

BodyActionModel

5.0

Generated by Doxygen 1.8.5

Wed Jun 1 2022 12:10:13

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	11
6.1	Models	11
6.1.1	Detailed Description	11
6.2	Dynamics	12
6.2.1	Detailed Description	12
6.3	BodyAction	13
6.3.1	Detailed Description	14
6.3.2	Macro Definition Documentation	14
6.3.2.1	PATH	14
7	Namespace Documentation	15
7.1	jeod Namespace Reference	15
7.1.1	Detailed Description	16
8	Data Structure Documentation	17
8.1	jeod::BodyAction Class Reference	17
8.1.1	Detailed Description	18
8.1.2	Constructor & Destructor Documentation	19
8.1.2.1	BodyAction	19

8.1.2.2	BodyAction	19
8.1.2.3	~BodyAction	19
8.1.3	Member Function Documentation	19
8.1.3.1	apply	19
8.1.3.2	get_identifier	19
8.1.3.3	initialize	19
8.1.3.4	is_ready	20
8.1.3.5	operator=	20
8.1.3.6	shutdown	20
8.1.3.7	validate_name	20
8.1.4	Friends And Related Function Documentation	21
8.1.4.1	init_attrjeod__BodyAction	21
8.1.4.2	InputProcessor	21
8.1.5	Field Documentation	21
8.1.5.1	action_identifier	21
8.1.5.2	action_name	21
8.1.5.3	active	21
8.1.5.4	dyn_subject	22
8.1.5.5	subject	22
8.1.5.6	terminate_on_error	22
8.2	jeod::BodyActionMessages Class Reference	22
8.2.1	Detailed Description	23
8.2.2	Constructor & Destructor Documentation	23
8.2.2.1	BodyActionMessages	23
8.2.2.2	BodyActionMessages	23
8.2.3	Member Function Documentation	23
8.2.3.1	operator=	23
8.2.4	Friends And Related Function Documentation	23
8.2.4.1	init_attrjeod__BodyActionMessages	24
8.2.4.2	InputProcessor	24
8.2.5	Field Documentation	24
8.2.5.1	fatal_error	24
8.2.5.2	illegal_value	24
8.2.5.3	invalid_name	24
8.2.5.4	invalid_object	25
8.2.5.5	not_performed	25
8.2.5.6	null_pointer	25
8.2.5.7	trace	25
8.3	jeod::BodyAttach Class Reference	26
8.3.1	Detailed Description	27

8.3.2	Constructor & Destructor Documentation	27
8.3.2.1	BodyAttach	27
8.3.2.2	BodyAttach	27
8.3.2.3	~BodyAttach	27
8.3.3	Member Function Documentation	27
8.3.3.1	apply	27
8.3.3.2	initialize	27
8.3.3.3	operator=	28
8.3.4	Friends And Related Function Documentation	28
8.3.4.1	init_attrjeod__BodyAttach	28
8.3.4.2	InputProcessor	28
8.3.5	Field Documentation	28
8.3.5.1	dyn_parent	28
8.3.5.2	parent	28
8.3.5.3	succeeded	28
8.4	jeod::BodyAttachAligned Class Reference	29
8.4.1	Detailed Description	30
8.4.2	Constructor & Destructor Documentation	30
8.4.2.1	BodyAttachAligned	30
8.4.2.2	~BodyAttachAligned	30
8.4.2.3	BodyAttachAligned	30
8.4.3	Member Function Documentation	30
8.4.3.1	apply	30
8.4.3.2	initialize	30
8.4.3.3	operator=	31
8.4.4	Friends And Related Function Documentation	31
8.4.4.1	init_attrjeod__BodyAttachAligned	31
8.4.4.2	InputProcessor	31
8.4.5	Field Documentation	31
8.4.5.1	parent_point_name	31
8.4.5.2	subject_point_name	31
8.5	jeod::BodyAttachMatrix Class Reference	31
8.5.1	Detailed Description	32
8.5.2	Constructor & Destructor Documentation	32
8.5.2.1	BodyAttachMatrix	32
8.5.2.2	~BodyAttachMatrix	33
8.5.3	Member Function Documentation	33
8.5.3.1	apply	33
8.5.4	Friends And Related Function Documentation	33
8.5.4.1	init_attrjeod__BodyAttachMatrix	33

8.5.4.2	InputProcessor	33
8.5.5	Field Documentation	33
8.5.5.1	offset_pstr_cstr_pstr	33
8.5.5.2	pstr_cstr	33
8.6	jeod::BodyDetach Class Reference	34
8.6.1	Detailed Description	34
8.6.2	Constructor & Destructor Documentation	34
8.6.2.1	BodyDetach	34
8.6.2.2	~BodyDetach	35
8.6.3	Member Function Documentation	35
8.6.3.1	apply	35
8.6.3.2	initialize	35
8.6.3.3	is_ready	35
8.6.4	Friends And Related Function Documentation	35
8.6.4.1	init_attrjeod__BodyDetach	35
8.6.4.2	InputProcessor	35
8.7	jeod::BodyDetachSpecific Class Reference	36
8.7.1	Detailed Description	36
8.7.2	Constructor & Destructor Documentation	37
8.7.2.1	BodyDetachSpecific	37
8.7.2.2	~BodyDetachSpecific	37
8.7.3	Member Function Documentation	37
8.7.3.1	apply	37
8.7.3.2	initialize	37
8.7.3.3	is_ready	37
8.7.4	Friends And Related Function Documentation	38
8.7.4.1	init_attrjeod__BodyDetachSpecific	38
8.7.4.2	InputProcessor	38
8.7.5	Field Documentation	38
8.7.5.1	detach_from	38
8.7.5.2	dyn_detach_from	38
8.8	jeod::BodyReattach Class Reference	38
8.8.1	Detailed Description	39
8.8.2	Constructor & Destructor Documentation	39
8.8.2.1	BodyReattach	39
8.8.2.2	~BodyReattach	40
8.8.3	Member Function Documentation	40
8.8.3.1	apply	40
8.8.4	Friends And Related Function Documentation	40
8.8.4.1	init_attrjeod__BodyReattach	40

8.8.4.2	InputProcessor	40
8.8.5	Field Documentation	40
8.8.5.1	offset_pstr_cstr_pstr	40
8.8.5.2	pstr_cstr	40
8.9	jeod::DynBodyFrameSwitch Class Reference	41
8.9.1	Detailed Description	42
8.9.2	Member Enumeration Documentation	42
8.9.2.1	SwitchSense	42
8.9.3	Constructor & Destructor Documentation	42
8.9.3.1	DynBodyFrameSwitch	42
8.9.3.2	~DynBodyFrameSwitch	42
8.9.4	Member Function Documentation	42
8.9.4.1	apply	42
8.9.4.2	initialize	43
8.9.4.3	is_ready	43
8.9.5	Friends And Related Function Documentation	43
8.9.5.1	init_attrjeod__DynBodyFrameSwitch	43
8.9.5.2	InputProcessor	43
8.9.6	Field Documentation	43
8.9.6.1	integ_frame	43
8.9.6.2	integ_frame_name	43
8.9.6.3	sort_grav_controls	44
8.9.6.4	switch_distance	44
8.9.6.5	switch_sense	44
8.10	jeod::DynBodyInit Class Reference	44
8.10.1	Detailed Description	46
8.10.2	Constructor & Destructor Documentation	46
8.10.2.1	DynBodyInit	46
8.10.2.2	DynBodyInit	46
8.10.2.3	~DynBodyInit	46
8.10.3	Member Function Documentation	46
8.10.3.1	apply	46
8.10.3.2	apply_user_inputs	47
8.10.3.3	compute_rotational_state	47
8.10.3.4	compute_translational_state	47
8.10.3.5	find_body_frame	47
8.10.3.6	find_dyn_body	48
8.10.3.7	find_planet	48
8.10.3.8	find_ref_frame	48
8.10.3.9	initialize	49

8.10.3.10	initializes_what	49
8.10.3.11	is_ready	49
8.10.3.12	operator=	50
8.10.3.13	report_failure	50
8.10.4	Friends And Related Function Documentation	50
8.10.4.1	init_attrjeod__DynBodyInit	50
8.10.4.2	InputProcessor	50
8.10.5	Field Documentation	50
8.10.5.1	ang_velocity	50
8.10.5.2	body_frame_id	51
8.10.5.3	body_ref_frame	51
8.10.5.4	orientation	51
8.10.5.5	position	51
8.10.5.6	rate_in_parent	51
8.10.5.7	reference_ref_frame	52
8.10.5.8	reference_ref_frame_name	52
8.10.5.9	reverse_sense	52
8.10.5.10	state	52
8.10.5.11	subscribed_frame	52
8.10.5.12	velocity	52
8.11	jeod::DynBodyInitLvHRotState Class Reference	53
8.11.1	Detailed Description	54
8.11.2	Constructor & Destructor Documentation	54
8.11.2.1	DynBodyInitLvHRotState	54
8.11.2.2	DynBodyInitLvHRotState	54
8.11.2.3	~DynBodyInitLvHRotState	54
8.11.3	Member Function Documentation	54
8.11.3.1	initialize	54
8.11.3.2	operator=	54
8.11.4	Friends And Related Function Documentation	54
8.11.4.1	init_attrjeod__DynBodyInitLvHRotState	54
8.11.4.2	InputProcessor	54
8.12	jeod::DynBodyInitLvHState Class Reference	55
8.12.1	Detailed Description	56
8.12.2	Constructor & Destructor Documentation	56
8.12.2.1	DynBodyInitLvHState	56
8.12.2.2	~DynBodyInitLvHState	56
8.12.2.3	DynBodyInitLvHState	56
8.12.3	Member Function Documentation	56
8.12.3.1	apply	56

8.12.3.2	initialize	57
8.12.3.3	operator=	58
8.12.3.4	set_lvlh_frame_object	58
8.12.4	Friends And Related Function Documentation	58
8.12.4.1	init_attrjeod__DynBodyInitLvLhState	58
8.12.4.2	InputProcessor	58
8.12.5	Field Documentation	58
8.12.5.1	lvlh_object_ptr	58
8.12.5.2	lvlh_type	58
8.13	jeod::DynBodyInitLvLhTransState Class Reference	59
8.13.1	Detailed Description	59
8.13.2	Constructor & Destructor Documentation	60
8.13.2.1	DynBodyInitLvLhTransState	60
8.13.2.2	DynBodyInitLvLhTransState	60
8.13.2.3	~DynBodyInitLvLhTransState	60
8.13.3	Member Function Documentation	60
8.13.3.1	initialize	60
8.13.3.2	operator=	60
8.13.4	Friends And Related Function Documentation	60
8.13.4.1	init_attrjeod__DynBodyInitLvLhTransState	60
8.13.4.2	InputProcessor	60
8.14	jeod::DynBodyInitNedRotState Class Reference	60
8.14.1	Detailed Description	61
8.14.2	Constructor & Destructor Documentation	61
8.14.2.1	DynBodyInitNedRotState	61
8.14.2.2	DynBodyInitNedRotState	61
8.14.2.3	~DynBodyInitNedRotState	62
8.14.3	Member Function Documentation	62
8.14.3.1	initialize	62
8.14.3.2	operator=	62
8.14.4	Friends And Related Function Documentation	62
8.14.4.1	init_attrjeod__DynBodyInitNedRotState	62
8.14.4.2	InputProcessor	62
8.15	jeod::DynBodyInitNedState Class Reference	62
8.15.1	Detailed Description	63
8.15.2	Constructor & Destructor Documentation	64
8.15.2.1	DynBodyInitNedState	64
8.15.2.2	DynBodyInitNedState	64
8.15.2.3	~DynBodyInitNedState	64
8.15.3	Member Function Documentation	64

8.15.3.1	apply	64
8.15.3.2	initialize	64
8.15.3.3	operator=	64
8.15.4	Friends And Related Function Documentation	64
8.15.4.1	init_attrjeod__DynBodyInitNedState	65
8.15.4.2	InputProcessor	65
8.15.5	Field Documentation	65
8.15.5.1	altlatlong_type	65
8.15.5.2	ref_point	65
8.16	jeod::DynBodyInitNedTransState Class Reference	65
8.16.1	Detailed Description	66
8.16.2	Constructor & Destructor Documentation	66
8.16.2.1	DynBodyInitNedTransState	66
8.16.2.2	DynBodyInitNedTransState	66
8.16.2.3	~DynBodyInitNedTransState	66
8.16.3	Member Function Documentation	66
8.16.3.1	initialize	66
8.16.3.2	operator=	67
8.16.4	Friends And Related Function Documentation	67
8.16.4.1	init_attrjeod__DynBodyInitNedTransState	67
8.16.4.2	InputProcessor	67
8.17	jeod::DynBodyInitOrbit Class Reference	67
8.17.1	Detailed Description	69
8.17.2	Member Enumeration Documentation	69
8.17.2.1	OrbitalSet	69
8.17.3	Constructor & Destructor Documentation	69
8.17.3.1	DynBodyInitOrbit	69
8.17.3.2	~DynBodyInitOrbit	70
8.17.4	Member Function Documentation	70
8.17.4.1	apply	70
8.17.4.2	initialize	70
8.17.5	Friends And Related Function Documentation	70
8.17.5.1	init_attrjeod__DynBodyInitOrbit	70
8.17.5.2	InputProcessor	70
8.17.6	Field Documentation	70
8.17.6.1	alt_apoapsis	71
8.17.6.2	alt_periapsis	71
8.17.6.3	arg_latitude	71
8.17.6.4	arg_periapsis	71
8.17.6.5	ascending_node	71

8.17.6.6	eccentricity	71
8.17.6.7	inclination	71
8.17.6.8	mean_anomaly	72
8.17.6.9	orb_radius	72
8.17.6.10	orbit_frame	72
8.17.6.11	orbit_frame_name	72
8.17.6.12	planet	72
8.17.6.13	planet_name	72
8.17.6.14	radial_vel	73
8.17.6.15	semi_latus_rectum	73
8.17.6.16	semi_major_axis	73
8.17.6.17	set	73
8.17.6.18	time_periapsis	73
8.17.6.19	true_anomaly	73
8.18	jeod::DynBodyInitPlanetDerived Class Reference	74
8.18.1	Detailed Description	75
8.18.2	Constructor & Destructor Documentation	75
8.18.2.1	DynBodyInitPlanetDerived	75
8.18.2.2	DynBodyInitPlanetDerived	75
8.18.2.3	~DynBodyInitPlanetDerived	75
8.18.3	Member Function Documentation	75
8.18.3.1	apply	75
8.18.3.2	initialize	76
8.18.3.3	is_ready	77
8.18.3.4	operator=	77
8.18.4	Friends And Related Function Documentation	77
8.18.4.1	init_attrjeod__DynBodyInitPlanetDerived	77
8.18.4.2	InputProcessor	77
8.18.5	Field Documentation	77
8.18.5.1	body_is_required	77
8.18.5.2	ref_body	77
8.18.5.3	ref_body_name	78
8.18.5.4	required_items	78
8.19	jeod::DynBodyInitRotState Class Reference	78
8.19.1	Detailed Description	79
8.19.2	Member Enumeration Documentation	79
8.19.2.1	StateItems	79
8.19.3	Constructor & Destructor Documentation	80
8.19.3.1	DynBodyInitRotState	80
8.19.3.2	DynBodyInitRotState	80

