when using a [RadiationSurface](#) (i.e. with facets).

**Parameters**

in	<i>dyn_mgr</i>	The dynamics manager
in	<i>surf_ptr</i>	pointer to the radiation surface
in	<i>center_grav</i>	position of center of mass Units: M

Definition at line 70 of file radiation\_pressure\_\_surface\_model.cc.

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

**8.8.3.4 initialize()** [2/2]

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

Initialize the radiation pressure model when using a [RadiationDefaultSurface](#) (i.e.

spherical, uniform properties.)

**Parameters**

in	<i>dyn_mgr</i>	The dynamics manager
in	<i>surf_ptr</i>	pointer to the radiation surface

Definition at line 69 of file radiation\_pressure\_\_default\_surface.cc.

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

**8.8.3.5 initialize\_environment()**

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

Initializes the source and third bodies.

**Parameters**

in	<i>dyn_mgr_ptr</i>	The dynamics manager
----	--------------------	----------------------

Definition at line 79 of file radiation\_pressure.cc.

References jeod::RadiationSource::initialize(), initialized, num\_third\_bodies, source, and third\_bodies.

Referenced by initialize().

#### 8.8.3.6 operator=()

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

#### 8.8.3.7 set\_calculate\_forces()

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

Sets the value calculate\_forces.

##### Parameters

in	<i>value</i>	whether forces are needed
----	--------------	---------------------------

Definition at line 382 of file radiation\_pressure.cc.

References calculate\_forces, force, and torque.

#### 8.8.3.8 set\_third\_body\_active()

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

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

##### Parameters

in	<i>third_body_name</i>	Name of ThirdBody
----	------------------------	-------------------

Definition at line 270 of file radiation\_pressure.cc.

References active, dyn\_manager\_ptr, find\_third\_body(), jeod::RadiationMessages::invalid\_function\_call, third\_bodies, and third\_bodies\_active.

#### 8.8.3.9 set\_third\_body\_inactive()

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

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

##### Parameters

in	<i>third_body_name</i>	Name of ThirdBody
----	------------------------	-------------------

Definition at line 310 of file radiation\_pressure.cc.

References active, dyn\_manager\_ptr, find\_third\_body(), jeod::RadiationMessages::invalid\_function\_call, num\_↔  
third\_bodies, third\_bodies, and third\_bodies\_active.

#### 8.8.3.10 third\_body\_adjustments()

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

Used to modify the flux for shadowing.

##### Parameters

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

Definition at line 136 of file radiation\_pressure.cc.

References active, jeod::RadiationSource::flux\_inertial, jeod::RadiationSource::flux\_mag, jeod::RadiationSource↔  
::flux\_struc, illum\_factor, num\_third\_bodies, source, and third\_bodies.

Referenced by update().

#### 8.8.3.11 update()

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

Updates the model at each time step.

## Parameters

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

Definition at line 97 of file radiation\_pressure.cc.

References active, jeod::RadiationSource::calculate\_flux(), source, surface\_ptr, third\_bodies\_active, third\_body\_↵ adjustments(), update\_default\_surface(), and update\_facet\_surface().

## 8.8.3.12 update\_default\_surface()

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

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

Definition at line 90 of file radiation\_pressure\_\_default\_surface.cc.

References active, calculate\_forces, default\_surface\_ptr, jeod::RadiationBaseFacet::F\_absorption, jeod::↵ RadiationBaseFacet::F\_diffuse, jeod::RadiationBaseFacet::F\_specular, jeod::RadiationSource::flux\_mag, jeod↵ ::RadiationSource::flux\_struc\_hat, force, jeod::RadiationDefaultSurface::incident\_radiation(), jeod::Radiation↵ BaseFacet::initialize\_runtime\_values(), jeod::RadiationBaseFacet::interact\_with\_third\_body(), num\_third\_bodies, source, jeod::RadiationDefaultSurface::thermal\_update(), third\_bodies, and torque.

Referenced by update().

## 8.8.3.13 update\_facet\_surface()

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

Used to update the model when the surface comprises facets.

Definition at line 98 of file radiation\_pressure\_\_surface\_model.cc.

References active, calculate\_forces, jeod::RadiationSurface::equalize\_absorption\_emission(), jeod::Radiation↵ Source::flux\_mag, jeod::RadiationSource::flux\_struc\_hat, force, jeod::RadiationSurface::force, jeod::Radiation↵ Surface::incident\_radiation(), jeod::RadiationSurface::initialize\_runtime\_values(), jeod::RadiationSurface::interact↵ \_with\_third\_body(), num\_third\_bodies, jeod::RadiationSurface::radiation\_pressure(), source, surface\_ptr, thermal, third\_bodies, torque, and jeod::RadiationSurface::torque.

Referenced by update().

## 8.8.4 Friends And Related Function Documentation

#### 8.8.4.1 `init_attrjeod__RadiationPressure`

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

#### 8.8.4.2 `InputProcessor`

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

Definition at line 92 of file `radiation_pressure.hh`.

### 8.8.5 Field Documentation

#### 8.8.5.1 `active`

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

Is radiation pressure desired?

`trick_units(-)`

Definition at line 97 of file `radiation_pressure.hh`.

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

#### 8.8.5.2 `calculate_forces`

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

Flag to indicate whether forces and torques should be calculated.

`trick_units(-)`

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

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



#### 8.8.5.3 default\_surface\_ptr

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

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

trick\_units(-)

Definition at line 163 of file radiation\_pressure.hh.

Referenced by initialize(), and update\_default\_surface().

#### 8.8.5.4 dyn\_manager\_ptr

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

Pointer to the Dynamics Manager.

trick\_units(-)

Definition at line 168 of file radiation\_pressure.hh.

Referenced by add\_third\_body(), initialize(), set\_third\_body\_active(), and set\_third\_body\_inactive().

#### 8.8.5.5 force

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

Net force due to radiation.

trick\_units(N)

Definition at line 102 of file radiation\_pressure.hh.

Referenced by set\_calculate\_forces(), update\_default\_surface(), and update\_facet\_surface().

#### 8.8.5.6 illum\_factor

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

fraction of primary flux that is not eclipsed

trick\_units(-)

Definition at line 112 of file radiation\_pressure.hh.

Referenced by third\_body\_adjustments().

#### 8.8.5.7 initialized

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

Has model been initialized?

trick\_units(-)

Definition at line 128 of file radiation\_pressure.hh.

Referenced by add\_third\_body(), and initialize\_environment().

#### 8.8.5.8 num\_third\_bodies

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

Number of ThirdBodies available.

trick\_units(count)

Definition at line 144 of file radiation\_pressure.hh.

Referenced by add\_third\_body(), find\_third\_body(), initialize\_environment(), set\_third\_body\_inactive(), third\_body\_adjustments(), update\_default\_surface(), and update\_facet\_surface().

#### 8.8.5.9 source

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

Collection of data for radiation source.

trick\_units(-)

Definition at line 117 of file radiation\_pressure.hh.

Referenced by initialize(), initialize\_environment(), third\_body\_adjustments(), update(), update\_default\_surface(), and update\_facet\_surface().

#### 8.8.5.10 surface\_ptr

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

The surface over which the radiation pressure will be collected.

If this is NULL, use the default radiation surface (which is a whole separate entity from a RadiationSurface.trick\_units(-)

Definition at line 157 of file radiation\_pressure.hh.

Referenced by initialize(), update(), and update\_facet\_surface().

#### 8.8.5.11 thermal

ThermalModelRider jeod::RadiationPressure::thermal

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

trick\_units(-)

Definition at line 122 of file radiation\_pressure.hh.

Referenced by initialize(), and update\_facet\_surface().

#### 8.8.5.12 third\_bodies

JeodPointerVector<RadiationThirdBody>::type jeod::RadiationPressure::third\_bodies [protected]

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

trick\_io(\*\*)

Definition at line 150 of file radiation\_pressure.hh.

Referenced by add\_third\_body(), find\_third\_body(), initialize(), initialize\_environment(), RadiationPressure(), set\_↔\_third\_body\_active(), set\_third\_body\_inactive(), third\_body\_adjustments(), update\_default\_surface(), update\_↔\_facet\_surface(), and ~RadiationPressure().

#### 8.8.5.13 third\_bodies\_active

bool jeod::RadiationPressure::third\_bodies\_active {} [protected]

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

trick\_units(-)

Definition at line 139 of file radiation\_pressure.hh.

Referenced by add\_third\_body(), set\_third\_body\_active(), set\_third\_body\_inactive(), and update().

#### 8.8.5.14 torque

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

Net torque due to radiation.

trick\_units(N\*m)

Definition at line 107 of file radiation\_pressure.hh.

Referenced by set\_calculate\_forces(), update\_default\_surface(), and update\_facet\_surface().

