

BodyActionModel

5.0

Generated by Doxygen 1.8.14

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	9
5.1	File List	9
6	Module Documentation	13
6.1	Models	13
6.1.1	Detailed Description	13
6.2	Dynamics	14
6.2.1	Detailed Description	14
6.3	BodyAction	15
6.3.1	Detailed Description	17
6.3.2	Macro Definition Documentation	17
6.3.2.1	PATH	17

7 Namespace Documentation	19
7.1 jeod Namespace Reference	19
7.1.1 Detailed Description	20
8 Data Structure Documentation	21
8.1 jeod::BodyAction Class Reference	21
8.1.1 Detailed Description	22
8.1.2 Constructor & Destructor Documentation	23
8.1.2.1 BodyAction() [1/2]	23
8.1.2.2 BodyAction() [2/2]	23
8.1.2.3 ~BodyAction()	23
8.1.3 Member Function Documentation	23
8.1.3.1 apply()	23
8.1.3.2 get_identifier()	24
8.1.3.3 get_subject_dyn_body()	24
8.1.3.4 initialize()	24
8.1.3.5 is_ready()	25
8.1.3.6 is_same_subject_body()	25
8.1.3.7 is_subject_dyn_body()	26
8.1.3.8 operator=()	26
8.1.3.9 set_subject_body() [1/2]	26
8.1.3.10 set_subject_body() [2/2]	26
8.1.3.11 shutdown()	27
8.1.3.12 validate_body_inputs()	27
8.1.3.13 validate_name()	27
8.1.4 Friends And Related Function Documentation	28
8.1.4.1 init_attrjeod__BodyAction	28
8.1.4.2 InputProcessor	28
8.1.5 Field Documentation	28
8.1.5.1 action_identifier	28
8.1.5.2 action_name	29

8.1.5.3	active	29
8.1.5.4	dyn_subject	29
8.1.5.5	mass_subject	30
8.1.5.6	terminate_on_error	30
8.2	jeod::BodyActionMessages Class Reference	30
8.2.1	Detailed Description	31
8.2.2	Constructor & Destructor Documentation	31
8.2.2.1	BodyActionMessages() [1/2]	31
8.2.2.2	BodyActionMessages() [2/2]	32
8.2.3	Member Function Documentation	32
8.2.3.1	operator=()	32
8.2.4	Friends And Related Function Documentation	32
8.2.4.1	init_attrjeod__BodyActionMessages	32
8.2.4.2	InputProcessor	32
8.2.5	Field Documentation	32
8.2.5.1	fatal_error	32
8.2.5.2	illegal_value	33
8.2.5.3	invalid_name	33
8.2.5.4	invalid_object	33
8.2.5.5	not_performed	34
8.2.5.6	null_pointer	34
8.2.5.7	trace	34
8.3	jeod::BodyAttach Class Reference	35
8.3.1	Detailed Description	36
8.3.2	Constructor & Destructor Documentation	36
8.3.2.1	BodyAttach() [1/2]	36
8.3.2.2	BodyAttach() [2/2]	36
8.3.2.3	~BodyAttach()	36
8.3.3	Member Function Documentation	37
8.3.3.1	apply()	37

8.3.3.2	initialize()	37
8.3.3.3	operator=()	38
8.3.3.4	set_parent_body() [1/2]	38
8.3.3.5	set_parent_body() [2/2]	38
8.3.4	Friends And Related Function Documentation	38
8.3.4.1	init_attrjeod__BodyAttach	38
8.3.4.2	InputProcessor	38
8.3.5	Field Documentation	39
8.3.5.1	dyn_parent	39
8.3.5.2	mass_parent	39
8.3.5.3	succeeded	39
8.4	jeod::BodyAttachAligned Class Reference	40
8.4.1	Detailed Description	41
8.4.2	Constructor & Destructor Documentation	41
8.4.2.1	BodyAttachAligned() [1/2]	41
8.4.2.2	~BodyAttachAligned()	41
8.4.2.3	BodyAttachAligned() [2/2]	41
8.4.3	Member Function Documentation	41
8.4.3.1	apply()	41
8.4.3.2	initialize()	42
8.4.3.3	operator=()	42
8.4.4	Friends And Related Function Documentation	42
8.4.4.1	init_attrjeod__BodyAttachAligned	42
8.4.4.2	InputProcessor	43
8.4.5	Field Documentation	43
8.4.5.1	parent_point_name	43
8.4.5.2	subject_point_name	43
8.5	jeod::BodyAttachMatrix Class Reference	44
8.5.1	Detailed Description	44
8.5.2	Constructor & Destructor Documentation	45

8.5.2.1	BodyAttachMatrix()	45
8.5.2.2	~BodyAttachMatrix()	45
8.5.3	Member Function Documentation	45
8.5.3.1	apply()	45
8.5.4	Friends And Related Function Documentation	46
8.5.4.1	init_attrjeod__BodyAttachMatrix	46
8.5.4.2	InputProcessor	46
8.5.5	Field Documentation	46
8.5.5.1	offset_pstr_cstr_pstr	46
8.5.5.2	pstr_cstr	46
8.6	jeod::BodyDetach Class Reference	47
8.6.1	Detailed Description	47
8.6.2	Constructor & Destructor Documentation	47
8.6.2.1	BodyDetach()	48
8.6.2.2	~BodyDetach()	48
8.6.3	Member Function Documentation	48
8.6.3.1	apply()	48
8.6.3.2	is_ready()	49
8.6.4	Friends And Related Function Documentation	49
8.6.4.1	init_attrjeod__BodyDetach	49
8.6.4.2	InputProcessor	49
8.7	jeod::BodyDetachSpecific Class Reference	49
8.7.1	Detailed Description	50
8.7.2	Constructor & Destructor Documentation	50
8.7.2.1	BodyDetachSpecific()	51
8.7.2.2	~BodyDetachSpecific()	51
8.7.3	Member Function Documentation	51
8.7.3.1	apply()	51
8.7.3.2	initialize()	52
8.7.3.3	is_ready()	52

8.7.3.4	set_detach_from_body() [1/2]	52
8.7.3.5	set_detach_from_body() [2/2]	53
8.7.4	Friends And Related Function Documentation	53
8.7.4.1	init_attrjeod__BodyDetachSpecific	53
8.7.4.2	InputProcessor	53
8.7.5	Field Documentation	53
8.7.5.1	dyn_detach_from	53
8.7.5.2	mass_detach_from	54
8.8	jeod::BodyReattach Class Reference	54
8.8.1	Detailed Description	55
8.8.2	Constructor & Destructor Documentation	55
8.8.2.1	BodyReattach()	55
8.8.2.2	~BodyReattach()	55
8.8.3	Member Function Documentation	55
8.8.3.1	apply()	55
8.8.4	Friends And Related Function Documentation	56
8.8.4.1	init_attrjeod__BodyReattach	56
8.8.4.2	InputProcessor	56
8.8.5	Field Documentation	56
8.8.5.1	offset_pstr_cstr_pstr	56
8.8.5.2	pstr_cstr	57
8.9	jeod::DynBodyFrameSwitch Class Reference	57
8.9.1	Detailed Description	58
8.9.2	Member Enumeration Documentation	58
8.9.2.1	SwitchSense	58
8.9.3	Constructor & Destructor Documentation	59
8.9.3.1	DynBodyFrameSwitch()	59
8.9.3.2	~DynBodyFrameSwitch()	59
8.9.4	Member Function Documentation	59
8.9.4.1	apply()	59

8.9.4.2	initialize()	60
8.9.4.3	is_ready()	60
8.9.5	Friends And Related Function Documentation	60
8.9.5.1	init_attrjeod__DynBodyFrameSwitch	60
8.9.5.2	InputProcessor	61
8.9.6	Field Documentation	61
8.9.6.1	integ_frame	61
8.9.6.2	integ_frame_name	61
8.9.6.3	sort_grav_controls	61
8.9.6.4	switch_distance	62
8.9.6.5	switch_sense	62
8.10	jeod::DynBodyInit Class Reference	62
8.10.1	Detailed Description	64
8.10.2	Constructor & Destructor Documentation	64
8.10.2.1	DynBodyInit() [1/2]	65
8.10.2.2	DynBodyInit() [2/2]	65
8.10.2.3	~DynBodyInit()	65
8.10.3	Member Function Documentation	65
8.10.3.1	apply()	65
8.10.3.2	apply_user_inputs()	66
8.10.3.3	compute_rotational_state()	66
8.10.3.4	compute_translational_state()	66
8.10.3.5	find_body_frame()	67
8.10.3.6	find_dyn_body()	67
8.10.3.7	find_planet()	68
8.10.3.8	find_ref_frame()	68
8.10.3.9	initialize()	69
8.10.3.10	initializes_what()	69
8.10.3.11	is_ready()	70
8.10.3.12	operator=()	70

8.10.3.13	report_failure()	70
8.10.4	Friends And Related Function Documentation	70
8.10.4.1	init_attrjeod__DynBodyInit	71
8.10.4.2	InputProcessor	71
8.10.5	Field Documentation	71
8.10.5.1	ang_velocity	71
8.10.5.2	body_frame_id	72
8.10.5.3	body_ref_frame	72
8.10.5.4	orientation	72
8.10.5.5	position	72
8.10.5.6	rate_in_parent	73
8.10.5.7	reference_ref_frame	73
8.10.5.8	reference_ref_frame_name	73
8.10.5.9	reverse_sense	73
8.10.5.10	state	74
8.10.5.11	subscribed_frame	74
8.10.5.12	velocity	74
8.11	jeod::DynBodyInitLvLhRotState Class Reference	75
8.11.1	Detailed Description	75
8.11.2	Constructor & Destructor Documentation	76
8.11.2.1	DynBodyInitLvLhRotState() [1/2]	76
8.11.2.2	DynBodyInitLvLhRotState() [2/2]	76
8.11.2.3	~DynBodyInitLvLhRotState()	76
8.11.3	Member Function Documentation	76
8.11.3.1	initialize()	76
8.11.3.2	operator=()	77
8.11.4	Friends And Related Function Documentation	77
8.11.4.1	init_attrjeod__DynBodyInitLvLhRotState	77
8.11.4.2	InputProcessor	77
8.12	jeod::DynBodyInitLvLhState Class Reference	77

8.12.1 Detailed Description	78
8.12.2 Constructor & Destructor Documentation	78
8.12.2.1 DynBodyInitLvLhState() [1/2]	79
8.12.2.2 ~DynBodyInitLvLhState()	79
8.12.2.3 DynBodyInitLvLhState() [2/2]	79
8.12.3 Member Function Documentation	79
8.12.3.1 apply()	79
8.12.3.2 initialize()	80
8.12.3.3 operator=()	80
8.12.3.4 set_lvLh_frame_object()	80
8.12.4 Friends And Related Function Documentation	81
8.12.4.1 init_attrjeod__DynBodyInitLvLhState	81
8.12.4.2 InputProcessor	81
8.12.5 Field Documentation	81
8.12.5.1 lvLh_object_ptr	81
8.12.5.2 lvLh_type	81
8.13 jeod::DynBodyInitLvLhTransState Class Reference	82
8.13.1 Detailed Description	82
8.13.2 Constructor & Destructor Documentation	83
8.13.2.1 DynBodyInitLvLhTransState() [1/2]	83
8.13.2.2 DynBodyInitLvLhTransState() [2/2]	83
8.13.2.3 ~DynBodyInitLvLhTransState()	83
8.13.3 Member Function Documentation	83
8.13.3.1 initialize()	83
8.13.3.2 operator=()	84
8.13.4 Friends And Related Function Documentation	84
8.13.4.1 init_attrjeod__DynBodyInitLvLhTransState	84
8.13.4.2 InputProcessor	84
8.14 jeod::DynBodyInitNedRotState Class Reference	84
8.14.1 Detailed Description	85

8.14.2	Constructor & Destructor Documentation	85
8.14.2.1	DynBodyInitNedRotState() [1/2]	85
8.14.2.2	DynBodyInitNedRotState() [2/2]	85
8.14.2.3	~DynBodyInitNedRotState()	86
8.14.3	Member Function Documentation	86
8.14.3.1	initialize()	86
8.14.3.2	operator=()	86
8.14.4	Friends And Related Function Documentation	86
8.14.4.1	init_attrjeod__DynBodyInitNedRotState	86
8.14.4.2	InputProcessor	87
8.15	jeod::DynBodyInitNedState Class Reference	87
8.15.1	Detailed Description	88
8.15.2	Constructor & Destructor Documentation	88
8.15.2.1	DynBodyInitNedState() [1/2]	88
8.15.2.2	DynBodyInitNedState() [2/2]	88
8.15.2.3	~DynBodyInitNedState()	89
8.15.3	Member Function Documentation	89
8.15.3.1	apply()	89
8.15.3.2	initialize()	89
8.15.3.3	operator=()	90
8.15.4	Friends And Related Function Documentation	90
8.15.4.1	init_attrjeod__DynBodyInitNedState	90
8.15.4.2	InputProcessor	90
8.15.5	Field Documentation	90
8.15.5.1	altlatlong_type	90
8.15.5.2	ref_point	91
8.16	jeod::DynBodyInitNedTransState Class Reference	91
8.16.1	Detailed Description	92
8.16.2	Constructor & Destructor Documentation	92
8.16.2.1	DynBodyInitNedTransState() [1/2]	92

8.16.2.2	DynBodyInitNedTransState() [2/2]	92
8.16.2.3	~DynBodyInitNedTransState()	92
8.16.3	Member Function Documentation	93
8.16.3.1	initialize()	93
8.16.3.2	operator=()	93
8.16.4	Friends And Related Function Documentation	93
8.16.4.1	init_attrjeod__DynBodyInitNedTransState	93
8.16.4.2	InputProcessor	93
8.17	jeod::DynBodyInitOrbit Class Reference	94
8.17.1	Detailed Description	95
8.17.2	Member Enumeration Documentation	96
8.17.2.1	OrbitalSet	96
8.17.3	Constructor & Destructor Documentation	96
8.17.3.1	DynBodyInitOrbit()	96
8.17.3.2	~DynBodyInitOrbit()	97
8.17.4	Member Function Documentation	97
8.17.4.1	apply()	97
8.17.4.2	initialize()	97
8.17.5	Friends And Related Function Documentation	98
8.17.5.1	init_attrjeod__DynBodyInitOrbit	98
8.17.5.2	InputProcessor	98
8.17.6	Field Documentation	98
8.17.6.1	alt_apoapsis	98
8.17.6.2	alt_periapsis	99
8.17.6.3	arg_latitude	99
8.17.6.4	arg_periapsis	99
8.17.6.5	ascending_node	99
8.17.6.6	eccentricity	100
8.17.6.7	inclination	100
8.17.6.8	mean_anomaly	100

8.17.6.9	<code>orb_radius</code>	100
8.17.6.10	<code>orbit_frame</code>	101
8.17.6.11	<code>orbit_frame_name</code>	101
8.17.6.12	<code>planet</code>	101
8.17.6.13	<code>planet_name</code>	101
8.17.6.14	<code>radial_vel</code>	102
8.17.6.15	<code>semi_latus_rectum</code>	102
8.17.6.16	<code>semi_major_axis</code>	102
8.17.6.17	<code>set</code>	102
8.17.6.18	<code>time_periapsis</code>	103
8.17.6.19	<code>true_anomaly</code>	103
8.18	<code>jeod::DynBodyInitPlanetDerived</code> Class Reference	103
8.18.1	Detailed Description	104
8.18.2	Constructor & Destructor Documentation	105
8.18.2.1	<code>DynBodyInitPlanetDerived()</code> [1/2]	105
8.18.2.2	<code>DynBodyInitPlanetDerived()</code> [2/2]	105
8.18.2.3	<code>~DynBodyInitPlanetDerived()</code>	105
8.18.3	Member Function Documentation	105
8.18.3.1	<code>apply()</code>	105
8.18.3.2	<code>initialize()</code>	106
8.18.3.3	<code>is_ready()</code>	106
8.18.3.4	<code>operator=()</code>	107
8.18.4	Friends And Related Function Documentation	107
8.18.4.1	<code>init_attrjeod__DynBodyInitPlanetDerived</code>	107
8.18.4.2	<code>InputProcessor</code>	107
8.18.5	Field Documentation	107
8.18.5.1	<code>body_is_required</code>	107
8.18.5.2	<code>ref_body</code>	108
8.18.5.3	<code>ref_body_name</code>	108
8.18.5.4	<code>required_items</code>	108

8.19	jeod::DynBodyInitRotState Class Reference	109
8.19.1	Detailed Description	110
8.19.2	Member Enumeration Documentation	110
8.19.2.1	StateItems	110
8.19.3	Constructor & Destructor Documentation	110
8.19.3.1	DynBodyInitRotState() [1/2]	110
8.19.3.2	DynBodyInitRotState() [2/2]	111
8.19.3.3	~DynBodyInitRotState()	111
8.19.4	Member Function Documentation	111
8.19.4.1	apply()	111
8.19.4.2	initialize()	111
8.19.4.3	initializes_what()	112
8.19.4.4	is_ready()	112
8.19.4.5	operator=()	113
8.19.5	Friends And Related Function Documentation	113
8.19.5.1	init_attrjeod__DynBodyInitRotState	113
8.19.5.2	InputProcessor	113
8.19.6	Field Documentation	113
8.19.6.1	state_items	113
8.20	jeod::DynBodyInitTransState Class Reference	114
8.20.1	Detailed Description	115
8.20.2	Member Enumeration Documentation	115
8.20.2.1	StateItems	115
8.20.3	Constructor & Destructor Documentation	115
8.20.3.1	DynBodyInitTransState() [1/2]	115
8.20.3.2	~DynBodyInitTransState()	116
8.20.3.3	DynBodyInitTransState() [2/2]	116
8.20.4	Member Function Documentation	116
8.20.4.1	apply()	116
8.20.4.2	initialize()	116

8.20.4.3	initializes_what()	117
8.20.4.4	is_ready()	117
8.20.4.5	operator=()	118
8.20.5	Friends And Related Function Documentation	118
8.20.5.1	init_attrjeod__DynBodyInitTransState	118
8.20.5.2	InputProcessor	118
8.20.6	Field Documentation	118
8.20.6.1	state_items	118
8.21	jeod::DynBodyInitWrtPlanet Class Reference	119
8.21.1	Detailed Description	120
8.21.2	Constructor & Destructor Documentation	120
8.21.2.1	DynBodyInitWrtPlanet() [1/2]	120
8.21.2.2	DynBodyInitWrtPlanet() [2/2]	120
8.21.2.3	~DynBodyInitWrtPlanet()	120
8.21.3	Member Function Documentation	120
8.21.3.1	apply()	120
8.21.3.2	initialize()	121
8.21.3.3	initializes_what()	121
8.21.3.4	is_ready()	122
8.21.3.5	operator=()	122
8.21.4	Friends And Related Function Documentation	122
8.21.4.1	init_attrjeod__DynBodyInitWrtPlanet	122
8.21.4.2	InputProcessor	122
8.21.5	Field Documentation	122
8.21.5.1	planet	123
8.21.5.2	planet_name	123
8.21.5.3	set_items	123
8.22	jeod::MassBodyInit Class Reference	124
8.22.1	Detailed Description	125
8.22.2	Constructor & Destructor Documentation	125
8.22.2.1	MassBodyInit() [1/2]	125
8.22.2.2	MassBodyInit() [2/2]	125
8.22.2.3	~MassBodyInit()	125
8.22.3	Member Function Documentation	125
8.22.3.1	apply()	125
8.22.3.2	operator=()	126
8.22.4	Friends And Related Function Documentation	126
8.22.4.1	init_attrjeod__MassBodyInit	126
8.22.4.2	InputProcessor	126
8.22.5	Field Documentation	126
8.22.5.1	num_points	126
8.22.5.2	points	127
8.22.5.3	properties	127

9	File Documentation	129
9.1	body_action.cc File Reference	129
9.1.1	Detailed Description	129
9.2	body_action.hh File Reference	129
9.2.1	Detailed Description	130
9.3	body_action_messages.cc File Reference	130
9.3.1	Detailed Description	130
9.4	body_action_messages.hh File Reference	130
9.4.1	Detailed Description	131
9.5	body_attach.cc File Reference	131
9.5.1	Detailed Description	131
9.6	body_attach.hh File Reference	131
9.6.1	Detailed Description	132
9.7	body_attach_aligned.cc File Reference	132
9.7.1	Detailed Description	132
9.8	body_attach_aligned.hh File Reference	132
9.8.1	Detailed Description	133
9.9	body_attach_matrix.cc File Reference	133
9.9.1	Detailed Description	133
9.10	body_attach_matrix.hh File Reference	133
9.10.1	Detailed Description	134
9.11	body_detach.cc File Reference	134
9.11.1	Detailed Description	134
9.12	body_detach.hh File Reference	134
9.12.1	Detailed Description	135
9.13	body_detach_specific.cc File Reference	135
9.13.1	Detailed Description	135
9.14	body_detach_specific.hh File Reference	135
9.14.1	Detailed Description	136
9.15	body_reattach.cc File Reference	136

9.15.1 Detailed Description	136
9.16 body_reattach.hh File Reference	136
9.16.1 Detailed Description	137
9.17 class_declarations.hh File Reference	137
9.17.1 Detailed Description	137
9.18 dyn_body_frame_switch.cc File Reference	137
9.18.1 Detailed Description	138
9.19 dyn_body_frame_switch.hh File Reference	138
9.19.1 Detailed Description	138
9.20 dyn_body_init.cc File Reference	138
9.20.1 Detailed Description	139
9.21 dyn_body_init.hh File Reference	139
9.21.1 Detailed Description	139
9.22 dyn_body_init_lvlh_rot_state.cc File Reference	139
9.22.1 Detailed Description	140
9.23 dyn_body_init_lvlh_rot_state.hh File Reference	140
9.23.1 Detailed Description	140
9.24 dyn_body_init_lvlh_state.cc File Reference	140
9.24.1 Detailed Description	141
9.25 dyn_body_init_lvlh_state.hh File Reference	141
9.25.1 Detailed Description	141
9.26 dyn_body_init_lvlh_trans_state.cc File Reference	141
9.26.1 Detailed Description	142
9.27 dyn_body_init_lvlh_trans_state.hh File Reference	142
9.27.1 Detailed Description	142
9.28 dyn_body_init_ned_rot_state.cc File Reference	142
9.28.1 Detailed Description	143
9.29 dyn_body_init_ned_rot_state.hh File Reference	143
9.29.1 Detailed Description	143
9.30 dyn_body_init_ned_state.cc File Reference	143

9.30.1 Detailed Description	144
9.31 dyn_body_init_ned_state.hh File Reference	144
9.31.1 Detailed Description	144
9.32 dyn_body_init_ned_trans_state.cc File Reference	144
9.32.1 Detailed Description	145
9.33 dyn_body_init_ned_trans_state.hh File Reference	145
9.33.1 Detailed Description	145
9.34 dyn_body_init_orbit.cc File Reference	145
9.34.1 Detailed Description	146
9.35 dyn_body_init_orbit.hh File Reference	146
9.35.1 Detailed Description	146
9.36 dyn_body_init_planet_derived.cc File Reference	146
9.36.1 Detailed Description	147
9.37 dyn_body_init_planet_derived.hh File Reference	147
9.37.1 Detailed Description	147
9.38 dyn_body_init_rot_state.cc File Reference	147
9.38.1 Detailed Description	148
9.39 dyn_body_init_rot_state.hh File Reference	148
9.39.1 Detailed Description	148
9.40 dyn_body_init_trans_state.cc File Reference	148
9.40.1 Detailed Description	149
9.41 dyn_body_init_trans_state.hh File Reference	149
9.41.1 Detailed Description	149
9.42 dyn_body_init_wrt_planet.cc File Reference	149
9.42.1 Detailed Description	150
9.43 dyn_body_init_wrt_planet.hh File Reference	150
9.43.1 Detailed Description	150
9.44 mass_body_init.cc File Reference	150
9.44.1 Detailed Description	151
9.45 mass_body_init.hh File Reference	151
9.45.1 Detailed Description	151

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models	13
Dynamics	14
BodyAction	15

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod	Namespace jeod	19
----------------------	--------------------------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

jeod::BodyAction	21
jeod::BodyAttach	35
jeod::BodyAttachAligned	40
jeod::BodyAttachMatrix	44
jeod::BodyDetach	47
jeod::BodyDetachSpecific	49
jeod::BodyReattach	54
jeod::DynBodyFrameSwitch	57
jeod::DynBodyInit	62
jeod::DynBodyInitRotState	109
jeod::DynBodyInitTransState	114
jeod::DynBodyInitOrbit	94
jeod::DynBodyInitWrtPlanet	119
jeod::DynBodyInitPlanetDerived	103
jeod::DynBodyInitLvLhState	77
jeod::DynBodyInitLvLhRotState	75
jeod::DynBodyInitLvLhTransState	82
jeod::DynBodyInitNedState	87
jeod::DynBodyInitNedRotState	84
jeod::DynBodyInitNedTransState	91
jeod::MassBodyInit	124
jeod::BodyActionMessages	30

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

jeod::BodyAction	BodyAction is the base class for the BodyAction model	21
jeod::BodyActionMessages	Specifies the message IDs used in the BodyAction model	30
jeod::BodyAttach	Provides the basic ability to attach one MassBody to another	35
jeod::BodyAttachAligned	Attaches a pair of MassBody objects at a pair of MassPoints	40
jeod::BodyAttachMatrix	Attaches a pair of MassBody objects using the offset+matrix attach mechanism	44
jeod::BodyDetach	Provides the basic ability to detach one MassBody from another	47
jeod::BodyDetachSpecific	Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach_from body	49
jeod::BodyReattach	Alters the nature of an existing attachment	54
jeod::DynBodyFrameSwitch	Switch a DynBody 's integration frame to a specified frame when the body switches to that integration frame's sphere of influence	57
jeod::DynBodyInit	Base class for initialize the state of a DynBody	62
jeod::DynBodyInitLvlhRotState	Initialize a vehicle's rotational state with respect to some vehicle's LVLH frame	75
jeod::DynBodyInitLvlhState	Initialize selected aspects of a vehicle's state with respect to some vehicle's LVLH frame	77
jeod::DynBodyInitLvlhTransState	Initialize a vehicle's translational state with respect to some other vehicle's LVLH frame	82
jeod::DynBodyInitNedRotState	Initialize a vehicle's rotational state wrt some vehicle's North-East-Down frame	84
jeod::DynBodyInitNedState	Initialize selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet	87
jeod::DynBodyInitNedTransState	Initialize a vehicle's translational state wrt some vehicle's North-East-Down frame	91

jeod::DynBodyInitOrbit	
Initialize a vehicle's translational state given an orbital specification	94
jeod::DynBodyInitPlanetDerived	
(Initialize selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet	103
jeod::DynBodyInitRotState	
Initialize aspects of a vehicle's rotational state	109
jeod::DynBodyInitTransState	
Initialize aspects of a vehicle's translational state	114
jeod::DynBodyInitWrtPlanet	
Initialize selected aspects of a vehicle's state with respect to some frame based on the planet .	119
jeod::MassBodyInit	
Base class for initializing a MassBody	124