8.19.3.3	<code>~DynBodyInitRotState</code>	80
8.19.4	Member Function Documentation	80
8.19.4.1	<code>apply</code>	80
8.19.4.2	<code>initialize</code>	80
8.19.4.3	<code>initializes_what</code>	80
8.19.4.4	<code>is_ready</code>	81
8.19.4.5	<code>operator=</code>	81
8.19.5	Friends And Related Function Documentation	81
8.19.5.1	<code>init_attrjeod__DynBodyInitRotState</code>	81
8.19.5.2	<code>InputProcessor</code>	81
8.19.6	Field Documentation	81
8.19.6.1	<code>state_items</code>	81
8.20	<code>jeod::DynBodyInitTransState</code> Class Reference	81
8.20.1	Detailed Description	83
8.20.2	Member Enumeration Documentation	83
8.20.2.1	<code>StateItems</code>	83
8.20.3	Constructor & Destructor Documentation	83
8.20.3.1	<code>DynBodyInitTransState</code>	83
8.20.3.2	<code>~DynBodyInitTransState</code>	83
8.20.3.3	<code>DynBodyInitTransState</code>	83
8.20.4	Member Function Documentation	83
8.20.4.1	<code>apply</code>	83
8.20.4.2	<code>initialize</code>	84
8.20.4.3	<code>initializes_what</code>	84
8.20.4.4	<code>is_ready</code>	84
8.20.4.5	<code>operator=</code>	84
8.20.5	Friends And Related Function Documentation	84
8.20.5.1	<code>init_attrjeod__DynBodyInitTransState</code>	84
8.20.5.2	<code>InputProcessor</code>	85
8.20.6	Field Documentation	85
8.20.6.1	<code>state_items</code>	85
8.21	<code>jeod::DynBodyInitWrtPlanet</code> Class Reference	85
8.21.1	Detailed Description	86
8.21.2	Constructor & Destructor Documentation	86
8.21.2.1	<code>DynBodyInitWrtPlanet</code>	86
8.21.2.2	<code>DynBodyInitWrtPlanet</code>	86
8.21.2.3	<code>~DynBodyInitWrtPlanet</code>	86
8.21.3	Member Function Documentation	86
8.21.3.1	<code>apply</code>	86
8.21.3.2	<code>initialize</code>	88

8.21.3.3	initializes_what	88
8.21.3.4	is_ready	88
8.21.3.5	operator=	89
8.21.4	Friends And Related Function Documentation	89
8.21.4.1	init_attrjeod__DynBodyInitWrtPlanet	89
8.21.4.2	InputProcessor	89
8.21.5	Field Documentation	89
8.21.5.1	planet	89
8.21.5.2	planet_name	89
8.21.5.3	set_items	89
8.22	jeod::MassBodyInit Class Reference	89
8.22.1	Detailed Description	90
8.22.2	Constructor & Destructor Documentation	91
8.22.2.1	MassBodyInit	91
8.22.2.2	MassBodyInit	91
8.22.2.3	~MassBodyInit	91
8.22.3	Member Function Documentation	91
8.22.3.1	apply	91
8.22.3.2	operator=	91
8.22.4	Friends And Related Function Documentation	91
8.22.4.1	init_attrjeod__MassBodyInit	91
8.22.4.2	InputProcessor	91
8.22.5	Field Documentation	91
8.22.5.1	num_points	91
8.22.5.2	points	91
8.22.5.3	properties	92
9	File Documentation	93
9.1	body_action.cc File Reference	93
9.1.1	Detailed Description	93
9.2	body_action.hh File Reference	93
9.2.1	Detailed Description	94
9.3	body_action_messages.cc File Reference	94
9.3.1	Detailed Description	94
9.4	body_action_messages.hh File Reference	94
9.4.1	Detailed Description	95
9.5	body_attach.cc File Reference	95
9.6	body_attach.hh File Reference	95
9.7	body_attach_aligned.cc File Reference	96
9.8	body_attach_aligned.hh File Reference	96

9.9	body_attach_matrix.cc File Reference	96
9.10	body_attach_matrix.hh File Reference	97
9.11	body_detach.cc File Reference	97
9.12	body_detach.hh File Reference	97
9.13	body_detach_specific.cc File Reference	98
9.14	body_detach_specific.hh File Reference	98
9.15	body_reattach.cc File Reference	98
9.16	body_reattach.hh File Reference	99
9.17	class_declarations.hh File Reference	99
9.17.1	Detailed Description	99
9.18	dyn_body_frame_switch.cc File Reference	99
9.18.1	Detailed Description	100
9.19	dyn_body_frame_switch.hh File Reference	100
9.19.1	Detailed Description	100
9.20	dyn_body_init.cc File Reference	100
9.20.1	Detailed Description	101
9.21	dyn_body_init.hh File Reference	101
9.21.1	Detailed Description	101
9.22	dyn_body_init_lvlh_rot_state.cc File Reference	102
9.22.1	Detailed Description	102
9.23	dyn_body_init_lvlh_rot_state.hh File Reference	102
9.23.1	Detailed Description	102
9.24	dyn_body_init_lvlh_state.cc File Reference	103
9.24.1	Detailed Description	103
9.25	dyn_body_init_lvlh_state.hh File Reference	103
9.25.1	Detailed Description	103
9.26	dyn_body_init_lvlh_trans_state.cc File Reference	104
9.26.1	Detailed Description	104
9.27	dyn_body_init_lvlh_trans_state.hh File Reference	104
9.27.1	Detailed Description	104
9.28	dyn_body_init_ned_rot_state.cc File Reference	105
9.28.1	Detailed Description	105
9.29	dyn_body_init_ned_rot_state.hh File Reference	105
9.29.1	Detailed Description	105
9.30	dyn_body_init_ned_state.cc File Reference	106
9.30.1	Detailed Description	106
9.31	dyn_body_init_ned_state.hh File Reference	106
9.31.1	Detailed Description	107
9.32	dyn_body_init_ned_trans_state.cc File Reference	107
9.32.1	Detailed Description	107

9.33	dyn_body_init_ned_trans_state.hh File Reference	107
9.33.1	Detailed Description	108
9.34	dyn_body_init_orbit.cc File Reference	108
9.34.1	Detailed Description	108
9.35	dyn_body_init_orbit.hh File Reference	108
9.35.1	Detailed Description	109
9.36	dyn_body_init_planet_derived.cc File Reference	109
9.36.1	Detailed Description	109
9.37	dyn_body_init_planet_derived.hh File Reference	109
9.37.1	Detailed Description	110
9.38	dyn_body_init_rot_state.cc File Reference	110
9.38.1	Detailed Description	110
9.39	dyn_body_init_rot_state.hh File Reference	110
9.39.1	Detailed Description	111
9.40	dyn_body_init_trans_state.cc File Reference	111
9.40.1	Detailed Description	111
9.41	dyn_body_init_trans_state.hh File Reference	111
9.41.1	Detailed Description	112
9.42	dyn_body_init_wrt_planet.cc File Reference	112
9.42.1	Detailed Description	112
9.43	dyn_body_init_wrt_planet.hh File Reference	112
9.43.1	Detailed Description	113
9.44	mass_body_init.cc File Reference	113
9.44.1	Detailed Description	113
9.45	mass_body_init.hh File Reference	113
9.45.1	Detailed Description	114

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models	11
Dynamics	12
BodyAction	13

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod	Namespace jeod	15
----------------------	--------------------------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

jeod::BodyAction	17
jeod::BodyAttach	26
jeod::BodyAttachAligned	29
jeod::BodyAttachMatrix	31
jeod::BodyDetach	34
jeod::BodyDetachSpecific	36
jeod::BodyReattach	38
jeod::DynBodyFrameSwitch	41
jeod::DynBodyInit	44
jeod::DynBodyInitRotState	78
jeod::DynBodyInitTransState	81
jeod::DynBodyInitOrbit	67
jeod::DynBodyInitWrtPlanet	85
jeod::DynBodyInitPlanetDerived	74
jeod::DynBodyInitLvLhState	55
jeod::DynBodyInitLvLhRotState	53
jeod::DynBodyInitLvLhTransState	59
jeod::DynBodyInitNedState	62
jeod::DynBodyInitNedRotState	60
jeod::DynBodyInitNedTransState	65
jeod::MassBodyInit	89
jeod::BodyActionMessages	22

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	17
jeod::BodyActionMessages	
Specifies the message IDs used in the BodyAction model	22
jeod::BodyAttach	
Provides the basic ability to attach one MassBody to another	26
jeod::BodyAttachAligned	
Attaches a pair of MassBody objects at a pair of MassPoints	29
jeod::BodyAttachMatrix	
Attaches a pair of MassBody objects using the offset+matrix attach mechanism	31
jeod::BodyDetach	
Provides the basic ability to detach one MassBody from another	34
jeod::BodyDetachSpecific	
Causes the subject body to detach from a specific body by severing the link immediately spawning from the detach_from body	36
jeod::BodyReattach	
Alters the nature of an existing attachment	38
jeod::DynBodyFrameSwitch	
Switch a DynBody 's integration frame to a specified frame when the body switches to that integration frame's sphere of influence	41
jeod::DynBodyInit	
Base class for initialize the state of a DynBody	44
jeod::DynBodyInitLvlhRotState	
Initialize a vehicle's rotational state with respect to some vehicle's LVLH frame	53
jeod::DynBodyInitLvlhState	
Initialize selected aspects of a vehicle's state with respect to some vehicle's LVLH frame	55
jeod::DynBodyInitLvlhTransState	
Initialize a vehicle's translational state with respect to some other vehicle's LVLH frame	59
jeod::DynBodyInitNedRotState	
Initialize a vehicle's rotational state wrt some vehicle's North-East-Down frame	60
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	62
jeod::DynBodyInitNedTransState	
Initialize a vehicle's translational state wrt some vehicle's North-East-Down frame	65
jeod::DynBodyInitOrbit	
Initialize a vehicle's translational state given an orbital specification	67

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	74
jeod::DynBodyInitRotState	
Initialize aspects of a vehicle's rotational state	78
jeod::DynBodyInitTransState	
Initialize aspects of a vehicle's translational state	81
jeod::DynBodyInitWrtPlanet	
Initialize selected aspects of a vehicle's state with respect to some frame based on the planet .	85
jeod::MassBodyInit	
Base class for initializing a MassBody	89

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	93
body_action.hh	Define the class BodyAction, the base class used for performing actions on a MassBody or DynBody object	93
body_action_messages.cc	Implement the class BodyActionMessages	94
body_action_messages.hh	Define the class BodyActionMessages, the class that specifies the message IDs used in the BodyAction model	94
body_attach.cc		95
body_attach.hh		95
body_attach_aligned.cc		96
body_attach_aligned.hh		96
body_attach_matrix.cc		96
body_attach_matrix.hh		97
body_detach.cc		97
body_detach.hh		97
body_detach_specific.cc		98
body_detach_specific.hh		98
body_reattach.cc		98
body_reattach.hh		99
class_declarations.hh	Forward declarations of classes defined in dyn_body_init_XXX.hh files	99
dyn_body_frame_switch.cc	Define methods for the class DynBodyFrameSwitch	99
dyn_body_frame_switch.hh	Define the class DynBodyFrameSwitch, the BodyAction derived class used for switch a DynBody's integration frame	100
dyn_body_init.cc	Define methods for the base body initialization class	100
dyn_body_init.hh	Define the class DynBodyInit, the base class used for initializing the state of a DynBody object	101
dyn_body_init_lvlh_rot_state.cc	Define methods for DynBodyInitLvLhRotState	102
dyn_body_init_lvlh_rot_state.hh	Define the class DynBodyInitLvLhRotState, which initialize a vehicle's rotational state with respect to some vehicle's LVLH frame	102

dyn_body_init_lvhl_state.cc	Define methods for the DynBodyInitLvhlState class	103
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	103
dyn_body_init_lvhl_trans_state.cc	Define methods for DynBodyInitLvhlTransState	104
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	104
dyn_body_init_ned_rot_state.cc	Define methods for DynBodyInitNedRotState	105
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	105
dyn_body_init_ned_state.cc	Define methods for DynBodyInitNedState	106
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	106
dyn_body_init_ned_trans_state.cc	Define methods for DynBodyInitNedTransState	107
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	107
dyn_body_init_orbit.cc	Define classes for items represented in some ephemeris model	108
dyn_body_init_orbit.hh	Define the class DynBodyInitOrbit, which initializes a vehicle in in some orbit	108
dyn_body_init_planet_derived.cc	Define methods for the DynBodyInitPlanetDerived class	109
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	109
dyn_body_init_rot_state.cc	Define methods for DynBodyInitRotState	110
dyn_body_init_rot_state.hh	Define the class DynBodyInitRotState that initialize aspects of a vehicle's rotational state	110
dyn_body_init_trans_state.cc	Define methods for DynBodyInitTransState	111
dyn_body_init_trans_state.hh	Define the class DynBodyInitTransState that initialize aspects of a vehicle's translational state	111
dyn_body_init_wrt_planet.cc	Define methods for the DynBodyInitWrtPlanet class	112
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	112
mass_body_init.cc	Define methods for the mass body initialization class	113
mass_body_init.hh	Define the class MassBodyInit, the base class used for initializing the core mass properties of a MassBody object	113

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

Definition at line 32 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.

Chapter 8

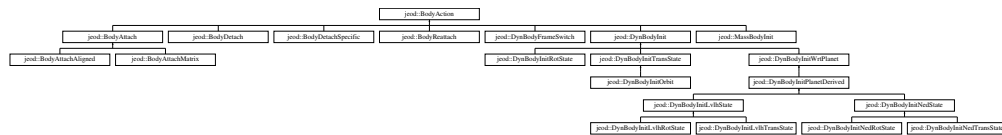
Data Structure Documentation

8.1 jeod::BodyAction Class Reference

[BodyAction](#) is the base class for the [BodyAction](#) model.

```
#include <body_action.hh>
```

Inheritance diagram for jeod::BodyAction:



Public Member Functions

- [BodyAction](#) ()
Construct a [BodyAction](#).
- virtual [~BodyAction](#) ()
Destruct a [BodyAction](#).
- virtual void [shutdown](#) ()
Release resources allocated by a [BodyAction](#) object.
- const char * [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

- MassBody * [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.
- 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.