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

- [radiation\\_pressure.hh](#)
- [radiation\\_pressure.cc](#)
- [radiation\\_pressure\\_\\_default\\_surface.cc](#)
- [radiation\\_pressure\\_\\_surface\\_model.cc](#)

## 8.9 jeod::RadiationSource Class Reference

Provides information on the source of the incident radiation.

```
#include <radiation_source.hh>
```

### Public Types

- enum `OldShadowGeometry` { `Cylindrical` = 0, `Cyl` = 1, `Conical` = 2, `Con` = 3 }  
*distinguishes between conical and cylindrical shadowing.*

### Public Member Functions

- `RadiationSource` ()=default
- virtual `~RadiationSource` ()=default
- `RadiationSource` (const `RadiationSource` &)=delete
- `RadiationSource` & `operator=` (const `RadiationSource` &)=delete
- virtual void `initialize` (DynManager \*dyn\_manager\_ptr)  
*Initializes the source object for use in the Radiation Pressure model.*
- virtual void `calculate_flux` (RefFrame &veh\_struc\_frame, const double center\_grav[3])  
*calculates the flux vector from the vehicle's position.*
- void `set_name` (std::string name\_in)  
*Setter for the name.*

### Data Fields

- const double `solar_luminosity` {3.827E+26}  
*Solar Luminosity.*
- const double `solar_radius` {6.98E+08}  
*Mean solar radius.*
- std::string `name` {"Sun"}  
*The name of the source of the illumination (usually Sun)*
- double `flux_hat` [3] {}  
*(inertial) unit vector associated with primary radiative flux.*
- double `flux_mag` {}  
*magnitude of principle radiative flux vector*
- double `d_source_to_cg` {}  
*distance from source to vehicle cg.*
- double `luminosity` {`solar_luminosity`}  
*Luminosity of primary source.*
- double `radius` {`solar_radius`}  
*Radius of primary source.*
- double `flux_inertial` [3] {}  
*(N/M/s) Power per unit area in the inertial reference frame*
- double `flux_struc` [3] {}  
*(N/M\*s) Power per unit area in the vehicle structural reference frame*
- double `flux_struc_hat` [3] {}  
*Unit vector representing flux vector in the vehicle structural reference frame.*
- double `inertial_cg` [3] {}

- vehicle cg position w.r.t vehicle strucural origin, expressed in inertial RF.*
- double [source\\_to\\_cg](#) [3] {}  
*vehicle cg position relative to the source*
- double [source\\_to\\_struc\\_origin](#) [3] {}  
*vehicle struc frame origin position relative to source.*
- bool [multiple\\_shadow\\_bodies](#) {}  
*flags that more than one body are casting shadows on the vehicle.*
- RefFrame \* [inertial\\_frame\\_ptr](#) {}  
*the inertial reference frame associated with this source*
- bool [bodies\\_active](#) {}  
*can there be 3rd-body interference, either shadowing or reflection.*
- unsigned int [num\\_bodies](#) {}  
*number of ThirdBodies available.*
- [RadiationThirdBody](#) \*\* [third\\_body](#) {}  
*Planetary bodies that provide shadowing or indirect, reflected, illumination.*
- [OldShadowGeometry](#) [shadow\\_geometry](#) {}  
*Flag indicating cylindrical / conical shadow geometry.*

## Friends

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

### 8.9.1 Detailed Description

Provides information on the source of the incident radiation.

Definition at line 90 of file radiation\_source.hh.

### 8.9.2 Member Enumeration Documentation

#### 8.9.2.1 OldShadowGeometry

enum [jeod::RadiationSource::OldShadowGeometry](#)

distinguishes between conical and cylindrical shadowing.

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

#### Enumerator

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

Definition at line 200 of file radiation\_source.hh.

### 8.9.3 Constructor & Destructor Documentation

#### 8.9.3.1 RadiationSource() [1/2]

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

#### 8.9.3.2 ~RadiationSource()

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

#### 8.9.3.3 RadiationSource() [2/2]

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

### 8.9.4 Member Function Documentation

#### 8.9.4.1 calculate\_flux()

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

calculates the flux vector from the vehicle's position.

##### Parameters

in	<i>veh_struc_frame</i>	the vehicle structural reference frame
in	<i>center_gravity</i>	position of the center of mass Units: M

Definition at line 72 of file radiation\_source.cc.

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

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

#### 8.9.4.2 initialize()

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

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

##### Parameters

in	<i>dyn_mgr_ptr</i>	pointer to the dynamics manager
----	--------------------	---------------------------------

Definition at line 117 of file radiation\_source.cc.

References jeod::RadiationMessages::incomplete\_setup\_error, inertial\_frame\_ptr, and name.

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

#### 8.9.4.3 operator=()

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

#### 8.9.4.4 set\_name()

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

Setter for the name.

Definition at line 227 of file radiation\_source.hh.

### 8.9.5 Friends And Related Function Documentation

#### 8.9.5.1 init\_attrjeod\_\_RadiationSource

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

### 8.9.5.2 InputProcessor

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

Definition at line 92 of file radiation\_source.hh.

## 8.9.6 Field Documentation

### 8.9.6.1 bodies\_active

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

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

trick\_units(-)

Definition at line 183 of file radiation\_source.hh.

### 8.9.6.2 d\_source\_to\_cg

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

distance from source to vehicle cg.

trick\_units(m)

Definition at line 121 of file radiation\_source.hh.

Referenced by calculate\_flux(), and jeod::RadiationThirdBody::calculate\_shadow().

### 8.9.6.3 flux\_hat

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

(inertial) unit vector associated with primary radiative flux.

trick\_units(-)

Definition at line 111 of file radiation\_source.hh.

Referenced by calculate\_flux().



#### 8.9.6.4 flux\_inertial

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

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

trick\_units(-)

Definition at line 136 of file radiation\_source.hh.

Referenced by calculate\_flux(), and jeod::RadiationPressure::third\_body\_adjustments().

#### 8.9.6.5 flux\_mag

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

magnitude of principle radiative flux vector

trick\_units(N/m2)

Definition at line 116 of file radiation\_source.hh.

Referenced by calculate\_flux(), jeod::RadiationPressure::third\_body\_adjustments(), jeod::RadiationPressure::update\_default\_surface(), and jeod::RadiationPressure::update\_facet\_surface().

#### 8.9.6.6 flux\_struc

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

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

trick\_units(-)

Definition at line 141 of file radiation\_source.hh.

Referenced by calculate\_flux(), and jeod::RadiationPressure::third\_body\_adjustments().

#### 8.9.6.7 flux\_struc\_hat

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

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

trick\_units(-)

Definition at line 147 of file radiation\_source.hh.

Referenced by calculate\_flux(), jeod::RadiationPressure::update\_default\_surface(), and jeod::RadiationPressure::update\_facet\_surface().

#### 8.9.6.8 inertial\_cg

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

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

trick\_units(m)

Definition at line 153 of file radiation\_source.hh.

Referenced by calculate\_flux().

#### 8.9.6.9 inertial\_frame\_ptr

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

the inertial reference frame associated with this source

trick\_units(-)

Definition at line 173 of file radiation\_source.hh.

Referenced by calculate\_flux(), initialize(), jeod::RadiationThirdBody::initialize(), and jeod::RadiationThirdBody↵  
::update\_third\_body\_state().

#### 8.9.6.10 luminosity

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

Luminosity of primary source.

trick\_units(-)

Definition at line 126 of file radiation\_source.hh.

Referenced by calculate\_flux().

#### 8.9.6.11 multiple\_shadow\_bodies

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

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

trick\_units(-)

Definition at line 168 of file radiation\_source.hh.

#### 8.9.6.12 name

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

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

trick\_units(—)

Definition at line 106 of file radiation\_source.hh.

Referenced by initialize().

#### 8.9.6.13 num\_bodies

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

number of ThirdBodies available.

trick\_units(count)

Definition at line 188 of file radiation\_source.hh.

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

#### 8.9.6.14 radius

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

Radius of primary source.

trick\_units(m)

Definition at line 131 of file radiation\_source.hh.

Referenced by jeod::RadiationThirdBody::calculate\_shadow(), and jeod::RadiationThirdBody::initialize().

#### 8.9.6.15 shadow\_geometry

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

Flag indicating cylindrical / conical shadow geometry.

trick\_units(—)

Definition at line 211 of file radiation\_source.hh.

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

#### 8.9.6.16 solar\_luminosity

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

Solar Luminosity.

trick\_units(-)

Definition at line 96 of file radiation\_source.hh.

#### 8.9.6.17 solar\_radius

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

Mean solar radius.

trick\_units(m)

Definition at line 101 of file radiation\_source.hh.

#### 8.9.6.18 source\_to\_cg

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

vehicle cg position relative to the source

trick\_units(m)

Definition at line 158 of file radiation\_source.hh.

Referenced by calculate\_flux(), and jeod::RadiationThirdBody::calculate\_shadow().

#### 8.9.6.19 source\_to\_struc\_origin

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

vehicle struc frame origin position relative to source.

trick\_units(m)

Definition at line 163 of file radiation\_source.hh.

Referenced by calculate\_flux().

## 8.9.6.20 third\_body

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

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

trick\_units(-)

Definition at line 194 of file radiation\_source.hh.