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

body_action.cc	Define methods for the BodyAction class	129
body_action.hh	Define the class BodyAction, the base class used for performing actions on a MassBody or DynBody object	129
body_action_messages.cc	Implement the class BodyActionMessages	130
body_action_messages.hh	Define the class BodyActionMessages, the class that specifies the message IDs used in the BodyAction model	130
body_attach.cc	Define methods for the mass body initialization class	131
body_attach.hh	Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another	131
body_attach_aligned.cc	Define methods for the mass body initialization class	132
body_attach_aligned.hh	Define the class MassBodyAttachAligned, which causes one MassBody to be attached to another at a pair of MassPoints	132
body_attach_matrix.cc	Define methods for the mass body initialization class	133
body_attach_matrix.hh	Define the class MassBodyAttachMatrix, which causes one MassBody to be attached given a transformation	133
body_detach.cc	Define methods for the MassBodyDetach class	134
body_detach.hh	Define the class MassBodyDetach, the base class used for detaching one MassBody object from one another	134
body_detach_specific.cc	Define methods for the BodyDetachSpecific class	135
body_detach_specific.hh	Define the class MassBodyDetachSpecific, the class used for detaching one MassBody object from another specified MassBody	135

body_reattach.cc	Define methods for the mass body initialization class	136
body_reattach.hh	Define the class MassBodyReattach, which causes one MassBody to be reattached given a transformation	136
class_declarations.hh	Forward declarations of classes defined in dyn_body_init_XXX.hh files	137
dyn_body_frame_switch.cc	Define methods for the class DynBodyFrameSwitch	137
dyn_body_frame_switch.hh	Define the class DynBodyFrameSwitch, the BodyAction derived class used for switch a DynBody's integration frame	138
dyn_body_init.cc	Define methods for the base body initialization class	138
dyn_body_init.hh	Define the class DynBodyInit, the base class used for initializing the state of a DynBody object	139
dyn_body_init_lvhl_rot_state.cc	Define methods for DynBodyInitLvhlRotState	139
dyn_body_init_lvhl_rot_state.hh	Define the class DynBodyInitLvhlRotState, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame	140
dyn_body_init_lvhl_state.cc	Define methods for the DynBodyInitLvhlState class	140
dyn_body_init_lvhl_state.hh	Define the class DynBodyInitLvhlState, the base class for initializing selected aspects of a vehicle's state with respect to some vehicle's LVLH frame	141
dyn_body_init_lvhl_trans_state.cc	Define methods for DynBodyInitLvhlTransState	141
dyn_body_init_lvhl_trans_state.hh	Define the class DynBodyInitLvhlTransState, which initialize a vehicle's translational state with respect to some other vehicle's LVLH frame	142
dyn_body_init_ned_rot_state.cc	Define methods for DynBodyInitNedRotState	142
dyn_body_init_ned_rot_state.hh	Define the class DynBodyInitNedRotState, which initialize a vehicle's rotational state wrt some other vehicle's North-East-Down frame	143
dyn_body_init_ned_state.cc	Define methods for DynBodyInitNedState	143
dyn_body_init_ned_state.hh	Define the class DynBodyInitNedState, the base class for initializing selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet	144
dyn_body_init_ned_trans_state.cc	Define methods for DynBodyInitNedTransState	144
dyn_body_init_ned_trans_state.hh	Define the class DynBodyInitNedTransState, which initialize a vehicle's translational state wrt some other vehicle's North-East-Down frame	145
dyn_body_init_orbit.cc	Define classes for items represented in some ephemeris model	145
dyn_body_init_orbit.hh	Define the class DynBodyInitOrbit, which initializes a vehicle in in some orbit	146
dyn_body_init_planet_derived.cc	Define methods for the DynBodyInitPlanetDerived class	146
dyn_body_init_planet_derived.hh	Define the class DynBodyInitPlanetDerived, the base class for initializing selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet	147

dyn_body_init_rot_state.cc	
Define methods for DynBodyInitRotState	147
dyn_body_init_rot_state.hh	
Define the class DynBodyInitRotState that initialize aspects of a vehicle's rotational state	148
dyn_body_init_trans_state.cc	
Define methods for DynBodyInitTransState	148
dyn_body_init_trans_state.hh	
Define the class DynBodyInitTransState that initialize aspects of a vehicle's translational state . .	149
dyn_body_init_wrt_planet.cc	
Define methods for the DynBodyInitWrtPlanet class	149
dyn_body_init_wrt_planet.hh	
Define the class DynBodyInitWrtPlanet, the base class for initializing selected aspects of a vehicle's state with respect to some state that is connected to a planet in some way	150
mass_body_init.cc	
Define methods for the mass body initialization class	150
mass_body_init.hh	
Define the class MassBodyInit, the base class used for initializing the core mass properties of a MassBody object	151

Chapter 6

Module Documentation

6.1 Models

Modules

- [Dynamics](#)

6.1.1 Detailed Description

6.2 Dynamics

Modules

- [BodyAction](#)

6.2.1 Detailed Description

6.3 BodyAction

Files

- file [body_action.hh](#)
Define the class BodyAction, the base class used for performing actions on a MassBody or DynBody object.
- file [body_action_messages.hh](#)
Define the class BodyActionMessages, the class that specifies the message IDs used in the BodyAction model.
- file [body_attach.hh](#)
Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another.
- file [body_attach_aligned.hh](#)
Define the class MassBodyAttachAligned, which causes one MassBody to be attached to another at a pair of Mass↔Points.
- file [body_attach_matrix.hh](#)
Define the class MassBodyAttachMatrix, which causes one MassBody to be attached given a transformation.
- file [body_detach.hh](#)
Define the class MassBodyDetach, the base class used for detaching one MassBody object from one another.
- file [body_detach_specific.hh](#)
Define the class MassBodyDetachSpecific, the class used for detaching one MassBody object from another specified MassBody.
- file [body_reattach.hh](#)
Define the class MassBodyReattach, which causes one MassBody to be reattached given a transformation.
- file [class_declarations.hh](#)
Forward declarations of classes defined in dyn_body_init_XXX.hh files.
- file [dyn_body_frame_switch.hh](#)
Define the class DynBodyFrameSwitch, the BodyAction derived class used for switch a DynBody's integration frame.
- file [dyn_body_init.hh](#)
Define the class DynBodyInit, the base class used for initializing the state of a DynBody object.
- file [dyn_body_init_lvhl_rot_state.hh](#)
Define the class DynBodyInitLvhlRotState, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.
- file [dyn_body_init_lvhl_state.hh](#)
Define the class DynBodyInitLvhlState, the base class for initializing selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.
- file [dyn_body_init_lvhl_trans_state.hh](#)
Define the class DynBodyInitLvhlTransState, which initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.
- file [dyn_body_init_ned_rot_state.hh](#)
Define the class DynBodyInitNedRotState, which initialize a vehicle's rotational state wrt some other vehicle's North-East-Down frame.
- file [dyn_body_init_ned_state.hh](#)
Define the class DynBodyInitNedState, the base class for initializing selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.
- file [dyn_body_init_ned_trans_state.hh](#)
Define the class DynBodyInitNedTransState, which initialize a vehicle's translational state wrt some other vehicle's North-East-Down frame.
- file [dyn_body_init_orbit.hh](#)
Define the class DynBodyInitOrbit, which initializes a vehicle in in some orbit.
- file [dyn_body_init_planet_derived.hh](#)
Define the class DynBodyInitPlanetDerived, the base class for initializing selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.
- file [dyn_body_init_rot_state.hh](#)

- Define the class `DynBodyInitRotState` that initialize aspects of a vehicle's rotational state.*

 - file [dyn_body_init_trans_state.hh](#)
- Define the class `DynBodyInitTransState` that initialize aspects of a vehicle's translational state.*

 - file [dyn_body_init_wrt_planet.hh](#)
- Define the class `DynBodyInitWrtPlanet`, the base class for initializing selected aspects of a vehicle's state with respect to some state that is connected to a planet in some way.*

 - file [mass_body_init.hh](#)
- Define the class `MassBodyInit`, the base class used for initializing the core mass properties of a `MassBody` object.*

 - file [body_action.cc](#)
- Define methods for the `BodyAction` class.*

 - file [body_action_messages.cc](#)
- Implement the class `BodyActionMessages`.*

 - file [body_attach.cc](#)
- Define methods for the mass body initialization class.*

 - file [body_attach_aligned.cc](#)
- Define methods for the mass body initialization class.*

 - file [body_attach_matrix.cc](#)
- Define methods for the mass body initialization class.*

 - file [body_detach.cc](#)
- Define methods for the `MassBodyDetach` class.*

 - file [body_detach_specific.cc](#)
- Define methods for the `BodyDetachSpecific` class.*

 - file [body_reattach.cc](#)
- Define methods for the mass body initialization class.*

 - file [dyn_body_frame_switch.cc](#)
- Define methods for the class `DynBodyFrameSwitch`.*

 - file [dyn_body_init.cc](#)
- Define methods for the base body initialization class.*

 - file [dyn_body_init_lvih_rot_state.cc](#)
- Define methods for `DynBodyInitLvIhRotState`.*

 - file [dyn_body_init_lvih_state.cc](#)
- Define methods for the `DynBodyInitLvIhState` class.*

 - file [dyn_body_init_lvih_trans_state.cc](#)
- Define methods for `DynBodyInitLvIhTransState`.*

 - file [dyn_body_init_ned_rot_state.cc](#)
- Define methods for `DynBodyInitNedRotState`.*

 - file [dyn_body_init_ned_state.cc](#)
- Define methods for `DynBodyInitNedState`.*

 - file [dyn_body_init_ned_trans_state.cc](#)
- Define methods for `DynBodyInitNedTransState`.*

 - file [dyn_body_init_orbit.cc](#)
- Define classes for items represented in some ephemeris model.*

 - file [dyn_body_init_planet_derived.cc](#)
- Define methods for the `DynBodyInitPlanetDerived` class.*

 - file [dyn_body_init_rot_state.cc](#)
- Define methods for `DynBodyInitRotState`.*

 - file [dyn_body_init_trans_state.cc](#)
- Define methods for `DynBodyInitTransState`.*

 - file [dyn_body_init_wrt_planet.cc](#)
- Define methods for the `DynBodyInitWrtPlanet` class.*

 - file [mass_body_init.cc](#)
- Define methods for the mass body initialization class.*

Namespaces

- [jeod](#)

Namespace jeod.

Macros

- `#define PATH "dynamics/body_action/"`

6.3.1 Detailed Description

6.3.2 Macro Definition Documentation

6.3.2.1 PATH

```
#define PATH "dynamics/body_action/"
```

Definition at line 31 of file body_action_messages.cc.

Chapter 7

Namespace Documentation

7.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class [BodyAction](#)
BodyAction is the base class for the BodyAction model.
- class [BodyActionMessages](#)
Specifies the message IDs used in the BodyAction model.
- class [BodyAttach](#)
Provides the basic ability to attach one MassBody to another.
- class [BodyAttachAligned](#)
Attaches a pair of MassBody objects at a pair of MassPoints.
- class [BodyAttachMatrix](#)
Attaches a pair of MassBody objects using the offset+matrix attach mechanism.
- class [BodyDetach](#)
Provides the basic ability to detach one MassBody from another.
- class [BodyDetachSpecific](#)
Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach←→_from body.
- class [BodyReattach](#)
Alters the nature of an existing attachment.
- class [DynBodyFrameSwitch](#)
Switch a DynBody's integration frame to a specified frame when the body switches to that integration frame's sphere of influence.
- class [DynBodyInit](#)
Base class for initialize the state of a DynBody.
- class [DynBodyInitLvLhRotState](#)
Initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.
- class [DynBodyInitLvLhState](#)
Initialize selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.
- class [DynBodyInitLvLhTransState](#)
initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.

- class [DynBodyInitNedRotState](#)
Initialize a vehicle's rotational state wrt some vehicle's North-East-Down frame.
- class [DynBodyInitNedState](#)
Initialize selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.
- class [DynBodyInitNedTransState](#)
Initialize a vehicle's translational state wrt some vehicle's North-East-Down frame.
- class [DynBodyInitOrbit](#)
Initialize a vehicle's translational state given an orbital specification.
- class [DynBodyInitPlanetDerived](#)
(Initialize selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.
- class [DynBodyInitRotState](#)
Initialize aspects of a vehicle's rotational state.
- class [DynBodyInitTransState](#)
Initialize aspects of a vehicle's translational state.
- class [DynBodyInitWrtPlanet](#)
Initialize selected aspects of a vehicle's state with respect to some frame based on the planet.
- class [MassBodyInit](#)
Base class for initializing a MassBody.

7.1.1 Detailed Description

Namespace jeod.

Data Structure Documentation

BodyAction is the base class for the **BodyAction** model.

Inheritance diagram for jeod::BodyAction:



- void `set_subject_body` (MassBody &mass_body_in)
Set the subject mass body of this action.
- void `set_subject_dyn_body` (DynBody &dyn_body_in)
Set the subject dyn body of this action.
- bool `is_same_subject_body` (MassBody &mass_body_in)
Test the input mass body against the subject body and return true if they are the same body.
- bool `is_subject_dyn_body` ()
Check if the subject is a DynBody.
- DynBody * `get_subject_dyn_body` ()
- `BodyAction` ()
Construct a `BodyAction`.
- virtual `~BodyAction` ()
Destruct a `BodyAction`.
- virtual void `shutdown` ()
Release resources allocated by a `BodyAction` object.
- const std::string & `get_identifier` (void) const
Accessor for action_identifier.
- virtual void `initialize` (DynManager &dyn_manager)
Begin initialization of a `BodyAction`.
- virtual bool `is_ready` (void)
In general, determine if the initializer is ready to be applied.
- virtual void `apply` (DynManager &dyn_manager)
Complete initialization.

Data Fields

- bool `active`
Controls when the action is performed.
- bool `terminate_on_error`
Indicates whether errors encountered while performing the action are to terminate the simulation.
- std::string `action_name`
An identifier for this action.

Protected Member Functions

- virtual void `validate_body_inputs` (DynBody *&dyn_body_in, MassBody *&mass_body_in, const std::string &body_base_name)
- void `validate_name` (const std::string &variable_value, const std::string &variable_type, const std::string &variable_name)
Ensure that a string is not trivially invalid.

Protected Attributes

- MassBody * `mass_subject`
The MassBody of the body that is the subject of this action.
- DynBody * `dyn_subject`
The DynBody of the body that is the subject of this action.
- std::string `action_identifier`
An identifier for this action, constructed from the class name and the action name at initialization time.

Private Member Functions

- `BodyAction` (const `BodyAction` &)
- `BodyAction` & `operator=` (const `BodyAction` &)

Friends

- class `InputProcessor`
- void `init_attrjeod__BodyAction` ()

8.1.1 Detailed Description

`BodyAction` is the base class for the `BodyAction` model.

A `BodyAction` instance that performs some operation on a `MassBody` object. The simulation Dynamics Manager object manages a collection of `BodyAction` objects for the purpose of initializing `MassBody` objects and later, for performing asynchronous actions on them.

The `BodyAction` model hinges on three methods:

- `initialize()` The `initialize()` method initializes the `BodyAction`. This method does not and must not operate on the subject of the action. All derived classes must forward the `initialize()` call to the immediate parent class and then perform class-dependent object initializations.
- `is_ready()` The `is_ready` method indicates whether the action is ready to be applied. For example, an action that initializes the translation state of a vehicle relative to some other vehicle cannot do its job until that other vehicle's translational state is set. The `is_ready()` method for such an action should return false until the other vehicle's translational state has been set.
- `apply()` The `apply()` method applies the action – it does something to the subject of the action. All derived classes must perform class-dependent actions and then must forward the `apply()` call to the immediate parent class.

Definition at line 108 of file `body_action.hh`.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 BodyAction() [1/2]

```
jeod::BodyAction::BodyAction (
    const BodyAction & ) [private]
```

8.1.2.2 BodyAction() [2/2]

```
jeod::BodyAction::BodyAction (
    void )
```

Construct a [BodyAction](#).

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

8.1.2.3 ~BodyAction()

```
jeod::BodyAction::~~BodyAction (
    void ) [virtual]
```

Destruct a [BodyAction](#).

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

References `shutdown()`.

8.1.3 Member Function Documentation

8.1.3.1 apply()

```
void jeod::BodyAction::apply (
    DynManager & dyn_manager ) [virtual]
```

Complete initialization.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Jeod manager
----------------------	---------------------------------	--------------

Reimplemented in [jeod::DynBodyInitOrbit](#), [jeod::DynBodyInit](#), [jeod::DynBodyFrameSwitch](#), [jeod::BodyAttach](#), [jeod::DynBodyInitPlanetDerived](#), [jeod::BodyDetachSpecific](#), [jeod::DynBodyInitWrtPlanet](#), [jeod::MassBodyInit](#), [jeod::DynBodyInitRotState](#), [jeod::DynBodyInitNedState](#), [jeod::BodyAttachAligned](#), [jeod::BodyReattach](#), [jeod::DynBodyInitTransState](#), [jeod::DynBodyInitLvlhState](#), [jeod::BodyAttachMatrix](#), and [jeod::BodyDetach](#).

Definition at line 123 of file `body_action.cc`.

References `shutdown()`.

Referenced by [jeod::BodyDetach::apply\(\)](#), [jeod::BodyReattach::apply\(\)](#), [jeod::MassBodyInit::apply\(\)](#), [jeod::BodyDetachSpecific::apply\(\)](#), [jeod::BodyAttach::apply\(\)](#), [jeod::DynBodyFrameSwitch::apply\(\)](#), and [jeod::DynBodyInit::apply\(\)](#).

8.1.3.2 `get_identifier()`

```
const std::string & jeod::BodyAction::get_identifier (
    void ) const [inline]
```

Accessor for `action_identifier`.

Returns

Action identifier

Definition at line 265 of file `body_action.hh`.

References `action_identifier`.

8.1.3.3 `get_subject_dyn_body()`

```
DynBody * jeod::BodyAction::get_subject_dyn_body ( )
```

Definition at line 218 of file `body_action.cc`.

References `dyn_subject`, and `mass_subject`.

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

8.1.3.4 `initialize()`

```
void jeod::BodyAction::initialize (
    DynManager & dyn_manager ) [virtual]
```

Begin initialization of a [BodyAction](#).

The `initialize` method for all subclasses of [BodyAction](#) *must* pass the `initialize` call to their immediate parent class, which in turn must do the same, eventually invoking this method.

Parameters

<code>in, out</code>	<code>dyn_manager</code>	Dynamics manager
----------------------	--------------------------	------------------

Reimplemented in [jeod::DynBodyInitOrbit](#), [jeod::DynBodyInit](#), [jeod::DynBodyFrameSwitch](#), [jeod::BodyAttach](#), [jeod::BodyDetachSpecific](#), [jeod::DynBodyInitPlanetDerived](#), [jeod::DynBodyInitWrtPlanet](#), [jeod::DynBodyInitRotState](#), [jeod::DynBodyInitNedState](#), [jeod::BodyAttachAligned](#), [jeod::DynBodyInitTransState](#), [jeod::DynBodyInitLvlhState](#), [jeod::DynBodyInitLvlhRotState](#), [jeod::DynBodyInitLvlhTransState](#), [jeod::DynBodyInitNedRotState](#), and [jeod::DynBodyInitNedTransState](#).

Definition at line 101 of file `body_action.cc`.

References `action_identifier`, `action_name`, `dyn_subject`, `mass_subject`, and `validate_body_inputs()`.

Referenced by [jeod::BodyDetachSpecific::initialize\(\)](#), [jeod::BodyAttach::initialize\(\)](#), [jeod::DynBodyFrameSwitch::initialize\(\)](#), and [jeod::DynBodyInit::initialize\(\)](#).

8.1.3.5 `is_ready()`

```
bool jeod::BodyAction::is_ready (
    void ) [virtual]
```

In general, determine if the initializer is ready to be applied.

This base class method simply queries the active flag. Subclasses should override this default method.

Returns

Can initializer run?

Reimplemented in [jeod::DynBodyInit](#), [jeod::DynBodyFrameSwitch](#), [jeod::BodyDetachSpecific](#), [jeod::DynBodyInitPlanetDerived](#), [jeod::DynBodyInitRotState](#), [jeod::DynBodyInitWrtPlanet](#), [jeod::DynBodyInitTransState](#), and [jeod::BodyDetach](#).

Definition at line 140 of file `body_action.cc`.

References `active`.

Referenced by [jeod::DynBodyFrameSwitch::is_ready\(\)](#), and [jeod::DynBodyInit::is_ready\(\)](#).

8.1.3.6 `is_same_subject_body()`

```
bool jeod::BodyAction::is_same_subject_body (
    MassBody & mass_body_in )
```

Test the input mass body against the subject body and return true if they are the same body.

Definition at line 190 of file `body_action.cc`.

References `dyn_subject`, and `mass_subject`.

8.1.3.7 is_subject_dyn_body()

```
bool jeod::BodyAction::is_subject_dyn_body ( )
```

Check if the subject is a DynBody.

Definition at line 202 of file body_action.cc.

References dyn_subject, and mass_subject.

8.1.3.8 operator=()

```
BodyAction& jeod::BodyAction::operator= (
    const BodyAction & ) [private]
```

8.1.3.9 set_subject_body() [1/2]

```
void jeod::BodyAction::set_subject_body (
    MassBody & mass_body_in )
```

Set the subject mass body of this action.

Resets dyn_subject to null

Definition at line 146 of file body_action.cc.

References dyn_subject, and mass_subject.

8.1.3.10 set_subject_body() [2/2]

```
void jeod::BodyAction::set_subject_body (
    DynBody & dyn_body_in )
```

Set the subject dyn body of this action.

Resets mass_subject to null

Definition at line 152 of file body_action.cc.

References dyn_subject, and mass_subject.

8.1.3.11 shutdown()

```
void jeod::BodyAction::shutdown (
    void ) [virtual]
```

Release resources allocated by a [BodyAction](#) object.

Definition at line 86 of file `body_action.cc`.

Referenced by `apply()`, and `~BodyAction()`.

8.1.3.12 validate_body_inputs()

```
void jeod::BodyAction::validate_body_inputs (
    DynBody *& dyn_body_in,
    MassBody *& mass_body_in,
    const std::string & body_base_name ) [protected], [virtual]
```

Definition at line 158 of file `body_action.cc`.

References `action_identifier`, `jeod::BodyActionMessages::fatal_error`, and `jeod::BodyActionMessages::null_pointer`.

Referenced by `jeod::BodyDetachSpecific::initialize()`, `jeod::BodyAttach::initialize()`, and `initialize()`.

8.1.3.13 validate_name()

```
void jeod::BodyAction::validate_name (
    const std::string & variable_value,
    const std::string & variable_name,
    const std::string & variable_type ) [protected]
```

Ensure that a string is not trivially invalid.

Parameters

in	<i>variable_value</i>	String to be checked
in	<i>variable_name</i>	For error reporting
in	<i>variable_type</i>	For error reporting

Definition at line 241 of file `body_action.cc`.

References `action_identifier`, and `jeod::BodyActionMessages::invalid_name`.

Referenced by `jeod::DynBodyInit::find_body_frame()`, `jeod::DynBodyInit::find_dyn_body()`, `jeod::DynBodyInit::find_planet()`, `jeod::DynBodyInit::find_ref_frame()`, and `jeod::DynBodyInitOrbit::initialize()`.

8.1.4 Friends And Related Function Documentation

8.1.4.1 init_attrjeod__BodyAction

```
void init_attrjeod__BodyAction ( ) [friend]
```

8.1.4.2 InputProcessor

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

Definition at line 110 of file body_action.hh.

8.1.5 Field Documentation

8.1.5.1 action_identifier

```
std::string jeod::BodyAction::action_identifier [protected]
```

An identifier for this action, constructed from the class name and the action name at initialization time.

This is used for generating error and debug messages.trick_units(–)

Definition at line 194 of file body_action.hh.

Referenced by jeod::BodyDetach::apply(), jeod::BodyReattach::apply(), jeod::DynBodyInitNedState::apply(), jeod::MassBodyInit::apply(), jeod::BodyDetachSpecific::apply(), jeod::BodyAttach::apply(), jeod::DynBodyFrameSwitch::apply(), jeod::DynBodyInit::apply(), jeod::DynBodyInitOrbit::apply(), jeod::DynBodyInit::find_body_frame(), jeod::DynBodyInit::find_dyn_body(), jeod::DynBodyInit::find_planet(), jeod::DynBodyInit::find_ref_frame(), get__identifier(), jeod::DynBodyInitLvlhTransState::initialize(), jeod::DynBodyInitNedRotState::initialize(), jeod::DynBodyInitNedTransState::initialize(), jeod::DynBodyInitLvlhRotState::initialize(), jeod::DynBodyInitTransState::initialize(), jeod::BodyAttachAligned::initialize(), jeod::DynBodyInitRotState::initialize(), jeod::DynBodyFrameSwitch::initialize(), jeod::DynBodyInit::initialize(), initialize(), jeod::DynBodyInitOrbit::initialize(), jeod::DynBodyInitTransState::is_ready(), jeod::DynBodyInitRotState::is_ready(), jeod::DynBodyInit::report_failure(), validate_body_inputs(), and validate_name().

8.1.5.2 action_name

```
std::string jeod::BodyAction::action_name
```

An identifier for this action.

This can be left as empty (default value). The action_name is used only when an error is detected. The generated error message identifies the action name if supplied. The intent is to generate an error message that helps the user pinpoint the source of the error.trick_units(-)

Definition at line 168 of file body_action.hh.

Referenced by initialize().

8.1.5.3 active

```
bool jeod::BodyAction::active
```

Controls when the action is performed.

The action will be performed when the action is activated via this flag and when all other prerequisites for the action have been satisfied. The default value for this flag is class-dependent, set in various constructors. The default is true for actions that can reasonably be performed during initialization time and false for actions that are most likely performed while the simulation is running.trick_units(-)

Definition at line 147 of file body_action.hh.

Referenced by jeod::BodyAttach::BodyAttach(), jeod::BodyDetach::BodyDetach(), jeod::BodyDetachSpecific::BodyDetachSpecific(), jeod::BodyReattach::BodyReattach(), jeod::BodyDetach::is_ready(), jeod::BodyDetachSpecific::is_ready(), and is_ready().

8.1.5.4 dyn_subject

```
DynBody* jeod::BodyAction::dyn_subject [protected]
```

The DynBody of the body that is the subject of this action.

This or the subject pointer must be supplied. Actions on the body are performed by the apply methods of specific class derived from the [BodyAction](#) class.trick_units(-)

Definition at line 187 of file body_action.hh.

Referenced by jeod::BodyDetach::apply(), jeod::BodyAttachMatrix::apply(), jeod::BodyAttachAligned::apply(), jeod::BodyDetachSpecific::apply(), jeod::DynBodyFrameSwitch::apply(), jeod::DynBodyInit::apply(), jeod::DynBodyInit::apply_user_inputs(), get_subject_dyn_body(), jeod::DynBodyFrameSwitch::initialize(), jeod::DynBodyInit::initialize(), initialize(), jeod::DynBodyFrameSwitch::is_ready(), is_same_subject_body(), is_subject_dyn_body(), and set_subject_body().

8.1.5.5 mass_subject

```
MassBody* jeod::BodyAction::mass_subject [protected]
```

The MassBody of the body that is the subject of this action.

This or the dyn_subject pointer must be supplied. Actions on the body are performed by the apply methods of specific class derived from the [BodyAction](#) class.trick_units(-)

Definition at line 179 of file body_action.hh.

Referenced by jeod::BodyDetach::apply(), jeod::BodyAttachMatrix::apply(), jeod::BodyAttachAligned::apply(), jeod::BodyReattach::apply(), jeod::MassBodyInit::apply(), jeod::BodyDetachSpecific::apply(), jeod::BodyAttach↵::apply(), get_subject_dyn_body(), jeod::DynBodyFrameSwitch::initialize(), jeod::DynBodyInit::initialize(), initialize(), is_same_subject_body(), is_subject_dyn_body(), and set_subject_body().

8.1.5.6 terminate_on_error

```
bool jeod::BodyAction::terminate_on_error
```

Indicates whether errors encountered while performing the action are to terminate the simulation.

Several of the low-level methods used to perform the action do not terminate the simulation on encountering an error condition. They instead leave states unchanged and return an error indicator. This flag, if set, causes the simulation to be terminated when such an error condition occurs. The default value for this flag is true, set in the constructor.trick_units(-)

Definition at line 159 of file body_action.hh.

Referenced by jeod::BodyDetach::apply(), jeod::BodyReattach::apply(), jeod::BodyDetachSpecific::apply(), and jeod::BodyAttach::apply().

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

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

8.2 jeod::BodyActionMessages Class Reference

Specifies the message IDs used in the [BodyAction](#) model.

```
#include <body_action_messages.hh>
```

Static Public Attributes

- static char const * [fatal_error](#)
Issued when performing an action results in an error return from the method performing the action.
- static char const * [illegal_value](#)
Issued when a simple type (e.g.
- static char const * [invalid_name](#)
Issued when a name is invalid (NULL, empty, or does not name an object of the specified type).
- static char const * [invalid_object](#)
Issued when a pointer points to an object of the wrong type.
- static char const * [null_pointer](#)
Error issued when a pointer is required but was not provided.
- static char const * [not_performed](#)
Issued when a [BodyAction](#) cannot be run.
- static char const * [trace](#)
Debug message issued to trace [BodyAction](#) actions.

Private Member Functions

- [BodyActionMessages](#) (void)
- [BodyActionMessages](#) (const [BodyActionMessages](#) &)
- [BodyActionMessages](#) & operator= (const [BodyActionMessages](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__BodyActionMessages](#) ()

8.2.1 Detailed Description

Specifies the message IDs used in the [BodyAction](#) model.

Assumptions and Limitations

- This is a complete catalog of all messages sent by the [BodyAction](#) model.
- This is not an exhaustive list of all the things that can go awry.

Definition at line 81 of file [body_action_messages.hh](#).

8.2.2 Constructor & Destructor Documentation

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

```
jeod::BodyActionMessages::BodyActionMessages (
    void ) [private]
```

8.2.2.2 BodyActionMessages() [2/2]

```
jeod::BodyActionMessages::BodyActionMessages (
    const BodyActionMessages & ) [private]
```

8.2.3 Member Function Documentation

8.2.3.1 operator=()

```
BodyActionMessages& jeod::BodyActionMessages::operator= (
    const BodyActionMessages & ) [private]
```

8.2.4 Friends And Related Function Documentation

8.2.4.1 init_attrjeod__BodyActionMessages

```
void init_attrjeod__BodyActionMessages ( ) [friend]
```

8.2.4.2 InputProcessor

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

Definition at line 84 of file body_action_messages.hh.