- char * `action_name`

An identifier for this action.

Protected Member Functions

- void `validate_name` (const char *variable_value, const char *variable_type, const char *variable_name)

Ensure that a string is not trivially invalid.

Protected Attributes

- char * `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 72 of file `body_action.hh`.

8.1.2 Constructor & Destructor Documentation

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

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

Construct a [BodyAction](#).

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

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

Destruct a [BodyAction](#).

Definition at line 76 of file `body_action.cc`.

References `shutdown()`.

8.1.3 Member Function Documentation

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

Complete initialization.

Parameters

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

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

Definition at line 151 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 `const char * jeod::BodyAction::get_identifier (void) const [inline]`

Accessor for `action_identifier`.

Returns

Action identifier

Definition at line 205 of file `body_action.hh`.

References `action_identifier`.

8.1.3.3 `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

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

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

Definition at line 107 of file `body_action.cc`.

References `action_identifier`, `action_name`, `dyn_subject`, `jeod::BodyActionMessages::fatal_error`, `jeod::BodyActionMessages::null_pointer`, and `subject`.

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

8.1.3.4 `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::DynBodyInitPlanetDerived](#), [jeod::BodyDetachSpecific](#), [jeod::DynBodyInitRotState](#), [jeod::DynBodyInitWrtPlanet](#), [jeod::DynBodyInitTransState](#), and [jeod::BodyDetach](#).

Definition at line 168 of file `body_action.cc`.

References `active`.

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

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

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

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

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

References `action_identifier`.

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

8.1.3.7 `void jeod::BodyAction::validate_name (const char * variable_value, const char * variable_name, const char * variable_type) [protected]`

Ensure that a string is not trivially invalid.

Parameters

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

<code>in</code>	<code>variable_type</code>	For error reporting
-----------------	----------------------------	---------------------

Definition at line 183 of file `body_action.cc`.

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

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 `void init_attrjeod__BodyAction () [friend]`

8.1.4.2 `friend class InputProcessor [friend]`

Definition at line 74 of file `body_action.hh`.

8.1.5 Field Documentation

8.1.5.1 `char* 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 138 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::DynBodyInitLvHTransState::initialize()`, `jeod::DynBodyInitNedRotState::initialize()`, `jeod::DynBodyInitNedTransState::initialize()`, `jeod::DynBodyInitLvHRotState::initialize()`, `jeod::DynBodyInitTransState::initialize()`, `jeod::BodyAttachAligned::initialize()`, `jeod::DynBodyInitRotState::initialize()`, `jeod::BodyDetachSpecific::initialize()`, `jeod::BodyAttach::initialize()`, `jeod::DynBodyFrameSwitch::initialize()`, `initialize()`, `jeod::DynBodyInit::initialize()`, `jeod::DynBodyInitOrbit::initialize()`, `jeod::DynBodyInitTransState::is_ready()`, `jeod::DynBodyInitRotState::is_ready()`, `jeod::DynBodyInit::report_failure()`, `shutdown()`, and `validate_name()`.

8.1.5.2 `char* jeod::BodyAction::action_name`

An identifier for this action.

This can be left as NULL (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 128 of file `body_action.hh`.

Referenced by `initialize()`.

8.1.5.3 `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 107 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 DynBody* jeod::BodyAction::dyn_subject

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

This or the subject pointer must be supplied. If both applied, they must be consistent between the two bodies. Actions on the body are performed by the apply methods of specific class derived from the [BodyAction](#) class. `trick_io(**)`

Definition at line 96 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()`, `jeod::DynBodyInitLvlhRotState::initialize()`, `jeod::BodyDetach::initialize()`, `jeod::BodyDetachSpecific::initialize()`, `jeod::BodyAttach::initialize()`, `jeod::DynBodyFrameSwitch::initialize()`, `initialize()`, `jeod::DynBodyInit::initialize()`, and `jeod::DynBodyFrameSwitch::is_ready()`.

8.1.5.5 MassBody* jeod::BodyAction::subject

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

This or the dyn_subject pointer must be supplied. If both applied, they must be consistent between the two bodies. Actions on the body are performed by the apply methods of specific class derived from the [BodyAction](#) class. `trick_units(-)`

Definition at line 87 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()`, `jeod::DynBodyInitLvlhRotState::initialize()`, `jeod::BodyDetach::initialize()`, `jeod::BodyDetachSpecific::initialize()`, `jeod::BodyAttach::initialize()`, `jeod::DynBodyFrameSwitch::initialize()`, `initialize()`, and `jeod::DynBodyInit::initialize()`.

8.1.5.6 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 119 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 46 of file [body_action_messages.hh](#).

8.2.2 Constructor & Destructor Documentation

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

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

8.2.3 Member Function Documentation

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

8.2.4 Friends And Related Function Documentation

8.2.4.1 `void init_attrjeod__BodyActionMessages () [friend]`

8.2.4.2 `friend class InputProcessor [friend]`

Definition at line 49 of file `body_action_messages.hh`.

8.2.5 Field Documentation

8.2.5.1 `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 60 of file `body_action_messages.hh`.

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

8.2.5.2 `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 65 of file `body_action_messages.hh`.

Referenced by `jeod::DynBodyInitLvLhState::apply()`, `jeod::DynBodyInitNedState::apply()`, `jeod::DynBodyInitOrbit::apply()`, `jeod::DynBodyInitNedRotState::initialize()`, `jeod::DynBodyInitLvLhTransState::initialize()`, `jeod::DynBodyInitNedTransState::initialize()`, `jeod::DynBodyInitLvLhRotState::initialize()`, `jeod::DynBodyInitTransState::initialize()`, `jeod::DynBodyInitRotState::initialize()`, and `jeod::DynBodyInitOrbit::initialize()`.

8.2.5.3 `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 71 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 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 76 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 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 86 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 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 81 of file body_action_messages.hh.

Referenced by jeod::DynBodyInitLvlhRotState::initialize(), jeod::BodyDetachSpecific::initialize(), jeod::BodyAttach::initialize(), jeod::BodyAction::initialize(), and jeod::BodyAction::validate_name().

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

Initial value:

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

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

trick_units(–)

Definition at line 91 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:

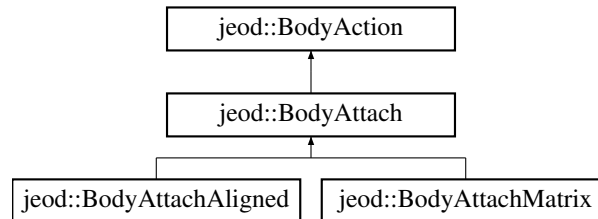
- [body_action_messages.hh](#)
- [body_action_messages.cc](#)

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

- [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

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

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 59 of file `body_attach.hh`.

8.3.2 Constructor & Destructor Documentation

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

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

Construct a MassBodyAttach.

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

References `jeod::BodyAction::active`.

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

Destructor.

Definition at line 122 of file `body_attach.hh`.

8.3.3 Member Function Documentation

8.3.3.1 `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

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

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

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

Definition at line 118 of file `body_attach.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyAction::apply()`, `jeod::BodyActionMessages::fatal_error`, `jeod::BodyActionMessages::not_performed`, `parent`, `jeod::BodyAction::subject`, `succeeded`, `jeod::BodyAction::terminate_on_error`, and `jeod::BodyActionMessages::trace`.

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

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

Initialize a MassBodyAttach.

Parameters

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

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

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

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

References [jeod::BodyAction::action_identifier](#), [dyn_parent](#), [jeod::BodyAction::dyn_subject](#), [jeod::BodyAction::initialize\(\)](#), [jeod::BodyActionMessages::null_pointer](#), [parent](#), and [jeod::BodyAction::subject](#).

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

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

8.3.4 Friends And Related Function Documentation

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

8.3.4.2 `friend class InputProcessor` `[friend]`

Definition at line 61 of file `body_attach.hh`.

8.3.5 Field Documentation

8.3.5.1 `DynBody* jeod::BodyAttach::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.

This or the parent pointer must be supplied. If both applied, they must be consistent between the two bodies.`trick_units(-)`

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

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

8.3.5.2 `MassBody* jeod::BodyAttach::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.

This or the `dyn_parent` pointer must be supplied. If both applied, they must be consistent between the two bodies.`-trick_units(-)`

Definition at line 75 of file `body_attach.hh`.

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

8.3.5.3 `bool jeod::BodyAttach::succeeded`

Did the attachment succeed?

`trick_units(-)`

Definition at line 89 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:

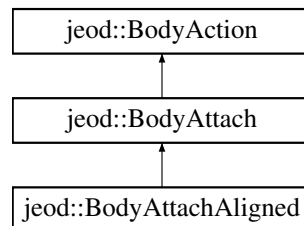
- [body_attach.hh](#)
- [body_attach.cc](#)

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

- char * [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.
- char * [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 52 of file body_attach_aligned.hh.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 jeod::BodyAttachAligned::BodyAttachAligned (void)

Construct a MassBodyAttachAligned.

Definition at line 55 of file body_attach_aligned.cc.

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

Destructor.

Definition at line 105 of file body_attach_aligned.hh.

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

8.4.3 Member Function Documentation

8.4.3.1 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 105 of file body_attach_aligned.cc.

References [jeod::BodyAttach::apply\(\)](#), [jeod::BodyAttach::dyn_parent](#), [jeod::BodyAction::dyn_subject](#), [jeod::BodyAttach::parent](#), [parent_point_name](#), [jeod::BodyAction::subject](#), [subject_point_name](#), and [jeod::BodyAttach::succeeded](#).

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

Initialize a MassBodyAttach.

Parameters

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

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

Definition at line 70 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 `BodyAttachAligned& jeod::BodyAttachAligned::operator= (const BodyAttachAligned &)` `[private]`

8.4.4 Friends And Related Function Documentation

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

8.4.4.2 `friend class InputProcessor` `[friend]`

Definition at line 54 of file `body_attach_aligned.hh`.

8.4.5 Field Documentation

8.4.5.1 `char* 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 73 of file `body_attach_aligned.hh`.

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

8.4.5.2 `char* 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 66 of file `body_attach_aligned.hh`.

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

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

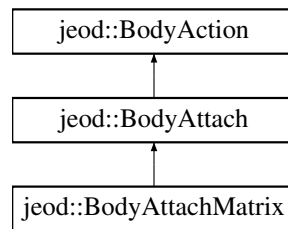
- [body_attach_aligned.hh](#)
- [body_attach_aligned.cc](#)

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 53 of file `body_attach_matrix.hh`.

8.5.2 Constructor & Destructor Documentation

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

Construct a MassBodyAttachMatrix.

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

References `offset_pstr_cstr_pstr`.

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

Destructor.

Definition at line 96 of file `body_attach_matrix.hh`.

8.5.3 Member Function Documentation

8.5.3.1 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 66 of file `body_attach_matrix.cc`.

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

8.5.4 Friends And Related Function Documentation

8.5.4.1 void init_attrjeod__BodyAttachMatrix () [friend]

8.5.4.2 friend class InputProcessor [friend]

Definition at line 55 of file `body_attach_matrix.hh`.

8.5.5 Field Documentation

8.5.5.1 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 67 of file `body_attach_matrix.hh`.

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

8.5.5.2 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 73 of file `body_attach_matrix.hh`.

Referenced by `apply()`.

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

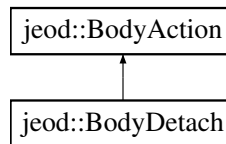
- [body_attach_matrix.hh](#)
- [body_attach_matrix.cc](#)

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 [initialize](#) (DynManager &dyn_manager)
Initialize a MassBodyDetach.
- 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 55 of file `body_detach.hh`.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 jeod::BodyDetach::BodyDetach (void)

Construct a MassBodyDetach.

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

References `jeod::BodyAction::active`.

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

Destructor.

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

8.6.3 Member Function Documentation**8.6.3.1** `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 95 of file `body_detach.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyAction::apply()`, `jeod::BodyAction::dyn_subject`, `jeod::BodyActionMessages::fatal_error`, `jeod::BodyActionMessages::not_performed`, `jeod::BodyAction::subject`, `jeod::BodyAction::terminate_on_error`, and `jeod::BodyActionMessages::trace`.

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

Initialize a MassBodyDetach.

Parameters

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

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

Definition at line 70 of file `body_detach.cc`.

References `jeod::BodyAction::dyn_subject`, `jeod::BodyAction::initialize()`, and `jeod::BodyAction::subject`.

8.6.3.3 `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 149 of file `body_detach.cc`.

References `jeod::BodyAction::active`.

8.6.4 Friends And Related Function Documentation**8.6.4.1** `void init_attrjeod__BodyDetach () [friend]`**8.6.4.2** `friend class InputProcessor [friend]`

Definition at line 57 of file `body_detach.hh`.

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

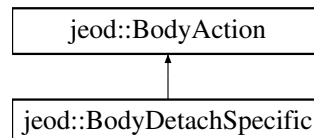
- [body_detach.hh](#)
- [body_detach.cc](#)

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

- [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.

Data Fields

- MassBody * [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 59 of file body_detach_specific.hh.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 jeod::BodyDetachSpecific::BodyDetachSpecific (void)

Construct a [BodyDetachSpecific](#).

Definition at line 55 of file body_detach_specific.cc.

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

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

Destructor.

Definition at line 115 of file body_detach_specific.hh.

8.7.3 Member Function Documentation

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

Detach the body from its parent.

Parameters

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

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

Definition at line 141 of file body_detach_specific.cc.

References [jeod::BodyAction::action_idenfier](#), [jeod::BodyAction::apply\(\)](#), [detach_from](#), [dyn_detach_from](#), [jeod::BodyAction::dyn_subject](#), [jeod::BodyActionMessages::fatal_error](#), [jeod::BodyActionMessages::not_performed](#), [jeod::BodyAction::subject](#), [jeod::BodyAction::terminate_on_error](#), and [jeod::BodyActionMessages::trace](#).

8.7.3.2 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 72 of file body_detach_specific.cc.

References [jeod::BodyAction::action_idenfier](#), [detach_from](#), [dyn_detach_from](#), [jeod::BodyAction::dyn_subject](#), [jeod::BodyAction::initialize\(\)](#), [jeod::BodyActionMessages::null_pointer](#), and [jeod::BodyAction::subject](#).

8.7.3.3 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 214 of file `body_detach_specific.cc`.

References `jeod::BodyAction::active`.

8.7.4 Friends And Related Function Documentation

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

8.7.4.2 `friend class InputProcessor` [[friend](#)]

Definition at line 61 of file `body_detach_specific.hh`.

8.7.5 Field Documentation

8.7.5.1 `MassBody* jeod::BodyDetachSpecific::detach_from`

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 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 76 of file `body_detach_specific.hh`.