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

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

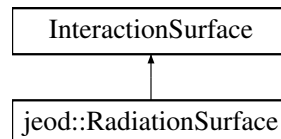
- [radiation\\_source.hh](#)
- [radiation\\_source.cc](#)

## 8.10 jeod::RadiationSurface Class Reference

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

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

Inheritance diagram for jeod::RadiationSurface:



## Public Member Functions

- [RadiationSurface](#) ()  
*Constructor for [RadiationSurface](#).*
- [~RadiationSurface](#) () override  
*Destructor for [RadiationSurface](#).*
- [RadiationSurface & operator=](#) (const [RadiationSurface](#) &)=delete
- [RadiationSurface](#) (const [RadiationSurface](#) &)=delete
- void [initialize](#) (double center\_grav[3])  
*Initializes the radiation surface.*
- void [allocate\\_array](#) (unsigned int size) override  
*Allocates memory for an array of radiation facets.*
- void [allocate\\_interaction\\_facet](#) (Facet \*facet, InteractionFacetFactory \*factory, FacetParams \*params, unsigned int index) override  
*Turns facet memory into radiation facet memory.*
- void [initialize\\_runtime\\_values](#) ()  
*To initialize the values during each update run.*
- void [incident\\_radiation](#) (double flux\_mag, const double flux\_struc\_hat[3], bool calculate\_forces)  
*systematically calls the method to calculate the interaction on each facet.*

- void `interact_with_third_body` (`RadiationThirdBody` \*third\_body\_ptr, const bool calculate\_forces)  
*systematically calls the method to calculate the interaction of each facet with ThirdBody flux.*
- void `accumulate_thermal_sources` () override  
*systematically calls the method to accumulate thermal sources on each facet.*
- void `thermal_integrator` () override  
*systematically calls the method to integrate the temperature variation for each facet.*
- void `equalize_absorption_emission` ()  
*systematically calls the method to ensure that the same for each facet.*
- void `radiation_pressure` ()  
*systematically calls the method to wrap up the radiation pressure calculation on each facet and accumulate forces and torques over all facets.*
- void `add_thermal_integrators_to` (`DynBody` \*dyn\_body)  
*To attach thermal integrators from each facet to the integration group of a DynBody.*

## Data Fields

- `RadiationFacet` \*\* `facets` {}  
*Array of pointers to the Radiation Facets.*
- unsigned int `num_facets` {}  
*number of facets in this surface.*
- bool `include_conduction` {}  
*Flag to instruct model to include conduction between plates in the thermal calculation.*
- double \*\* `thermal_conduction` {}  
*Conductivity values (Watts per Kelvin) between facets.*
- double `force` [3] {}  
*Force resulting from all radiative interactions.*
- double `torque` [3] {}  
*Torque resulting from all radiative interactions.*
- unsigned int `ii_facet` {}  
*Simple counter, used repeatedly.*

## Friends

- class `InputProcessor`
- void `init_attrjeod__RadiationSurface` ()

### 8.10.1 Detailed Description

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

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

### 8.10.2 Constructor & Destructor Documentation

**8.10.2.1 RadiationSurface()** [1/2]

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

Constructor for [RadiationSurface](#).

Definition at line 60 of file radiation\_surface.cc.

**8.10.2.2 ~RadiationSurface()**

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

Destructor for [RadiationSurface](#).

Definition at line 337 of file radiation\_surface.cc.

References facets, and num\_facets.

**8.10.2.3 RadiationSurface()** [2/2]

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

**8.10.3 Member Function Documentation****8.10.3.1 accumulate\_thermal\_sources()**

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

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

Definition at line 255 of file radiation\_surface.cc.

References facets, ii\_facet, num\_facets, and jeod::RadiationBaseFacet::thermal.

**8.10.3.2 add\_thermal\_integrators\_to()**

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

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

**Parameters**

<i>in, out</i>	<i>dyn_body</i>	Body to which integrators are to be added.
----------------	-----------------	--

Definition at line 325 of file radiation\_surface.cc.

References facets, num\_facets, and jeod::RadiationBaseFacet::thermal.

**8.10.3.3 allocate\_array()**

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

Allocates memory for an array of radiation facets.

**Parameters**

<i>in</i>	<i>size</i>	Size of array
-----------	-------------	---------------

Definition at line 134 of file radiation\_surface.cc.

References facets, ii\_facet, num\_facets, and jeod::RadiationMessages::operational\_setup\_error.

**8.10.3.4 allocate\_interaction\_facet()**

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

Turns facet memory into radiaiton facet memory.

**Parameters**

<i>in</i>	<i>facet</i>	pointer to the facet
<i>in</i>	<i>factory</i>	pointer to the facet factory
<i>in</i>	<i>params</i>	pointer to the generic facet parameter.
<i>in</i>	<i>index</i>	index value in the facet list.

Definition at line 166 of file radiation\_surface.cc.

References facets, jeod::RadiationMessages::invalid\_setup\_error, num\_facets, and jeod::RadiationMessages↵  
::operational\_setup\_error.



## 8.10.3.5 equalize\_absorption\_emission()

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

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

Definition at line 284 of file radiation\_surface.cc.

References facets, ii\_facet, num\_facets, and jeod::RadiationBaseFacet::thermal.

Referenced by jeod::RadiationPressure::update\_facet\_surface().

## 8.10.3.6 incident\_radiation()

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

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

## Parameters

in	<i>flux_mag</i>	Magnitude of incident flux
in	<i>flux_struc_hat</i>	unit vector of incident flux
in	<i>calculate_forces</i>	boolean indicating whether to calculate forces.

Definition at line 230 of file radiation\_surface.cc.

References facets, ii\_facet, jeod::RadiationBaseFacet::incident\_radiation(), and num\_facets.

Referenced by jeod::RadiationPressure::update\_facet\_surface().

## 8.10.3.7 initialize()

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

Initializes the radiation surface.

## Parameters

in	<i>center_grav</i>	position of center of mass Units: M
----	--------------------	--

Definition at line 69 of file radiation\_surface.cc.

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

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

#### 8.10.3.8 initialize\_runtime\_values()

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

To initialize the values during each update run.

Definition at line 312 of file `radiation_surface.cc`.

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

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

#### 8.10.3.9 interact\_with\_third\_body()

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

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

##### Parameters

<code>in, out</code>	<code>third_body_ptr</code>	Third body that emits radiation.
<code>in</code>	<code>calculate_forces</code>	Calculate forces on the surface if true.

Definition at line 244 of file `radiation_surface.cc`.

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

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

#### 8.10.3.10 operator=()

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

#### 8.10.3.11 radiation\_pressure()

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

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

Definition at line 296 of file radiation\_surface.cc.

References facets, force, ii\_facet, num\_facets, jeod::RadiationFacet::radiation\_pressure(), and torque.

Referenced by jeod::RadiationPressure::update\_facet\_surface().

#### 8.10.3.12 thermal\_integrator()

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

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

Definition at line 266 of file radiation\_surface.cc.

References facets, ii\_facet, num\_facets, and jeod::RadiationBaseFacet::thermal.

### 8.10.4 Friends And Related Function Documentation

#### 8.10.4.1 init\_attrjeod\_\_RadiationSurface

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

#### 8.10.4.2 InputProcessor

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

Definition at line 90 of file radiation\_surface.hh.

### 8.10.5 Field Documentation

#### 8.10.5.1 facets

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

Array of pointers to the Radiation Facets.

trick\_units(-)

Definition at line 94 of file radiation\_surface.hh.

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

#### 8.10.5.2 force

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

Force resulting from all radiative interactions.

trick\_units(-)

Definition at line 118 of file radiation\_surface.hh.

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

#### 8.10.5.3 ii\_facet

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

Simple counter, used repeatedly.

trick\_units(-)

Definition at line 128 of file radiation\_surface.hh.

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

#### 8.10.5.4 include\_conduction

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

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

Requires a `thermal_conduction` matrix.trick\_units(-)

Definition at line 105 of file radiation\_surface.hh.

#### 8.10.5.5 num\_facets

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

number of facets in this surface.

trick\_units(-)

Definition at line 99 of file radiation\_surface.hh.

Referenced by `accumulate_thermal_sources()`, `add_thermal_integrators_to()`, `allocate_array()`, `allocate_interaction_facet()`, `equalize_absorption_emission()`, `incident_radiation()`, `initialize()`, `initialize_runtime_values()`, `interact_with_third_body()`, `radiation_pressure()`, `thermal_integrator()`, and `~RadiationSurface()`.

#### 8.10.5.6 thermal\_conduction

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

Conductivity values (Watts per Kelvin) between facets.

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

Definition at line 113 of file radiation\_surface.hh.

#### 8.10.5.7 torque

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

Torque resulting from all radiative interactions.

trick\_units(-)

Definition at line 123 of file radiation\_surface.hh.

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

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

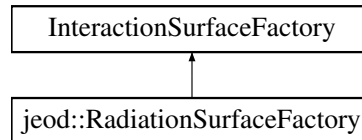
- [radiation\\_surface.hh](#)
- [radiation\\_surface.cc](#)

## 8.11 jeod::RadiationSurfaceFactory Class Reference

The factory for creating Radiation Surfaces.

```
#include <radiation_surface_factory.hh>
```

Inheritance diagram for jeod::RadiationSurfaceFactory:



### Public Member Functions

- [RadiationSurfaceFactory](#) ()  
*Constructor.*
- [~RadiationSurfaceFactory](#) () override  
*Destructor.*
- [RadiationSurfaceFactory](#) & [operator=](#) (const [RadiationSurfaceFactory](#) &)=delete
- [RadiationSurfaceFactory](#) (const [RadiationSurfaceFactory](#) &)=delete
- void [add\\_facet\\_params](#) (FacetParams \*to\_add) override  
*Add a set of radiation parameters to those available to Radiation Surface.*

### Protected Attributes

- [FlatPlateRadiationFactory](#) flat\_plate\_radiation\_factory  
*The factory to build flat plate thermal facets.*

### Friends

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

#### 8.11.1 Detailed Description

The factory for creating Radiation Surfaces.

Definition at line 85 of file radiation\_surface\_factory.hh.