8.2.5 Field Documentation

8.2.5.1 fatal_error

```
char const * jeod::BodyActionMessages::fatal_error [static]
```

Initial value:

```
=
    "dynamics/body_action/" "fatal_error"
```

Issued when performing an action results in an error return from the method performing the action.

trick_units(-)

Definition at line 95 of file body_action_messages.hh.

Referenced by jeod::BodyDetach::apply(), jeod::BodyReattach::apply(), jeod::BodyDetachSpecific::apply(), jeod::BodyAttach::apply(), and jeod::BodyAction::validate_body_inputs().

8.2.5.2 illegal_value

```
char const * jeod::BodyActionMessages::illegal_value [static]
```

Initial value:

```
=
    "dynamics/body_action/" "illegal_value"
```

Issued when a simple type (e.g.

an enum) has an illegal value. `trick_units(-)`

Definition at line 100 of file `body_action_messages.hh`.

Referenced by `jeod::DynBodyInitLvHState::apply()`, `jeod::DynBodyInitNedState::apply()`, `jeod::DynBodyInitOrbit::apply()`, `jeod::DynBodyInitNedRotState::initialize()`, `jeod::DynBodyInitLvHTransState::initialize()`, `jeod::DynBodyInitNedTransState::initialize()`, `jeod::DynBodyInitLvHRotState::initialize()`, `jeod::DynBodyInitTransState::initialize()`, `jeod::DynBodyInitRotState::initialize()`, and `jeod::DynBodyInitOrbit::initialize()`.

8.2.5.3 invalid_name

```
char const * jeod::BodyActionMessages::invalid_name [static]
```

Initial value:

```
=
    "dynamics/body_action/" "invalid_name"
```

Issued when a name is invalid (NULL, empty, or does not name an object of the specified type).

`trick_units(-)`

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

Referenced by `jeod::DynBodyInit::compute_rotational_state()`, `jeod::DynBodyInit::compute_translational_state()`, `jeod::DynBodyInit::find_body_frame()`, `jeod::DynBodyInit::find_dyn_body()`, `jeod::DynBodyInit::find_planet()`, `jeod::DynBodyInit::find_ref_frame()`, `jeod::BodyAttachAligned::initialize()`, `jeod::DynBodyFrameSwitch::initialize()`, `jeod::DynBodyInitOrbit::initialize()`, and `jeod::BodyAction::validate_name()`.

8.2.5.4 invalid_object

```
char const * jeod::BodyActionMessages::invalid_object [static]
```

Initial value:

```
=
    "dynamics/body_action/" "invalid_object"
```

Issued when a pointer points to an object of the wrong type.

`trick_units(-)`

Definition at line 111 of file `body_action_messages.hh`.

Referenced by `jeod::DynBodyFrameSwitch::initialize()`, `jeod::DynBodyInit::initialize()`, `jeod::DynBodyInitOrbit::initialize()`, `jeod::DynBodyInitTransState::is_ready()`, and `jeod::DynBodyInitRotState::is_ready()`.

8.2.5.5 not_performed

```
char const * jeod::BodyActionMessages::not_performed [static]
```

Initial value:

```
=
    "dynamics/body_action/" "not_performed"
```

Issued when a [BodyAction](#) cannot be run.

trick_units(-)

Definition at line 121 of file body_action_messages.hh.

Referenced by jeod::BodyDetach::apply(), jeod::BodyDetachSpecific::apply(), jeod::BodyAttach::apply(), and jeod::DynBodyInit::report_failure().

8.2.5.6 null_pointer

```
char const * jeod::BodyActionMessages::null_pointer [static]
```

Initial value:

```
=
    "dynamics/body_action/" "null_pointer"
```

Error issued when a pointer is required but was not provided.

trick_units(-)

Definition at line 116 of file body_action_messages.hh.

Referenced by jeod::DynBodyInitLvlhRotState::initialize(), and jeod::BodyAction::validate_body_inputs().

8.2.5.7 trace

```
char const * jeod::BodyActionMessages::trace [static]
```

Initial value:

```
=
    "dynamics/body_action/" "trace"
```

Debug message issued to trace [BodyAction](#) actions.

trick_units(-)

Definition at line 126 of file body_action_messages.hh.

Referenced by jeod::BodyDetach::apply(), jeod::BodyReattach::apply(), jeod::MassBodyInit::apply(), jeod::BodyDetachSpecific::apply(), jeod::BodyAttach::apply(), jeod::DynBodyFrameSwitch::apply(), and jeod::DynBodyInit::apply().

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

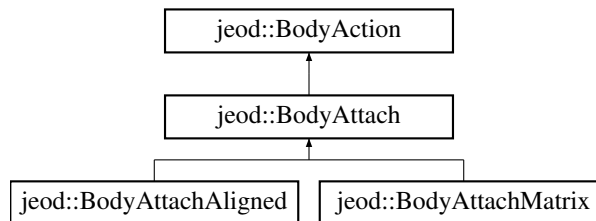
- [body_action_messages.hh](#)
- [body_action_messages.cc](#)

8.3 jeod::BodyAttach Class Reference

Provides the basic ability to attach one MassBody to another.

```
#include <body_attach.hh>
```

Inheritance diagram for jeod::BodyAttach:



Public Member Functions

- void [set_parent_body](#) (MassBody &mass_body_in)
Set the parent mass body of this action.
- void [set_parent_body](#) (DynBody &dyn_body_in)
Set the parent dyn body of this action.
- [BodyAttach](#) ()
Construct a MassBodyAttach.
- virtual [~BodyAttach](#) ()
Destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize a MassBodyAttach.
- virtual void [apply](#) (DynManager &dyn_manager)
A derived class presumably has performed the attachment, which may not have worked, and forwarded the apply call to this method.

Data Fields

- bool [succeeded](#)
Did the attachment succeed?

Protected Attributes

- MassBody * [mass_parent](#)
The MassBody corresponding to which the subject body is to be attached, directly if the subject body is a root body, and indirectly by attaching the subject body's root body to the parent body otherwise.
- DynBody * [dyn_parent](#)
The DynBody corresponding to which the subject body is to be attached, directly if the subject body is a root body, and indirectly by attaching the subject body's root body to the parent body otherwise.

Private Member Functions

- [BodyAttach](#) (const [BodyAttach](#) &)
- [BodyAttach](#) & [operator=](#) (const [BodyAttach](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__BodyAttach](#) ()

Additional Inherited Members

8.3.1 Detailed Description

Provides the basic ability to attach one MassBody to another.

This can be either an initialization or asynchronous [BodyAction](#). The action will be performed when the sim user or some simulation job enables the active flag.

MassBodyAttach actions that are ready at simulation initialization time are run as a part of the initialization process, sandwiched between initializing mass properties and initializing state. Attach actions that are not ready at initialization time remain in the pending actions queue until the active flag is set.

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

8.3.2 Constructor & Destructor Documentation

8.3.2.1 BodyAttach() [1/2]

```
jeod::BodyAttach::BodyAttach (
    const BodyAttach & ) [private]
```

8.3.2.2 BodyAttach() [2/2]

```
jeod::BodyAttach::BodyAttach (
    void )
```

Construct a MassBodyAttach.

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

References `jeod::BodyAction::active`.

8.3.2.3 ~BodyAttach()

```
jeod::BodyAttach::~BodyAttach (
    void ) [inline], [virtual]
```

Destructor.

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

8.3.3 Member Function Documentation

8.3.3.1 apply()

```
void jeod::BodyAttach::apply (
    DynManager & dyn_manager ) [virtual]
```

A derived class presumably has performed the attachment, which may not have worked, and forwarded the apply call to this method.

This method acts on the status from that child class attachment.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Jeod manager
----------------	--------------------	--------------

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

Reimplemented in [jeod::BodyAttachAligned](#), and [jeod::BodyAttachMatrix](#).

Definition at line 103 of file `body_attach.cc`.

References [jeod::BodyAction::action_identifier](#), [jeod::BodyAction::apply\(\)](#), [jeod::BodyActionMessages::fatal_error](#), [mass_parent](#), [jeod::BodyAction::mass_subject](#), [jeod::BodyActionMessages::not_performed](#), [succeeded](#), [jeod::BodyAction::terminate_on_error](#), and [jeod::BodyActionMessages::trace](#).

Referenced by [jeod::BodyAttachMatrix::apply\(\)](#), and [jeod::BodyAttachAligned::apply\(\)](#).

8.3.3.2 initialize()

```
void jeod::BodyAttach::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize a MassBodyAttach.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager
----------------	--------------------	------------------

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

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

Definition at line 72 of file `body_attach.cc`.

References [dyn_parent](#), [jeod::BodyAction::initialize\(\)](#), [mass_parent](#), and [jeod::BodyAction::validate_body_inputs\(\)](#).

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

8.3.3.3 operator=()

```
BodyAttach& jeod::BodyAttach::operator= (
    const BodyAttach & ) [private]
```

8.3.3.4 set_parent_body() [1/2]

```
void jeod::BodyAttach::set_parent_body (
    MassBody & mass_body_in )
```

Set the parent mass body of this action.

Resets dyn_parent to null

Definition at line 84 of file body_attach.cc.

References dyn_parent, and mass_parent.

8.3.3.5 set_parent_body() [2/2]

```
void jeod::BodyAttach::set_parent_body (
    DynBody & dyn_body_in )
```

Set the parent dyn body of this action.

Resets mass_parent to null

Definition at line 90 of file body_attach.cc.

References dyn_parent, and mass_parent.

8.3.4 Friends And Related Function Documentation**8.3.4.1 init_attrjeod__BodyAttach**

```
void init_attrjeod__BodyAttach ( ) [friend]
```

8.3.4.2 InputProcessor

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

Definition at line 96 of file body_attach.hh.

8.3.5 Field Documentation

8.3.5.1 dyn_parent

```
DynBody* jeod::BodyAttach::dyn_parent [protected]
```

The DynBody corresponding to which the subject body is to be attached, directly if the subject body is a root body, and indirectly by attaching the subject body's root body to the parent body otherwise.

This or the parent pointer must be supplied.trick_units(–)

Definition at line 134 of file body_attach.hh.

Referenced by jeod::BodyAttachMatrix::apply(), jeod::BodyAttachAligned::apply(), initialize(), and set_parent_↔body().

8.3.5.2 mass_parent

```
MassBody* jeod::BodyAttach::mass_parent [protected]
```

The MassBody corresponding to which the subject body is to be attached, directly if the subject body is a root body, and indirectly by attaching the subject body's root body to the parent body otherwise.

This or the dyn_parent pointer must be supplied.trick_units(–)

Definition at line 126 of file body_attach.hh.

Referenced by jeod::BodyAttachMatrix::apply(), jeod::BodyAttachAligned::apply(), apply(), initialize(), and set_↔parent_body().

8.3.5.3 succeeded

```
bool jeod::BodyAttach::succeeded
```

Did the attachment succeed?

trick_units(–)

Definition at line 116 of file body_attach.hh.

Referenced by jeod::BodyAttachMatrix::apply(), jeod::BodyAttachAligned::apply(), and apply().

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

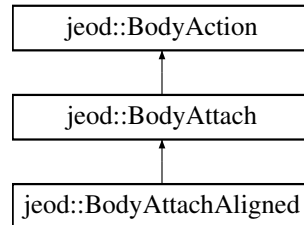
- [body_attach.hh](#)
- [body_attach.cc](#)

8.4 jeod::BodyAttachAligned Class Reference

Attaches a pair of MassBody objects at a pair of MassPoints.

```
#include <body_attach_aligned.hh>
```

Inheritance diagram for jeod::BodyAttachAligned:



Public Member Functions

- [BodyAttachAligned](#) ()
Construct a MassBodyAttachAligned.
- virtual [~BodyAttachAligned](#) ()
Destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize a MassBodyAttach.
- virtual void [apply](#) (DynManager &dyn_manager)
Initialize the core mass properties of the subject MassBody.

Data Fields

- std::string [subject_point_name](#)
The name of the mass point on the subject mass body to be attached to to the parent_point_name mass point on the parent mass body.
- std::string [parent_point_name](#)
The name of the mass point on the parent mass body to be attached to to the mass pointed named subject_point_↔ name on the subject mass body.

Private Member Functions

- [BodyAttachAligned](#) (const [BodyAttachAligned](#) &)
- [BodyAttachAligned](#) & [operator=](#) (const [BodyAttachAligned](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__BodyAttachAligned](#) ()

Additional Inherited Members

8.4.1 Detailed Description

Attaches a pair of MassBody objects at a pair of MassPoints.

When the action is ready, the attachment proceeds as follows:

- The points indicated by the subject and parent mass point names will be coincident after attachment is complete.
- The orientation between the two reference frames associated with the two attach points is a 180 degree yaw.

Definition at line 87 of file body_attach_aligned.hh.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 BodyAttachAligned() [1/2]

```
jeod::BodyAttachAligned::BodyAttachAligned (
    void )
```

Construct a MassBodyAttachAligned.

Definition at line 54 of file body_attach_aligned.cc.

8.4.2.2 ~BodyAttachAligned()

```
jeod::BodyAttachAligned::~~BodyAttachAligned (
    void ) [inline], [virtual]
```

Destructor.

Definition at line 140 of file body_attach_aligned.hh.

8.4.2.3 BodyAttachAligned() [2/2]

```
jeod::BodyAttachAligned::BodyAttachAligned (
    const BodyAttachAligned & ) [private]
```

8.4.3 Member Function Documentation

8.4.3.1 apply()

```
void jeod::BodyAttachAligned::apply (
    DynManager & dyn_manager ) [virtual]
```

Initialize the core mass properties of the subject MassBody.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Jeod manager
----------------	--------------------	--------------

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

Definition at line 104 of file `body_attach_aligned.cc`.

References `jeod::BodyAttach::apply()`, `jeod::BodyAttach::dyn_parent`, `jeod::BodyAction::dyn_subject`, `jeod::BodyAttach::mass_parent`, `jeod::BodyAction::mass_subject`, `parent_point_name`, `subject_point_name`, and `jeod::BodyAttach::succeeded`.

8.4.3.2 initialize()

```
void jeod::BodyAttachAligned::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize a MassBodyAttach.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager
----------------	--------------------	------------------

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

Definition at line 69 of file `body_attach_aligned.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyAttach::initialize()`, `jeod::BodyActionMessages::invalid_name`, `parent_point_name`, and `subject_point_name`.

8.4.3.3 operator=()

```
BodyAttachAligned& jeod::BodyAttachAligned::operator= (
    const BodyAttachAligned & ) [private]
```

8.4.4 Friends And Related Function Documentation**8.4.4.1 init_attrjeod__BodyAttachAligned**

```
void init_attrjeod__BodyAttachAligned ( ) [friend]
```

8.4.4.2 InputProcessor

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

Definition at line 89 of file `body_attach_aligned.hh`.

8.4.5 Field Documentation

8.4.5.1 parent_point_name

```
std::string jeod::BodyAttachAligned::parent_point_name
```

The name of the mass point on the parent mass body to be attached to to the mass pointed named `subject_point_name` on the subject mass body.

The supplied name can omit the parent mass body name dot prefix if desired. `trick_units(-)`

Definition at line 108 of file `body_attach_aligned.hh`.

Referenced by `apply()`, and `initialize()`.

8.4.5.2 subject_point_name

```
std::string jeod::BodyAttachAligned::subject_point_name
```

The name of the mass point on the subject mass body to be attached to to the `parent_point_name` mass point on the parent mass body.

The supplied name can omit the subject mass body name dot prefix if desired. `trick_units(-)`

Definition at line 101 of file `body_attach_aligned.hh`.

Referenced by `apply()`, and `initialize()`.

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

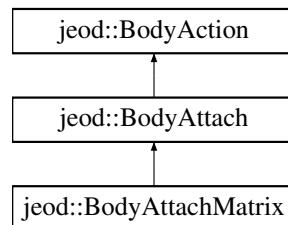
- [body_attach_aligned.hh](#)
- [body_attach_aligned.cc](#)

8.5 jeod::BodyAttachMatrix Class Reference

Attaches a pair of MassBody objects using the offset+matrix attach mechanism.

```
#include <body_attach_matrix.hh>
```

Inheritance diagram for jeod::BodyAttachMatrix:



Public Member Functions

- [BodyAttachMatrix](#) ()
Construct a MassBodyAttachMatrix.
- virtual [~BodyAttachMatrix](#) ()
Destructor.
- virtual void [apply](#) (DynManager &dyn_manager)
Initialize the core mass properties of the subject MassBody.

Data Fields

- double [offset_pstr_cstr_pstr](#) [3]
Location of this body's structural origin with respect to the new parent body's structural origin, specified in structural coordinates of the new parent body.
- Orientation [pstr_cstr](#)
Orientation of child's structural frame with respect to that of the new parent; sense is parent-to-child.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__BodyAttachMatrix](#) ()

Additional Inherited Members

8.5.1 Detailed Description

Attaches a pair of MassBody objects using the offset+matrix attach mechanism.

When the action is ready, the attachment is made such that:

- The displacement between the origins of the parent and subject bodies' structural frames is that given by the `offset_pstr_cstr_pstr` data member.
- The orientation between these two reference frames's axes is that given by the `pstr_cstr` data member.

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

8.5.2 Constructor & Destructor Documentation

8.5.2.1 BodyAttachMatrix()

```
jeod::BodyAttachMatrix::BodyAttachMatrix (
    void )
```

Construct a MassBodyAttachMatrix.

Definition at line 51 of file body_attach_matrix.cc.

References `offset_pstr_cstr_pstr`.

8.5.2.2 ~BodyAttachMatrix()

```
jeod::BodyAttachMatrix::~~BodyAttachMatrix (
    void ) [inline], [virtual]
```

Destructor.

Definition at line 130 of file body_attach_matrix.hh.

8.5.3 Member Function Documentation

8.5.3.1 apply()

```
void jeod::BodyAttachMatrix::apply (
    DynManager & dyn_manager ) [virtual]
```

Initialize the core mass properties of the subject MassBody.

Parameters

<code>in, out</code>	<code>dyn_manager</code>	Jeod manager
----------------------	--------------------------	--------------

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

Definition at line 65 of file body_attach_matrix.cc.

References `jeod::BodyAttach::apply()`, `jeod::BodyAttach::dyn_parent`, `jeod::BodyAction::dyn_subject`, `jeod::BodyAttach::mass_parent`, `jeod::BodyAction::mass_subject`, `offset_pstr_cstr_pstr`, `pstr_cstr`, and `jeod::BodyAttach::succeeded`.

8.5.4 Friends And Related Function Documentation

8.5.4.1 init_attrjeod__BodyAttachMatrix

```
void init_attrjeod__BodyAttachMatrix ( ) [friend]
```

8.5.4.2 InputProcessor

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

Definition at line 89 of file body_attach_matrix.hh.

8.5.5 Field Documentation

8.5.5.1 offset_pstr_cstr_pstr

```
double jeod::BodyAttachMatrix::offset_pstr_cstr_pstr[3]
```

Location of this body's structural origin with respect to the new parent body's structural origin, specified in structural coordinates of the new parent body.

trick_units(m)

Definition at line 101 of file body_attach_matrix.hh.

Referenced by apply(), and BodyAttachMatrix().

8.5.5.2 pstr_cstr

```
Orientation jeod::BodyAttachMatrix::pstr_cstr
```

Orientation of child's structural frame with respect to that of the new parent; sense is parent-to-child.

trick_units(-)

Definition at line 107 of file body_attach_matrix.hh.

Referenced by apply().

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

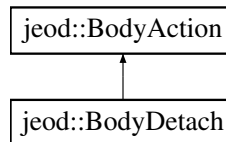
- [body_attach_matrix.hh](#)
- [body_attach_matrix.cc](#)

8.6 jeod::BodyDetach Class Reference

Provides the basic ability to detach one MassBody from another.

```
#include <body_detach.hh>
```

Inheritance diagram for jeod::BodyDetach:



Public Member Functions

- [BodyDetach](#) ()
Construct a MassBodyDetach.
- virtual [~BodyDetach](#) ()
Destructor.
- virtual void [apply](#) (DynManager &dyn_manager)
Detach the body from its parent.
- virtual bool [is_ready](#) (void)
Queries whether the "active" flag has been set.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__BodyDetach](#) ()

Additional Inherited Members

8.6.1 Detailed Description

Provides the basic ability to detach one MassBody from another.

This is inherently an asynchronous [BodyAction](#). The [is_ready\(\)](#) method simply returns the action's active flag. The action will be performed when the sim user or some simulation job enables the active flag.

The basic detachment action is to cause a body to detach from its immediate parent body. Subclasses can cause bodies to detach elsewhere.

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

8.6.2 Constructor & Destructor Documentation

8.6.2.1 BodyDetach()

```
jeod::BodyDetach::BodyDetach (
    void )
```

Construct a MassBodyDetach.

Definition at line 50 of file body_detach.cc.

References [jeod::BodyAction::active](#).

8.6.2.2 ~BodyDetach()

```
jeod::BodyDetach::~~BodyDetach (
    void ) [inline], [virtual]
```

Destructor.

Definition at line 123 of file body_detach.hh.

8.6.3 Member Function Documentation

8.6.3.1 apply()

```
void jeod::BodyDetach::apply (
    DynManager & dyn_manager ) [virtual]
```

Detach the body from its parent.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Jeod manager
----------------	--------------------	--------------

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

Definition at line 65 of file body_detach.cc.

References [jeod::BodyAction::action_identifier](#), [jeod::BodyAction::apply\(\)](#), [jeod::BodyAction::dyn_subject](#), [jeod::BodyActionMessages::fatal_error](#), [jeod::BodyAction::mass_subject](#), [jeod::BodyActionMessages::not_performed](#), [jeod::BodyAction::terminate_on_error](#), and [jeod::BodyActionMessages::trace](#).

8.6.3.2 is_ready()

```
bool jeod::BodyDetach::is_ready (
    void ) [virtual]
```

Queries whether the "active" flag has been set.

Returns

Can detach process run?

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

Definition at line 119 of file body_detach.cc.

References [jeod::BodyAction::active](#).

8.6.4 Friends And Related Function Documentation

8.6.4.1 init_attrjeod__BodyDetach

```
void init_attrjeod__BodyDetach ( ) [friend]
```

8.6.4.2 InputProcessor

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

Definition at line 92 of file body_detach.hh.

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

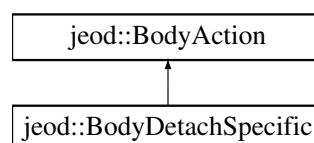
- [body_detach.hh](#)
- [body_detach.cc](#)

8.7 jeod::BodyDetachSpecific Class Reference

Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach↔_from body.

```
#include <body_detach_specific.hh>
```

Inheritance diagram for [jeod::BodyDetachSpecific](#):



Public Member Functions

- void [set_detach_from_body](#) (MassBody &mass_body_in)
Set the subject mass body of this action.
- void [set_detach_from_body](#) (DynBody &dyn_body_in)
Set the subject mass body of this action.
- [BodyDetachSpecific](#) ()
Construct a [BodyDetachSpecific](#).
- virtual [~BodyDetachSpecific](#) ()
Destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize a [BodyDetachSpecific](#).
- virtual void [apply](#) (DynManager &dyn_manager)
Detach the body from its parent.
- virtual bool [is_ready](#) (void)
Queries whether the "active" flag has been set.

Protected Attributes

- MassBody * [mass_detach_from](#)
The mass body from the subject of this action is to detach.
- DynBody * [dyn_detach_from](#)
The dynamic body from the subject of this action is to detach.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__BodyDetachSpecific](#) ()

Additional Inherited Members

8.7.1 Detailed Description

Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach↵
_from body.

This method works between two dynamic bodies (DynBody) or mass bodies (MassBody), but not mixtures of the two classes. The subject body itself is detached from its parent if and only if the specified detach_from body is the subject body's immediate parent. In the case that the detach_from body is some indirect parent, the body that detaches is the the immediate child body of the detach_from body that is along the connectivity path from the subject body to the detach_from * body. Specifying a detach_from body that is not a parent (direct or indirect) body of the subject body is an error.

Definition at line 94 of file body_detach_specific.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 BodyDetachSpecific()

```
jeod::BodyDetachSpecific::BodyDetachSpecific (
    void )
```

Construct a [BodyDetachSpecific](#).

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

References `jeod::BodyAction::active`.

8.7.2.2 ~BodyDetachSpecific()

```
jeod::BodyDetachSpecific::~~BodyDetachSpecific (
    void ) [inline], [virtual]
```

Destructor.

Definition at line 163 of file `body_detach_specific.hh`.

8.7.3 Member Function Documentation

8.7.3.1 apply()

```
void jeod::BodyDetachSpecific::apply (
    DynManager & dyn_manager ) [virtual]
```

Detach the body from its parent.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Definition at line 87 of file `body_detach_specific.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyAction::apply()`, `dyn_detach_from`, `jeod::BodyAction↵
::dyn_subject`, `jeod::BodyActionMessages::fatal_error`, `mass_detach_from`, `jeod::BodyAction::mass_subject`, `jeod↵
::BodyActionMessages::not_performed`, `jeod::BodyAction::terminate_on_error`, and `jeod::BodyActionMessages↵
::trace`.

8.7.3.2 initialize()

```
void jeod::BodyDetachSpecific::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize a [BodyDetachSpecific](#).

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager
----------------	--------------------	------------------

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

Definition at line 69 of file `body_detach_specific.cc`.

References `dyn_detach_from`, `jeod::BodyAction::initialize()`, `mass_detach_from`, and `jeod::BodyAction::validate_↔body_inputs()`.

8.7.3.3 is_ready()

```
bool jeod::BodyDetachSpecific::is_ready (
    void ) [virtual]
```

Queries whether the "active" flag has been set.

Returns

Can detach process run?

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

Definition at line 171 of file `body_detach_specific.cc`.

References `jeod::BodyAction::active`.

8.7.3.4 set_detach_from_body() [1/2]

```
void jeod::BodyDetachSpecific::set_detach_from_body (
    MassBody & mass_body_in )
```

Set the subject mass body of this action.

Resets `dyn_subject` to null

Definition at line 154 of file `body_detach_specific.cc`.

References `dyn_detach_from`, and `mass_detach_from`.

8.7.3.5 set_detach_from_body() [2/2]

```
void jeod::BodyDetachSpecific::set_detach_from_body (
    DynBody & dyn_body_in )
```

Set the subject mass body of this action.

Resets dyn_subject to null

Definition at line 160 of file body_detach_specific.cc.

References dyn_detach_from, and mass_detach_from.

8.7.4 Friends And Related Function Documentation

8.7.4.1 init_attrjeod__BodyDetachSpecific

```
void init_attrjeod__BodyDetachSpecific ( ) [friend]
```

8.7.4.2 InputProcessor

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

Definition at line 96 of file body_detach_specific.hh.

8.7.5 Field Documentation

8.7.5.1 dyn_detach_from

```
DynBody* jeod::BodyDetachSpecific::dyn_detach_from [protected]
```

The dynamic body from the subject of this action is to detach.

This pointer or the detach_from member must be supplied for dynamic body detachment. The detachment is performed between the mass_detach_from object and the direct descendant of the mass_detach_from object that is in the parental lineage from the subject body to the mass_detach_from body.trick_units(-)

Definition at line 134 of file body_detach_specific.hh.

Referenced by apply(), initialize(), and set_detach_from_body().

8.7.5.2 mass_detach_from

```
MassBody* jeod::BodyDetachSpecific::mass_detach_from [protected]
```

The mass body from the subject of this action is to detach.

This pointer must be supplied for pure MassBody detachments. The initialize method will attempt to determine if this MassBody refers to a DynBody. The detachment is performed between the mass_detach_from object and the direct descendant of the detach_from object that is in the parental lineage from the subject body to the detach_from body.trick_units(-)

Definition at line 124 of file body_detach_specific.hh.

Referenced by apply(), initialize(), and set_detach_from_body().

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

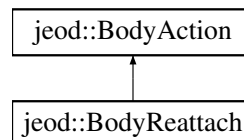
- [body_detach_specific.hh](#)
- [body_detach_specific.cc](#)

8.8 jeod::BodyReattach Class Reference

Alters the nature of an existing attachment.