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

8.7.5.2 `DynBody* jeod::BodyDetachSpecific::dyn_detach_from`

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 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 86 of file `body_detach_specific.hh`.

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

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

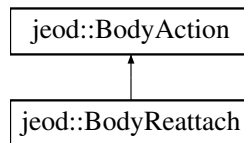
- [body_detach_specific.hh](#)
- [body_detach_specific.cc](#)

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 57 of file `body_reattach.hh`.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 jeod::BodyReattach::BodyReattach (void)

Construct a MassBodyReattach.

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

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

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

Destructor.

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

8.8.3 Member Function Documentation

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

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

Parameters

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

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

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

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

8.8.4 Friends And Related Function Documentation

8.8.4.1 `void init_attrjeod__BodyReattach () [friend]`

8.8.4.2 `friend class InputProcessor [friend]`

Definition at line 59 of file `body_reattach.hh`.

8.8.5 Field Documentation

8.8.5.1 `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 71 of file `body_reattach.hh`.

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

8.8.5.2 `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 77 of file `body_reattach.hh`.

Referenced by `apply()`.

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

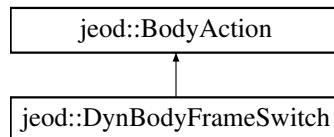
- [body_reattach.hh](#)
- [body_reattach.cc](#)

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

- char * [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 54 of file `dyn_body_frame_switch.hh`.

8.9.2 Member Enumeration Documentation

8.9.2.1 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 69 of file `dyn_body_frame_switch.hh`.

8.9.3 Constructor & Destructor Documentation

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

Construct a [DynBodyFrameSwitch](#) instance.

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

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

Destruct a [DynBodyFrameSwitch](#) instance.

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

8.9.4 Member Function Documentation

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

Switch reference frames.

Parameters

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

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

Definition at line 156 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 void jeod::DynBodyFrameSwitch::initialize (DynManager & *dyn_manager*) [virtual]

Initialization a [DynBodyFrameSwitch](#) instance.

Parameters

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

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

Definition at line 87 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::subject](#).

8.9.4.3 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 206 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** void init_attrjeod__DynBodyFrameSwitch () [friend]**8.9.5.2** friend class InputProcessor [friend]

Definition at line 56 of file `dyn_body_frame_switch.hh`.

8.9.6 Field Documentation**8.9.6.1** EphemerisRefFrame* jeod::DynBodyFrameSwitch::integ_frame [protected]

The reference frame corresponding to the input `integ_frame_name`.

`trick_io(**)`

Definition at line 117 of file `dyn_body_frame_switch.hh`.

Referenced by [apply\(\)](#), [initialize\(\)](#), and [is_ready\(\)](#).

8.9.6.2 char* 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 92 of file `dyn_body_frame_switch.hh`.

Referenced by [apply\(\)](#), and [initialize\(\)](#).

8.9.6.3 bool jeod::DynBodyFrameSwitch::sort_grav_controls

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

trick_units(-)

Definition at line 105 of file dyn_body_frame_switch.hh.

Referenced by apply().

8.9.6.4 double jeod::DynBodyFrameSwitch::switch_distance

The radius of the sphere of influence.

trick_units(m)

Definition at line 110 of file dyn_body_frame_switch.hh.

Referenced by is_ready().

8.9.6.5 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 99 of file dyn_body_frame_switch.hh.

Referenced by is_ready().

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

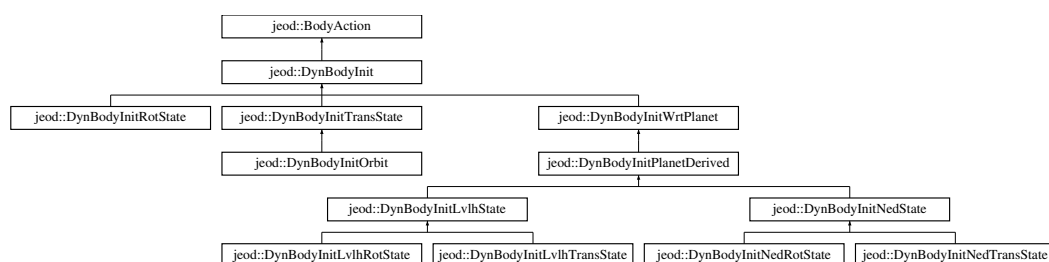
- [dyn_body_frame_switch.hh](#)
- [dyn_body_frame_switch.cc](#)

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

- char * [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.
- char * [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 char *planet_name, const char *variable_name)
Find the Planet with the given name, failing if not found.
- DynBody * [find_dyn_body](#) (DynManager &dyn_manager, const char *dyn_body_name, const char *variable_name)
Find the DynBody with the given name, failing if not found.
- RefFrame * [find_ref_frame](#) (DynManager &dyn_manager, const char *ref_frame_name, const char *variable_name)
Find the RefFrame with the given name, failing if not found.
- BodyRefFrame * [find_body_frame](#) (DynBody &frame_container, const char *body_frame_identifier, const char *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 55 of file dyn_body_init.hh.

8.10.2 Constructor & Destructor Documentation

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

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

Construct a [DynBodyInit](#).

Definition at line 61 of file dyn_body_init.cc.

References [ang_velocity](#), [position](#), and [velocity](#).

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

Destruct a [DynBodyInit](#).

Definition at line 86 of file dyn_body_init.cc.

References [body_frame_id](#).

8.10.3 Member Function Documentation

8.10.3.1 `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

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

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

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

Definition at line 268 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 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 334 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::DynBodyInitTransState::apply\(\)](#), [jeod::DynBodyInitLvlhState::apply\(\)](#), [jeod::DynBodyInitNedState::apply\(\)](#), [jeod::DynBodyInitRotState::apply\(\)](#), [compute_rotational_state\(\)](#), and [compute_translational_state\(\)](#).

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

This method is obsolete.

Use [apply_user_inputs](#) instead.

Definition at line 394 of file `dyn_body_init.cc`.

References [apply_user_inputs\(\)](#), and [jeod::BodyActionMessages::invalid_name](#).

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

This method is obsolete.

Use [apply_user_inputs](#) instead.

Definition at line 417 of file `dyn_body_init.cc`.

References [apply_user_inputs\(\)](#), and [jeod::BodyActionMessages::invalid_name](#).

8.10.3.5 BodyRefFrame * jeod::DynBodyInit::find_body_frame (DynBody & frame_container, const char * body_frame_identifier, const char * 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 556 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 DynBody * jeod::DynBodyInit::find_dyn_body (DynManager & *dyn_manager*, const char * *dyn_body_name*, const char * *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 482 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(), and initialize().

8.10.3.7 Planet * jeod::DynBodyInit::find_planet (DynManager & *dyn_manager*, const char * *planet_name*, const char * *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 445 of file dyn_body_init.cc.

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

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

8.10.3.8 RefFrame * jeod::DynBodyInit::find_ref_frame (DynManager & *dyn_manager*, const char * *ref_frame_name*, const char * *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 519 of file dyn_body_init.cc.

References [jeod::BodyAction::action_identifier](#), [jeod::BodyActionMessages::invalid_name](#), and [jeod::BodyAction::validate_name\(\)](#).

Referenced by [initialize\(\)](#).

8.10.3.9 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

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

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

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

Definition at line 104 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_dyn_body\(\)](#), [find_ref_frame\(\)](#), [jeod::BodyAction::initialize\(\)](#), [jeod::BodyActionMessages::invalid_object](#), [reference_ref_frame](#), [reference_ref_frame_name](#), [jeod::BodyAction::subject](#), and [subscribed_frame](#).

Referenced by [jeod::DynBodyInitTransState::initialize\(\)](#), [jeod::DynBodyInitRotState::initialize\(\)](#), and [jeod::DynBodyInitWrtPlanet::initialize\(\)](#).

8.10.3.10 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 189 of file dyn_body_init.cc.

Referenced by [apply\(\)](#), [apply_user_inputs\(\)](#), [is_ready\(\)](#), and [report_failure\(\)](#).

8.10.3.11 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 204 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 `DynBodyInit& jeod::DynBodyInit::operator= (const DynBodyInit &)` `[private]`

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

Report on an initializer that could not be processed.

Definition at line 308 of file `dyn_body_init.cc`.

References `jeod::BodyAction::action_identifier`, `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 `void init_attrjeod__DynBodyInit ()` `[friend]`

8.10.4.2 `friend class InputProcessor` `[friend]`

Definition at line 57 of file `dyn_body_init.hh`.

8.10.5 Field Documentation

8.10.5.1 `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(radian/s)`

Definition at line 127 of file `dyn_body_init.hh`.

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

8.10.5.2 `char* 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 66 of file `dyn_body_init.hh`.

Referenced by `initialize()`, and `~DynBodyInit()`.

8.10.5.3 `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 152 of file `dyn_body_init.hh`.

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

8.10.5.4 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 109 of file `dyn_body_init.hh`.

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

8.10.5.5 `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 91 of file `dyn_body_init.hh`.

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

8.10.5.6 `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 144 of file `dyn_body_init.hh`.

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

8.10.5.7 `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 158 of file `dyn_body_init.hh`.

Referenced by `jeod::DynBodyInitLvlhState::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 `char*` `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 72 of file `dyn_body_init.hh`.

Referenced by `initialize()`.

8.10.5.9 `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 137 of file `dyn_body_init.hh`.

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

8.10.5.10 `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 81 of file `dyn_body_init.hh`.

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

8.10.5.11 `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 168 of file `dyn_body_init.hh`.

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

8.10.5.12 `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 101 of file `dyn_body_init.hh`.

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

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

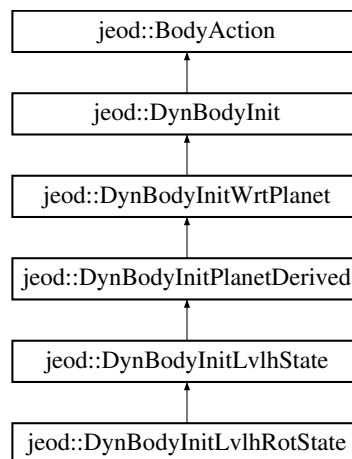
- [dyn_body_init.hh](#)
- [dyn_body_init.cc](#)

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 50 of file `dyn_body_init_lvih_rot_state.hh`.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 `jeod::DynBodyInitLvihRotState::DynBodyInitLvihRotState (const DynBodyInitLvihRotState &) [private]`

8.11.2.2 `jeod::DynBodyInitLvihRotState::DynBodyInitLvihRotState (void)`

[DynBodyInitLvihRotState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

8.11.2.3 `jeod::DynBodyInitLvihRotState::~~DynBodyInitLvihRotState (void) [virtual]`

[DynBodyInitLvihRotState](#) destructor.

Definition at line 74 of file `dyn_body_init_lvih_rot_state.cc`.

8.11.3 Member Function Documentation

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

Initialize the initializer.

Parameters

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

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

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

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

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

8.11.4 Friends And Related Function Documentation

8.11.4.1 `void init_attrjeod__DynBodyInitLvihRotState () [friend]`

8.11.4.2 `friend class InputProcessor [friend]`

Definition at line 52 of file `dyn_body_init_lvih_rot_state.hh`.

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

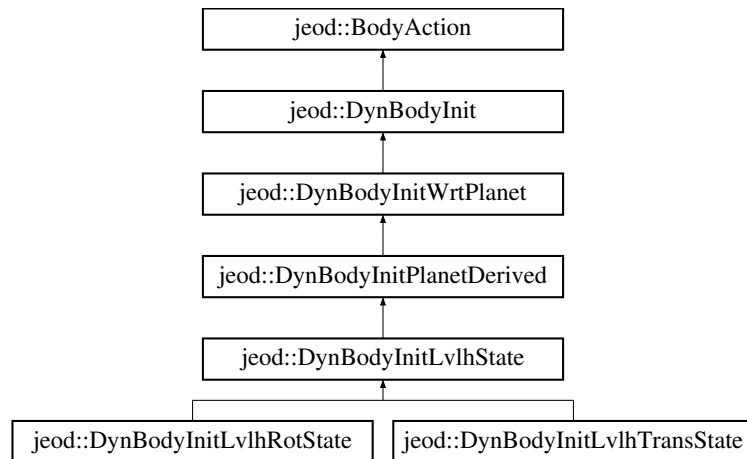
- [dyn_body_init_lvLh_rot_state.hh](#)
- [dyn_body_init_lvLh_rot_state.cc](#)

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

- [DynBodyInitLvLhState](#) ()
DynBodyInitLvLhState default constructor.
- virtual [~DynBodyInitLvLhState](#) ()
DynBodyInitLvLhState destructor.
- void [set_lvLh_frame_object](#) (LvLhFrame &lvh_frame_object)
Cache a pointer to a user-supplied LvLhFrame 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

- LvLhType::Type [lvLh_type](#)
Indicates type of LVLH coordinates desired.

Private Member Functions

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

Private Attributes

- `LvlhFrame * lvlh_object_ptr`

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

Friends

- class `InputProcessor`
- void `init_attrjeod__DynBodyInitLvlhState ()`

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 54 of file `dyn_body_init_lvlh_state.hh`.

8.12.2 Constructor & Destructor Documentation

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

`DynBodyInitLvlhState` default constructor.

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

References `jeod::DynBodyInitPlanetDerived::required_items`.

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

`DynBodyInitLvlhState` destructor.

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

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

8.12.3 Member Function Documentation

8.12.3.1 `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 118 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 void jeod::DynBodyInitLvHState::initialize (DynManager & *dyn_manager*) [virtual]

Initialize the initializer.

Parameters

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

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

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

Definition at line 91 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 DynBodyInitLvlhState& jeod::DynBodyInitLvlhState::operator= (const DynBodyInitLvlhState &)
[private]

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

Cache a pointer to a user-supplied LvlhFrame object.

Parameters

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

Definition at line 79 of file `dyn_body_init_lvlh_state.cc`.

References [lvlh_object_ptr](#).

8.12.4 Friends And Related Function Documentation

8.12.4.1 void init_attrjeod_DynBodyInitLvlhState () [friend]

8.12.4.2 friend class InputProcessor [friend]

Definition at line 56 of file `dyn_body_init_lvlh_state.hh`.

8.12.5 Field Documentation

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

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

`trick_units(-)`

Definition at line 73 of file `dyn_body_init_lvlh_state.hh`.

Referenced by [apply\(\)](#), [initialize\(\)](#), and [set_lvlh_frame_object\(\)](#).

8.12.5.2 LvlhType::Type jeod::DynBodyInitLvlhState::lvlh_type

Indicates type of LVLH coordinates desired.

Default is `rectilinear.trick_units(-)`

Definition at line 66 of file `dyn_body_init_lvlh_state.hh`.