#### 8.11.2 Constructor & Destructor Documentation

**8.11.2.1 RadiationSurfaceFactory()** [1/2]

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

Constructor.

Definition at line 52 of file radiation\_surface\_factory.cc.

References flat\_plate\_radiation\_factory.

**8.11.2.2 ~RadiationSurfaceFactory()**

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

Destructor.

Definition at line 91 of file radiation\_surface\_factory.cc.

**8.11.2.3 RadiationSurfaceFactory()** [2/2]

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

**8.11.3 Member Function Documentation****8.11.3.1 add\_facet\_params()**

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

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

Parameters

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

Definition at line 64 of file radiation\_surface\_factory.cc.

References jeod::RadiationMessages::invalid\_setup\_error.

### 8.11.3.2 operator=()

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

## 8.11.4 Friends And Related Function Documentation

### 8.11.4.1 init\_attrjeod\_\_RadiationSurfaceFactory

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

### 8.11.4.2 InputProcessor

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

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

## 8.11.5 Field Documentation

### 8.11.5.1 flat\_plate\_radiation\_factory

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

The factory to build flat plate thermal facets.

```
trick_units(-)
```

Definition at line 103 of file radiation\_surface\_factory.hh.

Referenced by RadiationSurfaceFactory().

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

- [radiation\\_surface\\_factory.hh](#)
- [radiation\\_surface\\_factory.cc](#)

## 8.12 jeod::RadiationThirdBody Class Reference

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

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



## Public Types

- enum [ShadowGeometry](#) { [Cylindrical](#) = 0, [Cyl](#) = 1, [Conical](#) = 2, [Con](#) = 3 }  
*distinguishes between conical and cylindrical shadowing*

## Public Member Functions

- [RadiationThirdBody](#) ()=default
- virtual [~RadiationThirdBody](#) ()=default
- [RadiationThirdBody](#) (const [RadiationThirdBody](#) &)=delete
- [RadiationThirdBody](#) & operator= (const [RadiationThirdBody](#) &)=delete
- virtual void [initialize](#) (DynManager \*dyn\_mgr\_ptr)  
*initializes the third body*
- void [calculate\\_shadow](#) ()  
*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 real\_time, RefFrame &veh\_struc\_frame)  
*Primary executable for handling the Third Body effects.*
- void [set\\_name](#) (std::string name\_in)  
*Setter for the name.*
- virtual void [accumulate\\_refl\\_flux](#) ([RadiationBaseFacet](#) \*veh\_surf\_elem, bool calculate\_forces)  
*To provide base class null implementation.*
- virtual void [accumulate\\_rad\\_flux](#) ([RadiationBaseFacet](#) \*veh\_surf\_elem, bool calculate\_forces)  
*To provide base class null implementation.*
- virtual bool [is\\_interactive](#) ()  
*Identifies this class as one that does not produce a radiation field.*
- bool [get\\_added\\_to\\_model](#) ()  
*Returns information on whether the body has already been used in one [RadiationPressure](#) model.*
- void [set\\_added\\_to\\_model](#) (bool value)  
*Sets the added\_to\_model value.*

## Data Fields

- [RadiationSource](#) \* [primary\\_source\\_ptr](#) {}  
*Pointer to the primary illumination source.*
- [ShadowGeometry](#) [shadow\\_geometry](#) {[Conical](#)}  
*Flag indicating cylindrical / conical shadow geometry.*
- bool [force\\_state\\_update](#) {true}  
*Flag to indicate whether to enforce an update to the third-body body-state.*
- std::string [name](#)  
*Name of planet.*
- bool [active](#) {true}  
*Model on/off flag.*
- double [update\\_interval](#) {}  
*Time interval between third body state updates.*
- double [radius](#) {}  
*Radius of [RadiationThirdBody](#).*
- double [output\\_flux](#) [3] {}  
*Calculated value.*
- RefFrame \* [local\\_frame\\_ptr](#) {}  
*Pointer to the local reference frame, for determining position of this object.*

## Protected Member Functions

- double [generate\\_alpha](#) (double rho\_adj, double delta)  
*generate\_alpha( rho\_adj, delta: generates an approximate alpha-value given a relative disk size and fraction of maximum possible eclipse*
- bool [test\\_for\\_state\\_update](#) (double time)  
*Tests for necessity of updating third body state, and calls appropriate update method (polymorphic) if needed.*
- virtual bool [update\\_third\\_body\\_state](#) ()  
*Updates the state of the Third Body.*

## Protected Attributes

- bool [added\\_to\\_model](#) {}  
*Flag to indicate whether body has already been added to a model.*
- bool [initialized](#) {}  
*Flag to indicate that initialization has been complete.*
- double [illum\\_factor](#) {1.0}  
*Value between 0 and 1 representing the fraction of possible flux that is actually incident after shadowing effects.*
- double [last\\_update\\_time](#) {}  
*Time at which the third body state was last updated.*
- Planet \* [planet\\_link](#) {}  
*Pointer to the associated Planet object.*
- double [r\\_plus](#) {}  
*The value (radius + R\_source)*
- double [r\\_minus](#) {}  
*The value (radius - R\_source)*
- double [r\\_ratio](#) {}  
*The ratio (radius / R\_source)*
- double [r\\_par](#) {}  
*Component of vector from [RadiationThirdBody](#) to vehicle that is parallel to vector from primary source to [RadiationThirdBody](#).*
- double [r\\_perp](#) {}  
*Component of vector from [RadiationThirdBody](#) to vehicle that is perpendicular to vector from primary source to [RadiationThirdBody](#).*
- double [third\\_to\\_cg\\_inrtl](#) [3] {}  
*Vector from [RadiationThirdBody](#) to vehicle center of gravity, in inertial RF.*
- double [source\\_to\\_third\\_inrtl](#) [3] {}  
*Vector from the primary source to the [RadiationThirdBody](#) in inertial RF.*
- double [d\\_source\\_to\\_third](#) {}  
*Magnitude of source\_to\_third.*
- double [source\\_to\\_third\\_hat\\_inrtl](#) [3] {}  
*Unit vector of source\_to\_third.*

## Friends

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

### 8.12.1 Detailed Description

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

Definition at line 84 of file radiation\_third\_body.hh.

### 8.12.2 Member Enumeration Documentation

#### 8.12.2.1 ShadowGeometry

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

distinguishes between conical and cylindrical shadowing

Enumerator

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

Definition at line 90 of file radiation\_third\_body.hh.

### 8.12.3 Constructor & Destructor Documentation

#### 8.12.3.1 RadiationThirdBody() [1/2]

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

#### 8.12.3.2 ~RadiationThirdBody()

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

#### 8.12.3.3 RadiationThirdBody() [2/2]

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

## 8.12.4 Member Function Documentation

### 8.12.4.1 accumulate\_rad\_flux()

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

To provide base class null implementation.

#### Parameters

<i>veh_surf_elem</i>	The facet of the vehicle.
<i>calculate_forces</i>	pass-through flag.

Definition at line 263 of file radiation\_third\_body.hh.

Referenced by jeod::RadiationBaseFacet::interact\_with\_third\_body().

### 8.12.4.2 accumulate\_refl\_flux()

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

To provide base class null implementation.

#### Parameters

<i>veh_surf_elem</i>	The facet of the vehicle.
<i>calculate_forces</i>	pass-through flag.

Definition at line 255 of file radiation\_third\_body.hh.

Referenced by jeod::RadiationBaseFacet::interact\_with\_third\_body().

### 8.12.4.3 calculate\_shadow()

```
void jeod::RadiationThirdBody::calculate_shadow ( )
```

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 198 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, jeod::RadiationSource::radius, radius, shadow\_geometry, jeod::RadiationSource::source\_to\_cg, source\_to\_third\_hat\_inrtl, source\_to\_third\_inrtl, and third\_to\_cg\_inrtl.

Referenced by process\_third\_body().

#### 8.12.4.4 convert\_shadow\_from\_int()

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

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

##### Parameters

in	<i>old_shadow_geometry</i>	integer representation of the shadow geometry enumeration.
----	----------------------------	--

Definition at line 396 of file radiation\_third\_body.cc.

References jeod::RadiationMessages::invalid\_setup\_error, and shadow\_geometry.

#### 8.12.4.5 generate\_alpha()

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

generate\_alpha( rho\_adj, delta: generates an approximate alpha-value given a relative disk size and fraction of maximum possible eclipse

##### Parameters

in	<i>rho_adj</i>	relative disk size
in	<i>delta</i>	fraction of eclipse achieved.

Definition at line 377 of file radiation\_third\_body.cc.

Referenced by calculate\_shadow().

#### 8.12.4.6 `get_added_to_model()`

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

Returns information on whether the body has already been used in one [RadiationPressure](#) model.

Protects against multiple use of one ThirdBody instance.

##### Returns

`added_to_model`

Definition at line 281 of file `radiation_third_body.hh`.

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

#### 8.12.4.7 `initialize()`

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

initializes the third body

##### Parameters

in	<i>dyn_mgr_ptr</i>	pointer to the dynamics manager
----	--------------------	---------------------------------

Definition at line 56 of file `radiation_third_body.cc`.

References `active`, `jeod::RadiationSource::inertial_frame_ptr`, `initialized`, `jeod::RadiationMessages::invalid_setup↵_error`, `local_frame_ptr`, `name`, `planet_link`, `primary_source_ptr`, `r_minus`, `r_plus`, `r_ratio`, `jeod::RadiationSource↵::radius`, and `radius`.