```
#include <body_reattach.hh>
```

Inheritance diagram for jeod::BodyReattach:



Public Member Functions

- [BodyReattach](#) ()
Construct a MassBodyReattach.
- virtual [~BodyReattach](#) ()
Destructor.
- virtual void [apply](#) (DynManager &dyn_manager)
Initialize the core mass properties of the subject MassBody.

Data Fields

- double [offset_pstr_cstr_pstr](#) [3]
Location of this body's structural origin with respect to the new parent body's structural origin, specified in structural coordinates of the new parent body.
- Orientation [pstr_cstr](#)
Orientation of child's structural frame with respect to that of the new parent; sense is parent-to-child.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__BodyReattach](#) ()

Additional Inherited Members

8.8.1 Detailed Description

Alters the nature of an existing attachment.

When the action is ready, the attachment is altered such that:

- The displacement between the origins of the parent and subject bodies' structural frames is that given by the `offset_pstr_cstr_pstr` data member.
- The orientation between these two reference frames's axes is that given by the `pstr_cstr` data member. Note that no parent body is specified. Reattachment does not change the attachment tree. It instead alters the physical relationships between a pair of objects that are already attached.

Definition at line 91 of file `body_reattach.hh`.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 BodyReattach()

```
jeod::BodyReattach::BodyReattach (
    void )
```

Construct a `MassBodyReattach`.

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

References `jeod::BodyAction::active`, and `offset_pstr_cstr_pstr`.

8.8.2.2 ~BodyReattach()

```
jeod::BodyReattach::~~BodyReattach (
    void ) [inline], [virtual]
```

Destructor.

Definition at line 134 of file `body_reattach.hh`.

8.8.3 Member Function Documentation

8.8.3.1 apply()

```
void jeod::BodyReattach::apply (
    DynManager & dyn_manager ) [virtual]
```

Initialize the core mass properties of the subject `MassBody`.

Parameters

<code>in, out</code>	<code>dyn_manager</code>	Jeod manager
----------------------	--------------------------	--------------

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

Definition at line 65 of file `body_reattach.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyAction::apply()`, `jeod::BodyActionMessages::fatal_error`, `jeod::BodyAction::mass_subject`, `offset_pstr_cstr_pstr`, `pstr_cstr`, `jeod::BodyAction::terminate_on_error`, and `jeod::BodyActionMessages::trace`.

8.8.4 Friends And Related Function Documentation

8.8.4.1 `init_attrjeod__BodyReattach`

```
void init_attrjeod__BodyReattach ( ) [friend]
```

8.8.4.2 `InputProcessor`

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

Definition at line 93 of file `body_reattach.hh`.

8.8.5 Field Documentation

8.8.5.1 `offset_pstr_cstr_pstr`

```
double jeod::BodyReattach::offset_pstr_cstr_pstr[3]
```

Location of this body's structural origin with respect to the new parent body's structural origin, specified in structural coordinates of the new parent body.

`trick_units(m)`

Definition at line 105 of file `body_reattach.hh`.

Referenced by `apply()`, and `BodyReattach()`.

8.8.5.2 pstr_cstr

Orientation jeod::BodyReattach::pstr_cstr

Orientation of child's structural frame with respect to that of the new parent; sense is parent-to-child.

trick_units(-)

Definition at line 111 of file body_reattach.hh.

Referenced by apply().

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

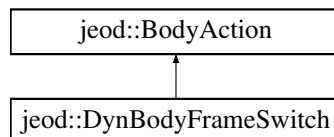
- [body_reattach.hh](#)
- [body_reattach.cc](#)

8.9 jeod::DynBodyFrameSwitch Class Reference

Switch a DynBody's integration frame to a specified frame when the body switches to that integration frame's sphere of influence.

```
#include <dyn_body_frame_switch.hh>
```

Inheritance diagram for jeod::DynBodyFrameSwitch:



Public Types

- enum [SwitchSense](#) { [SwitchOnApproach](#) = 0, [SwitchOnDeparture](#) = 1 }
- Specifies whether the [is_ready\(\)](#) method is to look for the vehicle entering ([SwitchOnApproach](#)) the new integration frame's sphere of influence versus leaving ([SwitchOnDeparture](#)) the current integration frame's sphere of influence.*

Public Member Functions

- [DynBodyFrameSwitch](#) ()
Construct a [DynBodyFrameSwitch](#) instance.
- virtual [~DynBodyFrameSwitch](#) ()
Destruct a [DynBodyFrameSwitch](#) instance.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialization a [DynBodyFrameSwitch](#) instance.
- virtual void [apply](#) (DynManager &dyn_manager)
Switch reference frames.
- virtual bool [is_ready](#) (void)
Determine whether it is time to switch frames.

Data Fields

- std::string [integ_frame_name](#)
The name of the new integration frame.
- [SwitchSense](#) [switch_sense](#)
Indicates whether the switch occurs when the subject DynBody enters a sphere of influence around the new integration frame or leaves a sphere of influence around of the current integration frame.
- bool [sort_grav_controls](#)
If set, the body's gravitational controls are sorted in ascending acceleration magnitude.
- double [switch_distance](#)
The radius of the sphere of influence.

Protected Attributes

- EphemerisRefFrame * [integ_frame](#)
The reference frame corresponding to the input integ_frame_name.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyFrameSwitch](#) ()

Additional Inherited Members

8.9.1 Detailed Description

Switch a DynBody's integration frame to a specified frame when the body switches to that integration frame's sphere of influence.

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

8.9.2 Member Enumeration Documentation

8.9.2.1 SwitchSense

```
enum jeod::DynBodyFrameSwitch::SwitchSense
```

Specifies whether the [is_ready\(\)](#) method is to look for the vehicle entering (SwitchOnApproach) the new integration frame's sphere of influence versus leaving (SwitchOnDeparture) the current integration frame's sphere of influence.

Enumerator

SwitchOnApproach	
SwitchOnDeparture	

Definition at line 103 of file dyn_body_frame_switch.hh.

8.9.3 Constructor & Destructor Documentation

8.9.3.1 DynBodyFrameSwitch()

```
jeod::DynBodyFrameSwitch::DynBodyFrameSwitch (
    void )
```

Construct a [DynBodyFrameSwitch](#) instance.

Definition at line 57 of file dyn_body_frame_switch.cc.

8.9.3.2 ~DynBodyFrameSwitch()

```
jeod::DynBodyFrameSwitch::~~DynBodyFrameSwitch (
    void ) [virtual]
```

Destruct a [DynBodyFrameSwitch](#) instance.

Definition at line 73 of file dyn_body_frame_switch.cc.

8.9.4 Member Function Documentation

8.9.4.1 apply()

```
void jeod::DynBodyFrameSwitch::apply (
    DynManager & dyn_manager ) [virtual]
```

Switch reference frames.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Jeod manager
----------------	--------------------	--------------

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

Definition at line 144 of file dyn_body_frame_switch.cc.

References [jeod::BodyAction::action_identifier](#), [jeod::BodyAction::apply\(\)](#), [jeod::BodyAction::dyn_subject](#), [integ_↔ frame](#), [integ_frame_name](#), [sort_grav_controls](#), and [jeod::BodyActionMessages::trace](#).

8.9.4.2 initialize()

```
void jeod::DynBodyFrameSwitch::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialization a [DynBodyFrameSwitch](#) instance.

Parameters

in, out	<i>dyn_manager</i>	Dynamics manager
---------	--------------------	------------------

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

Definition at line 85 of file `dyn_body_frame_switch.cc`.

References [jeod::BodyAction::action_identifier](#), [jeod::BodyAction::dyn_subject](#), [jeod::BodyAction::initialize\(\)](#), [integ_frame](#), [integ_frame_name](#), [jeod::BodyActionMessages::invalid_name](#), [jeod::BodyActionMessages::invalid↵_object](#), and [jeod::BodyAction::mass_subject](#).

8.9.4.3 is_ready()

```
bool jeod::DynBodyFrameSwitch::is_ready (
    void ) [virtual]
```

Determine whether it is time to switch frames.

A frame-switch action is ready if it is activated and if the vehicle has entered/left the appropriate sphere of influence.

Returns

Can action be applied?

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

Definition at line 194 of file `dyn_body_frame_switch.cc`.

References [jeod::BodyAction::dyn_subject](#), [integ_frame](#), [jeod::BodyAction::is_ready\(\)](#), [switch_distance](#), [switch_↵sense](#), and [SwitchOnApproach](#).

8.9.5 Friends And Related Function Documentation

8.9.5.1 init_attrjeod__DynBodyFrameSwitch

```
void init_attrjeod__DynBodyFrameSwitch ( ) [friend]
```


8.9.5.2 InputProcessor

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

Definition at line 90 of file dyn_body_frame_switch.hh.

8.9.6 Field Documentation

8.9.6.1 integ_frame

```
EphemerisRefFrame* jeod::DynBodyFrameSwitch::integ_frame [protected]
```

The reference frame corresponding to the input integ_frame_name.

trick_io(**)

Definition at line 151 of file dyn_body_frame_switch.hh.

Referenced by apply(), initialize(), and is_ready().

8.9.6.2 integ_frame_name

```
std::string jeod::DynBodyFrameSwitch::integ_frame_name
```

The name of the new integration frame.

This name must specify a valid valid integration frame. Failure to do so constitutes a fatal error.trick_units(-)

Definition at line 126 of file dyn_body_frame_switch.hh.

Referenced by apply(), and initialize().

8.9.6.3 sort_grav_controls

```
bool jeod::DynBodyFrameSwitch::sort_grav_controls
```

If set, the body's gravitational controls are sorted in ascending acceleration magnitude.

trick_units(-)

Definition at line 139 of file dyn_body_frame_switch.hh.

Referenced by apply().

8.9.6.4 switch_distance

```
double jeod::DynBodyFrameSwitch::switch_distance
```

The radius of the sphere of influence.

trick_units(m)

Definition at line 144 of file dyn_body_frame_switch.hh.

Referenced by is_ready().

8.9.6.5 switch_sense

```
SwitchSense jeod::DynBodyFrameSwitch::switch_sense
```

Indicates whether the switch occurs when the subject DynBody enters a sphere of influence around the new integration frame or leaves a sphere sphere of influence around of the current integration frame.

trick_units(-)

Definition at line 133 of file dyn_body_frame_switch.hh.

Referenced by is_ready().

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

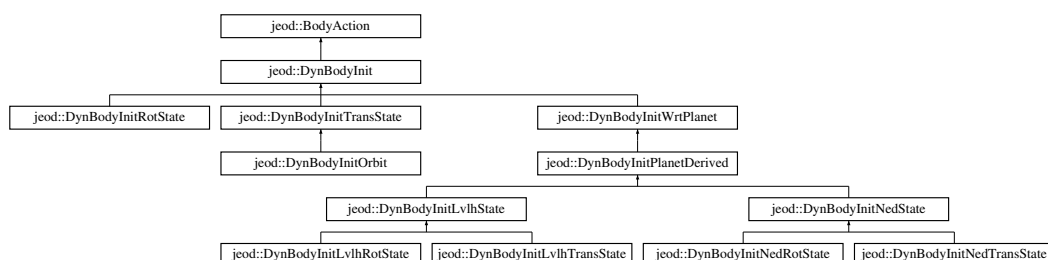
- [dyn_body_frame_switch.hh](#)
- [dyn_body_frame_switch.cc](#)

8.10 jeod::DynBodyInit Class Reference

Base class for initialize the state of a DynBody.

```
#include <dyn_body_init.hh>
```

Inheritance diagram for jeod::DynBodyInit:



Public Member Functions

- [DynBodyInit](#) ()
Construct a [DynBodyInit](#).
- virtual [~DynBodyInit](#) ()
Destruct a [DynBodyInit](#).
- virtual void [report_failure](#) (void)
Report on an initializer that could not be processed.
- virtual void [initialize](#) (DynManager &dyn_manager)
Complete initialization of a [DynBodyInit](#).
- virtual RefFrameItems::Items [initializes_what](#) (void)
In general, specify what state elements are to be initialized.
- virtual bool [is_ready](#) (void)
In general, determine if the initializer is ready to be applied.
- virtual void [apply](#) (DynManager &dyn_manager)
Complete initialization of the subject [DynBody](#).

Data Fields

- std::string [body_frame_id](#)
The suffix of the frame name (i.e., the part of the name after the vehicle identifier) to which this initializer pertains.
- std::string [reference_ref_frame_name](#)
The name of the reference frame against which state data specified in a [DynBodyInit](#) subclass are referenced.
- RefFrameState [state](#)
Contains state information set by the initializer, which is always a subclass of [DynBodyInit](#).
- double [position](#) [3]
Relative position between the subject and reference reference frame origins.
- double [velocity](#) [3]
Relative velocity between the subject and reference reference frame origins.
- Orientation [orientation](#)
Relative orientation between the subject and reference reference frame axes.
- double [ang_velocity](#) [3]
Relative angular velocity between the subject and reference axes.
- bool [reverse_sense](#)
Indicates how the user input state items are to be interpreted.
- bool [rate_in_parent](#)
Indicates how the user input angular velocity is to be interpreted.

Protected Member Functions

- void [apply_user_inputs](#) (void)
Compute the state wrt the reference reference frame, incorporate the user-input items to this relative state, and compute the state relative to the target frame's parent.
- void [compute_rotational_state](#) (void)
This method is obsolete.
- void [compute_translational_state](#) (void)
This method is obsolete.
- Planet * [find_planet](#) (DynManager &dyn_manager, const std::string &planet_name, const std::string &variable_name)
Find the Planet with the given name, failing if not found.

- DynBody * [find_dyn_body](#) (DynManager &dyn_manager, const std::string &dyn_body_name, const std::string &variable_name)
Find the DynBody with the given name, failing if not found.
- RefFrame * [find_ref_frame](#) (DynManager &dyn_manager, const std::string &ref_frame_name, const std::string &variable_name)
Find the RefFrame with the given name, failing if not found.
- BodyRefFrame * [find_body_frame](#) (DynBody &frame_container, const std::string &body_frame_identifier, const std::string &variable_name)
Find the RefFrame with the given name, failing if not found.

Protected Attributes

- BodyRefFrame * [body_ref_frame](#)
The reference frame whose name is vehicle_name.body_frame_id.
- RefFrame * [reference_ref_frame](#)
The reference frame whose name is reference_ref_frame_name.

Private Member Functions

- [DynBodyInit](#) (const [DynBodyInit](#) &)
- [DynBodyInit](#) & [operator=](#) (const [DynBodyInit](#) &)

Private Attributes

- RefFrame * [subscribed_frame](#)
The subscribed-to frame (the reference_ref_frame at initialization time), cached so that this frame will be unsubscribed at application time.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInit](#) ()

8.10.1 Detailed Description

Base class for initialize the state of a DynBody.

Definition at line 90 of file dyn_body_init.hh.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 DynBodyInit() [1/2]

```
jeod::DynBodyInit::DynBodyInit (
    const DynBodyInit & ) [private]
```

8.10.2.2 DynBodyInit() [2/2]

```
jeod::DynBodyInit::DynBodyInit (
    void )
```

Construct a [DynBodyInit](#).

Definition at line 60 of file `dyn_body_init.cc`.

References `ang_velocity`, `position`, and `velocity`.

8.10.2.3 ~DynBodyInit()

```
jeod::DynBodyInit::~~DynBodyInit (
    void ) [virtual]
```

Destruct a [DynBodyInit](#).

Definition at line 85 of file `dyn_body_init.cc`.

8.10.3 Member Function Documentation

8.10.3.1 apply()

```
void jeod::DynBodyInit::apply (
    DynManager & dyn_manager ) [virtual]
```

Complete initialization of the subject `DynBody`.

The `apply` method for all subclasses of [DynBodyInit](#) *must* pass the `apply` call to their immediate parent class, which in turn must do the same, eventually invoking this method.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Jeod manager
----------------------	---------------------------------	--------------

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

Reimplemented in [jeod::DynBodyInitOrbit](#), [jeod::DynBodyInitPlanetDerived](#), [jeod::DynBodyInitWrtPlanet](#), [jeod::DynBodyInitRotState](#), [jeod::DynBodyInitNedState](#), [jeod::DynBodyInitTransState](#), and [jeod::DynBodyInitLvlhState](#).

Definition at line 252 of file `dyn_body_init.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyAction::apply()`, `body_ref_frame`, `jeod::BodyAction::dyn_subject`, `initializes_what()`, `reference_ref_frame`, `state`, `subscribed_frame`, and `jeod::BodyActionMessages::trace`.

Referenced by `jeod::DynBodyInitTransState::apply()`, `jeod::DynBodyInitRotState::apply()`, and `jeod::DynBodyInitWrtPlanet::apply()`.

8.10.3.2 `apply_user_inputs()`

```
void jeod::DynBodyInit::apply_user_inputs (
    void ) [protected]
```

Compute the state wrt the reference reference frame, incorporate the user-input items to this relative state, and compute the state relative to the target frame's parent.

Definition at line 318 of file `dyn_body_init.cc`.

References `ang_velocity`, `body_ref_frame`, `jeod::BodyAction::dyn_subject`, `initializes_what()`, `orientation`, `position`, `rate_in_parent`, `reference_ref_frame`, `reverse_sense`, `state`, and `velocity`.

Referenced by `jeod::DynBodyInitLvlhState::apply()`, `jeod::DynBodyInitTransState::apply()`, `jeod::DynBodyInitNedState::apply()`, `jeod::DynBodyInitRotState::apply()`, `compute_rotational_state()`, and `compute_translational_state()`.

8.10.3.3 `compute_rotational_state()`

```
void jeod::DynBodyInit::compute_rotational_state (
    void ) [protected]
```

This method is obsolete.

Use `apply_user_inputs` instead.

Definition at line 378 of file `dyn_body_init.cc`.

References `apply_user_inputs()`, and `jeod::BodyActionMessages::invalid_name`.

8.10.3.4 `compute_translational_state()`

```
void jeod::DynBodyInit::compute_translational_state (
    void ) [protected]
```

This method is obsolete.

Use `apply_user_inputs` instead.

Definition at line 401 of file `dyn_body_init.cc`.

References `apply_user_inputs()`, and `jeod::BodyActionMessages::invalid_name`.

8.10.3.5 find_body_frame()

```
BodyRefFrame * jeod::DynBodyInit::find_body_frame (
    DynBody & frame_container,
    const std::string & body_frame_identifier,
    const std::string & variable_name ) [protected]
```

Find the RefFrame with the given name, failing if not found.

Returns

Found BodyRefFrame

Parameters

in	<i>frame_container</i>	Body containing frame
in	<i>body_frame_identifier</i>	BodyRefFrame identifier
in	<i>variable_name</i>	For error reporting

Definition at line 540 of file dyn_body_init.cc.

References `jeod::BodyAction::action_identifier`, `jeod::BodyActionMessages::invalid_name`, and `jeod::BodyAction↵::validate_name()`.

Referenced by `initialize()`.

8.10.3.6 find_dyn_body()

```
DynBody * jeod::DynBodyInit::find_dyn_body (
    DynManager & dyn_manager,
    const std::string & dyn_body_name,
    const std::string & variable_name ) [protected]
```

Find the DynBody with the given name, failing if not found.

Returns

Found DynBody

Parameters

in	<i>dyn_manager</i>	Dynamics manager
in	<i>dyn_body_name</i>	DynBody name
in	<i>variable_name</i>	For error reporting

Definition at line 466 of file dyn_body_init.cc.

References `jeod::BodyAction::action_identifier`, `jeod::BodyActionMessages::invalid_name`, and `jeod::BodyAction↵::validate_name()`.

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

8.10.3.7 `find_planet()`

```
Planet * jeod::DynBodyInit::find_planet (
    DynManager & dyn_manager,
    const std::string & planet_name,
    const std::string & variable_name ) [protected]
```

Find the Planet with the given name, failing if not found.

Returns

Found Planet

Parameters

in	<i>dyn_manager</i>	Dynamics manager
in	<i>planet_name</i>	Planet name
in	<i>variable_name</i>	For error reporting

Definition at line 429 of file `dyn_body_init.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyActionMessages::invalid_name`, and `jeod::BodyAction<T>::validate_name()`.

Referenced by `jeod::DynBodyInitWrtPlanet::initialize()`, and `jeod::DynBodyInitOrbit::initialize()`.

8.10.3.8 `find_ref_frame()`

```
RefFrame * jeod::DynBodyInit::find_ref_frame (
    DynManager & dyn_manager,
    const std::string & ref_frame_name,
    const std::string & variable_name ) [protected]
```

Find the RefFrame with the given name, failing if not found.

Returns

Found `ref_frame`

Parameters

in	<i>dyn_manager</i>	Dynamics manager
in	<i>ref_frame_name</i>	RefFrame name
in	<i>variable_name</i>	For error reporting

Definition at line 503 of file `dyn_body_init.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyActionMessages::invalid_name`, and `jeod::BodyActionMessages::validate_name()`.

Referenced by `initialize()`.

8.10.3.9 initialize()

```
void jeod::DynBodyInit::initialize (
    DynManager & dyn_manager ) [virtual]
```

Complete initialization of a [DynBodyInit](#).

The initialize method for all subclasses of [DynBodyInit](#) *must* pass the initialize call to their immediate parent class, which in turn must do the same, eventually invoking this method.

Parameters

<code>in, out</code>	<code>dyn_manager</code>	Dynamics manager
----------------------	--------------------------	------------------

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

Reimplemented in [jeod::DynBodyInitOrbit](#), [jeod::DynBodyInitPlanetDerived](#), [jeod::DynBodyInitWrtPlanet](#), [jeod::DynBodyInitRotState](#), [jeod::DynBodyInitNedState](#), [jeod::DynBodyInitTransState](#), [jeod::DynBodyInitLvlhState](#), [jeod::DynBodyInitLvlhRotState](#), [jeod::DynBodyInitLvlhTransState](#), [jeod::DynBodyInitNedRotState](#), and [jeod::DynBodyInitNedTransState](#).

Definition at line 100 of file `dyn_body_init.cc`.

References `jeod::BodyAction::action_identifier`, `body_frame_id`, `body_ref_frame`, `jeod::BodyAction::dyn_subject`, `find_body_frame()`, `find_ref_frame()`, `jeod::BodyAction::initialize()`, `jeod::BodyActionMessages::invalid_object`, `jeod::BodyAction::mass_subject`, `reference_ref_frame`, `reference_ref_frame_name`, and `subscribed_frame`.

Referenced by `jeod::DynBodyInitTransState::initialize()`, `jeod::DynBodyInitRotState::initialize()`, and `jeod::DynBodyInitWrtPlanet::initialize()`.

8.10.3.10 initializes_what()

```
RefFrameItems::Items jeod::DynBodyInit::initializes_what (
    void ) [virtual]
```

In general, specify what state elements are to be initialized.

This method indicates that no such elements are initialized. A subclass that does something *must* override this default method.

Returns

Initialized states

Reimplemented in [jeod::DynBodyInitRotState](#), [jeod::DynBodyInitWrtPlanet](#), and [jeod::DynBodyInitTransState](#).

Definition at line 173 of file `dyn_body_init.cc`.

Referenced by `apply()`, `apply_user_inputs()`, `is_ready()`, and `report_failure()`.

8.10.3.11 is_ready()

```
bool jeod::DynBodyInit::is_ready (
    void ) [virtual]
```

In general, determine if the initializer is ready to be applied.

This method determines whether the self-dependencies are satisfied. Dependencies on the reference reference frame are the responsibility of derived classes.

Returns

Can initializer run?

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

Reimplemented in [jeod::DynBodyInitPlanetDerived](#), [jeod::DynBodyInitRotState](#), [jeod::DynBodyInitWrtPlanet](#), and [jeod::DynBodyInitTransState](#).

Definition at line 188 of file `dyn_body_init.cc`.

References `body_ref_frame`, `initializes_what()`, `jeod::BodyAction::is_ready()`, `rate_in_parent`, and `reverse_sense`.

Referenced by `jeod::DynBodyInitTransState::is_ready()`, `jeod::DynBodyInitWrtPlanet::is_ready()`, and `jeod::DynBodyInitRotState::is_ready()`.

8.10.3.12 operator=()

```
DynBodyInit& jeod::DynBodyInit::operator= (
    const DynBodyInit & ) [private]
```

8.10.3.13 report_failure()

```
void jeod::DynBodyInit::report_failure (
    void ) [virtual]
```

Report on an initializer that could not be processed.

Definition at line 292 of file `dyn_body_init.cc`.

References `jeod::BodyAction::action_idenfier`, `body_ref_frame`, `initializes_what()`, `jeod::BodyActionMessages::not_performed`, and `reference_ref_frame`.

8.10.4 Friends And Related Function Documentation

8.10.4.1 init_attrjeod__DynBodyInit

```
void init_attrjeod__DynBodyInit ( ) [friend]
```

8.10.4.2 InputProcessor

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

Definition at line 92 of file dyn_body_init.hh.

8.10.5 Field Documentation

8.10.5.1 ang_velocity

```
double jeod::DynBodyInit::ang_velocity[3]
```

Relative angular velocity between the subject and reference axes.

The flags `reverse_sense` and `rate_in_parent` give four interpretations:

- Default (both `reverse_sense` and `rate_in_parent` are false):
Angular velocity of the subject frame with respect to the reference frame, expressed in subject frame coordinates.
- `reverse_sense` is clear, `rate_in_parent` is set:
Angular velocity of the subject frame with respect to the reference frame, expressed in reference frame coordinates.
- `reverse_sense` is set, `rate_in_parent` is clear:
Angular velocity of the reference frame with respect to the subject frame, expressed in reference frame coordinates.
- Both `reverse_sense` and `rate_in_parent` are set:
Angular velocity of the reference frame with respect to the subject frame, expressed in subject frame coordinates.
`trick_units(rad/s)`

Definition at line 162 of file dyn_body_init.hh.

Referenced by `jeod::DynBodyInitLvlhState::apply()`, `apply_user_inputs()`, and `DynBodyInit()`.

8.10.5.2 body_frame_id

```
std::string jeod::DynBodyInit::body_frame_id
```

The suffix of the frame name (i.e., the part of the name after the vehicle identifier) to which this initializer pertains.

trick_units(-)

Definition at line 101 of file dyn_body_init.hh.

Referenced by initialize().

8.10.5.3 body_ref_frame

```
BodyRefFrame* jeod::DynBodyInit::body_ref_frame [protected]
```

The reference frame whose name is vehicle_name.body_frame_id.

This is the frame to which the state is applied.trick_io(**)

Definition at line 187 of file dyn_body_init.hh.

Referenced by apply(), apply_user_inputs(), initialize(), is_ready(), and report_failure().

8.10.5.4 orientation

```
Orientation jeod::DynBodyInit::orientation
```

Relative orientation between the subject and reference reference frame axes.

The normal sense (reverse_sense is not set) is the transformation from reference to subject. The reverse meaning (reverse_sense set) is the transformation from subject to reference.trick_units(-)

Definition at line 144 of file dyn_body_init.hh.

Referenced by jeod::DynBodyInitLvHState::apply(), and apply_user_inputs().

8.10.5.5 position

```
double jeod::DynBodyInit::position[3]
```

Relative position between the subject and reference reference frame origins.

The normal sense (reverse_sense is not set) is the position of the subject origin with respect to the reference origin, expressed in reference coordinates. The reverse meaning (reverse_sense set) is the position of the reference origin with respect to the subject origin, expressed in subject coordinates.trick_units(m)

Definition at line 126 of file dyn_body_init.hh.

Referenced by jeod::DynBodyInitLvHState::apply(), jeod::DynBodyInitOrbit::apply(), apply_user_inputs(), and DynBodyInit().

8.10.5.6 rate_in_parent

```
bool jeod::DynBodyInit::rate_in_parent
```

Indicates how the user input angular velocity is to be interpreted.

This item works in conjunction with `reverse_sense`. See `ang_velocity` for a complete description. `trick_units(-)`

Definition at line 179 of file `dyn_body_init.hh`.

Referenced by `apply_user_inputs()`, and `is_ready()`.

8.10.5.7 reference_ref_frame

```
RefFrame* jeod::DynBodyInit::reference_ref_frame [protected]
```

The reference frame whose name is `reference_ref_frame_name`.

This is the frame against which the user state is reference. `trick_io(**)`

Definition at line 193 of file `dyn_body_init.hh`.

Referenced by `jeod::DynBodyInitLvIhState::apply()`, `jeod::DynBodyInitNedState::apply()`, `apply()`, `apply_user_inputs()`, `jeod::DynBodyInitWrtPlanet::initialize()`, `initialize()`, `jeod::DynBodyInitOrbit::initialize()`, `jeod::DynBodyInitTransState::is_ready()`, `jeod::DynBodyInitRotState::is_ready()`, and `report_failure()`.

8.10.5.8 reference_ref_frame_name

```
std::string jeod::DynBodyInit::reference_ref_frame_name
```

The name of the reference frame against which state data specified in a [DynBodyInit](#) subclass are referenced.

`trick_units(-)`

Definition at line 107 of file `dyn_body_init.hh`.

Referenced by `initialize()`.

8.10.5.9 reverse_sense

```
bool jeod::DynBodyInit::reverse_sense
```

Indicates how the user input state items are to be interpreted.

If clear (default setting), indicates that the user input position, velocity, orientation, and angular velocity are to be interpreted in the standard JEOD parent to child sense. The meaning is reversed when this flag is set. See the descriptions of the individual state items for details. `trick_units(-)`

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

Referenced by `apply_user_inputs()`, `jeod::DynBodyInitOrbit::initialize()`, and `is_ready()`.

8.10.5.10 state

```
RefFrameState jeod::DynBodyInit::state
```

Contains state information set by the initializer, which is always a subclass of [DynBodyInit](#).