Referenced by [apply\(\)](#).

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

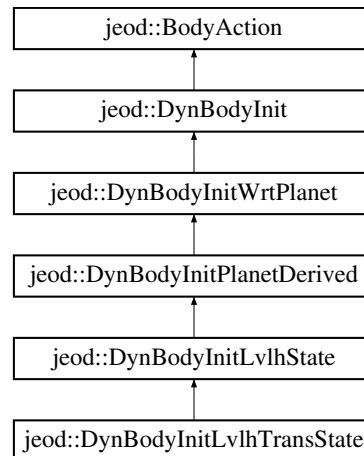
- [dyn_body_init_lvlh_state.hh](#)
- [dyn_body_init_lvlh_state.cc](#)

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 50 of file `dyn_body_init_lvlh_trans_state.hh`.

8.13.2 Constructor & Destructor Documentation

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

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

[DynBodyInitLvLhTransState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

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

[DynBodyInitLvLhTransState](#) destructor.

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

8.13.3 Member Function Documentation

8.13.3.1 `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 79 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 `DynBodyInitLvLhTransState& jeod::DynBodyInitLvLhTransState::operator= (const DynBodyInitLvLhTransState &)` [private]

8.13.4 Friends And Related Function Documentation

8.13.4.1 `void init_attrjeod__DynBodyInitLvLhTransState ()` [friend]

8.13.4.2 `friend class InputProcessor` [friend]

Definition at line 52 of file `dyn_body_init_lvLh_trans_state.hh`.

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

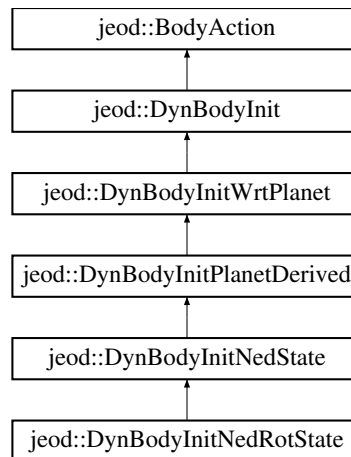
- [dyn_body_init_lvLh_trans_state.hh](#)
- [dyn_body_init_lvLh_trans_state.cc](#)

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 49 of file `dyn_body_init_ned_rot_state.hh`.

8.14.2 Constructor & Destructor Documentation

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

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

[DynBodyInitNedRotState](#) default constructor.

Definition at line 55 of file `dyn_body_init_ned_rot_state.cc`.

References `jeod::DynBodyInitWrtPlanet::set_items`.

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

[DynBodyInitNedRotState](#) destructor.

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

8.14.3 Member Function Documentation

8.14.3.1 `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 82 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 `DynBodyInitNedRotState& jeod::DynBodyInitNedRotState::operator= (const DynBodyInitNedRotState &)`
`[private]`

8.14.4 Friends And Related Function Documentation

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

8.14.4.2 `friend class InputProcessor` `[friend]`

Definition at line 51 of file `dyn_body_init_ned_rot_state.hh`.

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

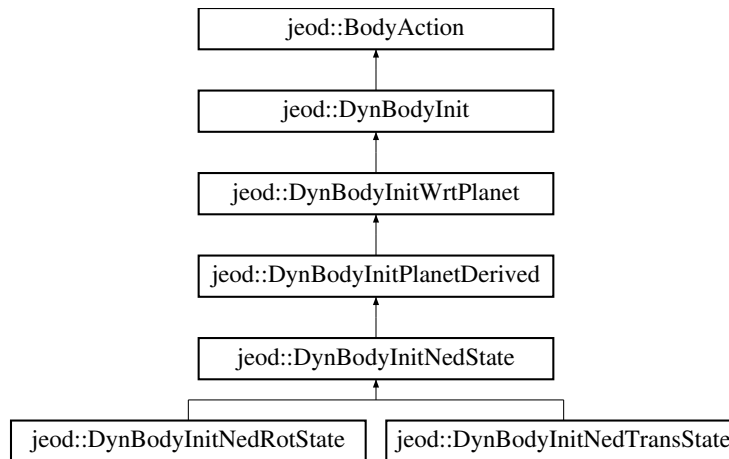
- [dyn_body_init_ned_rot_state.hh](#)
- [dyn_body_init_ned_rot_state.cc](#)

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 54 of file `dyn_body_init_ned_state.hh`.

8.15.2 Constructor & Destructor Documentation

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

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

[DynBodyInitNedState](#) default constructor.

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

References `jeod::DynBodyInitPlanetDerived::body_is_required`, and `jeod::DynBodyInitPlanetDerived::required_items`.

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

[DynBodyInitNedState](#) destructor.

Definition at line 83 of file `dyn_body_init_ned_state.cc`.

8.15.3 Member Function Documentation

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

Apply the initializer.

Parameters

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

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

Definition at line 114 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 `void jeod::DynBodyInitNedState::initialize (DynManager & dyn_manager) [virtual]`

Initialize the initializer.

Parameters

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

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

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

Definition at line 95 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 `DynBodyInitNedState& jeod::DynBodyInitNedState::operator= (const DynBodyInitNedState &) [private]`

8.15.4 Friends And Related Function Documentation

8.15.4.1 `void init_attrjeod__DynBodyInitNedState () [friend]`

8.15.4.2 `friend class InputProcessor [friend]`

Definition at line 56 of file `dyn_body_init_ned_state.hh`.

8.15.5 Field Documentation

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

Use spherical or elliptical coordinates?

`trick_units(-)`

Definition at line 72 of file `dyn_body_init_ned_state.hh`.

Referenced by `apply()`.

8.15.5.2 `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 67 of file `dyn_body_init_ned_state.hh`.

Referenced by `apply()`.

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

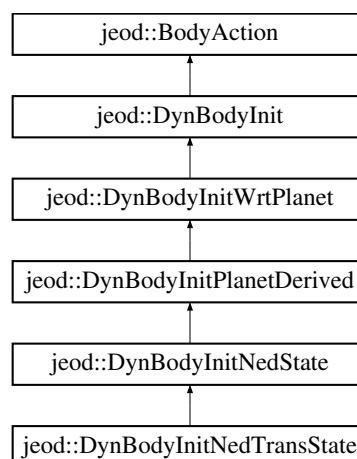
- [dyn_body_init_ned_state.hh](#)
- [dyn_body_init_ned_state.cc](#)

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 49 of file `dyn_body_init_ned_trans_state.hh`.

8.16.2 Constructor & Destructor Documentation

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

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

[DynBodyInitNedTransState](#) default constructor.

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

References `jeod::DynBodyInitWrtPlanet::set_items`.

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

[DynBodyInitNedTransState](#) destructor.

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

8.16.3 Member Function Documentation

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

Initialize the initializer.

Parameters

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

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

Definition at line 78 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 `DynBodyInitNedTransState& jeod::DynBodyInitNedTransState::operator= (const DynBodyInitNedTransState &) [private]`

8.16.4 Friends And Related Function Documentation

8.16.4.1 `void init_attrjeod__DynBodyInitNedTransState () [friend]`

8.16.4.2 `friend class InputProcessor [friend]`

Definition at line 51 of file `dyn_body_init_ned_trans_state.hh`.

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

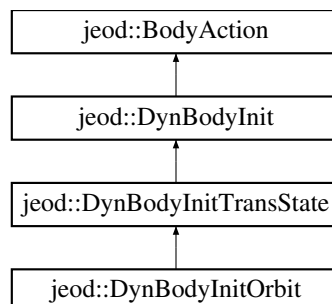
- [dyn_body_init_ned_trans_state.hh](#)
- [dyn_body_init_ned_trans_state.cc](#)

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

- char * [planet_name](#)
The name of the planet around which the orbit is to be established.
- char * [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 48 of file `dyn_body_init_orbit.hh`.

8.17.2 Member Enumeration Documentation

8.17.2.1 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
SmaEcclIncAscnodeArgperTimeperi
SmaEcclIncAscnodeArgperManom
SlrEcclIncAscnodeArgperTanom
IncAscnodeAltperAltapoArgperTanom
IncAscnodeAltperAltapoArgperTimeperi
SmaIncAscnodeArglatRadRadvel
SmaEcclIncAscnodeArgperTanom
CaseEleven

Definition at line 64 of file `dyn_body_init_orbit.hh`.

8.17.3 Constructor & Destructor Documentation

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

[DynBodyInitOrbit](#) default constructor.

Definition at line 60 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`, `orbit_frame_name`, `planet`, `planet_name`, `radial_vel`, `semi_latus_rectum`, `semi_major_axis`, `set`, `time_periapsis`, and `true_anomaly`.

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

[DynBodyInitOrbit](#) destructor.

Definition at line 91 of file `dyn_body_init_orbit.cc`.

References `orbit_frame_name`, and `planet_name`.

8.17.4 Member Function Documentation

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

Apply the initializer.

Parameters

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

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

Definition at line 201 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`, `mean_anomaly`, `orb_radius`, `orbit_frame`, `planet`, `jeod::DynBodyInit::position`, `radial_vel`, `semi_latus_rectum`, `semi_major_axis`, `set`, `SlrEcclncAscnodeArgperTanom`, `SmaEcclncAscnodeArgperManom`, `SmaEcclncAscnodeArgperTanom`, `SmaEcclncAscnodeArgperTimeperi`, `SmalncAscnodeArglatRadRadvel`, `time_periapsis`, `true_anomaly`, and `jeod::DynBodyInit::velocity`.

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

Initialize the initializer.

Parameters

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

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

Definition at line 108 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`, `orbit_frame`, `orbit_frame_name`, `planet`, `planet_name`, `jeod::DynBodyInit::reference_ref_frame`, `jeod::DynBodyInit::reverse_sense`, `set`, `SlrEcclncAscnodeArgperTanom`, `SmaEcclncAscnodeArgperManom`, `SmaEcclncAscnodeArgperTanom`, `SmaEcclncAscnodeArgperTimeperi`, `SmalncAscnodeArglatRadRadvel`, and `jeod::BodyAction::validate_name()`.

8.17.5 Friends And Related Function Documentation

8.17.5.1 void init_attrjeod__DynBodyInitOrbit () [friend]

8.17.5.2 friend class InputProcessor [friend]

Definition at line 50 of file `dyn_body_init_orbit.hh`.

8.17.6 Field Documentation

8.17.6.1 double jeod::DynBodyInitOrbit::alt_apoapsis

Apoapsis altitude.

trick_units(m)

Definition at line 175 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.2 double jeod::DynBodyInitOrbit::alt_periapsis

Periapsis altitude.

trick_units(m)

Definition at line 170 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.3 double jeod::DynBodyInitOrbit::arg_latitude

Argument of latitude.

trick_units(radian)

Definition at line 210 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.4 double jeod::DynBodyInitOrbit::arg_periapsis

Argument of periapsis.

trick_units(radian)

Definition at line 205 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.5 double jeod::DynBodyInitOrbit::ascending_node

Longitude (or right ascension) of ascending node.

trick_units(radian)

Definition at line 200 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.6 double jeod::DynBodyInitOrbit::eccentricity

Eccentricity.

trick_units(—)

Definition at line 190 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.7 double jeod::DynBodyInitOrbit::inclination

Inclination.

trick_units(radian)

Definition at line 195 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.8 double jeod::DynBodyInitOrbit::mean_anomaly

Mean anomaly.

trick_units(radian)

Definition at line 220 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.9 double jeod::DynBodyInitOrbit::orb_radius

Distance from center of planet.

trick_units(m)

Definition at line 180 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

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

The orbit reference frame (ignoring rotation)

trick_io(**)

Definition at line 238 of file dyn_body_init_orbit.hh.

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

8.17.6.11 char* 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 150 of file dyn_body_init_orbit.hh.

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

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

The planet.

trick_io(**)

Definition at line 233 of file dyn_body_init_orbit.hh.

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

8.17.6.13 char* 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 143 of file dyn_body_init_orbit.hh.

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

8.17.6.14 double jeod::DynBodyInitOrbit::radial_vel

Time derivative of the orbital radius.

trick_units(m/s)

Definition at line 185 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.15 double jeod::DynBodyInitOrbit::semi_latus_rectum

Semi-latus rectum.

trick_units(m)

Definition at line 165 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.16 double jeod::DynBodyInitOrbit::semi_major_axis

Semi-major axis.

trick_units(m)

Definition at line 160 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.17 OrbitalSet jeod::DynBodyInitOrbit::set

Specifies which set of orbital elements specify the orbit.

trick_units(-)

Definition at line 155 of file dyn_body_init_orbit.hh.

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

8.17.6.18 double jeod::DynBodyInitOrbit::time_periapsis

Time since periapsis passage.

trick_units(s)

Definition at line 215 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

8.17.6.19 double jeod::DynBodyInitOrbit::true_anomaly

True anomaly.

trick_units(radian)

Definition at line 225 of file dyn_body_init_orbit.hh.

Referenced by apply(), and DynBodyInitOrbit().

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

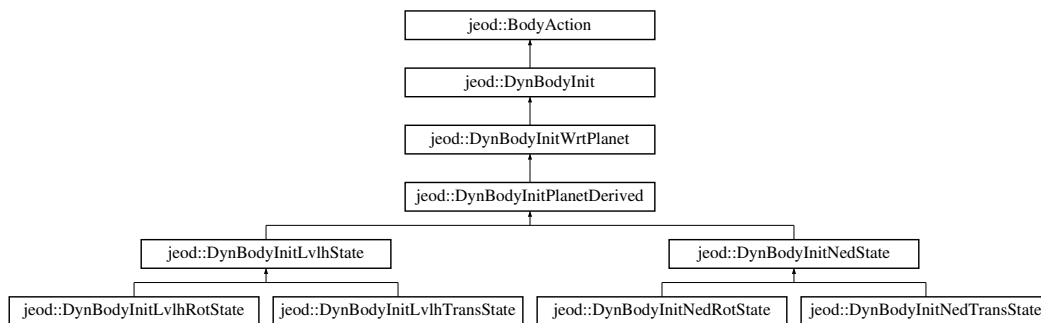
- [dyn_body_init_orbit.hh](#)
- [dyn_body_init_orbit.cc](#)

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

- char * [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 52 of file `dyn_body_init_planet_derived.hh`.

8.18.2 Constructor & Destructor Documentation

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

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

[DynBodyInitPlanetDerived](#) default constructor.

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

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

[DynBodyInitPlanetDerived](#) destructor.

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

8.18.3 Member Function Documentation

8.18.3.1 `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>dyn_manager</code>	Dynamics manager
----------------------	--------------------------	------------------

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

Reimplemented in [jeod::DynBodyInitNedState](#), and [jeod::DynBodyInitLvlhState](#).

Definition at line 131 of file `dyn_body_init_planet_derived.cc`.