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

#### 8.12.4.8 `is_interactive()`

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

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

##### Returns

`false`

Definition at line 270 of file `radiation_third_body.hh`.

## 8.12.4.9 operator=()

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

## 8.12.4.10 process\_third\_body()

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

Primary executable for handling the Third Body effects.

## Returns

shading fraction

## Parameters

in	<i>real_time</i>	Current time. Units: s
in	<i>veh_struc_frame</i>	Unused.

Definition at line 417 of file radiation\_third\_body.cc.

References active, calculate\_shadow(), illum\_factor, initialized, jeod::RadiationMessages::invalid\_setup\_error, name, jeod::RadiationMessages::operational\_setup\_error, and test\_for\_state\_update().

## 8.12.4.11 set\_added\_to\_model()

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

Sets the added\_to\_model value.

## Parameters

<i>value</i>	New value
--------------	-----------

Definition at line 290 of file radiation\_third\_body.hh.

Referenced by jeod::RadiationPressure::add\_third\_body().

**8.12.4.12 set\_name()**

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

Setter for the name.

Definition at line 245 of file radiation\_third\_body.hh.

**8.12.4.13 test\_for\_state\_update()**

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

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

**Parameters**

in	<i>real_time</i>	Current time Units: s
----	------------------	--------------------------

Definition at line 464 of file radiation\_third\_body.cc.

References force\_state\_update, last\_update\_time, update\_interval, and update\_third\_body\_state().

Referenced by process\_third\_body().

**8.12.4.14 update\_third\_body\_state()**

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

Updates the state of the Third Body.

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

Definition at line 486 of file radiation\_third\_body.cc.

References active, d\_source\_to\_third, jeod::RadiationSource::inertial\_frame\_ptr, jeod::RadiationMessages↔::invalid\_setup\_error, local\_frame\_ptr, name, primary\_source\_ptr, source\_to\_third\_hat\_inrtl, and source\_to\_↔third\_inrtl.

Referenced by test\_for\_state\_update().

**8.12.5 Friends And Related Function Documentation**



### 8.12.5.1 init\_attrjeod\_\_RadiationThirdBody

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

### 8.12.5.2 InputProcessor

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

Definition at line 86 of file radiation\_third\_body.hh.

## 8.12.6 Field Documentation

### 8.12.6.1 active

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

Model on/off flag.

trick\_units(-)

Definition at line 123 of file radiation\_third\_body.hh.

Referenced by jeod::RadiationPressure::add\_third\_body(), calculate\_shadow(), initialize(), process\_third\_body(), and update\_third\_body\_state().

### 8.12.6.2 added\_to\_model

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

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

trick\_units(-)

Definition at line 153 of file radiation\_third\_body.hh.

#### 8.12.6.3 d\_source\_to\_third

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

Magnitude of source\_to\_third.

trick\_units(m)

Definition at line 220 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), and update\_third\_body\_state().

#### 8.12.6.4 force\_state\_update

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

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

Default: true Self-resets to false after each use.trick\_units(-)

Definition at line 113 of file radiation\_third\_body.hh.

Referenced by test\_for\_state\_update().

#### 8.12.6.5 illum\_factor

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

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

trick\_units(-)

Definition at line 164 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), and process\_third\_body().

#### 8.12.6.6 initialized

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

Flag to indicate that initialization has been complete.

trick\_units(-)

Definition at line 158 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), initialize(), and process\_third\_body().

#### 8.12.6.7 last\_update\_time

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

Time at which the third body state was last updated.

trick\_units(s)

Definition at line 169 of file radiation\_third\_body.hh.

Referenced by test\_for\_state\_update().

#### 8.12.6.8 local\_frame\_ptr

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

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

trick\_units(-)

Definition at line 146 of file radiation\_third\_body.hh.

Referenced by initialize(), and update\_third\_body\_state().

#### 8.12.6.9 name

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

Name of planet.

trick\_units(-)

Definition at line 118 of file radiation\_third\_body.hh.

Referenced by jeod::RadiationPressure::add\_third\_body(), calculate\_shadow(), initialize(), process\_third\_body(), and update\_third\_body\_state().

#### 8.12.6.10 output\_flux

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

Calculated value.

Set only in derived class RadiationReflectingThirdBody.trick\_units(-)

Definition at line 140 of file radiation\_third\_body.hh.

#### 8.12.6.11 planet\_link

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

Pointer to the associated Planet object.

trick\_units(-)

Definition at line 175 of file radiation\_third\_body.hh.

Referenced by initialize().

#### 8.12.6.12 primary\_source\_ptr

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

Pointer to the primary illumination source.

trick\_units(-)

Definition at line 101 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), initialize(), and update\_third\_body\_state().

#### 8.12.6.13 r\_minus

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

The value (radius - R\_source)

trick\_units(-)

Definition at line 185 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), and initialize().

#### 8.12.6.14 r\_par

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

Component of vector from [RadiationThirdBody](#) to vehicle that is parallel to vector from primary source to [RadiationThirdBody](#).

trick\_units(m)

Definition at line 197 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow().

#### 8.12.6.15 r\_perp

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

Component of vector from [RadiationThirdBody](#) to vehicle that is perpendicular to vector from primary source to [RadiationThirdBody](#).

trick\_units(-)

Definition at line 203 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow().

#### 8.12.6.16 r\_plus

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

The value (radius + R\_source)

trick\_units(-)

Definition at line 180 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), and initialize().

#### 8.12.6.17 r\_ratio

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

The ratio (radius / R\_source)

trick\_units(-)

Definition at line 190 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), and initialize().

#### 8.12.6.18 radius

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

Radius of [RadiationThirdBody](#).

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

Definition at line 134 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), and initialize().

**8.12.6.19 shadow\_geometry**

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

Flag indicating cylindrical / conical shadow geometry.

trick\_units(-)

Definition at line 106 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), and convert\_shadow\_from\_int().

**8.12.6.20 source\_to\_third\_hat\_inrtl**

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

Unit vector of source\_to\_third.

trick\_units(-)

Definition at line 225 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), and update\_third\_body\_state().

**8.12.6.21 source\_to\_third\_inrtl**

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

Vector from the primary source to the [RadiationThirdBody](#) in inertial RF.

trick\_units(m)

Definition at line 215 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow(), and update\_third\_body\_state().

**8.12.6.22 third\_to\_cg\_inrtl**

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

Vector from [RadiationThirdBody](#) to vehicle center of gravity, in inertial RF.

trick\_units(m)

Definition at line 209 of file radiation\_third\_body.hh.

Referenced by calculate\_shadow().

**8.12.6.23 update\_interval**

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

Time interval between third body state updates.

trick\_units(s)

Definition at line 128 of file radiation\_third\_body.hh.

Referenced by test\_for\_state\_update().

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

- [radiation\\_third\\_body.hh](#)
- [radiation\\_third\\_body.cc](#)

## Chapter 9

# File Documentation

### 9.1 flat\_plate\_radiation\_facet.cc File Reference

Define member functions for class FlatPlateRadiationFacet.

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

#### Namespaces

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

#### 9.1.1 Detailed Description

Define member functions for class FlatPlateRadiationFacet.

### 9.2 flat\_plate\_radiation\_facet.hh File Reference

Individual facets for use with rad environment interaction models.

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

#### Data Structures

- class [jeod::FlatPlateRadiationFacet](#)  
*A flat plate facet to be used for radiation interaction.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.2.1 Detailed Description

Individual facets for use with rad environment interaction models.

## 9.3 flat\_plate\_radiation\_factory.cc File Reference

Factory that creates a FlatPlateRadiationFacet, from a facet model.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.3.1 Detailed Description

Factory that creates a FlatPlateRadiationFacet, from a facet model.

## 9.4 flat\_plate\_radiation\_factory.hh File Reference

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

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

## Data Structures

- class [jeod::FlatPlateRadiationFactory](#)

*The factory for building flat plate radiation facets.*



## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.4.1 Detailed Description

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

## 9.5 radiation\_base\_facet.cc File Reference

Define member functions for class RadiationBaseFacet.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.5.1 Detailed Description

Define member functions for class RadiationBaseFacet.

## 9.6 radiation\_base\_facet.hh File Reference

Individual facets for use with radiation environment interaction models.

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

## Data Structures

- class [jeod::RadiationBaseFacet](#)

*Generic type of facet for radiation pressure.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.6.1 Detailed Description

Individual facets for use with radiation environment interaction models.

## 9.7 radiation\_default\_surface.cc File Reference

Default surface for use with Radiation Pressure interaction model.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.7.1 Detailed Description

Default surface for use with Radiation Pressure interaction model.

## 9.8 radiation\_default\_surface.hh File Reference

Individual facets for use with radiation environment interaction models.

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

## Data Structures

- class [jeod::RadiationDefaultSurface](#)

*The default spherical surface for radiation pressure.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.8.1 Detailed Description

Individual facets for use with radiation environment interaction models.

## 9.9 radiation\_facet.cc File Reference

Individual facets for use with Radiation Pressure interaction models.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.9.1 Detailed Description

Individual facets for use with Radiation Pressure interaction models.

## 9.10 radiation\_facet.hh File Reference

Individual facets for use with radiation environment interaction models.

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

## Data Structures

- class [jeod::RadiationFacet](#)

*Generic type of facet for radiation pressure.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.10.1 Detailed Description

Individual facets for use with radiation environment interaction models.