The [DynBodyInit](#) apply method copies the state elements indicated by the initializer's `initializes_what` method to the frame indicated by the `frame_id` and then propagates the initialized states up/down the vehicle attachment tree.`trick_units(-)`

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

Referenced by `apply()`, and `apply_user_inputs()`.

8.10.5.11 subscribed_frame

```
RefFrame* jeod::DynBodyInit::subscribed_frame [private]
```

The subscribed-to frame (the `reference_ref_frame` at initialization time), cached so that this frame will be unsubscribed at application time.

`trick_io(**)`

Definition at line 203 of file `dyn_body_init.hh`.

Referenced by `apply()`, and `initialize()`.

8.10.5.12 velocity

```
double jeod::DynBodyInit::velocity[3]
```

Relative velocity between the subject and reference reference frame origins.

The normal sense (`reverse_sense` is not set) is the velocity of the subject origin with respect to the reference origin, expressed in and observed in reference coordinates. The reverse meaning (`reverse_sense` set) is the velocity of the reference origin with respect to the subject origin, expressed in and observed in subject coordinates.`trick_units(m/s)`

Definition at line 136 of file `dyn_body_init.hh`.

Referenced by `jeod::DynBodyInitLvHState::apply()`, `jeod::DynBodyInitOrbit::apply()`, `apply_user_inputs()`, and `DynBodyInit()`.

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

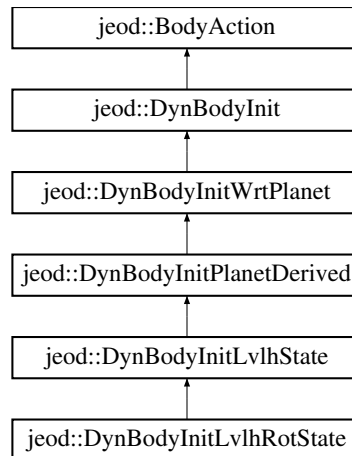
- [dyn_body_init.hh](#)
- [dyn_body_init.cc](#)

8.11 jeod::DynBodyInitLvlhRotState Class Reference

Initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.

```
#include <dyn_body_init_lvlh_rot_state.hh>
```

Inheritance diagram for jeod::DynBodyInitLvlhRotState:



Public Member Functions

- [DynBodyInitLvlhRotState](#) ()
DynBodyInitLvlhRotState default constructor.
- virtual [~DynBodyInitLvlhRotState](#) ()
DynBodyInitLvlhRotState destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize the initializer.

Private Member Functions

- [DynBodyInitLvlhRotState](#) (const [DynBodyInitLvlhRotState](#) &)
- [DynBodyInitLvlhRotState](#) & [operator=](#) (const [DynBodyInitLvlhRotState](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitLvlhRotState](#) ()

Additional Inherited Members

8.11.1 Detailed Description

Initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.

That some vehicle can be this vehicle itself.

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

8.11.2 Constructor & Destructor Documentation

8.11.2.1 DynBodyInitLvLhRotState() [1/2]

```
jeod::DynBodyInitLvLhRotState::DynBodyInitLvLhRotState (
    const DynBodyInitLvLhRotState & ) [private]
```

8.11.2.2 DynBodyInitLvLhRotState() [2/2]

```
jeod::DynBodyInitLvLhRotState::DynBodyInitLvLhRotState (
    void )
```

[DynBodyInitLvLhRotState](#) default constructor.

Definition at line 59 of file `dyn_body_init_lvLh_rot_state.cc`.

References `jeod::DynBodyInitWrtPlanet::set_items`.

8.11.2.3 ~DynBodyInitLvLhRotState()

```
jeod::DynBodyInitLvLhRotState::~~DynBodyInitLvLhRotState (
    void ) [virtual]
```

[DynBodyInitLvLhRotState](#) destructor.

Definition at line 73 of file `dyn_body_init_lvLh_rot_state.cc`.

8.11.3 Member Function Documentation

8.11.3.1 initialize()

```
void jeod::DynBodyInitLvLhRotState::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize the initializer.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Definition at line 86 of file `dyn_body_init_lvlh_rot_state.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyAction::get_subject_dyn_body()`, `jeod::BodyActionMessages::illegal_value`, `jeod::DynBodyInitLvlhState::initialize()`, `jeod::BodyActionMessages::null_pointer`, `jeod::DynBodyInitPlanetDerived::ref_body`, `jeod::DynBodyInitPlanetDerived::ref_body_name`, and `jeod::DynBodyInitWrtPlanet::set_items`.

8.11.3.2 operator=()

```
DynBodyInitLvlhRotState& jeod::DynBodyInitLvlhRotState::operator= (
    const DynBodyInitLvlhRotState & ) [private]
```

8.11.4 Friends And Related Function Documentation

8.11.4.1 init_attrjeod__DynBodyInitLvlhRotState

```
void init_attrjeod__DynBodyInitLvlhRotState ( ) [friend]
```

8.11.4.2 InputProcessor

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

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

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

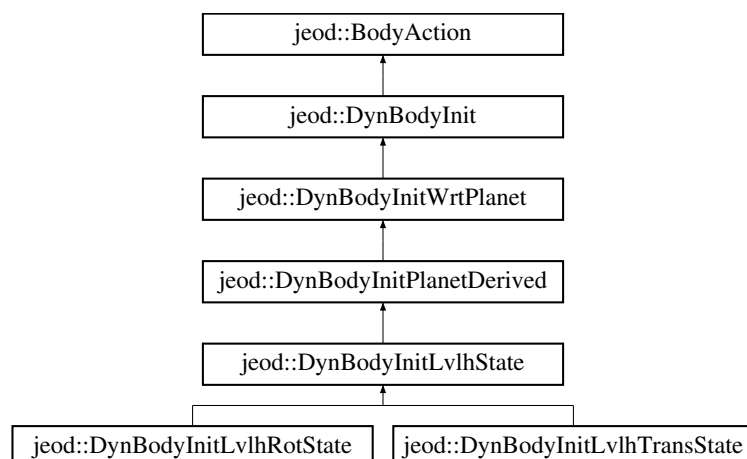
- [dyn_body_init_lvlh_rot_state.hh](#)
- [dyn_body_init_lvlh_rot_state.cc](#)

8.12 jeod::DynBodyInitLvlhState Class Reference

Initialize selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.

```
#include <dyn_body_init_lvlh_state.hh>
```

Inheritance diagram for `jeod::DynBodyInitLvlhState`:



Public Member Functions

- [DynBodyInitLvHState](#) ()
DynBodyInitLvHState default constructor.
- virtual [~DynBodyInitLvHState](#) ()
DynBodyInitLvHState destructor.
- void [set_lvH_frame_object](#) (LvHFrame &lvh_frame_object)
Cache a pointer to a user-supplied LvHFrame object.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize the initializer.
- virtual void [apply](#) (DynManager &dyn_manager)
Apply the initializer: Construct the reference LVLH frame so the parent initializer can compute the vehicle's state relative to the vehicle's inertial frame.

Data Fields

- LvHType::Type [lvh_type](#)
Indicates type of LVLH coordinates desired.

Private Member Functions

- [DynBodyInitLvHState](#) (const [DynBodyInitLvHState](#) &)
- [DynBodyInitLvHState](#) & [operator=](#) (const [DynBodyInitLvHState](#) &)

Private Attributes

- LvHFrame * [lvh_object_ptr](#)
A pointer to an LvHFrame which can be supplied by the user.

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitLvHState](#) ()

Additional Inherited Members

8.12.1 Detailed Description

Initialize selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.

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

8.12.2 Constructor & Destructor Documentation

8.12.2.1 DynBodyInitLvlhState() [1/2]

```
jeod::DynBodyInitLvlhState::DynBodyInitLvlhState (
    void )
```

[DynBodyInitLvlhState](#) default constructor.

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

References `jeod::DynBodyInitPlanetDerived::required_items`.

8.12.2.2 ~DynBodyInitLvlhState()

```
jeod::DynBodyInitLvlhState::~~DynBodyInitLvlhState (
    void ) [virtual]
```

[DynBodyInitLvlhState](#) destructor.

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

8.12.2.3 DynBodyInitLvlhState() [2/2]

```
jeod::DynBodyInitLvlhState::DynBodyInitLvlhState (
    const DynBodyInitLvlhState & ) [private]
```

8.12.3 Member Function Documentation

8.12.3.1 apply()

```
void jeod::DynBodyInitLvlhState::apply (
    DynManager & dyn_manager ) [virtual]
```

Apply the initializer: Construct the reference LVLH frame so the parent initializer can compute the vehicle's state relative to the vehicle's inertial frame.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Definition at line 114 of file `dyn_body_init_lvlh_state.cc`.

References `jeod::DynBodyInit::ang_velocity`, `jeod::DynBodyInitPlanetDerived::apply()`, `jeod::DynBodyInit::apply_↵ user_inputs()`, `jeod::BodyActionMessages::illegal_value`, `lvlh_object_ptr`, `lvlh_type`, `jeod::DynBodyInit::orientation`, `jeod::DynBodyInitWrtPlanet::planet`, `jeod::DynBodyInit::position`, `jeod::DynBodyInitPlanetDerived::ref_body`, `jeod↵::DynBodyInit::reference_ref_frame`, `jeod::DynBodyInitWrtPlanet::set_items`, and `jeod::DynBodyInit::velocity`.

8.12.3.2 initialize()

```
void jeod::DynBodyInitLvlhState::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize the initializer.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Reimplemented in [jeod::DynBodyInitLvlhRotState](#), and [jeod::DynBodyInitLvlhTransState](#).

Definition at line 87 of file `dyn_body_init_lvlh_state.cc`.

References `jeod::DynBodyInitPlanetDerived::body_is_required`, `jeod::DynBodyInitPlanetDerived::initialize()`, and `lvlh_object_ptr`.

Referenced by `jeod::DynBodyInitLvlhTransState::initialize()`, and `jeod::DynBodyInitLvlhRotState::initialize()`.

8.12.3.3 operator=()

```
DynBodyInitLvlhState& jeod::DynBodyInitLvlhState::operator= (
    const DynBodyInitLvlhState & ) [private]
```

8.12.3.4 set_lvlh_frame_object()

```
void jeod::DynBodyInitLvlhState::set_lvlh_frame_object (
    LvlhFrame & lvlh_frame_object )
```

Cache a pointer to a user-supplied LvlhFrame object.

Parameters

<code>in</code>	<code><i>lvlh_frame_object</i></code>	LVLH frame object
-----------------	---------------------------------------	-------------------

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

References `lvlh_object_ptr`.

8.12.4 Friends And Related Function Documentation

8.12.4.1 `init_attrjeod__DynBodyInitLvlhState`

```
void init_attrjeod__DynBodyInitLvlhState ( ) [friend]
```

8.12.4.2 `InputProcessor`

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

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

8.12.5 Field Documentation

8.12.5.1 `lvlh_object_ptr`

```
LvlhFrame* jeod::DynBodyInitLvlhState::lvlh_object_ptr [private]
```

A pointer to an `LvlhFrame` which can be supplied by the user.

`trick_units(-)`

Definition at line 105 of file `dyn_body_init_lvlh_state.hh`.

Referenced by `apply()`, `initialize()`, and `set_lvlh_frame_object()`.

8.12.5.2 `lvlh_type`

```
LvlhType::Type jeod::DynBodyInitLvlhState::lvlh_type
```

Indicates type of LVLH coordinates desired.

Default is `rectilinear.trick_units(-)`

Definition at line 98 of file `dyn_body_init_lvlh_state.hh`.

Referenced by `apply()`.

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

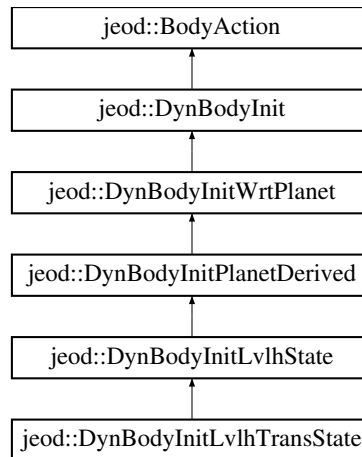
- [dyn_body_init_lvlh_state.hh](#)
- [dyn_body_init_lvlh_state.cc](#)

8.13 jeod::DynBodyInitLvlhTransState Class Reference

initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.

```
#include <dyn_body_init_lvlh_trans_state.hh>
```

Inheritance diagram for jeod::DynBodyInitLvlhTransState:



Public Member Functions

- [DynBodyInitLvlhTransState](#) ()
[DynBodyInitLvlhTransState](#) default constructor.
- virtual [~DynBodyInitLvlhTransState](#) ()
[DynBodyInitLvlhTransState](#) destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize the initializer.

Private Member Functions

- [DynBodyInitLvlhTransState](#) (const [DynBodyInitLvlhTransState](#) &)
- [DynBodyInitLvlhTransState](#) & [operator=](#) (const [DynBodyInitLvlhTransState](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitLvlhTransState](#) ()

Additional Inherited Members

8.13.1 Detailed Description

initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.

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

8.13.2 Constructor & Destructor Documentation

8.13.2.1 DynBodyInitLvLhTransState() [1/2]

```
jeod::DynBodyInitLvLhTransState::DynBodyInitLvLhTransState (
    const DynBodyInitLvLhTransState & ) [private]
```

8.13.2.2 DynBodyInitLvLhTransState() [2/2]

```
jeod::DynBodyInitLvLhTransState::DynBodyInitLvLhTransState (
    void )
```

[DynBodyInitLvLhTransState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

8.13.2.3 ~DynBodyInitLvLhTransState()

```
jeod::DynBodyInitLvLhTransState::~~DynBodyInitLvLhTransState (
    void ) [virtual]
```

[DynBodyInitLvLhTransState](#) destructor.

Definition at line 66 of file `dyn_body_init_lvLh_trans_state.cc`.

8.13.3 Member Function Documentation

8.13.3.1 initialize()

```
void jeod::DynBodyInitLvLhTransState::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize the initializer.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Definition at line 78 of file `dyn_body_init_lvlh_trans_state.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyActionMessages::illegal_value`, `jeod::DynBodyInitLvlhState::initialize()`, and `jeod::DynBodyInitWrtPlanet::set_items`.

8.13.3.2 operator=()

```
DynBodyInitLvlhTransState& jeod::DynBodyInitLvlhTransState::operator= (
    const DynBodyInitLvlhTransState & ) [private]
```

8.13.4 Friends And Related Function Documentation

8.13.4.1 init_attrjeod__DynBodyInitLvlhTransState

```
void init_attrjeod__DynBodyInitLvlhTransState ( ) [friend]
```

8.13.4.2 InputProcessor

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

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

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

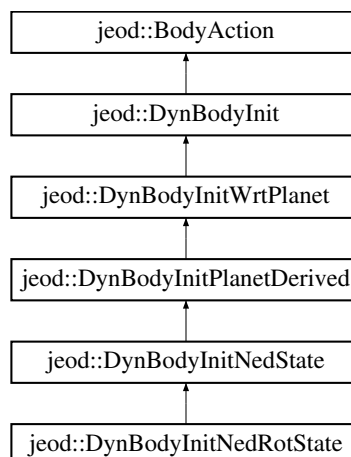
- [dyn_body_init_lvlh_trans_state.hh](#)
- [dyn_body_init_lvlh_trans_state.cc](#)

8.14 jeod::DynBodyInitNedRotState Class Reference

Initialize a vehicle's rotational state wrt some vehicle's North-East-Down frame.

```
#include <dyn_body_init_ned_rot_state.hh>
```

Inheritance diagram for `jeod::DynBodyInitNedRotState`:



Public Member Functions

- [DynBodyInitNedRotState](#) ()
DynBodyInitNedRotState default constructor.
- virtual [~DynBodyInitNedRotState](#) ()
DynBodyInitNedRotState destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize the initializer.

Private Member Functions

- [DynBodyInitNedRotState](#) (const [DynBodyInitNedRotState](#) &)
- [DynBodyInitNedRotState](#) & [operator=](#) (const [DynBodyInitNedRotState](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitNedRotState](#) ()

Additional Inherited Members

8.14.1 Detailed Description

Initialize a vehicle's rotational state wrt some vehicle's North-East-Down frame.

Definition at line 84 of file `dyn_body_init_ned_rot_state.hh`.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 [DynBodyInitNedRotState\(\)](#) [1/2]

```
jeod::DynBodyInitNedRotState::DynBodyInitNedRotState (
    const DynBodyInitNedRotState & ) [private]
```

8.14.2.2 [DynBodyInitNedRotState\(\)](#) [2/2]

```
jeod::DynBodyInitNedRotState::DynBodyInitNedRotState (
    void )
```

[DynBodyInitNedRotState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

8.14.2.3 ~DynBodyInitNedRotState()

```
jeod::DynBodyInitNedRotState::~~DynBodyInitNedRotState (
    void ) [virtual]
```

[DynBodyInitNedRotState](#) destructor.

Definition at line 68 of file `dyn_body_init_ned_rot_state.cc`.

8.14.3 Member Function Documentation

8.14.3.1 initialize()

```
void jeod::DynBodyInitNedRotState::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize the initializer.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Definition at line 81 of file `dyn_body_init_ned_rot_state.cc`.

References [jeod::BodyAction::action_identifier](#), [jeod::BodyActionMessages::illegal_value](#), [jeod::DynBodyInitNedState::initialize\(\)](#), and [jeod::DynBodyInitWrtPlanet::set_items](#).

8.14.3.2 operator=()

```
DynBodyInitNedRotState& jeod::DynBodyInitNedRotState::operator= (
    const DynBodyInitNedRotState & ) [private]
```

8.14.4 Friends And Related Function Documentation

8.14.4.1 init_attrjeod__DynBodyInitNedRotState

```
void init_attrjeod__DynBodyInitNedRotState ( ) [friend]
```

8.14.4.2 InputProcessor

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

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

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

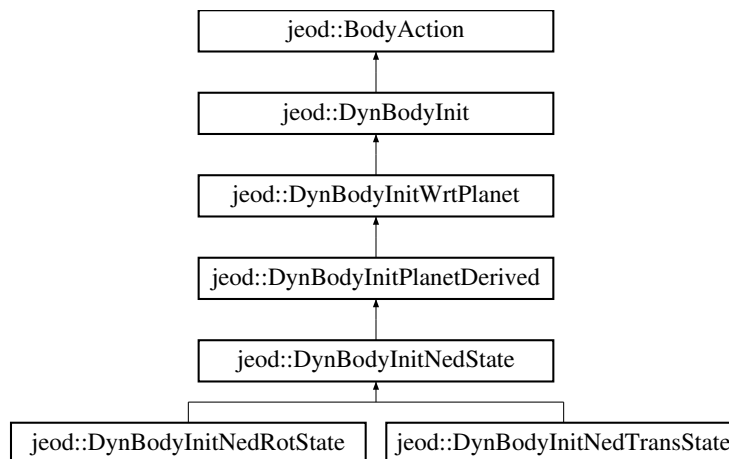
- [dyn_body_init_ned_rot_state.hh](#)
- [dyn_body_init_ned_rot_state.cc](#)

8.15 jeod::DynBodyInitNedState Class Reference

Initialize selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.

```
#include <dyn_body_init_ned_state.hh>
```

Inheritance diagram for `jeod::DynBodyInitNedState`:



Public Member Functions

- [DynBodyInitNedState](#) ()
DynBodyInitNedState default constructor.
- virtual [~DynBodyInitNedState](#) ()
DynBodyInitNedState destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize the initializer.
- virtual void [apply](#) (DynManager &dyn_manager)
Apply the initializer.

Data Fields

- AltLatLongState [ref_point](#)
Reference point for the local geodetic/geocentric, used only if the reference body is NULL.
- NorthEastDown::AltLatLongType [altlatlong_type](#)
Use spherical or elliptical coordinates?

Private Member Functions

- [DynBodyInitNedState](#) (const [DynBodyInitNedState](#) &)
- [DynBodyInitNedState](#) & operator= (const [DynBodyInitNedState](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitNedState](#) ()

Additional Inherited Members

8.15.1 Detailed Description

Initialize selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.

Definition at line 89 of file `dyn_body_init_ned_state.hh`.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 [DynBodyInitNedState](#)() [1/2]

```
jeod::DynBodyInitNedState::DynBodyInitNedState (
    const DynBodyInitNedState & ) [private]
```

8.15.2.2 [DynBodyInitNedState](#)() [2/2]

```
jeod::DynBodyInitNedState::DynBodyInitNedState (
    void )
```

[DynBodyInitNedState](#) default constructor.

Definition at line 60 of file `dyn_body_init_ned_state.cc`.

References `jeod::DynBodyInitPlanetDerived::body_is_required`, and `jeod::DynBodyInitPlanetDerived::required_`↔ items.

8.15.2.3 ~DynBodyInitNedState()

```
jeod::DynBodyInitNedState::~~DynBodyInitNedState (
    void ) [virtual]
```

[DynBodyInitNedState](#) destructor.

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

8.15.3 Member Function Documentation

8.15.3.1 apply()

```
void jeod::DynBodyInitNedState::apply (
    DynManager & dyn_manager ) [virtual]
```

Apply the initializer.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Definition at line 108 of file `dyn_body_init_ned_state.cc`.

References [jeod::BodyAction::action_identifier](#), [altlatlong_type](#), [jeod::DynBodyInitPlanetDerived::apply\(\)](#), [jeod::DynBodyInit::apply_user_inputs\(\)](#), [jeod::BodyActionMessages::illegal_value](#), [jeod::DynBodyInitWrtPlanet::planet](#), [jeod::DynBodyInitWrtPlanet::planet_name](#), [jeod::DynBodyInitPlanetDerived::ref_body](#), [ref_point](#), [jeod::DynBodyInit::reference_ref_frame](#), and [jeod::DynBodyInitWrtPlanet::set_items](#).

8.15.3.2 initialize()

```
void jeod::DynBodyInitNedState::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize the initializer.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Reimplemented in [jeod::DynBodyInitNedRotState](#), and [jeod::DynBodyInitNedTransState](#).

Definition at line 89 of file `dyn_body_init_ned_state.cc`.

References `jeod::DynBodyInitPlanetDerived::body_is_required`, `jeod::DynBodyInitPlanetDerived::initialize()`, and `jeod::DynBodyInitPlanetDerived::ref_body_name`.

Referenced by `jeod::DynBodyInitNedRotState::initialize()`, and `jeod::DynBodyInitNedTransState::initialize()`.

8.15.3.3 operator=()

```
DynBodyInitNedState& jeod::DynBodyInitNedState::operator= (
    const DynBodyInitNedState & ) [private]
```

8.15.4 Friends And Related Function Documentation

8.15.4.1 init_attrjeod__DynBodyInitNedState

```
void init_attrjeod__DynBodyInitNedState ( ) [friend]
```

8.15.4.2 InputProcessor

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

Definition at line 91 of file `dyn_body_init_ned_state.hh`.

8.15.5 Field Documentation

8.15.5.1 altlatlong_type

```
NorthEastDown::AltLatLongType jeod::DynBodyInitNedState::altlatlong_type
```

Use spherical or elliptical coordinates?

`trick_units(-)`

Definition at line 107 of file `dyn_body_init_ned_state.hh`.

Referenced by `apply()`.

8.15.5.2 ref_point

```
AltLatLongState jeod::DynBodyInitNedState::ref_point
```

Reference point for the local geodetic/geocentric, used only if the reference body is NULL.

trick_units(—)

Definition at line 102 of file dyn_body_init_ned_state.hh.

Referenced by apply().

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

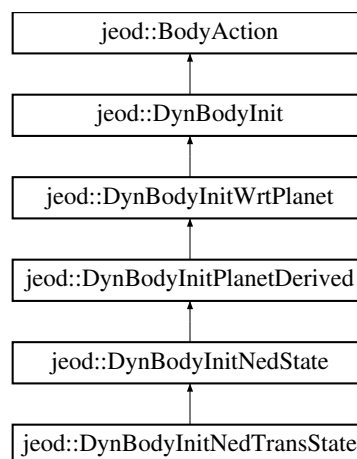
- [dyn_body_init_ned_state.hh](#)
- [dyn_body_init_ned_state.cc](#)

8.16 jeod::DynBodyInitNedTransState Class Reference

Initialize a vehicle's translational state wrt some vehicle's North-East-Down frame.

```
#include <dyn_body_init_ned_trans_state.hh>
```

Inheritance diagram for jeod::DynBodyInitNedTransState:



Public Member Functions

- [DynBodyInitNedTransState](#) ()
DynBodyInitNedTransState default constructor.
- virtual [~DynBodyInitNedTransState](#) ()
DynBodyInitNedTransState destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize the initializer.

Private Member Functions

- [DynBodyInitNedTransState](#) (const [DynBodyInitNedTransState](#) &)
- [DynBodyInitNedTransState](#) & operator= (const [DynBodyInitNedTransState](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitNedTransState](#) ()

Additional Inherited Members

8.16.1 Detailed Description

Initialize a vehicle's translational state wrt some vehicle's North-East-Down frame.

Definition at line 84 of file `dyn_body_init_ned_trans_state.hh`.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 [DynBodyInitNedTransState](#)() [1/2]

```
jeod::DynBodyInitNedTransState::DynBodyInitNedTransState (
    const DynBodyInitNedTransState & ) [private]
```

8.16.2.2 [DynBodyInitNedTransState](#)() [2/2]

```
jeod::DynBodyInitNedTransState::DynBodyInitNedTransState (
    void )
```

[DynBodyInitNedTransState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

8.16.2.3 [~DynBodyInitNedTransState](#)()

```
jeod::DynBodyInitNedTransState::~~DynBodyInitNedTransState (
    void ) [virtual]
```

[DynBodyInitNedTransState](#) destructor.

Definition at line 65 of file `dyn_body_init_ned_trans_state.cc`.

8.16.3 Member Function Documentation

8.16.3.1 initialize()

```
void jeod::DynBodyInitNedTransState::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize the initializer.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager
----------------	--------------------	------------------

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

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

References [jeod::BodyAction::action_identifier](#), [jeod::BodyActionMessages::illegal_value](#), [jeod::DynBodyInitNedState::initialize\(\)](#), and [jeod::DynBodyInitWrtPlanet::set_items](#).

8.16.3.2 operator=()

```
DynBodyInitNedTransState& jeod::DynBodyInitNedTransState::operator= (
    const DynBodyInitNedTransState & ) [private]
```

8.16.4 Friends And Related Function Documentation

8.16.4.1 init_attrjeod__DynBodyInitNedTransState

```
void init_attrjeod__DynBodyInitNedTransState ( ) [friend]
```

8.16.4.2 InputProcessor

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

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

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

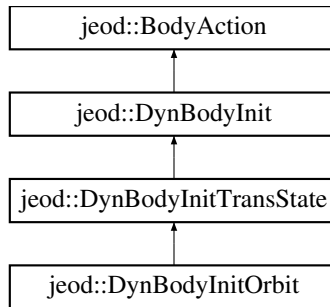
- [dyn_body_init_ned_trans_state.hh](#)
- [dyn_body_init_ned_trans_state.cc](#)

8.17 jeod::DynBodyInitOrbit Class Reference

Initialize a vehicle's translational state given an orbital specification.

```
#include <dyn_body_init_orbit.hh>
```

Inheritance diagram for jeod::DynBodyInitOrbit:



Public Types

- enum [OrbitalSet](#) {
[InvalidSet](#) = 0, [SmaEccIncAscnodeArgperTimeperi](#) = 1, [SmaEccIncAscnodeArgperManom](#) = 2,
[SlrEccIncAscnodeArgperTanom](#) = 3,
[IncAscnodeAltperAltapoArgperTanom](#) = 4, [IncAscnodeAltperAltapoArgperTimeperi](#) = 5, [SmaIncAscnodeArglatRadRadvel](#)
= 6, [SmaEccIncAscnodeArgperTanom](#) = 10,
[CaseEleven](#) = 11 }

Identifies which orbital elements define the orbit.

Public Member Functions

- [DynBodyInitOrbit](#) ()
[DynBodyInitOrbit](#) default constructor.
- virtual [~DynBodyInitOrbit](#) ()
[DynBodyInitOrbit](#) destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize the initializer.
- virtual void [apply](#) (DynManager &dyn_manager)
Apply the initializer.

Data Fields

- std::string [planet_name](#)
The name of the planet around which the orbit is to be established.
- std::string [orbit_frame_name](#)
Planet reference frame name, optionally dot-prefixed with the planet name.
- [OrbitalSet](#) [set](#)
Specifies which set of orbital elements specify the orbit.
- double [semi_major_axis](#)
Semi-major axis.