References [jeod::DynBodyInitWrtPlanet::apply\(\)](#).

Referenced by [jeod::DynBodyInitLvlhState::apply\(\)](#), and [jeod::DynBodyInitNedState::apply\(\)](#).

8.18.3.2 `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::DynBodyInitLvLhState](#), [jeod::DynBodyInitLvLhRotState](#), [jeod::DynBodyInitLvLhTransState](#), [jeod::DynBodyInitNedRotState](#), and [jeod::DynBodyInitNedTransState](#).

Definition at line 80 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::DynBodyInitLvLhState::initialize()`, and `jeod::DynBodyInitNedState::initialize()`.

8.18.3.3 `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 108 of file `dyn_body_init_planet_derived.cc`.

References `jeod::DynBodyInitWrtPlanet::is_ready()`, `ref_body`, and `required_items`.

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

8.18.4 Friends And Related Function Documentation

8.18.4.1 `void init_attrjeod__DynBodyInitPlanetDerived () [friend]`

8.18.4.2 `friend class InputProcessor [friend]`

Definition at line 54 of file `dyn_body_init_planet_derived.hh`.

8.18.5 Field Documentation

8.18.5.1 `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 `RefFrame`.-
`trick_io(**)`

Definition at line 89 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 `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 74 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 `char* 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 66 of file `dyn_body_init_planet_derived.hh`.

Referenced by `jeod::DynBodyInitLvHRotState::initialize()`, `jeod::DynBodyInitNedState::initialize()`, and `initialize()`.

8.18.5.4 `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 82 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:

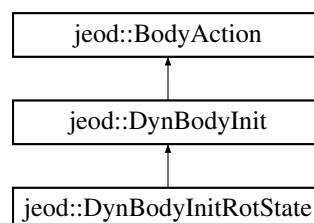
- [dyn_body_init_planet_derived.hh](#)
- [dyn_body_init_planet_derived.cc](#)

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 48 of file `dyn_body_init_rot_state.hh`.

8.19.2 Member Enumeration Documentation

8.19.2.1 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 60 of file `dyn_body_init_rot_state.hh`.

8.19.3 Constructor & Destructor Documentation

8.19.3.1 `jeod::DynBodyInitRotState::DynBodyInitRotState (const DynBodyInitRotState &) [private]`

8.19.3.2 `jeod::DynBodyInitRotState::DynBodyInitRotState (void)`

Construct a [DynBodyInitRotState](#) object.

Definition at line 55 of file `dyn_body_init_rot_state.cc`.

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

Destructor.

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

8.19.4 Member Function Documentation

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

Apply the initializer.

Parameters

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

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

Definition at line 172 of file `dyn_body_init_rot_state.cc`.

References `jeod::DynBodyInit::apply()`, and `jeod::DynBodyInit::apply_user_inputs()`.

8.19.4.2 `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

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

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

Definition at line 148 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 `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 71 of file `dyn_body_init_rot_state.cc`.

References Attitude, Both, Rate, and state_items.

Referenced by is_ready().

8.19.4.4 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 99 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 DynBodyInitRotState& jeod::DynBodyInitRotState::operator=(const DynBodyInitRotState &) [private]

8.19.5 Friends And Related Function Documentation

8.19.5.1 void init_attrjeod__DynBodyInitRotState() [friend]

8.19.5.2 friend class InputProcessor [friend]

Definition at line 50 of file dyn_body_init_rot_state.hh.

8.19.6 Field Documentation

8.19.6.1 StateItems jeod::DynBodyInitRotState::state_items

State items to be initialized – attitude, rate, or both.

trick_units(-)

Definition at line 74 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:

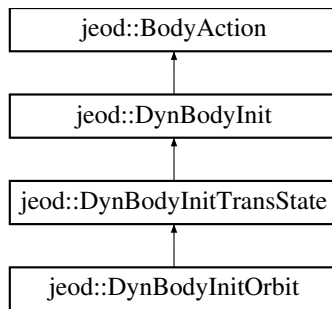
- [dyn_body_init_rot_state.hh](#)
- [dyn_body_init_rot_state.cc](#)

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 48 of file `dyn_body_init_trans_state.hh`.

8.20.2 Member Enumeration Documentation

8.20.2.1 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 58 of file `dyn_body_init_trans_state.hh`.

8.20.3 Constructor & Destructor Documentation

8.20.3.1 jeod::DynBodyInitTransState::DynBodyInitTransState (void)

Construct a [DynBodyInitTransState](#) object.

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

8.20.3.2 jeod::DynBodyInitTransState::~~DynBodyInitTransState (void) [inline], [virtual]

Destructor.

Definition at line 112 of file `dyn_body_init_trans_state.hh`.

8.20.3.3 jeod::DynBodyInitTransState::DynBodyInitTransState (const DynBodyInitTransState &) [private]

8.20.4 Member Function Documentation

8.20.4.1 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 178 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 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 153 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 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 71 of file `dyn_body_init_trans_state.cc`.

References `Both`, `Position`, `state_items`, and `Velocity`.

Referenced by `is_ready()`.

8.20.4.4 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 99 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 DynBodyInitTransState& jeod::DynBodyInitTransState::operator= (const DynBodyInitTransState &) [private]

8.20.5 Friends And Related Function Documentation

8.20.5.1 void init_attrjeod_DynBodyInitTransState () [friend]

8.20.5.2 friend class InputProcessor [friend]

Definition at line 50 of file dyn_body_init_trans_state.hh.

8.20.6 Field Documentation

8.20.6.1 StateItems jeod::DynBodyInitTransState::state_items

State items to be initialized – position, velocity, or both.

trick_units(–)

Definition at line 72 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:

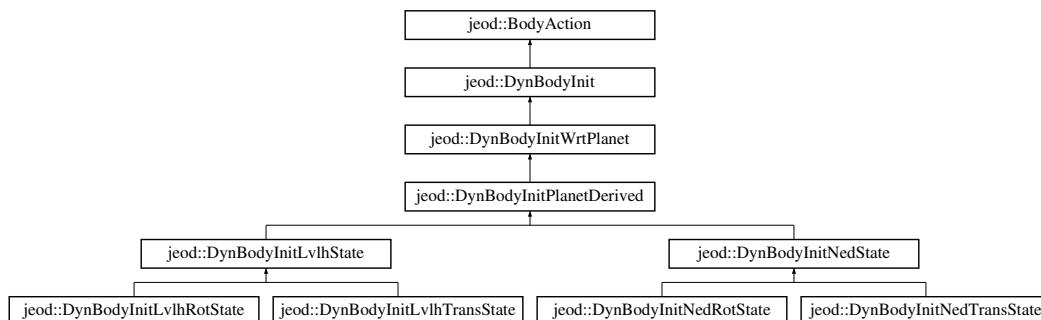
- [dyn_body_init_trans_state.hh](#)
- [dyn_body_init_trans_state.cc](#)

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

- char * [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 52 of file `dyn_body_init_wrt_planet.hh`.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 `jeod::DynBodyInitWrtPlanet::DynBodyInitWrtPlanet (const DynBodyInitWrtPlanet &) [private]`

8.21.2.2 `jeod::DynBodyInitWrtPlanet::DynBodyInitWrtPlanet (void)`

[DynBodyInitWrtPlanet](#) default constructor.

Note that by default, this class will try to set the whole enchilada.

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

8.21.2.3 `jeod::DynBodyInitWrtPlanet::~~DynBodyInitWrtPlanet (void) [virtual]`

[DynBodyInitWrtPlanet](#) destructor.

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

8.21.3 Member Function Documentation

8.21.3.1 `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 132 of file `dyn_body_init_wrt_planet.cc`.

References [jeod::DynBodyInit::apply\(\)](#).

Referenced by [jeod::DynBodyInitPlanetDerived::apply\(\)](#).

8.21.3.2 `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 78 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 `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 102 of file `dyn_body_init_wrt_planet.cc`.

References `set_items`.

8.21.3.4 `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 116 of file `dyn_body_init_wrt_planet.cc`.

References [jeod::DynBodyInit::is_ready\(\)](#).

Referenced by [jeod::DynBodyInitPlanetDerived::is_ready\(\)](#).

8.21.3.5 **DynBodyInitWrtPlanet& jeod::DynBodyInitWrtPlanet::operator= (const DynBodyInitWrtPlanet &)**
[private]

8.21.4 Friends And Related Function Documentation

8.21.4.1 **void init_attrjeod__DynBodyInitWrtPlanet ()** [friend]

8.21.4.2 **friend class InputProcessor** [friend]

Definition at line 54 of file dyn_body_init_wrt_planet.hh.

8.21.5 Field Documentation

8.21.5.1 **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 79 of file dyn_body_init_wrt_planet.hh.

Referenced by jeod::DynBodyInitLvlhState::apply(), jeod::DynBodyInitNedState::apply(), and initialize().

8.21.5.2 **char* 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 65 of file dyn_body_init_wrt_planet.hh.

Referenced by jeod::DynBodyInitNedState::apply(), and initialize().

8.21.5.3 **RefFrameItems::Items jeod::DynBodyInitWrtPlanet::set_items**

The state elements to be set by this initializer.

trick_units(-)

Definition at line 70 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:

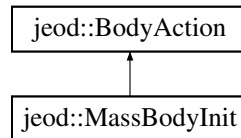
- [dyn_body_init_wrt_planet.hh](#)
- [dyn_body_init_wrt_planet.cc](#)

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 55 of file [mass_body_init.hh](#).

8.22.2 Constructor & Destructor Documentation

8.22.2.1 `jeod::MassBodyInit::MassBodyInit (const MassBodyInit &) [private]`

8.22.2.2 `jeod::MassBodyInit::MassBodyInit (void)`

Construct a [MassBodyInit](#).

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

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

Destructor.

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

8.22.3 Member Function Documentation

8.22.3.1 `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 72 of file `mass_body_init.cc`.

References `jeod::BodyAction::action_identifier`, `jeod::BodyAction::apply()`, `num_points`, `points`, `properties`, `jeod::BodyAction::subject`, and `jeod::BodyActionMessages::trace`.

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

8.22.4 Friends And Related Function Documentation

8.22.4.1 `void init_attrjeod__MassBodyInit () [friend]`

8.22.4.2 `friend class InputProcessor [friend]`

Definition at line 57 of file `mass_body_init.hh`.

8.22.5 Field Documentation

8.22.5.1 `unsigned int jeod::MassBodyInit::num_points`

Size of the points array.

`trick_units(-)`

Definition at line 77 of file `mass_body_init.hh`.

Referenced by `apply()`.

8.22.5.2 `MassPointInit* jeod::MassBodyInit::points`

Specifications for the subject mass body's mass points.

trick_units(-)

Definition at line 72 of file mass_body_init.hh.

Referenced by apply().

8.22.5.3 MassPropertiesInit jeod::MassBodyInit::properties

Specifications for the subject mass body's core mass properties.

trick_units(-)

Definition at line 67 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 <cstddef>
#include <cstdlib>
#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.

Definition in file [body_action.cc](#).

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 "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.
Definition in file [body_action.hh](#).

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.

Definition in file [body_action_messages.cc](#).

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.

Definition in file [body_action_messages.hh](#).

9.5 body_attach.cc File Reference

```
#include <cstdlib>
#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.

9.6 body_attach.hh File Reference

```
#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.

9.7 body_attach_aligned.cc File Reference

```
#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.

9.8 body_attach_aligned.hh File Reference

```
#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.

9.9 body_attach_matrix.cc File Reference

```
#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.

9.10 body_attach_matrix.hh File Reference

```
#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.

9.11 body_detach.cc File Reference

```
#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.

9.12 body_detach.hh File Reference

```
#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.

9.13 body_detach_specific.cc File Reference

```
#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.

9.14 body_detach_specific.hh File Reference

```
#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.

9.15 body_reattach.cc File Reference

```
#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.

9.16 body_reattach.hh File Reference

```
#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.

9.17 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.

Definition in file [class_declarations.hh](#).

9.18 dyn_body_frame_switch.cc File Reference

Define methods for the class DynBodyFrameSwitch.

```
#include <cstdlib>
#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.

Definition in file [dyn_body_frame_switch.cc](#).

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.

Definition in file [dyn_body_frame_switch.hh](#).

9.20 dyn_body_init.cc File Reference

Define methods for the base body initialization class.

```
#include <cstdint>
#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.

Definition in file [dyn_body_init.cc](#).

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.

Definition in file [dyn_body_init.hh](#).

9.22 dyn_body_init_lvlh_rot_state.cc File Reference

Define methods for DynBodyInitLvLhRotState.

```
#include <cstdint>
#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.

Definition in file [dyn_body_init_lvlh_rot_state.cc](#).

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.

Definition in file [dyn_body_init_lvlh_rot_state.hh](#).

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.

Definition in file [dyn_body_init_lvlh_state.cc](#).

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.

Definition in file [dyn_body_init_lvlh_state.hh](#).

9.26 dyn_body_init_lvlh_trans_state.cc File Reference

Define methods for DynBodyInitLvlhTransState.

```
#include <cstdlib>
#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.

Definition in file [dyn_body_init_lvlh_trans_state.cc](#).

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.

Definition in file [dyn_body_init_lvlh_trans_state.hh](#).

9.28 dyn_body_init_ned_rot_state.cc File Reference

Define methods for DynBodyInitNedRotState.

```
#include <cstdint>
#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.

Definition in file [dyn_body_init_ned_rot_state.cc](#).

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.

Definition in file [dyn_body_init_ned_rot_state.hh](#).

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.

Definition in file [dyn_body_init_ned_state.cc](#).

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.

Definition in file [dyn_body_init_ned_state.hh](#).

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.

Definition in file [dyn_body_init_ned_trans_state.cc](#).

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.

Definition in file [dyn_body_init_ned_trans_state.hh](#).

9.34 `dyn_body_init_orbit.cc` File Reference

Define classes for items represented in some ephemeris model.

```
#include <cstdint>
#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.

Definition in file [dyn_body_init_orbit.cc](#).

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.

Definition in file [dyn_body_init_orbit.hh](#).

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.

Definition in file [dyn_body_init_planet_derived.cc](#).

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.

Definition in file [dyn_body_init_planet_derived.hh](#).