## 9.11 radiation\_messages.cc File Reference

Implement the class RadiationMessages.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

## Macros

- `#define` [PATH](#) "interactions/radiation\_pressure/"

### 9.11.1 Detailed Description

Implement the class RadiationMessages.

## 9.12 radiation\_messages.hh File Reference

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

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

## Data Structures

- class [jeod::RadiationMessages](#)

*Provides error messages.*

## Namespaces

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

### 9.12.1 Detailed Description

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

## 9.13 radiation\_params.hh File Reference

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

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

## Data Structures

- class [jeod::RadiationParams](#)  
*Provides a parameter list to each facet, based on the facet material properties.*

## Namespaces

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

### 9.13.1 Detailed Description

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

## 9.14 radiation\_pressure.cc File Reference

Calculation of force and torque due to radiation pressure.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.14.1 Detailed Description

Calculation of force and torque due to radiation pressure.

## 9.15 radiation\_pressure.hh File Reference

Radiation pressure top-level definition.

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

## Data Structures

- class [jeod::RadiationPressure](#)

*Radiation pressure top-level definition.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.15.1 Detailed Description

Radiation pressure top-level definition.

## 9.16 radiation\_pressure\_\_default\_surface.cc File Reference

Calculation of force and torque due to radiation pressure.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.16.1 Detailed Description

Calculation of force and torque due to radiation pressure.

## 9.17 radiation\_pressure\_\_surface\_model.cc File Reference

Calculation of force and torque due to radiation pressure.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.17.1 Detailed Description

Calculation of force and torque due to radiation pressure.

## 9.18 radiation\_source.cc File Reference

Definition of methods associated with Primary Sources.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.18.1 Detailed Description

Definition of methods associated with Primary Sources.

## 9.19 radiation\_source.hh File Reference

Radiation pressure parameter and variable definitions.

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

## Data Structures

- class [jeod::RadiationSource](#)

*Provides information on the source of the incident radiation.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.19.1 Detailed Description

Radiation pressure parameter and variable definitions.

## 9.20 radiation\_surface.cc File Reference

Vehicle surface model for Radiation Pressure model.

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



## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.20.1 Detailed Description

Vehicle surface model for Radiation Pressure model.

## 9.21 radiation\_surface.hh File Reference

Vehicle surface model for general environment interaction models.

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

## Data Structures

- class [jeod::RadiationSurface](#)

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.21.1 Detailed Description

Vehicle surface model for general environment interaction models.

## 9.22 radiation\_surface\_factory.cc File Reference

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

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "../include/radiation_messages.hh"
#include "../include/radiation_params.hh"
#include "../include/radiation_surface_factory.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.22.1 Detailed Description

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

## 9.23 radiation\_surface\_factory.hh File Reference

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

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

## Data Structures

- class [jeod::RadiationSurfaceFactory](#)

*The factory for creating Radiation Surfaces.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.23.1 Detailed Description

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

## 9.24 radiation\_third\_body.cc File Reference

Provides the functionality associated with RadiationThirdBodys.

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.24.1 Detailed Description

Provides the functionality associated with RadiationThirdBodys.

## 9.25 radiation\_third\_body.hh File Reference

Define the class RadiationThirdBody.

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

## Data Structures

- class [jeod::RadiationThirdBody](#)

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

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.25.1 Detailed Description

Define the class RadiationThirdBody.



# Index

- ~FlatPlateRadiationFacet
  - jeod::FlatPlateRadiationFacet, [18](#)
- ~FlatPlateRadiationFactory
  - jeod::FlatPlateRadiationFactory, [22](#)
- ~RadiationBaseFacet
  - jeod::RadiationBaseFacet, [26](#)
- ~RadiationDefaultSurface
  - jeod::RadiationDefaultSurface, [33](#)
- ~RadiationFacet
  - jeod::RadiationFacet, [38](#)
- ~RadiationParams
  - jeod::RadiationParams, [46](#)
- ~RadiationPressure
  - jeod::RadiationPressure, [50](#)
- ~RadiationSource
  - jeod::RadiationSource, [62](#)
- ~RadiationSurface
  - jeod::RadiationSurface, [71](#)
- ~RadiationSurfaceFactory
  - jeod::RadiationSurfaceFactory, [79](#)
- ~RadiationThirdBody
  - jeod::RadiationThirdBody, [83](#)
- accumulate\_rad\_flux
  - jeod::RadiationThirdBody, [84](#)
- accumulate\_refl\_flux
  - jeod::RadiationThirdBody, [84](#)
- accumulate\_thermal\_sources
  - jeod::RadiationSurface, [71](#)
- active
  - jeod::RadiationPressure, [56](#)
  - jeod::RadiationThirdBody, [89](#)
- add\_facet\_params
  - jeod::RadiationSurfaceFactory, [79](#)
- add\_thermal\_integrator\_to
  - jeod::RadiationDefaultSurface, [33](#)
- add\_thermal\_integrators\_to
  - jeod::RadiationSurface, [71](#)
- add\_third\_body
  - jeod::RadiationPressure, [50](#)
- added\_to\_model
  - jeod::RadiationThirdBody, [89](#)
- albedo
  - jeod::RadiationBaseFacet, [28](#)
  - jeod::RadiationParams, [47](#)
- albedo\_IR
  - jeod::RadiationBaseFacet, [28](#)
- albedo\_vis
  - jeod::RadiationBaseFacet, [29](#)
- allocate\_array
  - jeod::RadiationSurface, [72](#)
- allocate\_interaction\_facet
  - jeod::RadiationSurface, [72](#)
- areaxflux
  - jeod::RadiationBaseFacet, [29](#)
- areaxflux\_e
  - jeod::RadiationBaseFacet, [29](#)
- bodies\_active
  - jeod::RadiationSource, [64](#)
- calculate\_flux
  - jeod::RadiationSource, [62](#)
- calculate\_forces
  - jeod::RadiationPressure, [56](#)
- calculate\_shadow
  - jeod::RadiationThirdBody, [84](#)
- center\_pressure
  - jeod::RadiationFacet, [40](#)
- convert\_shadow\_from\_int
  - jeod::RadiationThirdBody, [85](#)
- create\_facet
  - jeod::FlatPlateRadiationFactory, [23](#)
- crot\_to\_cp
  - jeod::RadiationFacet, [41](#)
- cx\_area
  - jeod::RadiationBaseFacet, [29](#)
- d\_source\_to\_cg
  - jeod::RadiationSource, [64](#)
- d\_source\_to\_third
  - jeod::RadiationThirdBody, [89](#)
- default\_surface\_ptr
  - jeod::RadiationPressure, [56](#)
- define\_facet
  - jeod::FlatPlateRadiationFacet, [18](#)
- define\_facet\_core
  - jeod::RadiationFacet, [38](#)
- diffuse
  - jeod::RadiationBaseFacet, [30](#)
  - jeod::RadiationParams, [47](#)
- dyn\_manager\_ptr
  - jeod::RadiationPressure, [57](#)
- equalize\_absorption\_emission
  - jeod::RadiationSurface, [72](#)
- F\_absorption
  - jeod::RadiationBaseFacet, [30](#)
- F\_diffuse
  - jeod::RadiationBaseFacet, [30](#)