- double [semi_latus_rectum](#)
Semi-latus rectum.
- double [alt_periapsis](#)
Periapsis altitude.
- double [alt_apoapsis](#)
Apoapsis altitude.
- double [orb_radius](#)
Distance from center of planet.
- double [radial_vel](#)
Time derivative of the orbital radius.
- double [eccentricity](#)
Eccentricity.
- double [inclination](#)
Inclination.
- double [ascending_node](#)
Longitude (or right ascension) of ascending node.
- double [arg_periapsis](#)
Argument of periapsis.
- double [arg_latitude](#)
Argument of latitude.
- double [time_periapsis](#)
Time since periapsis passage.
- double [mean_anomaly](#)
Mean anomaly.
- double [true_anomaly](#)
True anomaly.

Protected Attributes

- Planet * [planet](#)
The planet.
- EphemerisRefFrame * [orbit_frame](#)
The orbit reference frame (ignoring rotation)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitOrbit](#) ()

Additional Inherited Members

8.17.1 Detailed Description

Initialize a vehicle's translational state given an orbital specification.

Definition at line 83 of file `dyn_body_init_orbit.hh`.

8.17.2 Member Enumeration Documentation

8.17.2.1 OrbitalSet

```
enum jeod::DynBodyInitOrbit::OrbitalSet
```

Identifies which orbital elements define the orbit.

The goofy numbering scheme here is intentional. The numbers map directly to the corresponding orbital_set number in JEOD 1.4 / 1.5. NOTE: Orbital sets 4 and 11 are the same options.

Enumerator

InvalidSet	
SmaEccIncAscnodeArgperTimeperi	
SmaEccIncAscnodeArgperManom	
SlrEccIncAscnodeArgperTanom	
IncAscnodeAltperAltapoArgperTanom	
IncAscnodeAltperAltapoArgperTimeperi	
SmaIncAscnodeArglatRadRadvel	
SmaEccIncAscnodeArgperTanom	
CaseEleven	

Definition at line 99 of file dyn_body_init_orbit.hh.

8.17.3 Constructor & Destructor Documentation

8.17.3.1 DynBodyInitOrbit()

```
jeod::DynBodyInitOrbit::DynBodyInitOrbit (
    void )
```

[DynBodyInitOrbit](#) default constructor.

Definition at line 59 of file dyn_body_init_orbit.cc.

References [alt_apoapsis](#), [alt_periapsis](#), [arg_latitude](#), [arg_periapsis](#), [ascending_node](#), [eccentricity](#), [inclination](#), [InvalidSet](#), [mean_anomaly](#), [orb_radius](#), [orbit_frame](#), [planet](#), [radial_vel](#), [semi_latus_rectum](#), [semi_major_axis](#), [time_periapsis](#), and [true_anomaly](#).

8.17.3.2 ~DynBodyInitOrbit()

```
jeod::DynBodyInitOrbit::~~DynBodyInitOrbit (
    void ) [virtual]
```

[DynBodyInitOrbit](#) destructor.

Definition at line 87 of file `dyn_body_init_orbit.cc`.

8.17.4 Member Function Documentation

8.17.4.1 apply()

```
void jeod::DynBodyInitOrbit::apply (
    DynManager & dyn_manager ) [virtual]
```

Apply the initializer.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Definition at line 192 of file `dyn_body_init_orbit.cc`.

References `jeod::BodyAction::action_identifier`, `alt_apoapsis`, `alt_periapsis`, `jeod::DynBodyInitTransState::apply()`, `arg_latitude`, `arg_periapsis`, `ascending_node`, `CaseEleven`, `eccentricity`, `jeod::BodyActionMessages::illegal_value`, `IncAscnodeAltperAltapoArgperTanom`, `IncAscnodeAltperAltapoArgperTimeperi`, `inclination`, `InvalidSet`, `mean_↵ anomaly`, `orb_radius`, `orbit_frame`, `planet`, `jeod::DynBodyInit::position`, `radial_vel`, `semi_latus_rectum`, `semi_major_↵ _axis`, `SlrEccIncAscnodeArgperTanom`, `SmaEccIncAscnodeArgperManom`, `SmaEccIncAscnodeArgperTanom`, `SmaEccIncAscnodeArgperTimeperi`, `SmalncAscnodeArglatRadRadvel`, `time_periapsis`, `true_anomaly`, and `jeod_↵ ::DynBodyInit::velocity`.

8.17.4.2 initialize()

```
void jeod::DynBodyInitOrbit::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize the initializer.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Definition at line 98 of file `dyn_body_init_orbit.cc`.

References `jeod::BodyAction::action_identifier`, `CaseEleven`, `jeod::DynBodyInit::find_planet()`, `jeod::BodyActionMessages::illegal_value`, `IncAscnodeAltperAltapoArgperTanom`, `IncAscnodeAltperAltapoArgperTimeperi`, `jeod::DynBodyInitTransState::initialize()`, `jeod::BodyActionMessages::invalid_name`, `jeod::BodyActionMessages::invalid_object`, `InvalidSet`, `orbit_frame`, `orbit_frame_name`, `planet`, `planet_name`, `jeod::DynBodyInit::reference_ref_frame`, `jeod::DynBodyInit::reverse_sense`, `SlrEcclIncAscnodeArgperTanom`, `SmaEcclIncAscnodeArgperManom`, `SmaEcclIncAscnodeArgperTanom`, `SmaEcclIncAscnodeArgperTimeperi`, `SmaIncAscnodeArglatRadRadvel`, and `jeod::BodyAction::validate_name()`.

8.17.5 Friends And Related Function Documentation

8.17.5.1 `init_attrjeod__DynBodyInitOrbit`

```
void init_attrjeod__DynBodyInitOrbit ( ) [friend]
```

8.17.5.2 `InputProcessor`

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

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

8.17.6 Field Documentation

8.17.6.1 `alt_apoapsis`

```
double jeod::DynBodyInitOrbit::alt_apoapsis
```

Apoapsis altitude.

`trick_units(m)`

Definition at line 210 of file `dyn_body_init_orbit.hh`.

Referenced by `apply()`, and `DynBodyInitOrbit()`.

8.17.6.2 alt_periapsis

```
double jeod::DynBodyInitOrbit::alt_periapsis
```

Periapsis altitude.

trick_units(m)

Definition at line 205 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.3 arg_latitude

```
double jeod::DynBodyInitOrbit::arg_latitude
```

Argument of latitude.

trick_units(rad)

Definition at line 245 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.4 arg_periapsis

```
double jeod::DynBodyInitOrbit::arg_periapsis
```

Argument of periapsis.

trick_units(rad)

Definition at line 240 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.5 ascending_node

```
double jeod::DynBodyInitOrbit::ascending_node
```

Longitude (or right ascension) of ascending node.

trick_units(rad)

Definition at line 235 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.6 eccentricity

```
double jeod::DynBodyInitOrbit::eccentricity
```

Eccentricity.

trick_units(—)

Definition at line 225 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.7 inclination

```
double jeod::DynBodyInitOrbit::inclination
```

Inclination.

trick_units(rad)

Definition at line 230 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.8 mean_anomaly

```
double jeod::DynBodyInitOrbit::mean_anomaly
```

Mean anomaly.

trick_units(rad)

Definition at line 255 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.9 orb_radius

```
double jeod::DynBodyInitOrbit::orb_radius
```

Distance from center of planet.

trick_units(m)

Definition at line 215 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.10 orbit_frame

EphemerisRefFrame* jeod::DynBodyInitOrbit::orbit_frame [protected]

The orbit reference frame (ignoring rotation)

trick_io(**)

Definition at line 273 of file dyn_body_init_orbit.hh.

Referenced by apply(), DynBodyInitOrbit(), and initialize().

8.17.6.11 orbit_frame_name

std::string jeod::DynBodyInitOrbit::orbit_frame_name

Planet reference frame name, optionally dot-prefixed with the planet name.

If this specifies a rotating frame, a non-rotating frame instantaneously co-aligned with the rotating frame is assumed.trick_units(-)

Definition at line 185 of file dyn_body_init_orbit.hh.

Referenced by initialize().

8.17.6.12 planet

Planet* jeod::DynBodyInitOrbit::planet [protected]

The planet.

trick_io(**)

Definition at line 268 of file dyn_body_init_orbit.hh.

Referenced by apply(), DynBodyInitOrbit(), and initialize().

8.17.6.13 planet_name

std::string jeod::DynBodyInitOrbit::planet_name

The name of the planet around which the orbit is to be established.

This must be supplied, must name a planet, and the planet must have a gravity model.trick_units(-)

Definition at line 178 of file dyn_body_init_orbit.hh.

Referenced by initialize().

8.17.6.14 radial_vel

```
double jeod::DynBodyInitOrbit::radial_vel
```

Time derivative of the orbital radius.

trick_units(m/s)

Definition at line 220 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.15 semi_latus_rectum

```
double jeod::DynBodyInitOrbit::semi_latus_rectum
```

Semi-latus rectum.

trick_units(m)

Definition at line 200 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.16 semi_major_axis

```
double jeod::DynBodyInitOrbit::semi_major_axis
```

Semi-major axis.

trick_units(m)

Definition at line 195 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.17 set

```
OrbitalSet jeod::DynBodyInitOrbit::set
```

Specifies which set of orbital elements specify the orbit.

trick_units(-)

Definition at line 190 of file dyn_body_init_orbit.hh.

8.17.6.18 time_periapsis

```
double jeod::DynBodyInitOrbit::time_periapsis
```

Time since periapsis passage.

trick_units(s)

Definition at line 250 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.19 true_anomaly

```
double jeod::DynBodyInitOrbit::true_anomaly
```

True anomaly.

trick_units(rad)

Definition at line 260 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

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

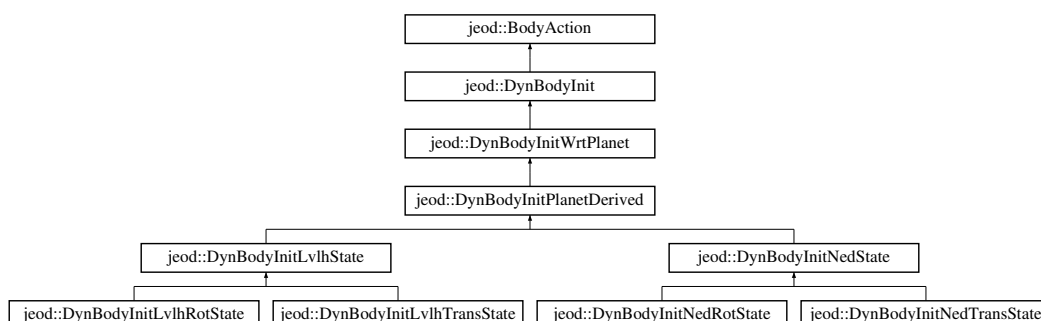
- [dyn_body_init_orbit.hh](#)
- [dyn_body_init_orbit.cc](#)

8.18 jeod::DynBodyInitPlanetDerived Class Reference

(Initialize selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.

```
#include <dyn_body_init_planet_derived.hh>
```

Inheritance diagram for jeod::DynBodyInitPlanetDerived:



Public Member Functions

- [DynBodyInitPlanetDerived](#) ()
[DynBodyInitPlanetDerived](#) default constructor.
- virtual [~DynBodyInitPlanetDerived](#) ()
[DynBodyInitPlanetDerived](#) destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize the initializer.
- virtual bool [is_ready](#) (void)
Indicate whether the initializer is ready to run.
- virtual void [apply](#) (DynManager &dyn_manager)
Apply the initializer: This is just a pass through.

Data Fields

- std::string [ref_body_name](#)
The name of the vehicle whose composite body frame is used to build the derived state with respect to which the vehicle initialization data are referenced.

Protected Attributes

- DynBody * [ref_body](#)
The vehicle corresponding to the [ref_body_name](#).
- RefFrameItems::Items [required_items](#)
The state elements in the reference body's composite body frame that must be set before this initializer can proceed.
- bool [body_is_required](#)
If true (default), the [ref_body](#) cannot be NULL.

Private Member Functions

- [DynBodyInitPlanetDerived](#) (const [DynBodyInitPlanetDerived](#) &)
- [DynBodyInitPlanetDerived](#) & [operator=](#) (const [DynBodyInitPlanetDerived](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitPlanetDerived](#) ()

Additional Inherited Members

8.18.1 Detailed Description

(Initialize selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.

Definition at line 87 of file [dyn_body_init_planet_derived.hh](#).

8.18.2 Constructor & Destructor Documentation

8.18.2.1 DynBodyInitPlanetDerived() [1/2]

```
jeod::DynBodyInitPlanetDerived::DynBodyInitPlanetDerived (
    const DynBodyInitPlanetDerived & ) [private]
```

8.18.2.2 DynBodyInitPlanetDerived() [2/2]

```
jeod::DynBodyInitPlanetDerived::DynBodyInitPlanetDerived (
    void )
```

[DynBodyInitPlanetDerived](#) default constructor.

Definition at line 51 of file `dyn_body_init_planet_derived.cc`.

8.18.2.3 ~DynBodyInitPlanetDerived()

```
jeod::DynBodyInitPlanetDerived::~DynBodyInitPlanetDerived (
    void ) [virtual]
```

[DynBodyInitPlanetDerived](#) destructor.

Definition at line 67 of file `dyn_body_init_planet_derived.cc`.

8.18.3 Member Function Documentation

8.18.3.1 apply()

```
void jeod::DynBodyInitPlanetDerived::apply (
    DynManager & dyn_manager ) [virtual]
```

Apply the initializer: This is just a pass through.

A derived class is responsible for setting the state that the [DynBodyInit](#) uses to initialize the state.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Reimplemented in [jeod::DynBodyInitNedState](#), and [jeod::DynBodyInitLvHState](#).

Definition at line 130 of file `dyn_body_init_planet_derived.cc`.

References [jeod::DynBodyInitWrtPlanet::apply\(\)](#).

Referenced by [jeod::DynBodyInitLvHState::apply\(\)](#), and [jeod::DynBodyInitNedState::apply\(\)](#).

8.18.3.2 initialize()

```
void jeod::DynBodyInitPlanetDerived::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize the initializer.

Parameters

<code>in, out</code>	<code>dyn_manager</code>	Dynamics manager
----------------------	--------------------------	------------------

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

Reimplemented in [jeod::DynBodyInitNedState](#), [jeod::DynBodyInitLvHState](#), [jeod::DynBodyInitLvHRotState](#), [jeod::DynBodyInitLvHTransState](#), [jeod::DynBodyInitNedRotState](#), and [jeod::DynBodyInitNedTransState](#).

Definition at line 79 of file `dyn_body_init_planet_derived.cc`.

References [body_is_required](#), [jeod::DynBodyInit::find_dyn_body\(\)](#), [jeod::DynBodyInitWrtPlanet::initialize\(\)](#), [ref_body](#), and [ref_body_name](#).

Referenced by [jeod::DynBodyInitLvHState::initialize\(\)](#), and [jeod::DynBodyInitNedState::initialize\(\)](#).

8.18.3.3 is_ready()

```
bool jeod::DynBodyInitPlanetDerived::is_ready (
    void ) [virtual]
```

Indicate whether the initializer is ready to run.

When the state is based on some reference body, that reference vehicle's composite body frame must contain the specified required items before the initializer can run.

Returns

Is initializer ready?

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

Definition at line 107 of file `dyn_body_init_planet_derived.cc`.

References [jeod::DynBodyInitWrtPlanet::is_ready\(\)](#), [ref_body](#), and [required_items](#).

8.18.3.4 operator=()

```
DynBodyInitPlanetDerived& jeod::DynBodyInitPlanetDerived::operator= (
    const DynBodyInitPlanetDerived & ) [private]
```

8.18.4 Friends And Related Function Documentation

8.18.4.1 init_attrjeod__DynBodyInitPlanetDerived

```
void init_attrjeod__DynBodyInitPlanetDerived ( ) [friend]
```

8.18.4.2 InputProcessor

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

Definition at line 89 of file dyn_body_init_planet_derived.hh.

8.18.5 Field Documentation

8.18.5.1 body_is_required

```
bool jeod::DynBodyInitPlanetDerived::body_is_required [protected]
```

If true (default), the ref_body cannot be NULL.

If false, the derived class must provide some means other than using a derived state to set the reference Ref↔
Frame.trick_io(**)

Definition at line 124 of file dyn_body_init_planet_derived.hh.

Referenced by jeod::DynBodyInitNedState::DynBodyInitNedState(), jeod::DynBodyInitLvlhState::initialize(), jeod↔
::DynBodyInitNedState::initialize(), and initialize().

8.18.5.2 ref_body

```
DynBody* jeod::DynBodyInitPlanetDerived::ref_body [protected]
```

The vehicle corresponding to the ref_body_name.

Note that this is not a user-inputtable item. `trick_io(**)`

Definition at line 109 of file `dyn_body_init_planet_derived.hh`.

Referenced by `jeod::DynBodyInitLvHState::apply()`, `jeod::DynBodyInitNedState::apply()`, `jeod::DynBodyInitLvHRotState::initialize()`, `initialize()`, and `is_ready()`.

8.18.5.3 ref_body_name

```
std::string jeod::DynBodyInitPlanetDerived::ref_body_name
```

The name of the vehicle whose composite body frame is used to build the derived state with respect to which the vehicle initialization data are referenced.

`trick_units(-)`

Definition at line 101 of file `dyn_body_init_planet_derived.hh`.

Referenced by `jeod::DynBodyInitLvHRotState::initialize()`, `jeod::DynBodyInitNedState::initialize()`, and `initialize()`.

8.18.5.4 required_items

```
RefFrameItems::Items jeod::DynBodyInitPlanetDerived::required_items [protected]
```

The state elements in the reference body's composite body frame that must be set before this initializer can proceed.

This is not user-inputtable; derived classes should set this item. The default is to require the full state to be set. `trick_io(**)`

Definition at line 117 of file `dyn_body_init_planet_derived.hh`.

Referenced by `jeod::DynBodyInitLvHState::DynBodyInitLvHState()`, `jeod::DynBodyInitNedState::DynBodyInitNedState()`, and `is_ready()`.

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

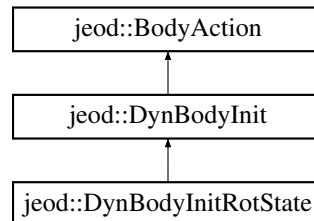
- [dyn_body_init_planet_derived.hh](#)
- [dyn_body_init_planet_derived.cc](#)

8.19 jeod::DynBodyInitRotState Class Reference

Initialize aspects of a vehicle's rotational state.

```
#include <dyn_body_init_rot_state.hh>
```

Inheritance diagram for jeod::DynBodyInitRotState:



Public Types

- enum [StateItems](#) { [Both](#) = 0, [Attitude](#) = 1, [Rate](#) = 2 }
- Identify which of attitude/rate is to be initialized.*

Public Member Functions

- [DynBodyInitRotState](#) ()
Construct a [DynBodyInitRotState](#) object.
- virtual [~DynBodyInitRotState](#) ()
Destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize aspects of this object and forward the initializer to the immediate parent class.
- virtual void [apply](#) (DynManager &dyn_manager)
Apply the initializer.
- virtual RefFrameItems::Items [initializes_what](#) (void)
Indicate what parts of the vehicle state this object initializes.
- virtual bool [is_ready](#) (void)
Indicate whether this initializer is ready to be applied.

Data Fields

- [StateItems](#) [state_items](#)
State items to be initialized – attitude, rate, or both.

Private Member Functions

- [DynBodyInitRotState](#) (const [DynBodyInitRotState](#) &)
- [DynBodyInitRotState](#) & [operator=](#) (const [DynBodyInitRotState](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitRotState](#) ()

Additional Inherited Members

8.19.1 Detailed Description

Initialize aspects of a vehicle's rotational state.

Definition at line 83 of file `dyn_body_init_rot_state.hh`.

8.19.2 Member Enumeration Documentation

8.19.2.1 StateItems

```
enum jeod::DynBodyInitRotState::StateItems
```

Identify which of attitude/rate is to be initialized.

Enumerator

Both	Initialize both attitude and rate.
Attitude	Initialize attitude only.
Rate	Initialize rate only.

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

8.19.3 Constructor & Destructor Documentation

8.19.3.1 DynBodyInitRotState() [1/2]

```
jeod::DynBodyInitRotState::DynBodyInitRotState (  
    const DynBodyInitRotState & ) [private]
```

8.19.3.2 DynBodyInitRotState() [2/2]

```
jeod::DynBodyInitRotState::DynBodyInitRotState (
    void )
```

Construct a [DynBodyInitRotState](#) object.

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

8.19.3.3 ~DynBodyInitRotState()

```
jeod::DynBodyInitRotState::~DynBodyInitRotState (
    void ) [inline], [virtual]
```

Destructor.

Definition at line 150 of file `dyn_body_init_rot_state.hh`.

8.19.4 Member Function Documentation

8.19.4.1 apply()

```
void jeod::DynBodyInitRotState::apply (
    DynManager & dyn_manager ) [virtual]
```

Apply the initializer.

Parameters

<code>in, out</code>	<code><i>dyn_manager</i></code>	Dynamics manager
----------------------	---------------------------------	------------------

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

Definition at line 171 of file `dyn_body_init_rot_state.cc`.

References `jeod::DynBodyInit::apply()`, and `jeod::DynBodyInit::apply_user_inputs()`.

8.19.4.2 initialize()

```
void jeod::DynBodyInitRotState::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize aspects of this object and forward the initializer to the immediate parent class.

This class needs no initialization per se.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager
----------------	--------------------	------------------

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

Definition at line 147 of file `dyn_body_init_rot_state.cc`.

References `jeod::BodyAction::action_identifier`, `Attitude`, `Both`, `jeod::BodyActionMessages::illegal_value`, `jeod::DynBodyInit::initialize()`, `Rate`, and `state_items`.

8.19.4.3 initializes_what()

```
RefFrameItems::Items jeod::DynBodyInitRotState::initializes_what (
    void ) [virtual]
```

Indicate what parts of the vehicle state this object initializes.

This is depends on the state specified by the user: `Both`=attitude and rate, `Attitude`=attitude, `Rate`=rate.

Returns

States initialized

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

Definition at line 70 of file `dyn_body_init_rot_state.cc`.

References `Attitude`, `Both`, `Rate`, and `state_items`.

Referenced by `is_ready()`.

8.19.4.4 is_ready()

```
bool jeod::DynBodyInitRotState::is_ready (
    void ) [virtual]
```

Indicate whether this initializer is ready to be applied.

The full rotational state of the reference reference frame must be known to compute the subject reference frame's rotational state.

Returns

Is initializer ready?

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

Definition at line 98 of file `dyn_body_init_rot_state.cc`.

References `jeod::BodyAction::action_identifier`, `initializes_what()`, `jeod::BodyActionMessages::invalid_object`, `jeod::DynBodyInit::is_ready()`, and `jeod::DynBodyInit::reference_ref_frame`.

8.19.4.5 operator=()

```
DynBodyInitRotState& jeod::DynBodyInitRotState::operator= (
    const DynBodyInitRotState & ) [private]
```

8.19.5 Friends And Related Function Documentation

8.19.5.1 init_attrjeod__DynBodyInitRotState

```
void init_attrjeod__DynBodyInitRotState ( ) [friend]
```

8.19.5.2 InputProcessor

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

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

8.19.6 Field Documentation

8.19.6.1 state_items

```
StateItems jeod::DynBodyInitRotState::state_items
```

State items to be initialized – attitude, rate, or both.

`trick_units(-)`

Definition at line 109 of file `dyn_body_init_rot_state.hh`.

Referenced by `initialize()`, and `initializes_what()`.

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

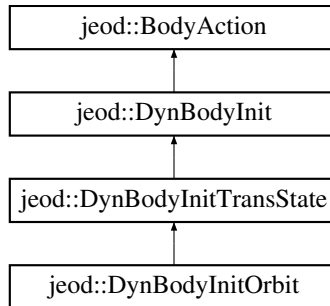
- [dyn_body_init_rot_state.hh](#)
- [dyn_body_init_rot_state.cc](#)

8.20 jeod::DynBodyInitTransState Class Reference

Initialize aspects of a vehicle's translational state.

```
#include <dyn_body_init_trans_state.hh>
```

Inheritance diagram for jeod::DynBodyInitTransState:



Public Types

- enum [StateItems](#) { [Both](#) = 0, [Position](#) = 1, [Velocity](#) = 2 }
Identify which of position/velocity is to be initialized.

Public Member Functions

- [DynBodyInitTransState](#) ()
Construct a [DynBodyInitTransState](#) object.
- virtual [~DynBodyInitTransState](#) ()
Destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize aspects of this object and forward the initializer to the immediate parent class.
- virtual void [apply](#) (DynManager &dyn_manager)
Apply the initializer.
- virtual RefFrameItems::Items [initializes_what](#) (void)
Indicate what parts of the vehicle state this object initializes.
- virtual bool [is_ready](#) (void)
Indicate whether this initializer is ready to be applied.

Data Fields

- [StateItems](#) [state_items](#)
State items to be initialized – position, velocity, or both.

Private Member Functions

- [DynBodyInitTransState](#) (const [DynBodyInitTransState](#) &)
- [DynBodyInitTransState](#) & [operator=](#) (const [DynBodyInitTransState](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitTransState](#) ()

Additional Inherited Members

8.20.1 Detailed Description

Initialize aspects of a vehicle's translational state.

Definition at line 83 of file `dyn_body_init_trans_state.hh`.

8.20.2 Member Enumeration Documentation

8.20.2.1 StateItems

```
enum jeod::DynBodyInitTransState::StateItems
```

Identify which of position/velocity is to be initialized.

Enumerator

Both	Initialize both position and velocity.
Position	Initialize position only.
Velocity	Initialize velocity only.

Definition at line 93 of file `dyn_body_init_trans_state.hh`.

8.20.3 Constructor & Destructor Documentation

8.20.3.1 DynBodyInitTransState() [1/2]

```
jeod::DynBodyInitTransState::DynBodyInitTransState (  
    void )
```

Construct a [DynBodyInitTransState](#) object.

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

8.20.3.2 ~DynBodyInitTransState()

```
jeod::DynBodyInitTransState::~~DynBodyInitTransState (
    void ) [inline], [virtual]
```

Destructor.

Definition at line 147 of file `dyn_body_init_trans_state.hh`.

8.20.3.3 DynBodyInitTransState() [2/2]

```
jeod::DynBodyInitTransState::DynBodyInitTransState (
    const DynBodyInitTransState & ) [private]
```

8.20.4 Member Function Documentation

8.20.4.1 apply()

```
void jeod::DynBodyInitTransState::apply (
    DynManager & dyn_manager ) [virtual]
```

Apply the initializer.

Parameters

<code>in, out</code>	<code>dyn_manager</code>	Dynamics manager
----------------------	--------------------------	------------------

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

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

Definition at line 177 of file `dyn_body_init_trans_state.cc`.

References [jeod::DynBodyInit::apply\(\)](#), and [jeod::DynBodyInit::apply_user_inputs\(\)](#).

Referenced by [jeod::DynBodyInitOrbit::apply\(\)](#).

8.20.4.2 initialize()

```
void jeod::DynBodyInitTransState::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize aspects of this object and forward the initializer to the immediate parent class.

This class needs no initialization per se.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager
----------------	--------------------	------------------

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

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

Definition at line 152 of file `dyn_body_init_trans_state.cc`.

References [jeod::BodyAction::action_identifier](#), [Both](#), [jeod::BodyActionMessages::illegal_value](#), [jeod::DynBodyInit::initialize\(\)](#), [Position](#), [state_items](#), and [Velocity](#).

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

8.20.4.3 initializes_what()

```
RefFrameItems::Items jeod::DynBodyInitTransState::initializes_what (
    void ) [virtual]
```

Indicate what parts of the vehicle state this object initializes.

This is depends on the state specified by the user: [Both](#)=position and velocity, [Position](#)=position, [Velocity](#)=velocity.

Returns

States initialized

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

Definition at line 70 of file `dyn_body_init_trans_state.cc`.

References [Both](#), [Position](#), [state_items](#), and [Velocity](#).

Referenced by [is_ready\(\)](#).

8.20.4.4 is_ready()

```
bool jeod::DynBodyInitTransState::is_ready (
    void ) [virtual]
```

Indicate whether this initializer is ready to be applied.

The full state of the reference reference frame must be known to compute the position and velocity of the subject reference frame.

Returns

Is initializer ready?

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

Definition at line 98 of file `dyn_body_init_trans_state.cc`.

References [jeod::BodyAction::action_identifier](#), [initializes_what\(\)](#), [jeod::BodyActionMessages::invalid_object](#), [jeod::DynBodyInit::is_ready\(\)](#), and [jeod::DynBodyInit::reference_ref_frame](#).

8.20.4.5 operator=()

```
DynBodyInitTransState& jeod::DynBodyInitTransState::operator= (
    const DynBodyInitTransState & ) [private]
```

8.20.5 Friends And Related Function Documentation

8.20.5.1 init_attrjeod__DynBodyInitTransState

```
void init_attrjeod__DynBodyInitTransState ( ) [friend]
```

8.20.5.2 InputProcessor

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

Definition at line 85 of file dyn_body_init_trans_state.hh.