9.38 dyn_body_init_rot_state.cc File Reference

Define methods for `DynBodyInitRotState`.

```
#include <cstdlib>
#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`.

Definition in file [dyn_body_init_rot_state.cc](#).

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.

Definition in file [dyn_body_init_rot_state.hh](#).

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.

Definition in file [dyn_body_init_trans_state.cc](#).

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.

Definition in file [dyn_body_init_trans_state.hh](#).

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.

Definition in file [dyn_body_init_wrt_planet.cc](#).

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.

Definition in file [dyn_body_init_wrt_planet.hh](#).

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.

Definition in file [mass_body_init.cc](#).

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.

Definition in file [mass_body_init.hh](#).

Index

- ~BodyAction
 - jeod::BodyAction, [19](#)
- ~BodyAttach
 - jeod::BodyAttach, [27](#)
- ~BodyAttachAligned
 - jeod::BodyAttachAligned, [30](#)
- ~BodyAttachMatrix
 - jeod::BodyAttachMatrix, [32](#)
- ~BodyDetach
 - jeod::BodyDetach, [34](#)
- ~BodyDetachSpecific
 - jeod::BodyDetachSpecific, [37](#)
- ~BodyReattach
 - jeod::BodyReattach, [39](#)
- ~DynBodyFrameSwitch
 - jeod::DynBodyFrameSwitch, [42](#)
- ~DynBodyInit
 - jeod::DynBodyInit, [46](#)
- ~DynBodyInitLvLhRotState
 - jeod::DynBodyInitLvLhRotState, [54](#)
- ~DynBodyInitLvLhState
 - jeod::DynBodyInitLvLhState, [56](#)
- ~DynBodyInitLvLhTransState
 - jeod::DynBodyInitLvLhTransState, [60](#)
- ~DynBodyInitNedRotState
 - jeod::DynBodyInitNedRotState, [61](#)
- ~DynBodyInitNedState
 - jeod::DynBodyInitNedState, [64](#)
- ~DynBodyInitNedTransState
 - jeod::DynBodyInitNedTransState, [66](#)
- ~DynBodyInitOrbit
 - jeod::DynBodyInitOrbit, [69](#)
- ~DynBodyInitPlanetDerived
 - jeod::DynBodyInitPlanetDerived, [75](#)
- ~DynBodyInitRotState
 - jeod::DynBodyInitRotState, [80](#)
- ~DynBodyInitTransState
 - jeod::DynBodyInitTransState, [83](#)
- ~DynBodyInitWrtPlanet
 - jeod::DynBodyInitWrtPlanet, [86](#)
- ~MassBodyInit
 - jeod::MassBodyInit, [91](#)
- action_identifier
 - jeod::BodyAction, [21](#)
- action_name
 - jeod::BodyAction, [21](#)
- active
 - jeod::BodyAction, [21](#)
- alt_apoapsis
 - jeod::DynBodyInitOrbit, [70](#)
- alt_periapsis
 - jeod::DynBodyInitOrbit, [71](#)
- altlatlong_type
 - jeod::DynBodyInitNedState, [65](#)
- ang_velocity
 - jeod::DynBodyInit, [50](#)
- apply
 - jeod::BodyAction, [19](#)
 - jeod::BodyAttach, [27](#)
 - jeod::BodyAttachAligned, [30](#)
 - jeod::BodyAttachMatrix, [33](#)
 - jeod::BodyDetach, [35](#)
 - jeod::BodyDetachSpecific, [37](#)
 - jeod::BodyReattach, [40](#)
 - jeod::DynBodyFrameSwitch, [42](#)
 - jeod::DynBodyInit, [46](#)
 - jeod::DynBodyInitLvLhState, [56](#)
 - jeod::DynBodyInitNedState, [64](#)
 - jeod::DynBodyInitOrbit, [70](#)
 - jeod::DynBodyInitPlanetDerived, [75](#)
 - jeod::DynBodyInitRotState, [80](#)
 - jeod::DynBodyInitTransState, [83](#)
 - jeod::DynBodyInitWrtPlanet, [86](#)
 - jeod::MassBodyInit, [91](#)
- apply_user_inputs
 - jeod::DynBodyInit, [47](#)
- arg_latitude
 - jeod::DynBodyInitOrbit, [71](#)
- arg_periapsis
 - jeod::DynBodyInitOrbit, [71](#)
- ascending_node
 - jeod::DynBodyInitOrbit, [71](#)
- Attitude
 - jeod::DynBodyInitRotState, [79](#)
- body_action.cc, [93](#)
- body_action.hh, [93](#)
- body_action_messages.cc, [94](#)
- body_action_messages.hh, [94](#)
- body_attach.cc, [95](#)
- body_attach.hh, [95](#)
- body_attach_aligned.cc, [96](#)
- body_attach_aligned.hh, [96](#)
- body_attach_matrix.cc, [96](#)
- body_attach_matrix.hh, [97](#)
- body_detach.cc, [97](#)
- body_detach.hh, [97](#)
- body_detach_specific.cc, [98](#)
- body_detach_specific.hh, [98](#)

- body_frame_id
 - jeod::DynBodyInit, 51
- body_is_required
 - jeod::DynBodyInitPlanetDerived, 77
- body_reattach.cc, 98
- body_reattach.hh, 99
- body_ref_frame
 - jeod::DynBodyInit, 51
- BodyAction, 13
 - jeod::BodyAction, 19
 - PATH, 14
- BodyActionMessages
 - jeod::BodyActionMessages, 23
- BodyAttach
 - jeod::BodyAttach, 27
- BodyAttachAligned
 - jeod::BodyAttachAligned, 30
- BodyAttachMatrix
 - jeod::BodyAttachMatrix, 32
- BodyDetach
 - jeod::BodyDetach, 34
- BodyDetachSpecific
 - jeod::BodyDetachSpecific, 37
- BodyReattach
 - jeod::BodyReattach, 39
- Both
 - jeod::DynBodyInitRotState, 79
 - jeod::DynBodyInitTransState, 83
- CaseEleven
 - jeod::DynBodyInitOrbit, 69
- class_declarations.hh, 99
- compute_rotational_state
 - jeod::DynBodyInit, 47
- compute_translational_state
 - jeod::DynBodyInit, 47
- detach_from
 - jeod::BodyDetachSpecific, 38
- dyn_body_frame_switch.cc, 99
- dyn_body_frame_switch.hh, 100
- dyn_body_init.cc, 100
- dyn_body_init.hh, 101
- dyn_body_init_lvih_rot_state.cc, 102
- dyn_body_init_lvih_rot_state.hh, 102
- dyn_body_init_lvih_state.cc, 103
- dyn_body_init_lvih_state.hh, 103
- dyn_body_init_lvih_trans_state.cc, 104
- dyn_body_init_lvih_trans_state.hh, 104
- dyn_body_init_ned_rot_state.cc, 105
- dyn_body_init_ned_rot_state.hh, 105
- dyn_body_init_ned_state.cc, 106
- dyn_body_init_ned_state.hh, 106
- dyn_body_init_ned_trans_state.cc, 107
- dyn_body_init_ned_trans_state.hh, 107
- dyn_body_init_orbit.cc, 108
- dyn_body_init_orbit.hh, 108
- dyn_body_init_planet_derived.cc, 109
- dyn_body_init_planet_derived.hh, 109
- dyn_body_init_rot_state.cc, 110
- dyn_body_init_rot_state.hh, 110
- dyn_body_init_trans_state.cc, 111
- dyn_body_init_trans_state.hh, 111
- dyn_body_init_wrt_planet.cc, 112
- dyn_body_init_wrt_planet.hh, 112
- dyn_detach_from
 - jeod::BodyDetachSpecific, 38
- dyn_parent
 - jeod::BodyAttach, 28
- dyn_subject
 - jeod::BodyAction, 22
- DynBodyFrameSwitch
 - jeod::DynBodyFrameSwitch, 42
- DynBodyInit
 - jeod::DynBodyInit, 46
- DynBodyInitLvihRotState
 - jeod::DynBodyInitLvihRotState, 54
- DynBodyInitLvihState
 - jeod::DynBodyInitLvihState, 56
- DynBodyInitLvihTransState
 - jeod::DynBodyInitLvihTransState, 60
- DynBodyInitNedRotState
 - jeod::DynBodyInitNedRotState, 61
- DynBodyInitNedState
 - jeod::DynBodyInitNedState, 64
- DynBodyInitNedTransState
 - jeod::DynBodyInitNedTransState, 66
- DynBodyInitOrbit
 - jeod::DynBodyInitOrbit, 69
- DynBodyInitPlanetDerived
 - jeod::DynBodyInitPlanetDerived, 75
- DynBodyInitRotState
 - jeod::DynBodyInitRotState, 80
- DynBodyInitTransState
 - jeod::DynBodyInitTransState, 83
- DynBodyInitWrtPlanet
 - jeod::DynBodyInitWrtPlanet, 86
- Dynamics, 12
- eccentricity
 - jeod::DynBodyInitOrbit, 71
- fatal_error
 - jeod::BodyActionMessages, 24
- find_body_frame
 - jeod::DynBodyInit, 47
- find_dyn_body
 - jeod::DynBodyInit, 48
- find_planet
 - jeod::DynBodyInit, 48
- find_ref_frame
 - jeod::DynBodyInit, 48
- get_identifier
 - jeod::BodyAction, 19
- illegal_value
 - jeod::BodyActionMessages, 24

IncAscnodeAltperAltapoArgperTanom
 jeod::DynBodyInitOrbit, 69
 IncAscnodeAltperAltapoArgperTimeperi
 jeod::DynBodyInitOrbit, 69
 inclination
 jeod::DynBodyInitOrbit, 71
 init_attrjeod__BodyAction
 jeod::BodyAction, 21
 init_attrjeod__BodyActionMessages
 jeod::BodyActionMessages, 23
 init_attrjeod__BodyAttach
 jeod::BodyAttach, 28
 init_attrjeod__BodyAttachAligned
 jeod::BodyAttachAligned, 31
 init_attrjeod__BodyAttachMatrix
 jeod::BodyAttachMatrix, 33
 init_attrjeod__BodyDetach
 jeod::BodyDetach, 35
 init_attrjeod__BodyDetachSpecific
 jeod::BodyDetachSpecific, 38
 init_attrjeod__BodyReattach
 jeod::BodyReattach, 40
 init_attrjeod__DynBodyFrameSwitch
 jeod::DynBodyFrameSwitch, 43
 init_attrjeod__DynBodyInit
 jeod::DynBodyInit, 50
 init_attrjeod__DynBodyInitLvLhRotState
 jeod::DynBodyInitLvLhRotState, 54
 init_attrjeod__DynBodyInitLvLhState
 jeod::DynBodyInitLvLhState, 58
 init_attrjeod__DynBodyInitLvLhTransState
 jeod::DynBodyInitLvLhTransState, 60
 init_attrjeod__DynBodyInitNedRotState
 jeod::DynBodyInitNedRotState, 62
 init_attrjeod__DynBodyInitNedState
 jeod::DynBodyInitNedState, 64
 init_attrjeod__DynBodyInitNedTransState
 jeod::DynBodyInitNedTransState, 67
 init_attrjeod__DynBodyInitOrbit
 jeod::DynBodyInitOrbit, 70
 init_attrjeod__DynBodyInitPlanetDerived
 jeod::DynBodyInitPlanetDerived, 77
 init_attrjeod__DynBodyInitRotState
 jeod::DynBodyInitRotState, 81
 init_attrjeod__DynBodyInitTransState
 jeod::DynBodyInitTransState, 84
 init_attrjeod__DynBodyInitWrtPlanet
 jeod::DynBodyInitWrtPlanet, 89
 init_attrjeod__MassBodyInit
 jeod::MassBodyInit, 91
 initialize
 jeod::BodyAction, 19
 jeod::BodyAttach, 27
 jeod::BodyAttachAligned, 30
 jeod::BodyDetach, 35
 jeod::BodyDetachSpecific, 37
 jeod::DynBodyFrameSwitch, 42
 jeod::DynBodyInit, 49
 jeod::DynBodyInitLvLhRotState, 54
 jeod::DynBodyInitLvLhState, 56
 jeod::DynBodyInitLvLhTransState, 60
 jeod::DynBodyInitNedRotState, 62
 jeod::DynBodyInitNedState, 64
 jeod::DynBodyInitNedTransState, 66
 jeod::DynBodyInitOrbit, 70
 jeod::DynBodyInitPlanetDerived, 75
 jeod::DynBodyInitRotState, 80
 jeod::DynBodyInitTransState, 83
 jeod::DynBodyInitWrtPlanet, 88
 initializes_what
 jeod::DynBodyInit, 49
 jeod::DynBodyInitRotState, 80
 jeod::DynBodyInitTransState, 84
 jeod::DynBodyInitWrtPlanet, 88
 InputProcessor
 jeod::BodyAction, 21
 jeod::BodyActionMessages, 24
 jeod::BodyAttach, 28
 jeod::BodyAttachAligned, 31
 jeod::BodyAttachMatrix, 33
 jeod::BodyDetach, 35
 jeod::BodyDetachSpecific, 38
 jeod::BodyReattach, 40
 jeod::DynBodyFrameSwitch, 43
 jeod::DynBodyInit, 50
 jeod::DynBodyInitLvLhRotState, 54
 jeod::DynBodyInitLvLhState, 58
 jeod::DynBodyInitLvLhTransState, 60
 jeod::DynBodyInitNedRotState, 62
 jeod::DynBodyInitNedState, 65
 jeod::DynBodyInitNedTransState, 67
 jeod::DynBodyInitOrbit, 70
 jeod::DynBodyInitPlanetDerived, 77
 jeod::DynBodyInitRotState, 81
 jeod::DynBodyInitTransState, 84
 jeod::DynBodyInitWrtPlanet, 89
 jeod::MassBodyInit, 91
 integ_frame
 jeod::DynBodyFrameSwitch, 43
 integ_frame_name
 jeod::DynBodyFrameSwitch, 43
 InvalidSet
 jeod::DynBodyInitOrbit, 69
 invalid_name
 jeod::BodyActionMessages, 24
 invalid_object
 jeod::BodyActionMessages, 24
 is_ready
 jeod::BodyAction, 20
 jeod::BodyDetach, 35
 jeod::BodyDetachSpecific, 37
 jeod::DynBodyFrameSwitch, 43
 jeod::DynBodyInit, 49
 jeod::DynBodyInitPlanetDerived, 77
 jeod::DynBodyInitRotState, 81
 jeod::DynBodyInitTransState, 84

- jeod::DynBodyInitWrtPlanet, 88
- jeod, 15
- jeod::DynBodyFrameSwitch
 - SwitchOnApproach, 42
 - SwitchOnDeparture, 42
- jeod::DynBodyInitOrbit
 - CaseEleven, 69
 - IncAscnodeAltperAltapoArgperTanom, 69
 - IncAscnodeAltperAltapoArgperTimeperi, 69