- F\_emission
  - jeod::RadiationBaseFacet, 31
- F\_specular
  - jeod::RadiationBaseFacet, 31
- facets
  - jeod::RadiationSurface, 75
- find\_third\_body
  - jeod::RadiationPressure, 51
- flat\_plate\_radiation\_facet.cc, 95
- flat\_plate\_radiation\_facet.hh, 95
- flat\_plate\_radiation\_factory
  - jeod::RadiationSurfaceFactory, 80
- flat\_plate\_radiation\_factory.cc, 96
- flat\_plate\_radiation\_factory.hh, 96
- FlatPlateRadiationFacet
  - jeod::FlatPlateRadiationFacet, 18
- FlatPlateRadiationFactory
  - jeod::FlatPlateRadiationFactory, 22, 23
- flux\_hat
  - jeod::RadiationSource, 64
- flux\_inertial
  - jeod::RadiationSource, 64
- flux\_mag
  - jeod::RadiationSource, 65
- flux\_struc
  - jeod::RadiationSource, 65
- flux\_struc\_hat
  - jeod::RadiationSource, 65
- force
  - jeod::RadiationPressure, 57
  - jeod::RadiationSurface, 76
- force\_state\_update
  - jeod::RadiationThirdBody, 90
- generate\_alpha
  - jeod::RadiationThirdBody, 85
- get\_added\_to\_model
  - jeod::RadiationThirdBody, 85
- get\_thermal\_integrator
  - jeod::RadiationFacet, 39
- ii\_facet
  - jeod::RadiationSurface, 76
- illum\_factor
  - jeod::RadiationPressure, 57
  - jeod::RadiationThirdBody, 90
- incident\_flux\_hat
  - jeod::FlatPlateRadiationFacet, 21
- incident\_radiation
  - jeod::FlatPlateRadiationFacet, 19
  - jeod::RadiationBaseFacet, 26
  - jeod::RadiationDefaultSurface, 34
  - jeod::RadiationSurface, 73
- include\_conduction
  - jeod::RadiationSurface, 76
- incomplete\_setup\_error
  - jeod::RadiationMessages, 43
- inertial\_cg
  - jeod::RadiationSource, 65
- inertial\_frame\_ptr
  - jeod::RadiationSource, 66
- init\_attrjeod\_\_FlatPlateRadiationFacet
  - jeod::FlatPlateRadiationFacet, 20
- init\_attrjeod\_\_FlatPlateRadiationFactory
  - jeod::FlatPlateRadiationFactory, 24
- init\_attrjeod\_\_RadiationBaseFacet
  - jeod::RadiationBaseFacet, 28
- init\_attrjeod\_\_RadiationDefaultSurface
  - jeod::RadiationDefaultSurface, 35
- init\_attrjeod\_\_RadiationFacet
  - jeod::RadiationFacet, 40
- init\_attrjeod\_\_RadiationMessages
  - jeod::RadiationMessages, 43
- init\_attrjeod\_\_RadiationParams
  - jeod::RadiationParams, 46
- init\_attrjeod\_\_RadiationPressure
  - jeod::RadiationPressure, 55
- init\_attrjeod\_\_RadiationSource
  - jeod::RadiationSource, 63
- init\_attrjeod\_\_RadiationSurface
  - jeod::RadiationSurface, 75
- init\_attrjeod\_\_RadiationSurfaceFactory
  - jeod::RadiationSurfaceFactory, 80
- init\_attrjeod\_\_RadiationThirdBody
  - jeod::RadiationThirdBody, 88
- initialize
  - jeod::RadiationBaseFacet, 27
  - jeod::RadiationDefaultSurface, 34
  - jeod::RadiationFacet, 39
  - jeod::RadiationPressure, 51, 52
  - jeod::RadiationSource, 63
  - jeod::RadiationSurface, 73
  - jeod::RadiationThirdBody, 86
- initialize\_environment
  - jeod::RadiationPressure, 52
- initialize\_geom
  - jeod::FlatPlateRadiationFacet, 19
  - jeod::RadiationFacet, 39
- initialize\_runtime\_values
  - jeod::RadiationBaseFacet, 27
  - jeod::RadiationSurface, 74
- initialized
  - jeod::RadiationPressure, 57
  - jeod::RadiationThirdBody, 90
- InputProcessor
  - jeod::FlatPlateRadiationFacet, 20
  - jeod::FlatPlateRadiationFactory, 24
  - jeod::RadiationBaseFacet, 28
  - jeod::RadiationDefaultSurface, 35
  - jeod::RadiationFacet, 40
  - jeod::RadiationMessages, 43
  - jeod::RadiationParams, 47
  - jeod::RadiationPressure, 56
  - jeod::RadiationSource, 63
  - jeod::RadiationSurface, 75
  - jeod::RadiationSurfaceFactory, 80
  - jeod::RadiationThirdBody, 89

- interact\_with\_third\_body
  - jeod::RadiationBaseFacet, 27
  - jeod::RadiationSurface, 74
- Interactions, 12
- invalid\_function\_call
  - jeod::RadiationMessages, 43
- invalid\_setup\_error
  - jeod::RadiationMessages, 44
- is\_correct\_factory
  - jeod::FlatPlateRadiationFactory, 23
- is\_interactive
  - jeod::RadiationThirdBody, 86
- jeod, 15
- jeod::FlatPlateRadiationFacet, 17
  - ~FlatPlateRadiationFacet, 18
  - define\_facet, 18
  - FlatPlateRadiationFacet, 18
  - incident\_flux\_hat, 21
  - incident\_radiation, 19
  - init\_attrjeod\_\_FlatPlateRadiationFacet, 20
  - initialize\_geom, 19
  - InputProcessor, 20
  - normal, 21
  - operator=, 20
  - radiation\_pressure, 20
  - sin\_theta, 21
- jeod::FlatPlateRadiationFactory, 22
  - ~FlatPlateRadiationFactory, 22
  - create\_facet, 23
  - FlatPlateRadiationFactory, 22, 23
  - init\_attrjeod\_\_FlatPlateRadiationFactory, 24
  - InputProcessor, 24
  - is\_correct\_factory, 23
  - operator=, 24
- jeod::RadiationBaseFacet, 24
  - ~RadiationBaseFacet, 26
  - albedo, 28
  - albedo\_IR, 28
  - albedo\_vis, 29
  - areaxflux, 29
  - areaxflux\_e, 29
  - cx\_area, 29
  - diffuse, 30
  - F\_absorption, 30
  - F\_diffuse, 30
  - F\_emission, 31
  - F\_specular, 31
  - incident\_radiation, 26
  - init\_attrjeod\_\_RadiationBaseFacet, 28
  - initialize, 27
  - initialize\_runtime\_values, 27
  - InputProcessor, 28
  - interact\_with\_third\_body, 27
  - operator=, 28
  - RadiationBaseFacet, 26
  - speed\_of\_light, 31
  - thermal, 31
- jeod::RadiationDefaultSurface, 32
  - ~RadiationDefaultSurface, 33
  - add\_thermal\_integrator\_to, 33
  - incident\_radiation, 34
  - init\_attrjeod\_\_RadiationDefaultSurface, 35
  - initialize, 34
  - InputProcessor, 35
  - name, 36
  - operator=, 35
  - rad\_coeff, 36
  - RadiationDefaultSurface, 33
  - set\_name, 35
  - surface\_area, 36
  - temperature, 36
  - thermal\_update, 35
- jeod::RadiationFacet, 37
  - ~RadiationFacet, 38
  - center\_pressure, 40
  - crot\_to\_cp, 41
  - define\_facet\_core, 38
  - get\_thermal\_integrator, 39
  - init\_attrjeod\_\_RadiationFacet, 40
  - initialize, 39
  - initialize\_geom, 39
  - InputProcessor, 40
  - operator=, 40
  - radiation\_pressure, 40
  - RadiationFacet, 38
  - two\_thirds, 41
- jeod::RadiationMessages, 42
  - incomplete\_setup\_error, 43
  - init\_attrjeod\_\_RadiationMessages, 43
  - InputProcessor, 43
  - invalid\_function\_call, 43
  - invalid\_setup\_error, 44
  - operational\_setup\_error, 44
  - operator=, 43
  - RadiationMessages, 42
  - unknown\_numerical\_error, 44
- jeod::RadiationParams, 45
  - ~RadiationParams, 46
  - albedo, 47
  - diffuse, 47
  - init\_attrjeod\_\_RadiationParams, 46
  - InputProcessor, 47
  - operator=, 46
  - RadiationParams, 46
  - thermal, 47
- jeod::RadiationPressure, 48
  - ~RadiationPressure, 50
  - active, 56
  - add\_third\_body, 50
  - calculate\_forces, 56
  - default\_surface\_ptr, 56
  - dyn\_manager\_ptr, 57
  - find\_third\_body, 51
  - force, 57
  - illum\_factor, 57
  - init\_attrjeod\_\_RadiationPressure, 55

initialize, 51, 52  
 initialize\_environment, 52  
 initialized, 57  
 InputProcessor, 56  
 num\_third\_bodies, 58  
 operator=, 53  
 RadiationPressure, 50  
 set\_calculate\_forces, 53  
 set\_third\_body\_active, 53  
 set\_third\_body\_inactive, 53  
 source, 58  
 surface\_ptr, 58  
 thermal, 58  
 third\_bodies, 59  
 third\_bodies\_active, 59  
 third\_body\_adjustments, 54  
 torque, 59  
 update, 54  
 update\_default\_surface, 55  
 update\_facet\_surface, 55  
 jeod::RadiationSource, 60  
   ~RadiationSource, 62  
   bodies\_active, 64  
   calculate\_flux, 62  
   d\_source\_to\_cg, 64  
   flux\_hat, 64  
   flux\_inertial, 64  
   flux\_mag, 65  
   flux\_struc, 65  
   flux\_struc\_hat, 65  
   inertial\_cg, 65  
   inertial\_frame\_ptr, 66  
   init\_attrjeod\_\_RadiationSource, 63  
   initialize, 63  
   InputProcessor, 63  
   luminosity, 66  
   multiple\_shadow\_bodies, 66  
   name, 66  
   num\_bodies, 67  
   OldShadowGeometry, 61  
   operator=, 63  
   RadiationSource, 62  
   radius, 67  
   set\_name, 63  
   shadow\_geometry, 67  
   solar\_luminosity, 67  
   solar\_radius, 68  
   source\_to\_cg, 68  
   source\_to\_struc\_origin, 68  
   third\_body, 68  
 jeod::RadiationSurface, 69  
   ~RadiationSurface, 71  
   accumulate\_thermal\_sources, 71  
   add\_thermal\_integrators\_to, 71  
   allocate\_array, 72  
   allocate\_interaction\_facet, 72  
   equalize\_absorption\_emission, 72  
   facets, 75  
   force, 76  
   ii\_facet, 76  
   incident\_radiation, 73  
   include\_conduction, 76  
   init\_attrjeod\_\_RadiationSurface, 75  
   initialize, 73  
   initialize\_runtime\_values, 74  
   InputProcessor, 75  
   interact\_with\_third\_body, 74  
   num\_facets, 76  
   operator=, 74  
   radiation\_pressure, 74  
   RadiationSurface, 70, 71  
   thermal\_conduction, 77  
   thermal\_integrator, 75  
   torque, 77  
 jeod::RadiationSurfaceFactory, 78  
   ~RadiationSurfaceFactory, 79  
   add\_facet\_params, 79  
   flat\_plate\_radiation\_factory, 80  
   init\_attrjeod\_\_RadiationSurfaceFactory, 80  
   InputProcessor, 80  
   operator=, 79  
   RadiationSurfaceFactory, 78, 79  
 jeod::RadiationThirdBody, 80  
   ~RadiationThirdBody, 83  
   accumulate\_rad\_flux, 84  
   accumulate\_refl\_flux, 84  
   active, 89  
   added\_to\_model, 89  
   calculate\_shadow, 84  
   convert\_shadow\_from\_int, 85  
   d\_source\_to\_third, 89  
   force\_state\_update, 90  
   generate\_alpha, 85  
   get\_added\_to\_model, 85  
   illum\_factor, 90  
   init\_attrjeod\_\_RadiationThirdBody, 88  
   initialize, 86  
   initialized, 90  
   InputProcessor, 89  
   is\_interactive, 86  
   last\_update\_time, 90  
   local\_frame\_ptr, 91  
   name, 91  
   operator=, 86  
   output\_flux, 91  
   planet\_link, 91  
   primary\_source\_ptr, 92  
   process\_third\_body, 87  
   r\_minus, 92  
   r\_par, 92  
   r\_perp, 92  
   r\_plus, 93  
   r\_ratio, 93  
   RadiationThirdBody, 83  
   radius, 93  
   set\_added\_to\_model, 87