8.20.6 Field Documentation

8.20.6.1 state_items

```
StateItems jeod::DynBodyInitTransState::state_items
```

State items to be initialized – position, velocity, or both.

trick_units(–)

Definition at line 107 of file dyn_body_init_trans_state.hh.

Referenced by initialize(), and initializes_what().

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

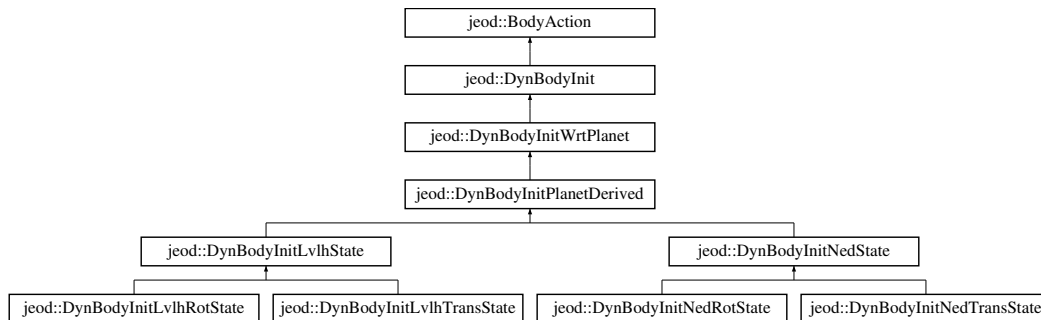
- [dyn_body_init_trans_state.hh](#)
- [dyn_body_init_trans_state.cc](#)

8.21 jeod::DynBodyInitWrtPlanet Class Reference

Initialize selected aspects of a vehicle's state with respect to some frame based on the planet.

```
#include <dyn_body_init_wrt_planet.hh>
```

Inheritance diagram for jeod::DynBodyInitWrtPlanet:



Public Member Functions

- [DynBodyInitWrtPlanet](#) ()
DynBodyInitWrtPlanet default constructor.
- virtual [~DynBodyInitWrtPlanet](#) ()
DynBodyInitWrtPlanet destructor.
- virtual void [initialize](#) (DynManager &dyn_manager)
Initialize the initializer.
- virtual RefFrameItems::Items [initializes_what](#) (void)
Indicate what parts of the vehicle state this object initializes.
- virtual bool [is_ready](#) (void)
Indicate whether the initializer is ready to run.
- virtual void [apply](#) (DynManager &dyn_manager)
Apply the initializer.

Data Fields

- std::string [planet_name](#)
The name of the planet about which the reference body's LVLH frame is to be computed.
- RefFrameItems::Items [set_items](#)
The state elements to be set by this initializer.

Protected Attributes

- Planet * [planet](#)
The planet corresponding to the planet_name.

Private Member Functions

- [DynBodyInitWrtPlanet](#) (const [DynBodyInitWrtPlanet](#) &)
- [DynBodyInitWrtPlanet](#) & [operator=](#) (const [DynBodyInitWrtPlanet](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__DynBodyInitWrtPlanet](#) ()

Additional Inherited Members

8.21.1 Detailed Description

Initialize selected aspects of a vehicle's state with respect to some frame based on the planet.

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

8.21.2 Constructor & Destructor Documentation

8.21.2.1 [DynBodyInitWrtPlanet\(\)](#) [1/2]

```
jeod::DynBodyInitWrtPlanet::DynBodyInitWrtPlanet (
    const DynBodyInitWrtPlanet & ) [private]
```

8.21.2.2 [DynBodyInitWrtPlanet\(\)](#) [2/2]

```
jeod::DynBodyInitWrtPlanet::DynBodyInitWrtPlanet (
    void )
```

[DynBodyInitWrtPlanet](#) default constructor.

Note that by default, this class will try to set the whole enchilada.

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

8.21.2.3 [~DynBodyInitWrtPlanet\(\)](#)

```
jeod::DynBodyInitWrtPlanet::~~DynBodyInitWrtPlanet (
    void ) [virtual]
```

[DynBodyInitWrtPlanet](#) destructor.

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

8.21.3 Member Function Documentation

8.21.3.1 [apply\(\)](#)

```
void jeod::DynBodyInitWrtPlanet::apply (
    DynManager & dyn_manager ) [virtual]
```

Apply the initializer.

This is just a pass-through. Some derived class must do the actual work.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager
----------------	--------------------	------------------

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

Reimplemented in [jeod::DynBodyInitPlanetDerived](#), [jeod::DynBodyInitNedState](#), and [jeod::DynBodyInitLvlhState](#).

Definition at line 131 of file `dyn_body_init_wrt_planet.cc`.

References [jeod::DynBodyInit::apply\(\)](#).

Referenced by [jeod::DynBodyInitPlanetDerived::apply\(\)](#).

8.21.3.2 initialize()

```
void jeod::DynBodyInitWrtPlanet::initialize (
    DynManager & dyn_manager ) [virtual]
```

Initialize the initializer.

Parameters

<i>in, out</i>	<i>dyn_manager</i>	Dynamics manager
----------------	--------------------	------------------

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

Reimplemented in [jeod::DynBodyInitPlanetDerived](#), [jeod::DynBodyInitNedState](#), [jeod::DynBodyInitLvlhState](#), [jeod::DynBodyInitLvlhRotState](#), [jeod::DynBodyInitLvlhTransState](#), [jeod::DynBodyInitNedRotState](#), and [jeod::DynBodyInitNedTransState](#).

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

References [jeod::DynBodyInit::find_planet\(\)](#), [jeod::DynBodyInit::initialize\(\)](#), [planet](#), [planet_name](#), and [jeod::DynBodyInit::reference_ref_frame](#).

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

8.21.3.3 initializes_what()

```
RefFrameItems::Items jeod::DynBodyInitWrtPlanet::initializes_what (
    void ) [virtual]
```

Indicate what parts of the vehicle state this object initializes.

Returns

States initialized

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

Definition at line 101 of file `dyn_body_init_wrt_planet.cc`.

References [set_items](#).

8.21.3.4 is_ready()

```
bool jeod::DynBodyInitWrtPlanet::is_ready (
    void ) [virtual]
```

Indicate whether the initializer is ready to run.

This particular implementation is just a pass-through.

Returns

Is initializer ready?

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

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

Definition at line 115 of file `dyn_body_init_wrt_planet.cc`.

References [jeod::DynBodyInit::is_ready\(\)](#).

Referenced by [jeod::DynBodyInitPlanetDerived::is_ready\(\)](#).

8.21.3.5 operator=()

```
DynBodyInitWrtPlanet& jeod::DynBodyInitWrtPlanet::operator= (
    const DynBodyInitWrtPlanet & ) [private]
```

8.21.4 Friends And Related Function Documentation

8.21.4.1 init_attrjeod__DynBodyInitWrtPlanet

```
void init_attrjeod__DynBodyInitWrtPlanet ( ) [friend]
```

8.21.4.2 InputProcessor

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

Definition at line 89 of file `dyn_body_init_wrt_planet.hh`.

8.21.5 Field Documentation

8.21.5.1 planet

```
Planet* jeod::DynBodyInitWrtPlanet::planet [protected]
```

The planet corresponding to the planet_name.

Note that this is not a user inputtable item.trick_io(**)

Definition at line 114 of file dyn_body_init_wrt_planet.hh.

Referenced by jeod::DynBodyInitLvlhState::apply(), jeod::DynBodyInitNedState::apply(), and initialize().

8.21.5.2 planet_name

```
std::string jeod::DynBodyInitWrtPlanet::planet_name
```

The name of the planet about which the reference body's LVLH frame is to be computed.

trick_units(-)

Definition at line 100 of file dyn_body_init_wrt_planet.hh.

Referenced by jeod::DynBodyInitNedState::apply(), and initialize().

8.21.5.3 set_items

```
RefFrameItems::Items jeod::DynBodyInitWrtPlanet::set_items
```

The state elements to be set by this initializer.

trick_units(-)

Definition at line 105 of file dyn_body_init_wrt_planet.hh.

Referenced by jeod::DynBodyInitLvlhState::apply(), jeod::DynBodyInitNedState::apply(), jeod::DynBodyInitLvlhRotState::DynBodyInitLvlhRotState(), jeod::DynBodyInitLvlhTransState::DynBodyInitLvlhTransState(), jeod::DynBodyInitNedRotState::DynBodyInitNedRotState(), jeod::DynBodyInitNedTransState::DynBodyInitNedTransState(), jeod::DynBodyInitNedRotState::initialize(), jeod::DynBodyInitLvlhTransState::initialize(), jeod::DynBodyInitNedTransState::initialize(), jeod::DynBodyInitLvlhRotState::initialize(), and initializes_what().

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

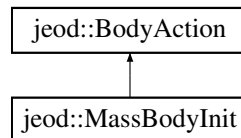
- [dyn_body_init_wrt_planet.hh](#)
- [dyn_body_init_wrt_planet.cc](#)

8.22 jeod::MassBodyInit Class Reference

Base class for initializing a MassBody.

```
#include <mass_body_init.hh>
```

Inheritance diagram for jeod::MassBodyInit:



Public Member Functions

- [MassBodyInit](#) ()
Construct a [MassBodyInit](#).
- virtual [~MassBodyInit](#) ()
Destructor.
- virtual void [apply](#) (DynManager &dyn_manager)
Initialize the core mass properties of the subject MassBody.

Data Fields

- MassPropertiesInit [properties](#)
Specifications for the subject mass body's core mass properties.
- MassPointInit * [points](#)
Specifications for the subject mass body's mass points.
- unsigned int [num_points](#)
Size of the points array.

Private Member Functions

- [MassBodyInit](#) (const [MassBodyInit](#) &)
- [MassBodyInit](#) & [operator=](#) (const [MassBodyInit](#) &)

Friends

- class [InputProcessor](#)
- void [init_attrjeod__MassBodyInit](#) ()

Additional Inherited Members

8.22.1 Detailed Description

Base class for initializing a MassBody.

Items initialized by this action are

- The body's core mass properties
- The body's mass points.

Definition at line 90 of file mass_body_init.hh.

8.22.2 Constructor & Destructor Documentation

8.22.2.1 MassBodyInit() [1/2]

```
jeod::MassBodyInit::MassBodyInit (
    const MassBodyInit & ) [private]
```

8.22.2.2 MassBodyInit() [2/2]

```
jeod::MassBodyInit::MassBodyInit (
    void )
```

Construct a [MassBodyInit](#).

Definition at line 55 of file mass_body_init.cc.

8.22.2.3 ~MassBodyInit()

```
jeod::MassBodyInit::~~MassBodyInit (
    void ) [inline], [virtual]
```

Destructor.

Definition at line 143 of file mass_body_init.hh.

8.22.3 Member Function Documentation

8.22.3.1 apply()

```
void jeod::MassBodyInit::apply (
    DynManager & dyn_manager ) [virtual]
```

Initialize the core mass properties of the subject MassBody.

Parameters

<code>in, out</code>	<code>dyn_manager</code>	Jeod manager
----------------------	--------------------------	--------------

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

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

References `jeod::BodyAction::action_identifier`, `jeod::BodyAction::apply()`, `jeod::BodyAction::mass_subject`, `num_points`, `points`, `properties`, and `jeod::BodyActionMessages::trace`.

8.22.3.2 operator=()

```
MassBodyInit& jeod::MassBodyInit::operator= (
    const MassBodyInit & ) [private]
```

8.22.4 Friends And Related Function Documentation

8.22.4.1 init_attrjeod__MassBodyInit

```
void init_attrjeod__MassBodyInit ( ) [friend]
```

8.22.4.2 InputProcessor

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

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

8.22.5 Field Documentation

8.22.5.1 num_points

```
unsigned int jeod::MassBodyInit::num_points
```

Size of the points array.

`trick_units(-)`

Definition at line 112 of file `mass_body_init.hh`.

Referenced by `apply()`.

8.22.5.2 points

```
MassPointInit* jeod::MassBodyInit::points
```

Specifications for the subject mass body's mass points.

trick_units(—)

Definition at line 107 of file mass_body_init.hh.

Referenced by apply().

8.22.5.3 properties

```
MassPropertiesInit jeod::MassBodyInit::properties
```

Specifications for the subject mass body's core mass properties.

trick_units(—)

Definition at line 102 of file mass_body_init.hh.

Referenced by apply().

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

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

Chapter 9

File Documentation

9.1 body_action.cc File Reference

Define methods for the BodyAction class.

```
#include <cstdlib>
#include <stdlib.h>
#include <string>
#include <typeinfo>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/body_action.hh"
#include "../include/body_action_messages.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.1.1 Detailed Description

Define methods for the BodyAction class.

9.2 body_action.hh File Reference

Define the class BodyAction, the base class used for performing actions on a MassBody or DynBody object.

```
#include <string>
#include "dynamics/mass/include/class_declarations.hh"
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "dynamics/mass/include/mass.hh"
```

Data Structures

- class [jeod::BodyAction](#)
[BodyAction](#) is the base class for the [BodyAction](#) model.

Namespaces

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

9.2.1 Detailed Description

Define the class `BodyAction`, the base class used for performing actions on a `MassBody` or `DynBody` object.

9.3 `body_action_messages.cc` File Reference

Implement the class `BodyActionMessages`.

```
#include "../include/body_action_messages.hh"
```

Namespaces

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

Macros

- `#define` [PATH](#) "dynamics/body_action/"

9.3.1 Detailed Description

Implement the class `BodyActionMessages`.

9.4 `body_action_messages.hh` File Reference

Define the class `BodyActionMessages`, the class that specifies the message IDs used in the `BodyAction` model.

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

Data Structures

- class [jeod::BodyActionMessages](#)
Specifies the message IDs used in the [BodyAction](#) model.

Namespaces

- [jeod](#)
Namespace jeod.

9.4.1 Detailed Description

Define the class BodyActionMessages, the class that specifies the message IDs used in the BodyAction model.

9.5 body_attach.cc File Reference

Define methods for the mass body initialization class.

```
#include <cstdint>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/mass/include/mass.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_attach.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.5.1 Detailed Description

Define methods for the mass body initialization class.

9.6 body_attach.hh File Reference

Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "dynamics/mass/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "body_action.hh"
```

Data Structures

- class [jeod::BodyAttach](#)

Provides the basic ability to attach one MassBody to another.

Namespaces

- [jeod](#)

Namespace jeod.

9.6.1 Detailed Description

Define the class MassBodyAttach, the base class used for attaching a pair of MassBody objects to one another.

9.7 body_attach_aligned.cc File Reference

Define methods for the mass body initialization class.

```
#include <cstddef>
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_attach_aligned.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.7.1 Detailed Description

Define methods for the mass body initialization class.

9.8 body_attach_aligned.hh File Reference

Define the class MassBodyAttachAligned, which causes one MassBody to be attached to another at a pair of MassPoints.

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


Data Structures

- class [jeod::BodyAttachAligned](#)
Attaches a pair of MassBody objects at a pair of MassPoints.

Namespaces

- [jeod](#)
Namespace jeod.

9.8.1 Detailed Description

Define the class MassBodyAttachAligned, which causes one MassBody to be attached to another at a pair of MassPoints.

9.9 body_attach_matrix.cc File Reference

Define methods for the mass body initialization class.

```
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/math/include/vector3.hh"
#include "../include/body_attach_matrix.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.9.1 Detailed Description

Define methods for the mass body initialization class.

9.10 body_attach_matrix.hh File Reference

Define the class MassBodyAttachMatrix, which causes one MassBody to be attached given a transformation.

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

Data Structures

- class [jeod::BodyAttachMatrix](#)

Attaches a pair of MassBody objects using the offset+matrix attach mechanism.

Namespaces

- [jeod](#)

Namespace jeod.

9.10.1 Detailed Description

Define the class MassBodyAttachMatrix, which causes one MassBody to be attached given a transformation.

9.11 body_detach.cc File Reference

Define methods for the MassBodyDetach class.

```
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_detach.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.11.1 Detailed Description

Define methods for the MassBodyDetach class.

9.12 body_detach.hh File Reference

Define the class MassBodyDetach, the base class used for detaching one MassBody object from one another.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "body_action.hh"
```

Data Structures

- class [jeod::BodyDetach](#)

Provides the basic ability to detach one MassBody from another.

Namespaces

- [jeod](#)

Namespace jeod.

9.12.1 Detailed Description

Define the class MassBodyDetach, the base class used for detaching one MassBody object from one another.

9.13 body_detach_specific.cc File Reference

Define methods for the BodyDetachSpecific class.

```
#include <cstdint>
#include <string>
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/body_detach_specific.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.13.1 Detailed Description

Define methods for the BodyDetachSpecific class.

9.14 body_detach_specific.hh File Reference

Define the class MassBodyDetachSpecific, the class used for detaching one MassBody object from another specified MassBody.

```
#include "dynamics/mass/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "body_action.hh"
```

Data Structures

- class [jeod::BodyDetachSpecific](#)

Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach←→_from body.

Namespaces

- [jeod](#)

Namespace jeod.

9.14.1 Detailed Description

Define the class MassBodyDetachSpecific, the class used for detaching one MassBody object from another specified MassBody.

9.15 body_reattach.cc File Reference

Define methods for the mass body initialization class.

```
#include "dynamics/mass/include/mass.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_reattach.hh"
#include "../include/body_action_messages.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.15.1 Detailed Description

Define methods for the mass body initialization class.

9.16 body_reattach.hh File Reference

Define the class MassBodyReattach, which causes one MassBody to be reattached given a transformation.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/orientation/include/orientation.hh"
#include "body_action.hh"
```

Data Structures

- class [jeod::BodyReattach](#)
Alters the nature of an existing attachment.

Namespaces

- [jeod](#)
Namespace jeod.

9.16.1 Detailed Description

Define the class MassBodyReattach, which causes one MassBody to be reattached given a transformation.

9.17 class_declarations.hh File Reference

Forward declarations of classes defined in dyn_body_init_XXX.hh files.

Namespaces

- [jeod](#)
Namespace jeod.

9.17.1 Detailed Description

Forward declarations of classes defined in dyn_body_init_XXX.hh files.

9.18 dyn_body_frame_switch.cc File Reference

Define methods for the class DynBodyFrameSwitch.

```
#include <cstddef>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↵
hh"
#include "environment/gravity/include/gravity_interaction.hh"
#include "environment/gravity/include/gravity_controls.hh"
#include "utils/math/include/vector3.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_frame_switch.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.18.1 Detailed Description

Define methods for the class DynBodyFrameSwitch.

9.19 dyn_body_frame_switch.hh File Reference

Define the class DynBodyFrameSwitch, the BodyAction derived class used for switch a DynBody's integration frame.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "utils/ref_frames/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "body_action.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↵
hh"
```

Data Structures

- class [jeod::DynBodyFrameSwitch](#)

Switch a DynBody's integration frame to a specified frame when the body switches to that integration frame's sphere of influence.

Namespaces

- [jeod](#)

Namespace jeod.

9.19.1 Detailed Description

Define the class DynBodyFrameSwitch, the BodyAction derived class used for switch a DynBody's integration frame.

9.20 dyn_body_init.cc File Reference

Define methods for the base body initialization class.

```
#include <cstdlib>
#include <typeinfo>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↵
hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.20.1 Detailed Description

Define methods for the base body initialization class.

9.21 dyn_body_init.hh File Reference

Define the class DynBodyInit, the base class used for initializing the state of a DynBody object.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "environment/planet/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/orientation/include/orientation.hh"
#include "utils/ref_frames/include/class_declarations.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "class_declarations.hh"
#include "body_action.hh"
```

Data Structures

- class [jeod::DynBodyInit](#)

Base class for initialize the state of a DynBody.

Namespaces

- [jeod](#)

Namespace jeod.

9.21.1 Detailed Description

Define the class DynBodyInit, the base class used for initializing the state of a DynBody object.

9.22 dyn_body_init_lvlh_rot_state.cc File Reference

Define methods for DynBodyInitLvLhRotState.

```
#include <cstdlib>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_lvlh_rot_state.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.22.1 Detailed Description

Define methods for DynBodyInitLvlhRotState.

9.23 dyn_body_init_lvlh_rot_state.hh File Reference

Define the class DynBodyInitLvlhRotState, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_lvlh_state.hh"
```

Data Structures

- class [jeod::DynBodyInitLvlhRotState](#)

Initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.

Namespaces

- [jeod](#)

Namespace jeod.

9.23.1 Detailed Description

Define the class DynBodyInitLvlhRotState, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame.

9.24 dyn_body_init_lvlh_state.cc File Reference

Define methods for the DynBodyInitLvlhState class.

```
#include <cstdint>
#include "dynamics/derived_state/include/lvlh_relative_derived_state.hh"
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_lvlh_state.hh"
```


Namespaces

- [jeod](#)

Namespace jeod.

9.24.1 Detailed Description

Define methods for the DynBodyInitLvlhState class.

9.25 dyn_body_init_lvlh_state.hh File Reference

Define the class DynBodyInitLvlhState, the base class for initializing selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.

```
#include "utils/lvlh_frame/include/lvlh_type.hh"
#include "utils/lvlh_frame/include/lvlh_frame.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_planet_derived.hh"
```

Data Structures

- class [jeod::DynBodyInitLvlhState](#)

Initialize selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.

Namespaces

- [jeod](#)

Namespace jeod.

9.25.1 Detailed Description

Define the class DynBodyInitLvlhState, the base class for initializing selected aspects of a vehicle's state with respect to some vehicle's LVLH frame.

9.26 dyn_body_init_lvlh_trans_state.cc File Reference

Define methods for DynBodyInitLvlhTransState.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_lvlh_trans_state.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.26.1 Detailed Description

Define methods for DynBodyInitLvlhTransState.

9.27 dyn_body_init_lvlh_trans_state.hh File Reference

Define the class DynBodyInitLvlhTransState, which initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_lvlh_state.hh"
```

Data Structures

- class [jeod::DynBodyInitLvlhTransState](#)

initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.

Namespaces

- [jeod](#)

Namespace jeod.

9.27.1 Detailed Description

Define the class DynBodyInitLvlhTransState, which initialize a vehicle's translational state with respect to some other vehicle's LVLH frame.

9.28 dyn_body_init_ned_rot_state.cc File Reference

Define methods for DynBodyInitNedRotState.

```
#include <cstdlib>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_ned_rot_state.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

9.28.1 Detailed Description

Define methods for DynBodyInitNedRotState.

9.29 dyn_body_init_ned_rot_state.hh File Reference

Define the class DynBodyInitNedRotState, which initialize a vehicle's rotational state wrt some other vehicle's North-East-Down frame.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_ned_state.hh"
```

Data Structures

- class [jeod::DynBodyInitNedRotState](#)
Initialize a vehicle's rotational state wrt some vehicle's North-East-Down frame.

Namespaces

- [jeod](#)
Namespace jeod.

9.29.1 Detailed Description

Define the class DynBodyInitNedRotState, which initialize a vehicle's rotational state wrt some other vehicle's North-East-Down frame.

9.30 dyn_body_init_ned_state.cc File Reference

Define methods for DynBodyInitNedState.

```
#include <cstdint>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/planet_fixed/north_east_down/include/north_east_down.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "utils/ref_frames/include/ref_frame_state.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_ned_state.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.30.1 Detailed Description

Define methods for DynBodyInitNedState.

9.31 dyn_body_init_ned_state.hh File Reference

Define the class DynBodyInitNedState, the base class for initializing selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/planet_fixed/planet_fixed_posn/include/alt_lat_long_state.↵
hh"
#include "utils/planet_fixed/north_east_down/include/north_east_down.hh"
#include "dyn_body_init_planet_derived.hh"
```

Data Structures

- class [jeod::DynBodyInitNedState](#)

Initialize selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.

Namespaces

- [jeod](#)

Namespace jeod.

9.31.1 Detailed Description

Define the class DynBodyInitNedState, the base class for initializing selected aspects of a vehicle's state with respect to either some vehicle's North-East-Down frame or the North-East-Down frame for a specified location on the planet.

9.32 dyn_body_init_ned_trans_state.cc File Reference

Define methods for DynBodyInitNedTransState.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_ned_trans_state.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.32.1 Detailed Description

Define methods for DynBodyInitNedTransState.

9.33 dyn_body_init_ned_trans_state.hh File Reference

Define the class DynBodyInitNedTransState, which initialize a vehicle's translational state wrt some other vehicle's North-East-Down frame.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_ned_state.hh"
```

Data Structures

- class [jeod::DynBodyInitNedTransState](#)

Initialize a vehicle's translational state wrt some vehicle's North-East-Down frame.

Namespaces

- [jeod](#)

Namespace jeod.

9.33.1 Detailed Description

Define the class DynBodyInitNedTransState, which initialize a vehicle's translational state wrt some other vehicle's North-East-Down frame.

9.34 dyn_body_init_orbit.cc File Reference

Define classes for items represented in some ephemeris model.

```
#include <cstdlib>
#include <math.h>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↵
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/named_item/include/named_item.hh"
#include "utils/orbital_elements/include/orbital_elements.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_orbit.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.34.1 Detailed Description

Define classes for items represented in some ephemeris model.

9.35 dyn_body_init_orbit.hh File Reference

Define the class DynBodyInitOrbit, which initializes a vehicle in in some orbit.

```
#include "environment/ephemerides/ephem_interface/include/class_declarations.hh"
#include "environment/planet/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "dyn_body_init_trans_state.hh"
```

Data Structures

- class [jeod::DynBodyInitOrbit](#)

Initialize a vehicle's translational state given an orbital specification.

Namespaces

- [jeod](#)

Namespace jeod.

9.35.1 Detailed Description

Define the class DynBodyInitOrbit, which initializes a vehicle in in some orbit.

9.36 dyn_body_init_planet_derived.cc File Reference

Define methods for the DynBodyInitPlanetDerived class.

```
#include <cstdlib>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_planet_derived.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.36.1 Detailed Description

Define methods for the DynBodyInitPlanetDerived class.

9.37 dyn_body_init_planet_derived.hh File Reference

Define the class DynBodyInitPlanetDerived, the base class for initializing selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.

```
#include "dynamics/dyn_body/include/class_declarations.hh"
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "dyn_body_init_wrt_planet.hh"
```

Data Structures

- class [jeod::DynBodyInitPlanetDerived](#)

(Initialize selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.

Namespaces

- [jeod](#)

Namespace jeod.

9.37.1 Detailed Description

Define the class DynBodyInitPlanetDerived, the base class for initializing selected aspects of a vehicle's state with respect to some state that is derived from some vehicle's state in conjunction with a planet.

9.38 dyn_body_init_rot_state.cc File Reference

Define methods for DynBodyInitRotState.

```
#include <cstddef>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↵
hh"
#include "utils/ref_frames/include/ref_frame.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_rot_state.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.38.1 Detailed Description

Define methods for DynBodyInitRotState.

9.39 dyn_body_init_rot_state.hh File Reference

Define the class DynBodyInitRotState that initialize aspects of a vehicle's rotational state.

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

Data Structures

- class [jeod::DynBodyInitRotState](#)

Initialize aspects of a vehicle's rotational state.

Namespaces

- [jeod](#)

Namespace jeod.

9.39.1 Detailed Description

Define the class DynBodyInitRotState that initialize aspects of a vehicle's rotational state.

9.40 dyn_body_init_trans_state.cc File Reference

Define methods for DynBodyInitTransState.

```
#include <cstdint>
#include "dynamics/dyn_body/include/dyn_body.hh"
#include "environment/ephemerides/ephem_interface/include/ephem_ref_frame.↵
hh"
#include "utils/ref_frames/include/ref_frame.hh"
#include "utils/math/include/vector3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_trans_state.hh"
```


Namespaces

- [jeod](#)

Namespace jeod.

9.40.1 Detailed Description

Define methods for DynBodyInitTransState.

9.41 dyn_body_init_trans_state.hh File Reference

Define the class DynBodyInitTransState that initialize aspects of a vehicle's translational state.

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

Data Structures

- class [jeod::DynBodyInitTransState](#)

Initialize aspects of a vehicle's translational state.

Namespaces

- [jeod](#)

Namespace jeod.

9.41.1 Detailed Description

Define the class DynBodyInitTransState that initialize aspects of a vehicle's translational state.

9.42 dyn_body_init_wrt_planet.cc File Reference

Define methods for the DynBodyInitWrtPlanet class.

```
#include <cstdlib>
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "environment/planet/include/planet.hh"
#include "utils/message/include/message_handler.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "../include/body_action_messages.hh"
#include "../include/dyn_body_init_wrt_planet.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.42.1 Detailed Description

Define methods for the DynBodyInitWrtPlanet class.