 - InvalidSet, 69
 - SlrEcclIncAscnodeArgperTanom, 69
 - SmaEcclIncAscnodeArgperManom, 69
 - SmaEcclIncAscnodeArgperTanom, 69
 - SmaEcclIncAscnodeArgperTimeperi, 69
 - SmaIncAscnodeArglatRadRadvel, 69
- jeod::DynBodyInitRotState
 - Attitude, 79
 - Both, 79
 - Rate, 79
- jeod::DynBodyInitTransState
 - Both, 83
 - Position, 83
 - Velocity, 83
- jeod::BodyAction, 17
 - ~BodyAction, 19
 - action_identifier, 21
 - action_name, 21
 - active, 21
 - apply, 19
 - BodyAction, 19
 - dyn_subject, 22
 - get_identifier, 19
 - init_attrjeod__BodyAction, 21
 - initialize, 19
 - InputProcessor, 21
 - is_ready, 20
 - operator=, 20
 - shutdown, 20
 - subject, 22
 - terminate_on_error, 22
 - validate_name, 20
- jeod::BodyActionMessages, 22
 - BodyActionMessages, 23
 - fatal_error, 24
 - illegal_value, 24
 - init_attrjeod__BodyActionMessages, 23
 - InputProcessor, 24
 - invalid_name, 24
 - invalid_object, 24
 - not_performed, 25
 - null_pointer, 25
 - operator=, 23
 - trace, 25
- jeod::BodyAttach, 26
 - ~BodyAttach, 27
 - apply, 27
 - BodyAttach, 27
 - dyn_parent, 28
 - init_attrjeod__BodyAttach, 28
 - initialize, 27
 - InputProcessor, 28
 - operator=, 28
 - parent, 28
 - succeeded, 28
- jeod::BodyAttachAligned, 29
 - ~BodyAttachAligned, 30
 - apply, 30
 - BodyAttachAligned, 30
 - init_attrjeod__BodyAttachAligned, 31
 - initialize, 30
 - InputProcessor, 31
 - operator=, 31
 - parent_point_name, 31
 - subject_point_name, 31
- jeod::BodyAttachMatrix, 31
 - ~BodyAttachMatrix, 32
 - apply, 33
 - BodyAttachMatrix, 32
 - init_attrjeod__BodyAttachMatrix, 33
 - InputProcessor, 33
 - offset_pstr_cstr_pstr, 33
 - pstr_cstr, 33
- jeod::BodyDetach, 34
 - ~BodyDetach, 34
 - apply, 35
 - BodyDetach, 34
 - init_attrjeod__BodyDetach, 35
 - initialize, 35
 - InputProcessor, 35
 - is_ready, 35
- jeod::BodyDetachSpecific, 36
 - ~BodyDetachSpecific, 37
 - apply, 37
 - BodyDetachSpecific, 37
 - detach_from, 38
 - dyn_detach_from, 38
 - init_attrjeod__BodyDetachSpecific, 38
 - initialize, 37
 - InputProcessor, 38
 - is_ready, 37
- jeod::BodyReattach, 38
 - ~BodyReattach, 39
 - apply, 40
 - BodyReattach, 39
 - init_attrjeod__BodyReattach, 40
 - InputProcessor, 40
 - offset_pstr_cstr_pstr, 40
 - pstr_cstr, 40
- jeod::DynBodyFrameSwitch, 41
 - ~DynBodyFrameSwitch, 42
 - apply, 42
 - DynBodyFrameSwitch, 42
 - init_attrjeod__DynBodyFrameSwitch, 43
 - initialize, 42
 - InputProcessor, 43
 - integ_frame, 43

- integ_frame_name, 43
- is_ready, 43
- sort_grav_controls, 43
- switch_distance, 44
- switch_sense, 44
- SwitchSense, 42
- jeod::DynBodyInit, 44
 - ~DynBodyInit, 46
 - ang_velocity, 50
 - apply, 46
 - apply_user_inputs, 47
 - body_frame_id, 51
 - body_ref_frame, 51
 - compute_rotational_state, 47
 - compute_translational_state, 47
 - DynBodyInit, 46
 - find_body_frame, 47
 - find_dyn_body, 48
 - find_planet, 48
 - find_ref_frame, 48
 - init_attrjeod__DynBodyInit, 50
 - initialize, 49
 - initializes_what, 49
 - InputProcessor, 50
 - is_ready, 49
 - operator=, 50
 - orientation, 51
 - position, 51
 - rate_in_parent, 51
 - reference_ref_frame, 51
 - reference_ref_frame_name, 52
 - report_failure, 50
 - reverse_sense, 52
 - state, 52
 - subscribed_frame, 52
 - velocity, 52
- jeod::DynBodyInitLvHRotState, 53
 - ~DynBodyInitLvHRotState, 54
 - DynBodyInitLvHRotState, 54
 - init_attrjeod__DynBodyInitLvHRotState, 54
 - initialize, 54
 - InputProcessor, 54
 - operator=, 54
- jeod::DynBodyInitLvHState, 55
 - ~DynBodyInitLvHState, 56
 - apply, 56
 - DynBodyInitLvHState, 56
 - init_attrjeod__DynBodyInitLvHState, 58
 - initialize, 56
 - InputProcessor, 58
 - lvh_object_ptr, 58
 - lvh_type, 58
 - operator=, 58
 - set_lvH_frame_object, 58
- jeod::DynBodyInitLvHTransState, 59
 - ~DynBodyInitLvHTransState, 60
 - DynBodyInitLvHTransState, 60
 - init_attrjeod__DynBodyInitLvHTransState, 60
- initialize, 60
- InputProcessor, 60
- operator=, 60
- jeod::DynBodyInitNedRotState, 60
 - ~DynBodyInitNedRotState, 61
 - DynBodyInitNedRotState, 61
 - init_attrjeod__DynBodyInitNedRotState, 62
 - initialize, 62
 - InputProcessor, 62
 - operator=, 62
- jeod::DynBodyInitNedState, 62
 - ~DynBodyInitNedState, 64
 - altlatlong_type, 65
 - apply, 64
 - DynBodyInitNedState, 64
 - init_attrjeod__DynBodyInitNedState, 64
 - initialize, 64
 - InputProcessor, 65
 - operator=, 64
 - ref_point, 65
- jeod::DynBodyInitNedTransState, 65
 - ~DynBodyInitNedTransState, 66
 - DynBodyInitNedTransState, 66
 - init_attrjeod__DynBodyInitNedTransState, 67
 - initialize, 66
 - InputProcessor, 67
 - operator=, 67
- jeod::DynBodyInitOrbit, 67
 - ~DynBodyInitOrbit, 69
 - alt_apoapsis, 70
 - alt_periapsis, 71
 - apply, 70
 - arg_latitude, 71
 - arg_periapsis, 71
 - ascending_node, 71
 - DynBodyInitOrbit, 69
 - eccentricity, 71
 - inclination, 71
 - init_attrjeod__DynBodyInitOrbit, 70
 - initialize, 70
 - InputProcessor, 70
 - mean_anomaly, 72
 - orb_radius, 72
 - orbit_frame, 72
 - orbit_frame_name, 72
 - OrbitalSet, 69
 - planet, 72
 - planet_name, 72
 - radial_vel, 73
 - semi_latus_rectum, 73
 - semi_major_axis, 73
 - set, 73
 - time_periapsis, 73
 - true_anomaly, 73
- jeod::DynBodyInitPlanetDerived, 74
 - ~DynBodyInitPlanetDerived, 75
 - apply, 75
 - body_is_required, 77

- DynBodyInitPlanetDerived, 75
- init_attrjeod__DynBodyInitPlanetDerived, 77
- initialize, 75
- InputProcessor, 77
- is_ready, 77
- operator=, 77
- ref_body, 77
- ref_body_name, 78
- required_items, 78
- jeod::DynBodyInitRotState, 78
 - ~DynBodyInitRotState, 80
 - apply, 80
 - DynBodyInitRotState, 80
 - init_attrjeod__DynBodyInitRotState, 81
 - initialize, 80
 - initializes_what, 80
 - InputProcessor, 81
 - is_ready, 81
 - operator=, 81
 - state_items, 81
 - StateItems, 79
- jeod::DynBodyInitTransState, 81
 - ~DynBodyInitTransState, 83
 - apply, 83
 - DynBodyInitTransState, 83
 - init_attrjeod__DynBodyInitTransState, 84
 - initialize, 83
 - initializes_what, 84
 - InputProcessor, 84
 - is_ready, 84
 - operator=, 84
 - state_items, 85
 - StateItems, 83
- jeod::DynBodyInitWrtPlanet, 85
 - ~DynBodyInitWrtPlanet, 86
 - apply, 86
 - DynBodyInitWrtPlanet, 86
 - init_attrjeod__DynBodyInitWrtPlanet, 89
 - initialize, 88
 - initializes_what, 88
 - InputProcessor, 89
 - is_ready, 88
 - operator=, 88
 - planet, 89
 - planet_name, 89
 - set_items, 89
- jeod::MassBodyInit, 89
 - ~MassBodyInit, 91
 - apply, 91
 - init_attrjeod__MassBodyInit, 91
 - InputProcessor, 91
 - MassBodyInit, 91
 - num_points, 91
 - operator=, 91
 - points, 91
 - properties, 92
- lvlh_object_ptr
 - jeod::DynBodyInitLvlhState, 58
- lvlh_type
 - jeod::DynBodyInitLvlhState, 58
- mass_body_init.cc, 113
- mass_body_init.hh, 113
- MassBodyInit
 - jeod::MassBodyInit, 91
- mean_anomaly
 - jeod::DynBodyInitOrbit, 72
- Models, 11
- not_performed
 - jeod::BodyActionMessages, 25
- null_pointer
 - jeod::BodyActionMessages, 25
- num_points
 - jeod::MassBodyInit, 91
- offset_pstr_cstr_pstr
 - jeod::BodyAttachMatrix, 33
 - jeod::BodyReattach, 40
- operator=
 - jeod::BodyAction, 20
 - jeod::BodyActionMessages, 23
 - jeod::BodyAttach, 28
 - jeod::BodyAttachAligned, 31
 - jeod::DynBodyInit, 50
 - jeod::DynBodyInitLvlhRotState, 54
 - jeod::DynBodyInitLvlhState, 58
 - jeod::DynBodyInitLvlhTransState, 60
 - jeod::DynBodyInitNedRotState, 62
 - jeod::DynBodyInitNedState, 64
 - jeod::DynBodyInitNedTransState, 67
 - jeod::DynBodyInitPlanetDerived, 77
 - jeod::DynBodyInitRotState, 81
 - jeod::DynBodyInitTransState, 84
 - jeod::DynBodyInitWrtPlanet, 88
 - jeod::MassBodyInit, 91
- orb_radius
 - jeod::DynBodyInitOrbit, 72
- orbit_frame
 - jeod::DynBodyInitOrbit, 72
- orbit_frame_name
 - jeod::DynBodyInitOrbit, 72
- OrbitalSet
 - jeod::DynBodyInitOrbit, 69
- orientation
 - jeod::DynBodyInit, 51
- PATH
 - BodyAction, 14
- parent
 - jeod::BodyAttach, 28
- parent_point_name
 - jeod::BodyAttachAligned, 31
- planet
 - jeod::DynBodyInitOrbit, 72
 - jeod::DynBodyInitWrtPlanet, 89
- planet_name

- jeod::DynBodyInitOrbit, [72](#)
- jeod::DynBodyInitWrtPlanet, [89](#)
- points
 - jeod::MassBodyInit, [91](#)
- Position
 - jeod::DynBodyInitTransState, [83](#)
- position
 - jeod::DynBodyInit, [51](#)
- properties
 - jeod::MassBodyInit, [92](#)
- pstr_cstr
 - jeod::BodyAttachMatrix, [33](#)
 - jeod::BodyReattach, [40](#)
- radial_vel
 - jeod::DynBodyInitOrbit, [73](#)
- Rate
 - jeod::DynBodyInitRotState, [79](#)
- rate_in_parent
 - jeod::DynBodyInit, [51](#)
- ref_body
 - jeod::DynBodyInitPlanetDerived, [77](#)
- ref_body_name
 - jeod::DynBodyInitPlanetDerived, [78](#)
- ref_point
 - jeod::DynBodyInitNedState, [65](#)
- reference_ref_frame
 - jeod::DynBodyInit, [51](#)
- reference_ref_frame_name
 - jeod::DynBodyInit, [52](#)
- report_failure
 - jeod::DynBodyInit, [50](#)
- required_items
 - jeod::DynBodyInitPlanetDerived, [78](#)
- reverse_sense
 - jeod::DynBodyInit, [52](#)
- semi_latus_rectum
 - jeod::DynBodyInitOrbit, [73](#)
- semi_major_axis
 - jeod::DynBodyInitOrbit, [73](#)
- set
 - jeod::DynBodyInitOrbit, [73](#)
- set_items
 - jeod::DynBodyInitWrtPlanet, [89](#)
- set_lvlh_frame_object
 - jeod::DynBodyInitLvlhState, [58](#)
- shutdown
 - jeod::BodyAction, [20](#)
- SlrEcclIncAscnodeArgperTanom
 - jeod::DynBodyInitOrbit, [69](#)
- SmaEcclIncAscnodeArgperManom
 - jeod::DynBodyInitOrbit, [69](#)
- SmaEcclIncAscnodeArgperTanom
 - jeod::DynBodyInitOrbit, [69](#)
- SmaEcclIncAscnodeArgperTimeperi
 - jeod::DynBodyInitOrbit, [69](#)
- SmaIncAscnodeArglatRadRadvel
 - jeod::DynBodyInitOrbit, [69](#)
- sort_grav_controls
 - jeod::DynBodyFrameSwitch, [43](#)
- state
 - jeod::DynBodyInit, [52](#)
- state_items
 - jeod::DynBodyInitRotState, [81](#)
 - jeod::DynBodyInitTransState, [85](#)
- StateItems
 - jeod::DynBodyInitRotState, [79](#)
 - jeod::DynBodyInitTransState, [83](#)
- subject
 - jeod::BodyAction, [22](#)
- subject_point_name
 - jeod::BodyAttachAligned, [31](#)
- subscribed_frame
 - jeod::DynBodyInit, [52](#)
- succeeded
 - jeod::BodyAttach, [28](#)
- SwitchOnApproach
 - jeod::DynBodyFrameSwitch, [42](#)
- SwitchOnDeparture
 - jeod::DynBodyFrameSwitch, [42](#)
- switch_distance
 - jeod::DynBodyFrameSwitch, [44](#)
- switch_sense
 - jeod::DynBodyFrameSwitch, [44](#)
- SwitchSense
 - jeod::DynBodyFrameSwitch, [42](#)
- terminate_on_error
 - jeod::BodyAction, [22](#)
- time_periapsis
 - jeod::DynBodyInitOrbit, [73](#)
- trace
 - jeod::BodyActionMessages, [25](#)
- true_anomaly
 - jeod::DynBodyInitOrbit, [73](#)
- validate_name
 - jeod::BodyAction, [20](#)
- Velocity
 - jeod::DynBodyInitTransState, [83](#)
- velocity
 - jeod::DynBodyInit, [52](#)