- set\_name, [87](#)
  - shadow\_geometry, [93](#)
  - ShadowGeometry, [83](#)
  - source\_to\_third\_hat\_inrtl, [94](#)
  - source\_to\_third\_inrtl, [94](#)
  - test\_for\_state\_update, [88](#)
  - third\_to\_cg\_inrtl, [94](#)
  - update\_interval, [94](#)
  - update\_third\_body\_state, [88](#)
- last\_update\_time
  - jeod::RadiationThirdBody, [90](#)
- local\_frame\_ptr
  - jeod::RadiationThirdBody, [91](#)
- luminosity
  - jeod::RadiationSource, [66](#)
- Models, [11](#)
- multiple\_shadow\_bodies
  - jeod::RadiationSource, [66](#)
- name
  - jeod::RadiationDefaultSurface, [36](#)
  - jeod::RadiationSource, [66](#)
  - jeod::RadiationThirdBody, [91](#)
- normal
  - jeod::FlatPlateRadiationFacet, [21](#)
- num\_bodies
  - jeod::RadiationSource, [67](#)
- num\_facets
  - jeod::RadiationSurface, [76](#)
- num\_third\_bodies
  - jeod::RadiationPressure, [58](#)
- OldShadowGeometry
  - jeod::RadiationSource, [61](#)
- operational\_setup\_error
  - jeod::RadiationMessages, [44](#)
- operator=
  - jeod::FlatPlateRadiationFacet, [20](#)
  - jeod::FlatPlateRadiationFactory, [24](#)
  - jeod::RadiationBaseFacet, [28](#)
  - jeod::RadiationDefaultSurface, [35](#)
  - jeod::RadiationFacet, [40](#)
  - jeod::RadiationMessages, [43](#)
  - jeod::RadiationParams, [46](#)
  - jeod::RadiationPressure, [53](#)
  - jeod::RadiationSource, [63](#)
  - jeod::RadiationSurface, [74](#)
  - jeod::RadiationSurfaceFactory, [79](#)
  - jeod::RadiationThirdBody, [86](#)
- output\_flux
  - jeod::RadiationThirdBody, [91](#)
- PATH
  - RadiationPressure, [14](#)
- planet\_link
  - jeod::RadiationThirdBody, [91](#)
- primary\_source\_ptr
  - jeod::RadiationThirdBody, [92](#)
- process\_third\_body
  - jeod::RadiationThirdBody, [87](#)
- r\_minus
  - jeod::RadiationThirdBody, [92](#)
- r\_par
  - jeod::RadiationThirdBody, [92](#)
- r\_perp
  - jeod::RadiationThirdBody, [92](#)
- r\_plus
  - jeod::RadiationThirdBody, [93](#)
- r\_ratio
  - jeod::RadiationThirdBody, [93](#)
- rad\_coeff
  - jeod::RadiationDefaultSurface, [36](#)
- radiation\_base\_facet.cc, [97](#)
- radiation\_base\_facet.hh, [97](#)
- radiation\_default\_surface.cc, [98](#)
- radiation\_default\_surface.hh, [98](#)
- radiation\_facet.cc, [99](#)
- radiation\_facet.hh, [99](#)
- radiation\_messages.cc, [100](#)
- radiation\_messages.hh, [100](#)
- radiation\_params.hh, [101](#)
- radiation\_pressure
  - jeod::FlatPlateRadiationFacet, [20](#)
  - jeod::RadiationFacet, [40](#)
  - jeod::RadiationSurface, [74](#)
- radiation\_pressure.cc, [101](#)
- radiation\_pressure.hh, [102](#)
- radiation\_pressure\_\_default\_surface.cc, [102](#)
- radiation\_pressure\_\_surface\_model.cc, [103](#)
- radiation\_source.cc, [103](#)
- radiation\_source.hh, [104](#)
- radiation\_surface.cc, [104](#)
- radiation\_surface.hh, [105](#)
- radiation\_surface\_factory.cc, [105](#)
- radiation\_surface\_factory.hh, [106](#)
- radiation\_third\_body.cc, [106](#)
- radiation\_third\_body.hh, [107](#)
- RadiationBaseFacet
  - jeod::RadiationBaseFacet, [26](#)
- RadiationDefaultSurface
  - jeod::RadiationDefaultSurface, [33](#)
- RadiationFacet
  - jeod::RadiationFacet, [38](#)
- RadiationMessages
  - jeod::RadiationMessages, [42](#)
- RadiationParams
  - jeod::RadiationParams, [46](#)
- RadiationPressure, [13](#)
  - jeod::RadiationPressure, [50](#)
  - PATH, [14](#)
- RadiationSource
  - jeod::RadiationSource, [62](#)
- RadiationSurface
  - jeod::RadiationSurface, [70, 71](#)
- RadiationSurfaceFactory

- jeod::RadiationSurfaceFactory, [78](#), [79](#)
- RadiationThirdBody
  - jeod::RadiationThirdBody, [83](#)
- radius
  - jeod::RadiationSource, [67](#)
  - jeod::RadiationThirdBody, [93](#)
- set\_added\_to\_model
  - jeod::RadiationThirdBody, [87](#)
- set\_calculate\_forces
  - jeod::RadiationPressure, [53](#)
- set\_name
  - jeod::RadiationDefaultSurface, [35](#)
  - jeod::RadiationSource, [63](#)
  - jeod::RadiationThirdBody, [87](#)
- set\_third\_body\_active
  - jeod::RadiationPressure, [53](#)
- set\_third\_body\_inactive
  - jeod::RadiationPressure, [53](#)
- shadow\_geometry
  - jeod::RadiationSource, [67](#)
  - jeod::RadiationThirdBody, [93](#)
- ShadowGeometry
  - jeod::RadiationThirdBody, [83](#)
- sin\_theta
  - jeod::FlatPlateRadiationFacet, [21](#)
- solar\_luminosity
  - jeod::RadiationSource, [67](#)
- solar\_radius
  - jeod::RadiationSource, [68](#)
- source
  - jeod::RadiationPressure, [58](#)
- source\_to\_cg
  - jeod::RadiationSource, [68](#)
- source\_to\_struc\_origin
  - jeod::RadiationSource, [68](#)
- source\_to\_third\_hat\_inrtl
  - jeod::RadiationThirdBody, [94](#)
- source\_to\_third\_inrtl
  - jeod::RadiationThirdBody, [94](#)
- speed\_of\_light
  - jeod::RadiationBaseFacet, [31](#)
- surface\_area
  - jeod::RadiationDefaultSurface, [36](#)
- surface\_ptr
  - jeod::RadiationPressure, [58](#)
- temperature
  - jeod::RadiationDefaultSurface, [36](#)
- test\_for\_state\_update
  - jeod::RadiationThirdBody, [88](#)
- thermal
  - jeod::RadiationBaseFacet, [31](#)
  - jeod::RadiationParams, [47](#)
  - jeod::RadiationPressure, [58](#)
- thermal\_conduction
  - jeod::RadiationSurface, [77](#)
- thermal\_integrator
  - jeod::RadiationSurface, [75](#)
- thermal\_update
  - jeod::RadiationDefaultSurface, [35](#)
- third\_bodies
  - jeod::RadiationPressure, [59](#)
- third\_bodies\_active
  - jeod::RadiationPressure, [59](#)
- third\_body
  - jeod::RadiationSource, [68](#)
- third\_body\_adjustments
  - jeod::RadiationPressure, [54](#)
- third\_to\_cg\_inrtl
  - jeod::RadiationThirdBody, [94](#)
- torque
  - jeod::RadiationPressure, [59](#)
  - jeod::RadiationSurface, [77](#)
- two\_thirds
  - jeod::RadiationFacet, [41](#)
- unknown\_numerical\_error
  - jeod::RadiationMessages, [44](#)
- update
  - jeod::RadiationPressure, [54](#)
- update\_default\_surface
  - jeod::RadiationPressure, [55](#)
- update\_facet\_surface
  - jeod::RadiationPressure, [55](#)
- update\_interval
  - jeod::RadiationThirdBody, [94](#)
- update\_third\_body\_state
  - jeod::RadiationThirdBody, [88](#)