9.43 dyn_body_init_wrt_planet.hh File Reference

Define the class DynBodyInitWrtPlanet, the base class for initializing selected aspects of a vehicle's state with respect to some state that is connected to a planet in some way.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "environment/planet/include/class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/ref_frames/include/ref_frame_items.hh"
#include "dyn_body_init.hh"
```

Data Structures

- class [jeod::DynBodyInitWrtPlanet](#)

Initialize selected aspects of a vehicle's state with respect to some frame based on the planet.

Namespaces

- [jeod](#)

Namespace jeod.

9.43.1 Detailed Description

Define the class DynBodyInitWrtPlanet, the base class for initializing selected aspects of a vehicle's state with respect to some state that is connected to a planet in some way.

9.44 mass_body_init.cc File Reference

Define methods for the mass body initialization class.

```
#include <cstdint>
#include "dynamics/mass/include/mass.hh"
#include "dynamics/dyn_manager/include/dyn_manager.hh"
#include "utils/math/include/vector3.hh"
#include "utils/math/include/matrix3x3.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/body_action_messages.hh"
#include "../include/mass_body_init.hh"
```

Namespaces

- [jeod](#)

Namespace jeod.

9.44.1 Detailed Description

Define methods for the mass body initialization class.

9.45 mass_body_init.hh File Reference

Define the class MassBodyInit, the base class used for initializing the core mass properties of a MassBody object.

```
#include "dynamics/dyn_manager/include/class_declarations.hh"
#include "dynamics/mass/include/class_declarations.hh"
#include "dynamics/mass/include/mass_properties_init.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "class_declarations.hh"
#include "body_action.hh"
```

Data Structures

- class [jeod::MassBodyInit](#)

Base class for initializing a MassBody.

Namespaces

- [jeod](#)

Namespace jeod.

9.45.1 Detailed Description

Define the class MassBodyInit, the base class used for initializing the core mass properties of a MassBody object.

Index

- ~BodyAction
 - jeod::BodyAction, [23](#)
- ~BodyAttach
 - jeod::BodyAttach, [36](#)
- ~BodyAttachAligned
 - jeod::BodyAttachAligned, [41](#)
- ~BodyAttachMatrix
 - jeod::BodyAttachMatrix, [45](#)
- ~BodyDetach
 - jeod::BodyDetach, [48](#)
- ~BodyDetachSpecific
 - jeod::BodyDetachSpecific, [51](#)
- ~BodyReattach
 - jeod::BodyReattach, [55](#)
- ~DynBodyFrameSwitch
 - jeod::DynBodyFrameSwitch, [59](#)
- ~DynBodyInit
 - jeod::DynBodyInit, [65](#)
- ~DynBodyInitLvLhRotState
 - jeod::DynBodyInitLvLhRotState, [76](#)
- ~DynBodyInitLvLhState
 - jeod::DynBodyInitLvLhState, [79](#)
- ~DynBodyInitLvLhTransState
 - jeod::DynBodyInitLvLhTransState, [83](#)
- ~DynBodyInitNedRotState
 - jeod::DynBodyInitNedRotState, [85](#)
- ~DynBodyInitNedState
 - jeod::DynBodyInitNedState, [88](#)
- ~DynBodyInitNedTransState
 - jeod::DynBodyInitNedTransState, [92](#)
- ~DynBodyInitOrbit
 - jeod::DynBodyInitOrbit, [96](#)
- ~DynBodyInitPlanetDerived
 - jeod::DynBodyInitPlanetDerived, [105](#)
- ~DynBodyInitRotState
 - jeod::DynBodyInitRotState, [111](#)
- ~DynBodyInitTransState
 - jeod::DynBodyInitTransState, [115](#)
- ~DynBodyInitWrtPlanet
 - jeod::DynBodyInitWrtPlanet, [120](#)
- ~MassBodyInit
 - jeod::MassBodyInit, [125](#)
- action_identifier
 - jeod::BodyAction, [28](#)
- action_name
 - jeod::BodyAction, [28](#)
- active
 - jeod::BodyAction, [29](#)
- alt_apoapsis
 - jeod::DynBodyInitOrbit, [98](#)
- alt_periapsis
 - jeod::DynBodyInitOrbit, [98](#)
- altlatlong_type
 - jeod::DynBodyInitNedState, [90](#)
- ang_velocity
 - jeod::DynBodyInit, [71](#)
- apply
 - jeod::BodyAction, [23](#)
 - jeod::BodyAttach, [37](#)
 - jeod::BodyAttachAligned, [41](#)
 - jeod::BodyAttachMatrix, [45](#)
 - jeod::BodyDetach, [48](#)
 - jeod::BodyDetachSpecific, [51](#)
 - jeod::BodyReattach, [55](#)
 - jeod::DynBodyFrameSwitch, [59](#)
 - jeod::DynBodyInit, [65](#)
 - jeod::DynBodyInitLvLhState, [79](#)
 - jeod::DynBodyInitNedState, [89](#)
 - jeod::DynBodyInitOrbit, [97](#)
 - jeod::DynBodyInitPlanetDerived, [105](#)
 - jeod::DynBodyInitRotState, [111](#)
 - jeod::DynBodyInitTransState, [116](#)
 - jeod::DynBodyInitWrtPlanet, [120](#)
 - jeod::MassBodyInit, [125](#)
- apply_user_inputs
 - jeod::DynBodyInit, [66](#)
- arg_latitude
 - jeod::DynBodyInitOrbit, [99](#)
- arg_periapsis
 - jeod::DynBodyInitOrbit, [99](#)
- ascending_node
 - jeod::DynBodyInitOrbit, [99](#)
- body_action.cc, [129](#)
- body_action.hh, [129](#)
- body_action_messages.cc, [130](#)
- body_action_messages.hh, [130](#)
- body_attach.cc, [131](#)
- body_attach.hh, [131](#)
- body_attach_aligned.cc, [132](#)
- body_attach_aligned.hh, [132](#)
- body_attach_matrix.cc, [133](#)
- body_attach_matrix.hh, [133](#)
- body_detach.cc, [134](#)
- body_detach.hh, [134](#)
- body_detach_specific.cc, [135](#)
- body_detach_specific.hh, [135](#)
- body_frame_id
 - jeod::DynBodyInit, [71](#)

- body_is_required
 - jeod::DynBodyInitPlanetDerived, 107
- body_reattach.cc, 136
- body_reattach.hh, 136
- body_ref_frame
 - jeod::DynBodyInit, 72
- BodyAction, 15
 - jeod::BodyAction, 23
 - PATH, 17
- BodyActionMessages
 - jeod::BodyActionMessages, 31
- BodyAttach
 - jeod::BodyAttach, 36
- BodyAttachAligned
 - jeod::BodyAttachAligned, 41
- BodyAttachMatrix
 - jeod::BodyAttachMatrix, 45
- BodyDetach
 - jeod::BodyDetach, 47
- BodyDetachSpecific
 - jeod::BodyDetachSpecific, 50
- BodyReattach
 - jeod::BodyReattach, 55
- class_declarations.hh, 137
- compute_rotational_state
 - jeod::DynBodyInit, 66
- compute_translational_state
 - jeod::DynBodyInit, 66
- dyn_body_frame_switch.cc, 137
- dyn_body_frame_switch.hh, 138
- dyn_body_init.cc, 138
- dyn_body_init.hh, 139
- dyn_body_init_lvhl_rot_state.cc, 139
- dyn_body_init_lvhl_rot_state.hh, 140
- dyn_body_init_lvhl_state.cc, 140
- dyn_body_init_lvhl_state.hh, 141
- dyn_body_init_lvhl_trans_state.cc, 141
- dyn_body_init_lvhl_trans_state.hh, 142
- dyn_body_init_ned_rot_state.cc, 142
- dyn_body_init_ned_rot_state.hh, 143
- dyn_body_init_ned_state.cc, 143
- dyn_body_init_ned_state.hh, 144
- dyn_body_init_ned_trans_state.cc, 144
- dyn_body_init_ned_trans_state.hh, 145
- dyn_body_init_orbit.cc, 145
- dyn_body_init_orbit.hh, 146
- dyn_body_init_planet_derived.cc, 146
- dyn_body_init_planet_derived.hh, 147
- dyn_body_init_rot_state.cc, 147
- dyn_body_init_rot_state.hh, 148
- dyn_body_init_trans_state.cc, 148
- dyn_body_init_trans_state.hh, 149
- dyn_body_init_wrt_planet.cc, 149
- dyn_body_init_wrt_planet.hh, 150
- dyn_detach_from
 - jeod::BodyDetachSpecific, 53
- dyn_parent
 - jeod::BodyAttach, 39
- dyn_subject
 - jeod::BodyAction, 29
- DynBodyFrameSwitch
 - jeod::DynBodyFrameSwitch, 59
- DynBodyInit
 - jeod::DynBodyInit, 64, 65
- DynBodyInitLvhlRotState
 - jeod::DynBodyInitLvhlRotState, 76
- DynBodyInitLvhlState
 - jeod::DynBodyInitLvhlState, 78, 79
- DynBodyInitLvhlTransState
 - jeod::DynBodyInitLvhlTransState, 83
- DynBodyInitNedRotState
 - jeod::DynBodyInitNedRotState, 85
- DynBodyInitNedState
 - jeod::DynBodyInitNedState, 88
- DynBodyInitNedTransState
 - jeod::DynBodyInitNedTransState, 92
- DynBodyInitOrbit
 - jeod::DynBodyInitOrbit, 96
- DynBodyInitPlanetDerived
 - jeod::DynBodyInitPlanetDerived, 105
- DynBodyInitRotState
 - jeod::DynBodyInitRotState, 110
- DynBodyInitTransState
 - jeod::DynBodyInitTransState, 115, 116
- DynBodyInitWrtPlanet
 - jeod::DynBodyInitWrtPlanet, 120
- Dynamics, 14
- eccentricity
 - jeod::DynBodyInitOrbit, 99
- fatal_error
 - jeod::BodyActionMessages, 32
- find_body_frame
 - jeod::DynBodyInit, 66
- find_dyn_body
 - jeod::DynBodyInit, 67
- find_planet
 - jeod::DynBodyInit, 68
- find_ref_frame
 - jeod::DynBodyInit, 68
- get_identifier
 - jeod::BodyAction, 24
- get_subject_dyn_body
 - jeod::BodyAction, 24
- illegal_value
 - jeod::BodyActionMessages, 32
- inclination
 - jeod::DynBodyInitOrbit, 100
- init_attrjeod__BodyAction
 - jeod::BodyAction, 28
- init_attrjeod__BodyActionMessages
 - jeod::BodyActionMessages, 32
- init_attrjeod__BodyAttach

- jeod::BodyAttach, 38
- init_attrjeod__BodyAttachAligned
 - jeod::BodyAttachAligned, 42
- init_attrjeod__BodyAttachMatrix
 - jeod::BodyAttachMatrix, 46
- init_attrjeod__BodyDetach
 - jeod::BodyDetach, 49
- init_attrjeod__BodyDetachSpecific
 - jeod::BodyDetachSpecific, 53
- init_attrjeod__BodyReattach
 - jeod::BodyReattach, 56
- init_attrjeod__DynBodyFrameSwitch
 - jeod::DynBodyFrameSwitch, 60
- init_attrjeod__DynBodyInit
 - jeod::DynBodyInit, 70
- init_attrjeod__DynBodyInitLvLhRotState
 - jeod::DynBodyInitLvLhRotState, 77
- init_attrjeod__DynBodyInitLvLhState
 - jeod::DynBodyInitLvLhState, 81
- init_attrjeod__DynBodyInitLvLhTransState
 - jeod::DynBodyInitLvLhTransState, 84
- init_attrjeod__DynBodyInitNedRotState
 - jeod::DynBodyInitNedRotState, 86
- init_attrjeod__DynBodyInitNedState
 - jeod::DynBodyInitNedState, 90
- init_attrjeod__DynBodyInitNedTransState
 - jeod::DynBodyInitNedTransState, 93
- init_attrjeod__DynBodyInitOrbit
 - jeod::DynBodyInitOrbit, 98
- init_attrjeod__DynBodyInitPlanetDerived
 - jeod::DynBodyInitPlanetDerived, 107
- init_attrjeod__DynBodyInitRotState
 - jeod::DynBodyInitRotState, 113
- init_attrjeod__DynBodyInitTransState
 - jeod::DynBodyInitTransState, 118
- init_attrjeod__DynBodyInitWrtPlanet
 - jeod::DynBodyInitWrtPlanet, 122
- init_attrjeod__MassBodyInit
 - jeod::MassBodyInit, 126
- initialize
 - jeod::BodyAction, 24
 - jeod::BodyAttach, 37
 - jeod::BodyAttachAligned, 42
 - jeod::BodyDetachSpecific, 51
 - jeod::DynBodyFrameSwitch, 59
 - jeod::DynBodyInit, 69
 - jeod::DynBodyInitLvLhRotState, 76
 - jeod::DynBodyInitLvLhState, 80
 - jeod::DynBodyInitLvLhTransState, 83
 - jeod::DynBodyInitNedRotState, 86
 - jeod::DynBodyInitNedState, 89
 - jeod::DynBodyInitNedTransState, 93
 - jeod::DynBodyInitOrbit, 97
 - jeod::DynBodyInitPlanetDerived, 106
 - jeod::DynBodyInitRotState, 111
 - jeod::DynBodyInitTransState, 116
 - jeod::DynBodyInitWrtPlanet, 121
- initializes_what
 - jeod::DynBodyInit, 69
 - jeod::DynBodyInitRotState, 112
 - jeod::DynBodyInitTransState, 117
 - jeod::DynBodyInitWrtPlanet, 121
- InputProcessor
 - jeod::BodyAction, 28
 - jeod::BodyActionMessages, 32
 - jeod::BodyAttach, 38
 - jeod::BodyAttachAligned, 42
 - jeod::BodyAttachMatrix, 46
 - jeod::BodyDetach, 49
 - jeod::BodyDetachSpecific, 53
 - jeod::BodyReattach, 56
 - jeod::DynBodyFrameSwitch, 60
 - jeod::DynBodyInit, 71
 - jeod::DynBodyInitLvLhRotState, 77
 - jeod::DynBodyInitLvLhState, 81
 - jeod::DynBodyInitLvLhTransState, 84
 - jeod::DynBodyInitNedRotState, 86
 - jeod::DynBodyInitNedState, 90
 - jeod::DynBodyInitNedTransState, 93
 - jeod::DynBodyInitOrbit, 98
 - jeod::DynBodyInitPlanetDerived, 107
 - jeod::DynBodyInitRotState, 113
 - jeod::DynBodyInitTransState, 118
 - jeod::DynBodyInitWrtPlanet, 122
 - jeod::MassBodyInit, 126
- integ_frame
 - jeod::DynBodyFrameSwitch, 61
- integ_frame_name
 - jeod::DynBodyFrameSwitch, 61
- invalid_name
 - jeod::BodyActionMessages, 33
- invalid_object
 - jeod::BodyActionMessages, 33
- is_ready
 - jeod::BodyAction, 25
 - jeod::BodyDetach, 48
 - jeod::BodyDetachSpecific, 52
 - jeod::DynBodyFrameSwitch, 60
 - jeod::DynBodyInit, 69
 - jeod::DynBodyInitPlanetDerived, 106
 - jeod::DynBodyInitRotState, 112
 - jeod::DynBodyInitTransState, 117
 - jeod::DynBodyInitWrtPlanet, 121
- is_same_subject_body
 - jeod::BodyAction, 25
- is_subject_dyn_body
 - jeod::BodyAction, 25
- jeod, 19
 - jeod::BodyAction, 21
 - ~BodyAction, 23
 - action_identifier, 28
 - action_name, 28
 - active, 29
 - apply, 23
 - BodyAction, 23
 - dyn_subject, 29

- get_identifier, 24
- get_subject_dyn_body, 24
- init_attrjeod__BodyAction, 28
- initialize, 24
- InputProcessor, 28
- is_ready, 25
- is_same_subject_body, 25
- is_subject_dyn_body, 25
- mass_subject, 29
- operator=, 26
- set_subject_body, 26
- shutdown, 26
- terminate_on_error, 30
- validate_body_inputs, 27
- validate_name, 27
- jeod::BodyActionMessages, 30
 - BodyActionMessages, 31
 - fatal_error, 32
 - illegal_value, 32
 - init_attrjeod__BodyActionMessages, 32
 - InputProcessor, 32
 - invalid_name, 33
 - invalid_object, 33
 - not_performed, 33
 - null_pointer, 34
 - operator=, 32
 - trace, 34
- jeod::BodyAttach, 35
 - ~BodyAttach, 36
 - apply, 37
 - BodyAttach, 36
 - dyn_parent, 39
 - init_attrjeod__BodyAttach, 38
 - initialize, 37
 - InputProcessor, 38
 - mass_parent, 39
 - operator=, 37
 - set_parent_body, 38
 - succeeded, 39
- jeod::BodyAttachAligned, 40
 - ~BodyAttachAligned, 41
 - apply, 41
 - BodyAttachAligned, 41
 - init_attrjeod__BodyAttachAligned, 42
 - initialize, 42
 - InputProcessor, 42
 - operator=, 42
 - parent_point_name, 43
 - subject_point_name, 43
- jeod::BodyAttachMatrix, 44
 - ~BodyAttachMatrix, 45
 - apply, 45
 - BodyAttachMatrix, 45
 - init_attrjeod__BodyAttachMatrix, 46
 - InputProcessor, 46
 - offset_pstr_cstr_pstr, 46
 - pstr_cstr, 46
- jeod::BodyDetach, 47
 - ~BodyDetach, 48
 - apply, 48
 - BodyDetach, 47
 - init_attrjeod__BodyDetach, 49
 - InputProcessor, 49
 - is_ready, 48
- jeod::BodyDetachSpecific, 49
 - ~BodyDetachSpecific, 51
 - apply, 51
 - BodyDetachSpecific, 50
 - dyn_detach_from, 53
 - init_attrjeod__BodyDetachSpecific, 53
 - initialize, 51
 - InputProcessor, 53
 - is_ready, 52
 - mass_detach_from, 53
 - set_detach_from_body, 52
- jeod::BodyReattach, 54
 - ~BodyReattach, 55
 - apply, 55
 - BodyReattach, 55
 - init_attrjeod__BodyReattach, 56
 - InputProcessor, 56
 - offset_pstr_cstr_pstr, 56
 - pstr_cstr, 56
- jeod::DynBodyFrameSwitch, 57
 - ~DynBodyFrameSwitch, 59
 - apply, 59
 - DynBodyFrameSwitch, 59
 - init_attrjeod__DynBodyFrameSwitch, 60
 - initialize, 59
 - InputProcessor, 60
 - integ_frame, 61
 - integ_frame_name, 61
 - is_ready, 60
 - sort_grav_controls, 61
 - switch_distance, 61
 - switch_sense, 62
 - SwitchSense, 58
- jeod::DynBodyInit, 62
 - ~DynBodyInit, 65
 - ang_velocity, 71
 - apply, 65
 - apply_user_inputs, 66
 - body_frame_id, 71
 - body_ref_frame, 72
 - compute_rotational_state, 66
 - compute_translational_state, 66
 - DynBodyInit, 64, 65
 - find_body_frame, 66
 - find_dyn_body, 67
 - find_planet, 68
 - find_ref_frame, 68
 - init_attrjeod__DynBodyInit, 70
 - initialize, 69
 - initializes_what, 69
 - InputProcessor, 71
 - is_ready, 69

- operator=, 70
- orientation, 72
- position, 72
- rate_in_parent, 72
- reference_ref_frame, 73
- reference_ref_frame_name, 73
- report_failure, 70
- reverse_sense, 73
- state, 73
- subscribed_frame, 74
- velocity, 74
- jeod::DynBodyInitLvHRotState, 75
 - ~DynBodyInitLvHRotState, 76
 - DynBodyInitLvHRotState, 76
 - init_attrjeod__DynBodyInitLvHRotState, 77
 - initialize, 76
 - InputProcessor, 77
 - operator=, 77
- jeod::DynBodyInitLvHState, 77
 - ~DynBodyInitLvHState, 79
 - apply, 79
 - DynBodyInitLvHState, 78, 79
 - init_attrjeod__DynBodyInitLvHState, 81
 - initialize, 80
 - InputProcessor, 81
 - lvh_object_ptr, 81
 - lvh_type, 81
 - operator=, 80
 - set_lvh_frame_object, 80
- jeod::DynBodyInitLvHTransState, 82
 - ~DynBodyInitLvHTransState, 83
 - DynBodyInitLvHTransState, 83
 - init_attrjeod__DynBodyInitLvHTransState, 84
 - initialize, 83
 - InputProcessor, 84
 - operator=, 84
- jeod::DynBodyInitNedRotState, 84
 - ~DynBodyInitNedRotState, 85
 - DynBodyInitNedRotState, 85
 - init_attrjeod__DynBodyInitNedRotState, 86
 - initialize, 86
 - InputProcessor, 86
 - operator=, 86
- jeod::DynBodyInitNedState, 87
 - ~DynBodyInitNedState, 88
 - altlatlong_type, 90
 - apply, 89
 - DynBodyInitNedState, 88
 - init_attrjeod__DynBodyInitNedState, 90
 - initialize, 89
 - InputProcessor, 90
 - operator=, 90
 - ref_point, 90
- jeod::DynBodyInitNedTransState, 91
 - ~DynBodyInitNedTransState, 92
 - DynBodyInitNedTransState, 92
 - init_attrjeod__DynBodyInitNedTransState, 93
 - initialize, 93
- InputProcessor, 93
 - operator=, 93
- jeod::DynBodyInitOrbit, 94
 - ~DynBodyInitOrbit, 96
 - alt_apoapsis, 98
 - alt_periapsis, 98
 - apply, 97
 - arg_latitude, 99
 - arg_periapsis, 99
 - ascending_node, 99
 - DynBodyInitOrbit, 96
 - eccentricity, 99
 - inclination, 100
 - init_attrjeod__DynBodyInitOrbit, 98
 - initialize, 97
 - InputProcessor, 98
 - mean_anomaly, 100
 - orb_radius, 100
 - orbit_frame, 100
 - orbit_frame_name, 101
 - OrbitalSet, 96
 - planet, 101
 - planet_name, 101
 - radial_vel, 101
 - semi_latus_rectum, 102
 - semi_major_axis, 102
 - set, 102
 - time_periapsis, 102
 - true_anomaly, 103
- jeod::DynBodyInitPlanetDerived, 103
 - ~DynBodyInitPlanetDerived, 105
 - apply, 105
 - body_is_required, 107
 - DynBodyInitPlanetDerived, 105
 - init_attrjeod__DynBodyInitPlanetDerived, 107
 - initialize, 106
 - InputProcessor, 107
 - is_ready, 106
 - operator=, 106
 - ref_body, 107
 - ref_body_name, 108
 - required_items, 108
- jeod::DynBodyInitRotState, 109
 - ~DynBodyInitRotState, 111
 - apply, 111
 - DynBodyInitRotState, 110
 - init_attrjeod__DynBodyInitRotState, 113
 - initialize, 111
 - initializes_what, 112
 - InputProcessor, 113
 - is_ready, 112
 - operator=, 112
 - state_items, 113
 - StateItems, 110
- jeod::DynBodyInitTransState, 114
 - ~DynBodyInitTransState, 115
 - apply, 116
 - DynBodyInitTransState, 115, 116

- init_attrjeod__DynBodyInitTransState, 118
- initialize, 116
- initializes_what, 117
- InputProcessor, 118
- is_ready, 117
- operator=, 117
- state_items, 118
- Stateltems, 115
- jeod::DynBodyInitWrtPlanet, 119
 - ~DynBodyInitWrtPlanet, 120
 - apply, 120
 - DynBodyInitWrtPlanet, 120
 - init_attrjeod__DynBodyInitWrtPlanet, 122
 - initialize, 121
 - initializes_what, 121
 - InputProcessor, 122
 - is_ready, 121
 - operator=, 122
 - planet, 122
 - planet_name, 123
 - set_items, 123
- jeod::MassBodyInit, 124
 - ~MassBodyInit, 125
 - apply, 125
 - init_attrjeod__MassBodyInit, 126
 - InputProcessor, 126
 - MassBodyInit, 125
 - num_points, 126
 - operator=, 126
 - points, 126
 - properties, 127
- lvh_object_ptr
 - jeod::DynBodyInitLvHState, 81
- lvh_type
 - jeod::DynBodyInitLvHState, 81
- mass_body_init.cc, 150
- mass_body_init.hh, 151
- mass_detach_from
 - jeod::BodyDetachSpecific, 53
- mass_parent
 - jeod::BodyAttach, 39
- mass_subject
 - jeod::BodyAction, 29
- MassBodyInit
 - jeod::MassBodyInit, 125
- mean_anomaly
 - jeod::DynBodyInitOrbit, 100
- Models, 13
- not_performed
 - jeod::BodyActionMessages, 33
- null_pointer
 - jeod::BodyActionMessages, 34
- num_points
 - jeod::MassBodyInit, 126
- offset_pstr_cstr_pstr
 - jeod::BodyAttachMatrix, 46
 - jeod::BodyReattach, 56
- operator=
 - jeod::BodyAction, 26
 - jeod::BodyActionMessages, 32
 - jeod::BodyAttach, 37
 - jeod::BodyAttachAligned, 42
 - jeod::DynBodyInit, 70
 - jeod::DynBodyInitLvHRotState, 77
 - jeod::DynBodyInitLvHState, 80
 - jeod::DynBodyInitLvHTransState, 84
 - jeod::DynBodyInitNedRotState, 86
 - jeod::DynBodyInitNedState, 90
 - jeod::DynBodyInitNedTransState, 93
 - jeod::DynBodyInitPlanetDerived, 106
 - jeod::DynBodyInitRotState, 112
 - jeod::DynBodyInitTransState, 117
 - jeod::DynBodyInitWrtPlanet, 122
 - jeod::MassBodyInit, 126
- orb_radius
 - jeod::DynBodyInitOrbit, 100
- orbit_frame
 - jeod::DynBodyInitOrbit, 100
- orbit_frame_name
 - jeod::DynBodyInitOrbit, 101
- OrbitalSet
 - jeod::DynBodyInitOrbit, 96
- orientation
 - jeod::DynBodyInit, 72
- PATH
 - BodyAction, 17
- parent_point_name
 - jeod::BodyAttachAligned, 43
- planet
 - jeod::DynBodyInitOrbit, 101
 - jeod::DynBodyInitWrtPlanet, 122
- planet_name
 - jeod::DynBodyInitOrbit, 101
 - jeod::DynBodyInitWrtPlanet, 123
- points
 - jeod::MassBodyInit, 126
- position
 - jeod::DynBodyInit, 72
- properties
 - jeod::MassBodyInit, 127
- pstr_cstr
 - jeod::BodyAttachMatrix, 46
 - jeod::BodyReattach, 56
- radial_vel
 - jeod::DynBodyInitOrbit, 101
- rate_in_parent
 - jeod::DynBodyInit, 72
- ref_body
 - jeod::DynBodyInitPlanetDerived, 107
- ref_body_name
 - jeod::DynBodyInitPlanetDerived, 108
- ref_point

- jeod::DynBodyInitNedState, 90
- reference_ref_frame
 - jeod::DynBodyInit, 73
- reference_ref_frame_name
 - jeod::DynBodyInit, 73
- report_failure
 - jeod::DynBodyInit, 70
- required_items
 - jeod::DynBodyInitPlanetDerived, 108
- reverse_sense
 - jeod::DynBodyInit, 73
- semi_latus_rectum
 - jeod::DynBodyInitOrbit, 102
- semi_major_axis
 - jeod::DynBodyInitOrbit, 102
- set
 - jeod::DynBodyInitOrbit, 102
- set_detach_from_body
 - jeod::BodyDetachSpecific, 52
- set_items
 - jeod::DynBodyInitWrtPlanet, 123
- set_lvlh_frame_object
 - jeod::DynBodyInitLvlhState, 80
- set_parent_body
 - jeod::BodyAttach, 38
- set_subject_body
 - jeod::BodyAction, 26
- shutdown
 - jeod::BodyAction, 26
- sort_grav_controls
 - jeod::DynBodyFrameSwitch, 61
- state
 - jeod::DynBodyInit, 73
- state_items
 - jeod::DynBodyInitRotState, 113
 - jeod::DynBodyInitTransState, 118
- StateItems
 - jeod::DynBodyInitRotState, 110
 - jeod::DynBodyInitTransState, 115
- subject_point_name
 - jeod::BodyAttachAligned, 43
- subscribed_frame
 - jeod::DynBodyInit, 74
- succeeded
 - jeod::BodyAttach, 39
- switch_distance
 - jeod::DynBodyFrameSwitch, 61
- switch_sense
 - jeod::DynBodyFrameSwitch, 62
- SwitchSense
 - jeod::DynBodyFrameSwitch, 58
- terminate_on_error
 - jeod::BodyAction, 30
- time_periapsis
 - jeod::DynBodyInitOrbit, 102
- trace
 - jeod::BodyActionMessages, 34
- true_anomaly
 - jeod::DynBodyInitOrbit, 103
- validate_body_inputs
 - jeod::BodyAction, 27
- validate_name
 - jeod::BodyAction, 27
- velocity
 - jeod::DynBodyInit, 74